# Internet Authentication for Remote Access

Paulo Sergio Pagliusi

Technical Report
RHUL–MA–2008–15
20 March 2008


Department of Mathematics
Royal Holloway, University of London
Egham, Surrey TW20 0EX, England
http://www.rhul.ac.uk/mathematics/techreports

# Internet Authentication for Remote Access

by

Paulo Sergio Pagliusi

Thesis submitted to the University of London
for the degree of Doctor of Philosophy

Department of Mathematics
Royal Holloway, University of London
2008

*In memoriam* **Vice Admiral**
**Adolf Magnus Moniz Ostwald**

# Declaration

These doctoral studies were conducted under the supervision of Professor Chris Mitchell and Professor Peter Wild.

The work presented in this thesis is the result of original research carried out by myself, in collaboration with others, whilst enrolled in the Department of Mathematics as a candidate for the degree of Doctor of Philosophy. This work has not been submitted for any other degree or award in any other university or educational establishment.

Paulo Sergio Pagliusi
25 January 2008

# Abstract

It is expected that future IP devices will employ a variety of different network access technologies to gain ubiquitous connectivity. Currently there are no authentication protocols available that are lightweight, can be carried over arbitrary access networks, and are flexible enough to be re-used in the many different contexts that are likely to arise in future Internet remote access. Furthermore, existing access procedures need to be enhanced to offer protection against Denial-of-Service (DoS) attacks, and do not provide non-repudiation. In addition to being limited to specific access media, some of these protocols are limited to specific network topologies and are not scalable.

This thesis reviews the authentication infrastructure challenges for future Internet remote access supporting ubiquitous client mobility, and proposes a series of solutions obtained by adapting and reinforcing security techniques arising from a variety of different sources. The focus is on entity authentication protocols that can be carried both by the IETF PANA authentication carrier and by the EAP mechanisms, and possibly making use of an AAA infrastructure. The core idea is to adapt authentication protocols arising from the mobile telecommunications sphere to Internet remote access. A proposal is also given for Internet access using a public key based authentication protocol. The subsequent security analysis of the proposed authentication protocols covers a variety of aspects, including: key freshness, DoS-resistance, and "false-entity-in-the-middle" attacks, in addition to identity privacy of users accessing the Internet via mobile devices.

This work aims primarily at contributing to ongoing research on the authentication infrastructure for the Internet remote access environment, and at reviewing and adapting authentication solutions implemented in other spheres, for instance in mobile telecommunications systems, for use in Internet remote access networks supporting ubiquitous mobility.

# Acknowledgements

First, I would like to thank my supervisor, Professor Chris Mitchell, for his help, constant support, for always being available, and for encouraging the conclusion of this work. After that, I would like to acknowledge the many helpful insights and corrections provided by Hannes Tschöfenig and Yoshihiro Ohba for some of the articles written during my research. Next, my thanks go to the Brazilian Navy (particularly GCM, EMA, DAbM, DEnsM, DTM, CASNAV, CCIM and DepSMRJ) for funding and motivating my research. The staff and fellow students within the Mathematics Department, especially from the Information Security Group, by their warmth and friendship, have provided a most pleasant environment during my years of study at Royal Holloway — my thanks go to all of them. I would also like to thank my friends, in especial Williamson de Lima Santos, João Augusto Gomes de Queiroz, and Aldemir Lima Nunes for their encouragement and support.

I must thank my dear wife Márcia, for allowing me the time and space to pursue projects such as this, while she gets on with running the house, the family, and much else besides. Last, but by no means least, I want to thank my sons Daniel and Rodrigo for their caress and patience, and my parents Adauto and Nilza, who have supported me in whatever decisions I have taken throughout my life, and whose support has also been important during my studies in England.

# List of Publications

A number of papers resulting from this work have been presented in refereed conferences.

- C. J. Mitchell and P. S. Pagliusi. Is entity authentication necessary? In Bruce Christianson, James A. Malcolm, Bruno Crispo, and Michael Roe, editors, *Security Protocols: 10th International Workshop on Security Protocols, Proceedings, Lecture Notes in Computer Science 2845*, pages 20–33, Cambridge, UK, December 2003. Springer-Verlag.

- P. S. Pagliusi. A contemporary foreword on GSM security. In G. Davida, Y. Frankel, and O. Rees, editors, *Infrastructure Security: International Conference — InfraSec 2002, Proceedings, Lecture Notes in Computer Science 2437*, pages 129–144, Bristol, UK, October 2002. Springer-Verlag.

- P. S. Pagliusi and C. J. Mitchell. PANA/IKEv2: an Internet authentication protocol for heterogeneous access. In K. Chae and M. Yung, editors, *4th International Workshop on Information Security Applications — WISA 2003, Proceedings, Lecture Notes in Computer Science 2908*, pages 135–149, Jeju Island, Korea, August 2003. Springer-Verlag.

- P. S. Pagliusi and C. J. Mitchell. PANA/GSM authentication for Internet access. In P. Farkas, editor, *Joint 1st Workshop on Mobile Future & Symposium on Trends in Communications — SympoTIC'03, Proceedings, IEEE Catalog Number 03EX727*, pages 146–152, Bratislava, Slovakia, October 2003. Institute of Electrical and Electronics Engineers.

- P. S. Pagliusi and C. J. Mitchell. Heterogeneous Internet access via PANA/UMTS. In *the Proceedings of 3rd International Conference on Information Security, Hardware/Software Codesign And Computers Network — ISCOCO 2004*. Rio De Janeiro, Brazil, October 2004. World Scientific And Engineering Academy And Society. To be published in the WSEAS Transactions.

# Abbreviations

| | |
|---|---|
| 2G | Second Generation |
| 2.5G | Second-and-a-Half Generation |
| 3G | Third Generation |
| 3GPP | Third Generation Partnership Project |
| AAA | Authentication, Authorisation and Accounting |
| AES | Advanced Encryption Standard |
| AH | Authentication Header |
| AK | Authentication Key |
| AKA | Authentication and Key Agreement |
| AMSK | Application MSK |
| AP | Authentication Proxy |
| AS | Authentication Server |
| AuC | Authentication Centre |
| AUTH | Authentication Payload |
| AUTN | Network Authentication Token |
| AVP | Attribute Value Pair |
| BS | Base Station |
| BSC | Base Station Controller |
| BSF | Bootstrapping Server Function |
| B-TID | Bootstrapping Transaction Identifier |

| | |
|---|---|
| CA | Certification Authority |
| CAVE | Cellular Authentication and Voice Encryption |
| CB | Contact Book |
| CDMA | Code Division Multiple Access |
| CDMA2000 | Code Division Multiple Access 2000 |
| CERT | Certificate |
| CHAP | Challenge-Handshake Authentication Protocol |
| CMEA | Cellular Message Encryption Algorithm |
| DDoS | Distributed-Denial-of-Service |
| DEA | Diameter-EAP-Answer |
| DER | Diameter-EAP-Request |
| DES | Data Encryption Standard |
| DFD | Data Flow Diagram |
| DHCP | Dynamic Host Control Protocol |
| DI | Device Identifier |
| DoS | Denial-of-Service |
| DSA | Digital Signature Algorithm |
| DSL | Direct Subscriber Line |
| EAP | Extensible Authentication Protocol |
| EAP-PSK | EAP-Pre-Shared Key |
| EAP-TTLS | EAP Tunnelled TLS Authentication Protocol |
| ECMEA | Enhanced CMEA |
| EMSK | Extended MSK |
| EP | Enforcement Point |
| ESN | Electronic Serial Number |
| ESP | Encapsulating Security Payload |
| ETSI | European Telecommunications Standards Institute |

| | |
|---|---|
| FA | Foreign Agent |
| FQDN | Fully-Qualified Domain Name |
| GAA | Generic Authentication Architecture |
| GBA | Generic Bootstrapping Architecture |
| GL | Geolocation |
| GPRS | General Packet Radio Service |
| GSM | Global System for Mobile Communications |
| GTC | Generic Token Card |
| GUSS | GBA User Security Settings |
| HA | Home Agent |
| HDLC | High-level Data Link Control |
| HLR | Home Location Register |
| HMAC | Hash Message Authentication Code |
| HN | Home Network |
| HPLMN | Home PLMN |
| HSS | Home Subscriber System |
| HTML | HyperText Markup Language |
| HTTP | Hypertext Transfer |
| HTTPS | HTTP over TLS |
| ID | Identifier |
| ID-FF | Liberty Identity Federation Framework |
| IdP | Identity Provider |
| ID-SIS | Liberty Identity Service Interface Specifications |
| ID-WSF | Liberty Identity Web Services Framework |
| IETF | Internet Engineering Task Force |
| IKE | Internet Key Exchange |
| IKEv2 | Internet Key Exchange version 2 |

| | |
|---|---|
| IMEI | International Mobile Equipment Identity |
| IMSI | International Mobile Subscriber Identity |
| IMT | International Mobile Telecommunication |
| IP | Internet Protocol |
| IPsec | IP security Protocol |
| ISAKMP | Internet Security Association and Key Management Protocol |
| ISP | Internet Service Provider |
| ITU | International Telecommunication Union |
| KDK | Key-Derivation Key |
| LAI | Location Area Identifier |
| LAN | Local Area Network |
| LCP | Link Control Phase |
| LECP | Liberty Enhanced Client Profile |
| LUA | Liberty-enabled User Agent |
| MAC | Message Authentication Code |
| ME | Mobile Equipment |
| MIN | Mobile Identification Number |
| MitM | Man-in-the-Middle |
| MN | Mobile Node |
| MS | Mobile Station |
| MSC | Mobile Switching Centre |
| MSK | Master Session Key |
| MT | Mobile Termination |
| NAF | Network Application Function |
| NAI | Network Access Identifier |
| NAK | Not Acknowledged |
| NAP | Network Access Provider |

| | |
|---|---|
| NAS | Network Access Server |
| NE | Network Element |
| OMA | Open Mobile Alliance |
| OTASP | Over The Air Service Provisioning |
| OTP | One-Time Password |
| PAA | PANA Authentication Agent |
| PaC | PANA Client |
| PAMPAS | Pioneering Advanced Mobile Privacy and Security |
| PAN | Personal Area Network |
| PANA | Protocol for carrying Authentication for Network Access |
| PANATLS | PANA over TLS |
| PAP | Password Authentication Protocol |
| PC | Personal Computer |
| PDA | Personal Digital Assistant |
| PDSN | Packet Data Serving Node |
| PEAP | Protected EAP Protocol |
| PIC | Pre-IKE Credential |
| PKI | Public Key Infrastructure |
| PLMN | Public Land Mobile Network |
| PP | Personal Profile |
| PPP | Point-to-Point Protocol |
| PSK | Pre-Shared Key |
| RA | Routing Area |
| RADIUS | Remote Access Dial in User Service |
| RAI | Routing Area Identity |
| RFC | Request For Comments |
| RNC | Radio Network Controller |

| | |
|---|---|
| RPC | Remote-Procedure-Call |
| RSA | Rivest – Shamir – Adleman |
| (R)UIM | (Removable) User Identity Module |
| SA | Security Association |
| SAML | Security Assertion Markup Language |
| SCTP | Stream Control Transmission Protocol |
| SeNAA | Secure Network Access Authentication |
| SEND | SEcure Neighbor Discovery |
| SGSN | Serving GPRS Support Node |
| SHA-1 | Secure Hash Algorithm revision 1 |
| SHAMAN | Security for Heterogeneous Access in Mobile Applications and Networks |
| SIM | Subscriber Identity Module |
| SLF | Subscriber Locator Function |
| SP | Service Provider |
| SPI | Security Parameter Index |
| SQN | Sequence Number |
| SSC | Support for Subscriber Certificates |
| SSD | Shared Secret Data |
| SSL | Secure Sockets Layer |
| SSO | Single Sign-On |
| TCP | Transmission Control Protocol |
| TE | Terminal Equipment |
| TEK | Transient EAP Key |
| TLLI | Temporary Logical Link Identity |
| TLS | Transport Layer Security protocol |
| TMSI | Temporary Mobile Subscriber Identity |
| TS | Technical Specification |

| | |
|---|---|
| TSK | Transient Session Key |
| TTP | Trusted Third Party |
| UDP | User Datagram Protocol |
| UE | User Equipment |
| UID | User IDentifier |
| UMTS | Universal Mobile Telecommunications System |
| URL | Uniform Resource Locator |
| USIM | Universal Subscriber Identity Module |
| UTRAN | UMTS Radio Access Network |
| VLR | Visitor Location Register |
| VN | Visited Network |
| WAP | Wireless Application Protocol |
| WEP | Wired Equivalent Privacy |
| WG | Working Group |
| WISP | Wireless ISP |
| WLAN | Wireless LAN |
| WTLS | Wireless TLS |
| XAUTH | Extended Authentication within ISAKMP/Oakley |
| XML | Extensible Markup Language |

# Contents

# List of Tables

# List of Figures

# Part I

# Overview of Entity Authentication

# Chapter 1

# Introduction

## Contents



The aim of this chapter is to provide an introduction, and also to present the overall structure of the thesis. Section 1.1 provides the motivation and main challenges addressed by the thesis. Sections 1.2 and 1.3 describe, respectively, the principal contributions and the structure of this thesis. In fact this thesis is divided into three main parts: Part I — Overview of Entity Authentication, Part II — Internet Remote Access Authentication, and Part III — Internet Authentication Protocols & Assessments.

## 1.1 Motivation and Challenges

It is expected that future IP devices will use a variety of network access technologies to support ubiquitous connectivity, and security is clearly a very important factor in these scenarios. According to the Pioneering Advanced Mobile Privacy and Security (PAMPAS) Project [82], "the increasing heterogeneity of the networking environment is one of the long-term trends which requires new security approaches".

The main challenges include the investigation and development of unified, secure and convenient authentication mechanisms that can be used in access networks. In this context, *authentication* and *key agreement* are the central components of secure access procedures for heterogeneous network access supporting ubiquitous mobility. By *heterogeneous network access* we mean to cover the situation where arbitrary network types are being accessed, through diverse interfaces, by a number of users located in various places, in different situations and with a variety of preferences. By *ubiquitous mobility* we mean the capability for providing a universal and ever-present global mobility service to a valid user via a variety of different networks.

For example, one future requirement identified by the Security for Heterogeneous Access in Mobile Applications and Networks (SHAMAN[1]) Project is to provide flexible security means for accessing heterogeneous mobile networks, including not only GSM [179], GPRS [61] and UMTS [8], but also WLAN [71], Bluetooth[2], and other network technologies. Moreover, "heterogeneous network access control security" received the highest rating value in the list of open research issues for future mobile communication systems produced by the PAMPAS Project [82].

Currently there are no authentication protocols available that are lightweight,

[1] http://www.ist-shaman.org/
[2] www.bluetooth.com

can be carried over arbitrary access networks, and are flexible enough for use with all the various access technologies likely to be deployed to support future ubiquitous mobility. Furthermore, existing access procedures need to be made resistant to Denial-of-Service (DoS) attacks; they also do not provide non-repudiation. In addition to being limited to specific access media (e.g. 802.1X–2004 [84] for IEEE 802 links), some of these protocols are limited to specific network topologies (e.g. PPP [168] for point-to-point links) and are not scalable.

## 1.2 The Contribution of this Thesis

This thesis reviews the authentication infrastructure challenges for future heterogeneous Internet remote access supporting ubiquitous client mobility, and proposes a series of new solutions by adapting and reinforcing security techniques arising from a variety of different sources.

Firstly the thesis provides background information on security services, and establishes a general entity authentication model. In order to highlight the issues involved, we next focus on the mechanisms most widely discussed in the context of Internet authentication for remote access. The advantages and disadvantages of these schemes are assessed and compared. Much of this information is based on existing work in the Internet entity authentication literature.

Secondly the thesis defines the problem domain, establishes the usage scenarios, and defines the requirements for authentication mechanisms for Internet remote access. The authentication services and properties needed to address the threats and to achieve the desired implementation features are then specified. Finally, after the selection of a common target transport environment, this thesis proposes, evaluates and compares four new Internet authentication schemes for heterogeneous remote access, designed to meet the established requirements.

The focus of this thesis is on authentication protocols that can be carried both by the IETF Protocol for carrying Authentication for Network Access (PANA) [65, 151] authentication carrier and Extensible Authentication Protocol (EAP) [13] mechanisms, and possibly making use of an Authentication, Authorisation and Accounting (AAA) infrastructure, e.g. the Diameter protocol [34]. The core idea is to adapt authentication protocols arising from the mobile telecommunications sphere to provide security mechanisms for heterogeneous Internet remote access. A new proposal is also given for Internet access using a public key based authentication protocol.

The security analysis of the proposed authentication protocols is performed using a threat modelling technique described in Chapter 4 of Howard and LeBlanc [81, p69-124]. The analysis addresses a variety of aspects, including: key freshness, DoS-resistance and resistance to "false-entity-in-the-middle" attacks, in addition to identity privacy of users accessing the Internet via mobile devices.

## 1.3 Organisation of this Thesis

This thesis is divided into three main parts. Part I is a preliminary part containing this introduction, background material regarding cryptographic techniques, and a technical overview of entity authentication. Part I also includes a review of existing authentication protocols relevant to this thesis. Part II covers Internet remote access authentication, establishing the problem domain, usage scenarios, requirements, and service properties for new Internet remote access authentication mechanisms. Part III contains the four new protocols, and assesses them using the formal threat modelling process mentioned in section 1.2.

Part I consists of Chapters 1, 2 and 3. Chapter 2 provides background on security services and cryptographic techniques, in addition to a technical overview

of entity authentication. A number of properties of authentication protocols, such as temporality, implicit key authentication, and key freshness establishment, are identified. A general model for entity authentication mechanisms is given. The techniques, definitions and schemes discussed in this chapter are used throughout this thesis.

In Chapter 3, we review authentication protocols in the context of Internet remote access. Different perspectives related to Internet remote access are distinguished. We then describe a number of possible approaches to constructing authentication protocols, and divide initial authentication for Internet remote access into two parts. The need for a higher layer authentication procedure for heterogeneous Internet access is discussed. Possible tunnelled authentication mechanisms are considered, and a wide range of potential alternatives are reviewed. We then summarise some of the existing authentication protocols relevant to this thesis, including legacy processes, public key based procedures, and mobile telecommunications methods.

Part II consists of Chapters 4, 5 and 6. In Chapter 4, the problem domain for Internet entity authentication is established. In addition, a number of authentication scenarios for Internet remote access are described; the first two of them are categorised in terms of the layer of the protocol stack in which security is provided. Next we depict a scenario covering the absence of lower layer security. We then describe further scenarios involving respectively mobile IP, personal area networks, and limited free access.

In Chapter 5 we develop means to assess entity authentication protocols against Internet remote access requirements. We define two main sets of requirements, namely security and implementation requirements. To establish the security requirements we analyse and compare potential risks associated with entity authentication protocols, examining a number of aspects of entity authentication security for Internet remote access. To obtain the implementa-

tion requirements we analyse and compare features such as complexity, flexibility and performance. The result of this critical analysis is used later in the thesis to determine the security and implementation services and properties required of new entity authentication schemes for Internet access. In Chapter 6, we discuss the selection of the PANA protocol as the target environment for transporting the new Internet authentication schemes proposed here. This chapter describes the PANA protocol in more detail, as well as explaining the reasons for its choice as the transport environment.

Part III consists of Chapters 7 to 11. Chapter 7 presents a proposal for combining the Global System for Mobile communication (GSM) [179] authentication mechanism with PANA, which we call PANA/GSM. This scheme adapts the security techniques used in GSM to PANA. Chapter 8 presents a new means of combining the Universal Mobile Telecommunications System (UMTS) Authentication and Key Agreement (AKA) mechanism [8] with PANA, which we call PANA/UMTS. This scheme adapts the security techniques used in UMTS to the PANA environment. Chapter 9 presents a proposal for combining the Liberty Alliance Project and Third Generation Partnership Project (3GPP) Generic Authentication Architecture (GAA) security mechanisms [11] with PANA, which we call PANA/Liberty. This scheme adapts the security techniques used in Liberty and 3GPP GAA to the PANA infrastructure. Chapter 10 presents a fourth new scheme that combines the Internet Key Exchange version 2 (IKEv2) [49] public key based authentication mechanism with PANA, which we call PANA/IKEv2. This scheme adapts the security techniques used in the IKEv2 public key based scheme to the PANA framework.

In Chapter 11, we perform threat modelling and comparative analyses of the four new Internet entity authentication techniques proposed in this thesis. The goal of this chapter is to determine which of them are secure, lightweight, flexible and scalable methods allowing a client to be authenticated in a heterogeneous Internet access environment supporting ubiquitous mobility.

Finally, in Chapter 12, we summarise the findings of the thesis. In addition, suggestions for future work are provided.

# Chapter 2

# Entity Authentication

## Contents



The aim of this chapter is to provide background information on security services and cryptographic tools, together with a technical overview of entity

authentication. Section 2.1 describes the set of basic building blocks used in security protocols, including the security services and mechanisms, in addition to the (symmetric and asymmetric) cryptographic techniques of relevance to this thesis. Section 2.2 identifies a number of basic concepts underlying authentication, and describes important properties of entity authentication protocols, such as temporality, implicit key authentication and the provision of key freshness. Section 2.3 provides a general authentication model. The techniques, definitions and schemes discussed in this chapter will be used throughout this thesis.

## 2.1 Security Building Blocks

*The art of war teaches us to rely not on the chance of the enemy's not attacking, but rather on the fact that we have made our position unassailable.*

**- The Art of War, Sun Tzu**

When designing a *security protocol*, it is important initially to outline its security goals. In order to meet each of these goals, one or more *security services* need to be provided. A series of fundamental building blocks can then be deployed to implement these security services.

The set of basic building blocks used in security protocols includes *algorithms*. Cryptographic algorithms are specific instances of *security mechanisms*. A security mechanism is a general term encompassing protocols, algorithms, cryptographic tools and even non-cryptographic techniques. One or more mechanisms can be used to build a *security service*. One or more security services may be provided by a security protocol.

Figure 2.1 illustrates this idea. A typical security protocol provides one or more services. Services are provided using security mechanisms. As stated above, cryptographic algorithms are one very important category of security mechanisms[1].

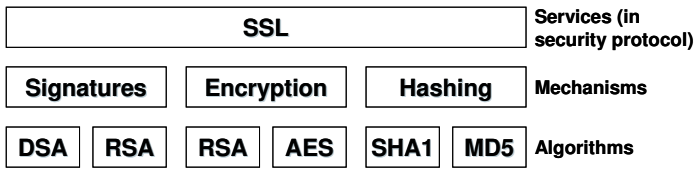


Figure 2.1: **Security building blocks**

[1]Peter Gutmann's Crypto Tutorial, http://www.cs.auckland.ac.nz/~pgut001/tutorial/ index.html.

In this section, therefore, we first define the security services of relevance to this thesis (subsection 2.1.1). In subsection 2.1.2, some of the mechanisms which can be used to provide the security services are listed. In subsection 2.1.3, the basic cryptographic techniques used throughout this thesis are briefly described.

### 2.1.1 Security Services

There are six main security services which are of importance when designing security protocols, and are consequently relevant to this thesis. They are: confidentiality, authentication, integrity, non-repudiation, access control, and availability [170, p9-11].

According to [137], 'it is important to note that all these services can be provided by a variety of different techniques, not just cryptographic means. This is one reason why it is important to distinguish between cryptographic techniques, designed to provide services, and the services themselves. Identifying which security services are needed comes from a requirements analysis of a system — deciding which cryptographic techniques should be employed to provide the services, and how they should be managed, is an implementation decision'.

The following definitions are based on those given in [39, 64, 72, 107, 132, 155, 170]. It is worth observing that these services are often combined. For instance, entity authentication can be used to support access control.

#### 2.1.1.1 Confidentiality

Confidentiality means keeping information secret from all but those who are authorised to see it [132]. In other words, it means that the assets of a computer system and transmitted information and/or data are protected against disclosure to unauthorised entities. Possible methods of confidentiality compro-

mise include printing, displaying, and other forms of disclosure, such as simply revealing the existence of the information.

In summary, confidentiality services protect against information being disclosed or revealed to entities not authorised to have that information.

#### 2.1.1.2 Authentication

Authentication is a service related to identification. However, although the terms identification and entity authentication are used synonymously by a number of authors, e.g. Menezes, Oorschot, and Vanstone [132, p386], in some places elsewhere in the literature, identification refers to learning a claimed or stated identity whereas entity authentication is the corroboration of a claimed identity. That is, identification involves *learning* an identifier (possibly a pseudonym) for an entity, e.g. a communicating party, whereas entity authentication is about *verifying* that this identifier does indeed belong to the entity who has claimed it. This thesis will use these latter definitions for these two terms.

Authentication applies to both entities and information. Two parties entering into a communication should authenticate each other. Information delivered over a channel should be authenticated, for instance, as to its origin, date of origin, and data content [132, p4]. For these reasons, this service is usually subdivided as follows (see section 2.2):

- *Entity authentication* ensures that an identity presented by a remote party participating in a communication connection or session is genuinely associated with that party [64]. It is 'an ability to verify an entity's claimed identity, by another entity' [72].
- *Message authentication*, otherwise known as data origin authentication, provides evidence to an entity that the source of a received message is as claimed [107]. The message authentication service thus provides confir-

mation of the source of a data unit. However the service in itself does not provide protection against duplication or modification of data units. Nevertheless one may argue that, implicitly at least, a message authentication service also provides data integrity since, if a message is modified, in some sense the source has changed.

According to Menezes, Oorschot, and Vanstone [132, p385], a major difference between entity authentication and message authentication is that message authentication provides no timeliness guarantees with respect to when a message was created, whereas entity authentication involves corroboration of a claimant's identity by a verifier through actual communications with the claimant at the instant of execution of the protocol. Conversely, entity authentication typically involves no meaningful message being transferred other than the claim of being a particular entity, whereas message authentication does. Nevertheless, entity authentication is often combined with key establishment; that is, as a result of executing a protocol, not only is one or both of the parties authenticated, but a secret key (known to be authentic and fresh) is established between the two parties.

#### 2.1.1.3 Integrity

Integrity ensures that information has not been altered by unauthorised means [132, p3]. In other words, it means that the assets of a computer system, transmitted information and/or data can be modified only by authorised parties and only in authorised ways. Data integrity services therefore are 'safeguards against the threat that the value or existence of data might be changed in a way inconsistent with the recognized security policy' [64].

To protect the integrity of data sent via untrusted communications channels, one must have the ability to detect data manipulation by unauthorised parties. Data manipulation includes writing, changing, changing the status,

deleting, substituting, inserting, reordering, and delaying or replaying of transmitted messages [64]. The Clark-Wilson model [39] defines integrity as those qualities which give data and systems both internal consistency and a good correspondence to real world expectations for the systems and data. Controls are needed for both internal and external reliability.

#### 2.1.1.4 Non-Repudiation

Non-repudiation prevents the denial of previous commitments or actions [132]. In other words, it is the ability to prove that an action or event has taken place, so that this event or action cannot be repudiated later. A non-repudiation service provides protection against a party to a communication exchange later falsely denying that the exchange occurred. Non-repudiation of receipt or transmission provides the sender or the receiver, respectively, with the means to establish that a message was indeed received or transmitted.

According to Ford [64], a non-repudiation service, in itself, does not eliminate repudiation. He states that 'it does not prevent any party from denying another party's claim that something occurred. What it does is ensure the availability of irrefutable evidence to support the speedy resolution of any such disagreement.' For example, one entity may authorise the purchase of property by another entity and later deny such authorisation was granted. A procedure involving examination of evidence of the authorisation by a trusted third party would typically be needed to resolve the dispute.

#### 2.1.1.5 Access Control

Access control restricts access to resources to authorised entities. The goal of an access control service is to protect against unauthorised access to any resource, e.g. a computing resource, communications resource, or information resource

[64]. Unauthorised access includes: unauthorised use, disclosure, modification, destruction, and issuing of commands. This service requires that access to the protected resources be controlled.

#### 2.1.1.6 Availability

Availability services require that computer system assets be available to authorised parties when needed. A variety of attacks can result in the loss of, or a reduction in, availability. Some of these attacks are amenable to automated countermeasures, such as authentication and encryption, whereas others require some sort of physical action to prevent or recover from the loss of availability of the elements of a distributed system [170].

### 2.1.2 Security Mechanisms

A *mechanism* is a general term encompassing protocols, algorithms, and non-cryptographic techniques (e.g. hardware protection and procedural controls) to achieve specific security objectives [132, p33]. In order to provide and support a security service, one or more security mechanisms are often combined. ISO 7498-2 [86] divides security mechanisms into two types: *specific security mechanisms*, i.e. those specific to providing certain security services, and *pervasive security mechanisms*, i.e. those not specific to the provision of individual security services [43, p33].

ISO 7498-2 then defines and describes eight types of specific security mechanism and five types of pervasive security mechanism. The eight types of specific security mechanism are: encipherment, digital signature, access control, data integrity (which includes message authentication codes), authentication exchange, traffic padding, routing control, and notarisation. For a more detailed description of the concepts underlying security mechanism standards see, for example,

Dent and Mitchell [43].

Four very important cryptographic security mechanisms relevant to this thesis are as follows:

- *Encipherment* is used to provide confidentiality; encipherment (or encryption) can also be used to help provide authentication and integrity services.
- *Digital signatures* are used to provide authentication, integrity protection, and non-repudiation services.
- *Message authentication codes* (MACs) are used to provide integrity protection and message authentication; MACs can also be used to help provide entity authentication services.
- *Authentication exchanges* are used to provide entity authentication and authenticated session key establishment.

In this thesis, a variety of different security mechanisms are discussed. It is important to observe that no single mechanism can provide all the security services.

Because of their importance, the next subsection focuses on the development, use, and management of cryptographic tools used as components in security protocols.

### 2.1.3 Cryptographic Tools

In this subsection we first outline the role of cryptography and its use of keys. After that, we briefly describe some of the basic symmetric and asymmetric cryptographic techniques used to provide security services. We also give definitions used throughout this thesis. We briefly describe each cryptographic tool

of importance to this thesis in terms of what they are, rather than the details of their operation.

#### 2.1.3.1 Cryptography

*Cryptography*, from Greek *kryptos* (hidden), and *graphein* (to write), is 'the science of designing of cipher systems' [155, p8]. Cryptography is thus 'the study of mathematical techniques related to aspects of information security such as confidentiality, data integrity, entity authentication, and data origin authentication' ([132, p4]). For a more thorough introduction to all the necessary cryptographic concepts see, for example, [132].

Cryptography involves applying an *algorithm* (a specified sequence of computational steps) to a data string to obtain a cryptographically protected version of the data string. Depending on the type of algorithm, the data string may or may not be recoverable from the transformed version. The operation of the algorithm almost always also takes as input a *key* (a sequence of symbols), which parameterises the operation of the algorithm.

Cryptographic algorithms can be divided into two main classes: *symmetric* and *asymmetric* techniques. These two classes are discussed in the following sections.

#### 2.1.3.2 Symmetric Cryptography

Symmetric (otherwise known as secret key or conventional) cryptography involves algorithms where the same key (a secret key), or two keys which can be easily computed from each other, are used as input to both the originator's and the recipient's transformation. Only the originator and the recipient know the shared secret key, which needs protection against accidental or malicious disclosure.

Therefore, in a symmetric cryptographic scheme, communications security depends on the strength of protection for the shared secret key. In fact, as mentioned in [87], 'without knowledge of the secret key, it is computationally infeasible to compute either the originator's or the recipient's transformation'.

There are a number of different types of symmetric cryptographic technique, including encryption schemes, cryptographic hash functions, and message authentication codes.

**Symmetric Encryption** Symmetric encryption, or secret-key encryption, is a symmetric cryptographic technique which can be used to provide confidentiality services. It uses either a single key for both the encryption and decryption transformations, or a pair of keys for encryption and decryption, where one is easily derived from the other [88]. According to Menezes et al. [132], the encryption is said to be symmetric if, for each associated encryption/decryption key pair, it is computationally 'easy' to determine a decryption key knowing only the encryption key, and vice versa.

There are two frequently used types of symmetric encryption scheme, namely block ciphers and stream ciphers:

- A *block cipher* is an encryption scheme which breaks up the plaintext to be transmitted into strings, or blocks, of a fixed length (e.g. of 64 or 128 bits) and encrypts them one block at a time [89].
- Conversely, a *stream cipher* is an encryption mechanism in which, using a running key or a fresh one-time-pad key stream, an encryption encrypts a plaintext in bit-wise or block-wise manner [90].

As stated by Mitchell [137], a block cipher algorithm possesses two related operations — an encryption operation, which will take as input a block of plaintext and a secret key and output a block of ciphertext, and a decryption opera-

tion which, when given the same secret key, will always map a ciphertext block back to the correct plaintext block.

Also according to Mitchell [137], 'the best known [block cipher] is almost certainly the Data Encryption Standard (DES) algorithm' [140]. This algorithm has been a de facto standard for over 20 years. However, DES secret keys only contain 56 bits, which means that, with modern technology, it is possible to search through all possible keys until the correct one has been found. As a result, DES is decreasingly often used, at least in its basic form — however, the use of DES in a compound form known as 'triple DES', with two or three different DES keys, has given the algorithm a new lease of life.

Another block cipher of increasing importance is the so called Advanced Encryption Standard (AES) algorithm, which was developed as a replacement for DES. Rijndael [42] was chosen as the AES and published as FIPS 197 [142]. This algorithm uses much longer keys than DES (of at least 128 bits) and also has a 128-bit block length, as opposed to the 64-bit blocks used by DES. One other block cipher algorithm of importance in a mobile context is KASUMI [9] (based on MISTY1 [130]), an algorithm with a 128-bit cipher key length that is incorporated into the 3GPP specifications (see section 3.5.3).

An example of a stream cipher is the A5 algorithm used in GSM (see section 3.5.1). Another example is the Ron's Code #4 (RC4) algorithm [164], which was designed by Rivest in 1987 and is one of the stream ciphers most widely used in software applications. A stream cipher is different from a block cipher in that data is encrypted 'bit by bit'. As described in [137], the major component of a stream cipher algorithm is a sequence generator, that takes a secret key as input and generates a pseudo-random sequence of bits as output. This sequence is bit-wise ex-ored with the plaintext bit sequence to derive the ciphertext. Decryption uses exactly the same process as encryption.

**Cryptographic Hash Functions** Hash functions can be used to help provide integrity and authentication services, although, since they do not use a key, they are typically used in conjunction with other security algorithms. 'A hash function is a computationally efficient function which maps strings of bits to fixed-length strings of bits' [91, 132]. Hash functions take a message as input and produce an output referred to as a *hash-code*, *hash-result*, *hash-value*, *message digest*, or just a *hash*.

Cryptographic hash functions must satisfy three properties, namely that it must be computationally infeasible to find: *(i)* for a given output, an input which maps to this output; *(ii)* for a given input, a second input which maps to the same output; and *(iii)* two different inputs which map to the same output.

Hash functions form an important part of almost all commonly used digital signature schemes. There are a number of types of hash function, including those based on block ciphers, those based on modular arithmetic, and dedicated hash functions. One well-known and widely used example of a hash-function is the Secure Hash Algorithm revision 1 (SHA-1) function, defined in FIPS 180-1 [139], which gives a 20-byte output. Another well-known example is the MD5 message-digest algorithm, defined in RFC 1321 [162], which gives a 16-byte output. The use of this latter cryptographic hash function is not recommended, since MD5 has been broken by Wang and Yu [180]. For cryptographic hash function standards, see for example [91, 92, 93, 94].

**Message Authentication Codes** Message Authentication Codes (MACs) are symmetric cryptographic techniques which can be used to provide both data origin authentication and integrity services. The data originator inputs the data to be protected into a MAC function, together with a secret key. The resulting output, a short fixed-length bit string, is known as the MAC. This MAC can be sent or stored with the data being protected.

The MAC verifier uses the same secret key to recompute a MAC value on the data. The data is only accepted as valid if the recomputed MAC agrees with the value sent or stored with the data. There are a number of widely used mechanisms for computing MACs; see for example [132]. Many of them are based on either block ciphers or cryptographic hash functions. An example of a MAC mechanism using cryptographic hash functions is the Keyed-Hashing for Message Authentication (HMAC[2]) method [123]. There are also standards for such schemes, notably ISO/IEC 9797 parts 1 and 2 [95, 96].

#### 2.1.3.3 Asymmetric Cryptography

An asymmetric cryptographic technique uses two related transformations, namely a public transformation and a private transformation. The two transformations have the property that, given the public transformation, it is computationally infeasible to derive the private transformation [87].

Diffie and Hellman [45] first introduced the concept of asymmetric cryptography in 1976. Asymmetric (or public-key) cryptography involves the use of key pairs, where each pair is made of a public key and a private key. The private key (which defines the private transformation) is kept secret by its owner, while the public key (which defines the public transformation) can be freely shared with everyone in the communications system.

**Key Management** In a large network, the number of key pairs that must be selected in order to support use of an asymmetric cryptosystem may be considerably smaller than the number of keys required to support use of a symmetric cryptosystem [132, p32]. But whilst asymmetric cryptography does not, like symmetric cryptography, rely on the sender and receiver agreeing on a shared

[2]RFC 2104 [123] specifies HMAC using a generic cryptographic hash function (denoted by H); the specific instantiation of HMAC using the MD5 or SHA-1 cryptographic hash function is known, respectively, as HMAC-MD5 or HMAC-SHA1.

secret, the user of a public key must nevertheless ensure that the correct key is used.

That is, although confidentiality is not important for the public key, it is important to ensure its origin and integrity. Public Key Infrastructures (PKIs) are used for this purpose. PKIs are systems consisting of trusted third parties (TTPs)[3], together with the services they make available to provide certified public keys. The concept of a PKI has been introduced as a means to generate, distribute and manage 'public key certificates' [64].

In a PKI, Certification Authorities (CAs) issue digitally signed certificates which bind a public key to an identifier and possibly other information (e.g. the certificate expiry date). 'In fact, a CA is a centre trusted to create and assign public key certificates. Optionally, the CA may create and assign keys to the entities' [98]. X.509 [108] is a widely adopted standard specifying the format of public key certificates. Standards also exist for other aspects of the operation of a PKI; see, for example, IETF PKIX[4].

There are a number of different types of public-key cryptographic tools, including encryption schemes, digital signature mechanisms, and key establishment techniques.

**Asymmetric Encryption** *Asymmetric encryption* algorithms can be used to provide confidentiality services. In an asymmetric encryption scheme, the public key is used for encryption and the private key for decryption. The best known algorithm for public key encryption is RSA, which was proposed in 1978 by Rivest, Shamir, and Adleman [163]. There are a number of standards describing how to use public-key encryption, including the use of RSA; see, for example, [83, 99].

[3] *A Trusted Third Party* (TTP) is a security authority, or its representative, trusted by other entities with respect to security related activities [97].

[4] http://www.ietf.org/html.charters/pkix-charter.html

**Digital Signatures** A *digital signature* is a cryptographic mechanism which can be used to help provide entity authentication, data origin authentication, integrity and non-repudiation services. ITU-T X.800 [107] defines a digital signature as 'data appended to, or a cryptographic transformation of, a data unit, that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery'. 'The process of signing entails transforming the message and some secret information held by the entity into a tag called a *signature*' [132, p22].

A signature mechanism consists of two components, namely *signing* and *verification* algorithms. The signing algorithm involves the transformation of the message into a signature, using the signing entity's *private key*. For a digital signature mechanism to work, there is a need for a verification process, so that it is possible to verify whether a signature on a message was genuinely created by the claimed entity.

This verification process takes as input the signature, the message, and the signer's *public verification key*, and outputs an indication as to whether or not the signature on the message is valid. Typically a digital signature functions as a check value on data. That is, when sending a digital signature on data, both the data and the signature need to be transmitted. Signature mechanisms do exist where part or all of the data can be recovered from the signature itself, but these are less commonly used.

Many digital signature schemes have been proposed over the last 25 years (for example, see [132]). For digital signature standards, see for instance [97, 100, 101, 141].

**Key Establishment Schemes** A *key establishment* mechanism is a process whereby a shared secret key is made available to two or more parties, typically for subsequent use with a symmetric cryptographic algorithm, such as an en-

cryption or MAC scheme. Key establishment schemes can be broadly subdivided into two kinds of mechanisms [132]:

- *key agreement mechanisms* — key establishment techniques in which a shared secret is derived by two or more parties as a function of information contributed by, or associated with, each of the parties, (ideally) such that no party can predetermine the resulting value (see, for example, the Diffie-Hellman key agreement algorithm [45]); and
- *key transport mechanisms* — key establishment techniques where one party creates or otherwise obtains a secret value, and securely transfers it to the other(s). Key transport mechanisms directly employ asymmetric or symmetric encryption.

Asymmetric cryptography based key establishment techniques, including both key agreement and key transport mechanisms involving various combinations of encryption and signatures, are standardised in ISO/IEC 11770-3 [102].

Menezes, van Oorschot and Vanstone [132] state that authenticated key transport may be regarded as a special case of message authentication with confidentiality, where the message includes a cryptographic key. Key establishment protocols involving authentication typically require a set-up phase whereby authentic and possibly secret initial keying material is distributed.

*Key pre-distribution* mechanisms are key establishment protocols whereby the resulting established keys are completely determined a priori by initial keying material. In contrast, *dynamic key establishment* mechanisms are those whereby the key established by a fixed pair (or group) of users varies on subsequent executions. Dynamic key establishment is also referred to as *session key establishment* [132, p490–491].

Many key establishment protocols based on public-key techniques (see, for example, [132, p515–524]) employ digital signatures for message authentication. Additional variations beyond key transport and key agreement exist, including various forms of key update, such as key derivation [132, p490]. In this case, the nature of the derived keying material depends on whether or not perfect forward secrecy is required.

As stated by Harkins and Carrel [76], *perfect forward secrecy* refers to the notion that compromise of a single key will only permit access to data protected by that key. For perfect forward secrecy to exist, the key used to protect transmission of data must not be used to derive any additional keys, and if the key used to protect transmission of data was derived from some other keying material, that material must not be used to derive any more keys.

## 2.2 Authentication: Basic Concepts

*We cannot enter into alliance with neighbouring princes until we are acquainted with their designs.*

**- The Art of War, Sun Tzu**

This section identifies a number of basic concepts relating to authentication, and also describes certain key properties of entity authentication protocols. We first state a number of concepts and thoughts related to the identification process (section 2.2.1); we then mention the distinct senses of authentication, and describe various different approaches to authentication, categorised by the type of evidence involved (section 2.2.2). Section 2.2.3 next defines what we mean here by an authentication protocol, and lists its possible states. In sections 2.2.4 and 2.2.5, certain basic properties of authentication protocols, including temporality, implicit key authentication and key freshness establishment, are briefly

described.

### 2.2.1 The Identification Process

The concepts and definitions described in this section were mostly stated by Clarke [40, 41]. The term *entity* encompasses all manner of real-world things, including objects, devices, animals, people, and 'legal persons' such as corporations, trusts, and incorporated associations. An entity has a range of characteristics, features or attributes. An *identity* is a particular presentation of an entity. An entity does not necessarily have a single identity, but may have many. Therefore individual entities of all kinds may have multiple identities, rather than just one.

An *identifier* is one or more data-items concerning an identity that are sufficient to distinguish it from other instances of its particular class, and that is used to signify that identity. Conventional identifiers such as names and codes are associated with identities rather than with entities. However, some identifiers are not only capable of distinguishing between identities, but can only characterise the entity itself, e.g. biometrics, because they measure some feature of the individual, or of the individual's behaviour. *Identification* is therefore the process whereby an identifier is acquired, and an association achieved between an identity and stored information, e.g. in a database.

*Identity authentication* is the further verification process, whereby a sufficient degree of confidence is established that the identification process has delivered a correct result; this can be performed by collecting multiple identifiers, acquiring knowledge that only the right entity is expected to have, or inspecting tokens that only the individual entity is expected to possess. Also known as *entity authentication*, identity authentication then refers to a process designed to cross-check against additional evidence the identity signified by the identifier acquired during the identification process. An item of evidence in this

context is usefully referred to as an *authenticator* or a *credential*.

It is important to state that the concept of certainty of identity is an unrealisable hope, because all identification and authentication techniques are subject to error. In addition to accidental errors, all are capable of being circumvented with varying degrees of ease. False inclusions arise, including successful masquerades, and the tighter that the tolerances are set, the greater is the frequency of false exclusions. Rather than the naive concept of proof of identity, the focus of this thesis is thus on evidence of identity.

### 2.2.2 Authentication and Credentials

As explained in section 2.1.1, in the context of communications security the term *authentication* has two distinct senses. One is *message authentication*, which is concerned with verifying the origin of received data, and, typically, involves a process for confirming the integrity of the data.

The other sense is *entity authentication*, described in section 2.2.1, where one entity (the verifier) gains assurance, through acquisition of corroborative evidence and/or supporting *credentials*, that the identity of another entity (the claimant) is as declared at the instant of execution of the mechanism, thereby preventing impersonation [155, p92]. This thesis will focus on authentication in this second sense. Thus in the remainder of this thesis, the word authentication, when used without further clarification, is always used to mean entity authentication.

When the claimant is a human user, the *credentials* can be categorised into one of "something you know" (e.g. a password), "something you have" (e.g. a token or smart card), or "something you are or you do" (e.g. biometrics), leading respectively to three approaches to user authentication: proof by *knowledge*, proof by *possession* or proof by *property* [38, p119–124]. There is also an alter-

native approach based on *user location* ("where you are"), e.g. in a physically secured terminal room of a bank [155, p93].

When the claimant is a machine, authentication processes can be divided into two types: *cryptographic* (or *strong*), e.g. challenge-response mechanisms, and *other* (or *weak*), e.g. password schemes. Some authors, like Menezes, Oorschot and Vanstone [132], include here a third authentication type, making a distinction between weak, strong, and zero-knowledge based authentication.

Zero-knowledge authentication protocols are similar in some regards to the challenge-response protocols, but are based on the ideas of interactive proof systems and zero-knowledge proofs, employing random numbers not only as challenges, but also as *commitments* to prevent cheating. For further details see [132, p405–417].

### 2.2.3 Authentication Protocols

As described in section 2.1.1.2, authentication can be summarised as identification plus verification. *Identification* is the procedure whereby an entity claims a certain identity ("Who are you?"), while *verification* is the procedure whereby that claim is checked ("Can you prove it?"). Thus the correctness of authentication relies heavily on the verification method employed.

When the verification method is based on cryptography, authentication tends to rely on an exchange of messages between the pair of entities through a communications medium. This exchange is called an *authentication protocol*.

An authentication protocol is a special type of communications protocol, i.e. a precisely defined sequence of communication and computation steps. A communication step transfers messages from one entity (the sender) to another (the receiver), while a computation step updates an entity's internal state.

Two distinct states can be identified at the verifier upon termination of an authentication protocol, one signifying successful authentication and the other failure. These states will be useful in two subsequent steps called *authorisation*, which is the "act of determining whether a requesting identity will be allowed access to a resource"; and *accounting*, which is "the act of collection of information on resource usage for the purpose of capacity planning, auditing, billing or cost allocation" [34].

### 2.2.4 Temporality

Authentication protocols provide assurance regarding the identity of an entity *only* at a given instant in time. Thus the authenticity of the entity can be ascertained just for the instance of the authentication exchange. If the continuity of such an assurance is required, use of additional techniques is necessary. For example, authentication can be repeated periodically, or the authentication protocol could be linked to an ongoing integrity service. In the latter case, the authentication protocol needs to be integrated with a key establishment mechanism, such that a by-product of successful entity authentication is a shared secret, a *session key*, appropriate for use with an integrity mechanism used to protect subsequently exchanged data [132, p385-388].

Therefore, when an entire communication session has to be authenticated, typically the initial message exchange will serve to set up a session key between the entities. Further messages are then protected by an integrity mechanism employing the session key. In this case, an authentication protocol meets its objective if it can be demonstrated that it establishes a *fresh* session key, known only to the participants in the session and possibly some TTPs [72]. The next section discusses this in more detail.

### 2.2.5 Implicit Key Authentication and Key Freshness Establishment

Following [136], we next consider the case where a protocol simultaneously provides entity authentication and session key establishment, and this session key is used to protect data subsequently transferred. See section 2.1.3.3 for details regarding key establishment schemes.

*Implicit key authentication* is the property whereby one party is assured that no other party aside from a specifically identified second party (and possibly an additional identified TTP) may gain access to a particular secret key. *Key confirmation* is the property whereby one party is assured that a second party actually has possession of a particular secret key. *Explicit key authentication* is the property obtained when both implicit key authentication and key confirmation hold.

A further property, useful in some applications, is key freshness. 'A key is fresh (from the viewpoint of one party) if it can be guaranteed to be new, as opposed to possibly an old key being reused through actions of either an adversary or authorized party' [132, p494]. In other words, *key freshness* is the property that the party to a key establishment process knows that the key is a 'new' key. Above all, the party should have evidence that the messages received during the protocol by which the key has been established are 'fresh' messages, i.e. they are not replays of 'old' messages from a previous instance of the protocol.

The absence of key freshness would enable an interceptor to force the verifier to keep re-using an 'old' session key, which might have been compromised. It would therefore seem reasonable to make key freshness a requirement for most applications of key establishment protocols.

To conclude this discussion, we note that the two critically important prop-

erties for most key establishment protocols would seem to be implicit key authentication and key freshness. Explicit key authentication is not always so important, and is, in any case, reached once a party receives evidence of use of a key.

## 2.3 General Authentication Model

The provision of an entity authentication service "almost inevitably involves a series of messages being exchanged between the parties concerned, each transfer of a message being known as a *pass* of the protocol. Such a sequence of messages is normally called an entity authentication protocol, or simply an authentication protocol (we use this shorter term throughout). For historical reasons, the term 'authentication mechanism' is instead used throughout ISO/IEC 9798. However, the term authentication protocol is used almost universally elsewhere" [43, p196].

A general model for authentication protocols taken from ISO/IEC 9798-1 [87] is shown in Figure 2.2. In this picture, the lines indicate potential information flows. Entities A and B may either directly interact with the trusted third party TP, indirectly interact with the trusted third party through B or A respectively, or use some information issued by the trusted third party.

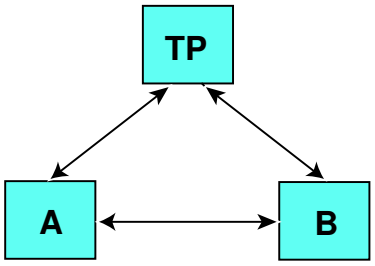


Figure 2.2: **Authentication model**

According to ISO/IEC 9798-1 [87], it is not essential that all the entities and

exchanges are present in every authentication mechanism. For *unilateral* authentication, a type of authentication which provides one entity with assurance of the other's identity but not vice-versa, entity A is considered the claimant, whereas entity B is considered the verifier. For *mutual* authentication, a type of authentication which provides both entities with assurance of each other's identity, A and B each take the roles of both claimant and verifier.

To meet the goals of an authentication protocol, the entities generate and exchange standardised messages. It takes the exchange of at least one message for unilateral authentication and the exchange of at least two messages for mutual authentication. An additional step may be needed if a challenge has to be sent to initiate the authentication exchange. Additional steps may also be needed if a TTP is involved.

# Chapter 3

# Authentication Protocols for Internet Remote Access

## Contents

---

The aim of this chapter is to review authentication protocols in the context of Internet remote access. Firstly, a variety of different perspectives related to Internet remote access are distinguished; we also describe a number of possible approaches to constructing authentication protocols (section 3.1). Secondly, we divide the initial authentication and key establishment processes for network access into two parts. The need for a higher layer authentication procedure in the first phase is discussed. Possible tunnelled authentication mechanisms are considered, taking into account the vulnerabilities arising from their use, and possible solutions to these problems (section 3.2). We then summarise a number of existing authentication protocols relevant to this thesis, including legacy processes (section 3.3), the EAP architecture (section 3.4), mobile authentication

methods (section 3.5), tunnel and key generation schemes (section 3.6), compound tunnelled alternatives (section 3.7), public key based procedures (section 3.8), the AAA backend infrastructure (section 3.9), and the Liberty Alliance Project architecture (section 3.10).

We focus here on the use of the EAP architecture as a format to carry authentication information, not only on Point-to-Point Protocol (PPP) links but also on wired IEEE 802 networks, wireless Local Area Networks (LANs), and the Internet. The definitions and schemes discussed in this chapter will be used throughout the remainder of this thesis.

## 3.1 Internet Access and Authentication

In most cases, network access requires some form of authentication of the end user to the network. Hence, many networks require entities to provide their credentials before being allowed access to network resources. *Network resources* could include: basic network access (sometimes meaning Internet access), access to LAN services (such as printer servers, file servers, database servers), or more specific communication services in the network (e.g. electronic mail, FTP connections, web servers), or even a certain grade of service (e.g. free vs. paid services).

We explain here the basic concepts underlying Internet remote access and the authentication process involved. Section 3.1.1 distinguishes between two different types of Internet remote access. Section 3.1.2 describes possible approaches to constructing authentication protocols for network access.

### 3.1.1 Internet Remote Access Perspectives

The term *remote access* has two distinct meanings in the context of network authentication. This is illustrated by contrasting definitions of authentication for network remote access present in two standards documents: namely ISO/IEC 18028-4 [103] and the IETF PANA RFC 4016 [151].

The ISO/IEC document considers authentication for network remote access from the point of view of a roaming user that already has access to a public network, such as the Internet. This user wishes to connect to a specific remote network and use its resources just as if a direct LAN link existed. Thus, in this case, the user does not need to be authenticated to achieve Internet network connectivity, since such access is already available, but instead needs to be authenticated in order to gain access to a remote network using the Internet.

By contrast, the IETF PANA Working Group (WG) document considers authentication for network remote access from the point of view of a roaming entity $A$ (a user or a device such as a notebook computer, or a Personal Digital Assistant (PDA), acting on behalf of a user) that needs to be authenticated to an entity $B$ within an access network in order to be provided with network connectivity. This viewpoint is shown in Figure 3.1, where entity $B$, which belongs to the access network, authenticates remote entity $A$ (i.e. its identity, signified by the recorded identifier), using credentials held by $A$, to provide Internet network connectivity. This perspective is very similar to GSM and 3GPP scenarios, where a user owning a device (a mobile station) needs to be authenticated because she is provided with connectivity in a telecommunications network.

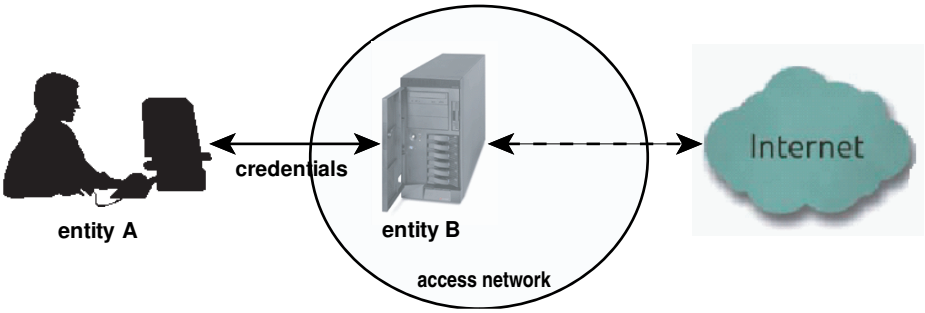


Figure 3.1: **Authentication for Internet remote access**

This thesis will focus on authentication for network remote access in the second sense. Thus the term *authentication for Internet remote access* is used in the text below to avoid any ambiguity.

### 3.1.2 Authentication Approaches

When a remote entity attaches to a visited network it has never been in contact with before, typically the network wants assurance that it will be properly paid

for the services granted to the entity. In addition, the entity may want assurance that the network will not tamper with any data that the entity transmits via the access network. This requires some form of authentication between the two parties, which can be carried out in a variety of ways. The general authentication model discussed in section 2.3 supports a number of alternative approaches, including unilateral and mutual authentication, with or without making use of a TTP.

The typical scenario for network remote access is the case where a subscription-like relationship exists between the remote entity and a home network, which involves the prior set-up of security information, such as algorithms and keys. The cryptographic mechanisms used in the authentication protocol lead to another means of classifying techniques, i.e. between methods based on symmetric or asymmetric cryptography. As explained in the SHAMAN Final Technical Report[1], "whereas the former (method) requires the involvement of the home network during the initial authentication process between the remote entity and the visited (access) network, the latter allows for architectures that avoid an on-line involvement of the home network, since the authentication protocol may then be based on certificates". In this latter case, however, a public key infrastructure is required to support certificate verification.

Another distinction can be made between one-step and two-step schemes. Whereas the former use a single protocol for mutual authentication, the latter use two separate authentication protocols, one for network authentication and the second for authentication of the remote entity. In this latter case, the network authentication protocol is typically executed first and is then used to create a protected tunnel through which the remote entity authentication protocol is run. In particular, such a tunnel provides confidentiality protection for the remote entity identity and other access negotiation information against active attacks during the initial access phase.

[1] http://www.ist-shaman.org/

Finally, in addition to the subscription-based cases, alternative remote access scenarios can be considered, where payment is provided by means other than relying on the subscription relation between the remote entity and a home network. For example, this could be achieved using credit cards or various forms of electronic money, leading to quite different security architectures, e.g. the frameworks for secure mobile commerce described by Knospe and Schwiderski-Grosche [120, 121, 122]. These scenarios, however, are outside the scope of this thesis. The authentication scenarios covered by this thesis will be discussed in more detail in Chapter 4.

In the next section we discuss the two basic phases into which the initial authentication process for Internet remote access can be divided.

## 3.2 Initial Authentication

From an architectural point of view, the process of initial authentication and key establishment for Internet remote access can be divided into two phases, as shown in Figure 3.2. The first phase takes place between the remote entity and the access network, and the second phase between that network and a backend AAA[2] infrastructure (discussed in section 3.9).

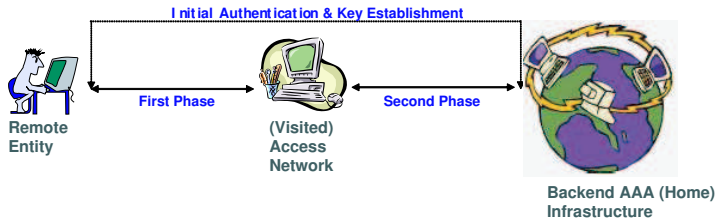


Figure 3.2: **Internet authentication**

[2]AAA is an acronym for Authentication, Authorisation and Accounting, which is a term used to describe the backend framework for applications such as network access or IP mobility [34].

One standardisation forum addressing the first phase is the PANA working group (see section 3.7.5 and Chapter 6). The main goal of this working group is to design a protocol that transports authentication data between a remote entity seeking access to a network and another entity located in the access network. More specifically, the objective of PANA is to devise a simple model, independent of the access network type, for transferring user authentication information to the access network and, optionally, to the AAA infrastructure. The protocol used by PANA is EAP (see section 3.4), which was originally designed for use with the Point-to-Point Protocol [168] (see section 3.3.2).

EAP does not specify any authentication method, but is simply a transport mechanism, allowing concrete authentication methods for EAP, such as legacy authentication protocols, public key based authentication procedures, and even methods from the mobile telecommunications area, to be defined separately. We discuss these specific authentication schemes in more detail in the following sections.

Although this thesis focuses on the first phase of the Internet authentication process, it is also important to consider the second phase, and in particular how it may be combined with the first phase. Therefore we also describe (in section 3.9) certain specific AAA backend protocols, i.e. RADIUS [161], Diameter [34], and Diameter EAP Application [59].

### 3.2.1 A Higher Layer for Internet Authentication

One simply way to carry out the first phase of the internet remote access authentication process, i.e. the authentication exchange between the remote entity and the network, is as follows. The remote entity establishes a connection with an entity in the access network, e.g. a user with a PC connects to the telephone network using a modem, and employs PPP authentication [168] to set up a dial-up connection to an Internet Service Provider (ISP). This direct connection may

exist for as long as is necessary, and functions somewhat like a leased line which is only active on demand. It may also become a permanent connection when Direct Subscriber Line (DSL[3]) or other broadband technology is used. Another example would be the use of IEEE 802.1X [84], which provides port-based network access control with peer authentication in point-to-point LAN or WLAN segments. IEEE 802.1X can be used to authenticate a remote entity to an IEEE 802.11 network.

Using today's technology, as in the examples above, authentication is generally performed at the time of link establishment. Moreover, authentication for Internet remote access is usually tied to the access technology itself. As a result, specific authentication schemes are implemented that depend on the type of network being accessed. The examples above show this access technology dependence in the case of the use of IEEE 802.1X by Wireless Internet Service Providers (WISP) for authenticating an entity to an IEEE 802.11 network, and PPP authentication in the case of a dial-up connection to an ISP.

However, according to Ohba et al. [144], authentication for Internet remote access may be performed at a higher layer, either at the network (IP) or the application layer. More evidence on why higher layer authentication is needed when link layer authentication is available can be found, for example, in the SHAMAN Final Technical Report[4]. This has the advantage of decoupling authentication from the access technology. The supposition here is that link layer connectivity is provided by the Internet access network operator. Thus common compound authentication protocols, e.g. the tunnelled authentication solutions located at the network (IP) layer or above, mentioned in the next section and currently being designed by the IETF, might be good candidates to solve this problem.

[3] http://www.dslforum.org/
[4] http://www.ist-shaman.org/

### 3.2.2 Tunnelled Authentication Mechanisms

A number of tunnelled authentication mechanisms have been proposed by the IETF for use when connecting remote entities for Internet remote access, including XAUTH [131], PIC [16], PANATLS [143], EAPTTLS [70] and PEAP [149]. Each of these protocols supports tunnelling of legacy one-way authentication methods in order to provide a number of benefits, including access technology independence, well understood key derivation, replay and dictionary attack protection, and privacy support.

Nevertheless, it is important to consider a larger spectrum of solutions to the problem of managing legacy authentication methods. This is supported by the fact that tunnelled protocols such as PANATLS (see section 3.7.6), although aiming to address problems such as access technology dependence, can be considered as part of a transition from legacy one-way authentication methods to certificate-based authentication [158].

It is important to take into account the vulnerabilities arising from the use of tunnelled authentication mechanisms in certain circumstances, as well as possible solutions to these problems, as described in the next section.

### 3.2.3 Vulnerabilities in Tunnelled Protocols

It has been discovered that the use of tunnelled protocols in the first phase, together with legacy client authentication protocols in the second phase, creates a vulnerability to an active Man-in-the-Middle (MitM) attack, which allows the attacker to impersonate the remote entity (see [21, 158]). The attack becomes possible if the legacy client authentication protocol is used in multiple environments (e.g. with and without tunnel-protection).

As stated by Asokan, Niemi and Nyberg [21], the MitM attack can occur

when using tunnelled authentication protocols constructed as combinations of two protocols: an inner protocol, and an outer protocol. The inner protocol, which provides authentication of the client to the network, consists of the legacy client authentication method. The outer protocol, which provides authentication of the network to the client, is used to protect the exchange of the inner protocol messages. The outer protocol is solely responsible for the generation of session key material.

Therefore, the session key material is based only on a unilateral authentication, in which the network server is authenticated to the client. The combination of the facts that firstly, the client authentication protocol can be used in multiple environments, secondly, the session keys are derived solely on the basis of the network authentication protocol, and thirdly, the client authentication protocol is not aware of the protection protocol, opens up the opportunity for a man-in-the-middle to impersonate the legitimate client. The active MitM attack proceeds as follows:

1. The MitM waits for a legitimate device to enter an untunnelled legacy remote authentication protocol and captures the initial message sent by the legitimate client.

2. The MitM initiates a tunnelled authentication protocol with an authentication agent.

3. After the tunnel is set up between the MitM and the authentication agent, the MitM starts forwarding the legitimate client's authentication protocol messages through the tunnel.

4. The MitM unwraps the legacy authentication protocol messages received through the tunnel from the authentication agent and forwards them to the legitimate client.

5. After the remote authentication has ended successfully, the MitM derives

the session keys from the same keys it is using for the tunnel.

Asokan, Niemi and Nyberg [21] have shown that the MitM problem can be addressed by either restricting the use of the legacy authentication protocol to a specific environment only, or by implementing a cryptographic binding between the protocols used in the first and second phases. The latter is deemed to be the recommended solution, as it allows more flexible use of existing strong EAP methods (section 3.4).

Tunnelled authentication protocols may also be vulnerable to a particular type of Denial-of-Service (DoS) attack, known as a 'blind resource consumption DoS attack' [65]. Such an attack would typically be launched during the initial handshake phase of the authentication process, by attackers masquerading as remote clients. The attackers would then bombard the access network with messages in order to swamp it, causing it to exhaust all available resources, and preventing network access by legitimate clients.

One means of mitigating this type of DoS attack, or at least making it more difficult to conduct effectively, requires the access network to generate a random value, referred to as a *cookie*, as described in [65]. During the initial handshake phase, the access network sends a request message carrying the cookie, and then checks whether the client answer message contains the expected cookie value. If the cookie is valid, the access network enters the authentication and authorisation phase. Otherwise, it discards the received message.

## 3.3 Legacy One-Way Authentication Protocols

Currently a number of legacy one-way (user) authentication methods are in use, including PAP [126] & CHAP [169], EAP-MD5 [27], One-Time-Password (OTP) [28, 75], and Generic Token Card (GTC) [28]. They all provide unilateral (user

to access network) authentication, and none of them derive keys that could be used in constructing compound MACs and/or compound keys, or provide keying material for authentication and/or encryption of a subsequent data stream. These legacy authentication methods can be used in at least two of the following three modes:

- plain mode;
- EAP encapsulated; or
- EAP encapsulated and tunnelled within a secure channel set up as a result of the initial authentication protocol.

Some legacy authentication methods encapsulated as EAP types, such as OTP, should not be used without a specific form of tunnelling. OTPs are typically created for a specific application, which can be contacted only through a unique form of protected tunnel. However, the situation is quite different for more sophisticated authentication methods, which are used with and without tunnels. This may open the system up to the vulnerability described in [21], which allows a MitM (see section 3.2.3) to impersonate the remote entity.

In the following sections we summarise some of the existing one-way legacy authentication protocols that do not involve the generation of keys.

### 3.3.1 PPP PAP & CHAP

The Point-to-Point Protocol (PPP) [168] provides a standard method of encapsulating network layer protocol information over point-to-point links. PPP also defines an extensible link control protocol, which allows negotiation of an authentication protocol for authenticating its remote entity (called the *peer*[5]),

[5]The end of the point-to-point link which is being authenticated by the authenticator.

before allowing network layer protocols to communicate over the link with an entity in the access network (called the *authenticator*[6]).

The Password Authentication Protocol (PAP) [126] provides a simple method for the remote entity or peer to establish its identity to the authenticator in the access network using a two-way handshake. This is done only upon initial link establishment. After the link establishment phase is complete, an *identifier/password* pair is repeatedly sent by the peer to the authenticator until authentication is acknowledged or the connection is terminated. The peer is in control of the frequency and timing of the attempts. PAP is obviously not a strong authentication method. Passwords are sent in clear over the circuit, and there is no protection from playback or repeated trial and error attacks[7].

The PPP Challenge Handshake Authentication Protocol (CHAP) [169] is a stronger legacy authentication method using PPP, which uses a random *challenge*, with a cryptographically hashed *response*, which depends on the challenge and a secret key. CHAP is used to periodically verify the identity of the peer using a three-way handshake (see Figure 3.3). This is done upon initial link establishment, and may be repeated at any time after the link has been established. After the link establishment phase is complete (0), the authenticator sends a 'challenge' message to the peer (1). The peer responds with a value calculated using a one-way hash function (see section 2.1.3.2). The authenticator checks the response against its own calculation of the expected hash value (2). If the values match, the authentication is acknowledged (3a); otherwise the connection should be terminated (3b). At random intervals, the authenticator sends a new challenge to the peer (4), and repeats steps 1 to 3.

[6]The end of the link that requires authentication to be performed.

[7]The PAP legacy authentication method is most appropriate for use where a plaintext password must be available to simulate a login at a remote host. In such use, this method provides a similar level of security to the usual user login at the remote host.

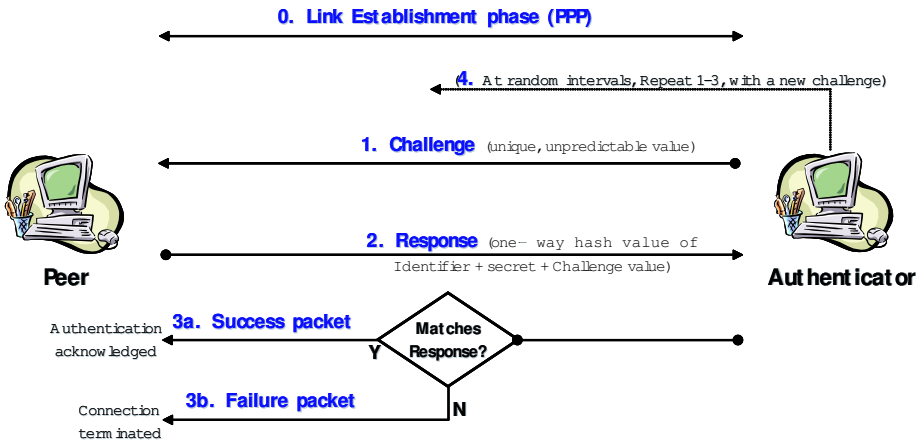


Figure 3.3: **CHAP typical steps**

**Main Advantage of CHAP:** CHAP provides protection against playback attacks by the peer through the use of an incrementally changing identifier and a variable challenge value. This method depends upon a 'secret' known only to the authenticator and that peer. The secret is not sent over the link.

**Main Disadvantages of CHAP:** CHAP requires that the secret be available to the authenticator in plaintext form. As a result, irreversibly encrypted password databases, as commonly used, e.g. in Unix, and which protect passwords against snooping by systems administrators, cannot be used. Hence CHAP is not ideally suited for large installations, since every possible secret is maintained at both ends of the link. CHAP is also incapable of protecting against real time active wiretapping attacks.

### 3.3.2 PPP EAP-MD5

RFC 2284 [27] defines the PPP Extensible Authentication Protocol (EAP), which is a general protocol for PPP authentication [168] which supports multiple authentication mechanisms. EAP does not select a specific authentication

mechanism at the Link Control Phase (LCP), but rather postpones this until the Authentication Phase. This allows the authenticator to request more information before determining the specific authentication mechanism. This also permits the use of a 'backend' server which actually implements the various mechanisms, while the PPP authenticator merely passes through the authentication exchange.

The PPP EAP authentication exchange proceeds as follows:

- After the Link Establishment Phase is complete, the authenticator sends one or more Requests to authenticate the peer. The Request has a type field to indicate what is being requested. Examples of Request types include Identity, MD5-challenge, One-Time Passwords, Generic Token Card, etc. All EAP implementations must support the MD5-Challenge mechanism, which corresponds closely to the CHAP authentication protocol (see section 3.3.1). Typically, the authenticator will send an initial Identity Request followed by one or more Requests for authentication information[8].
- The peer sends a Response packet in reply to each Request. As with the Request packet, the Response packet contains a type field which corresponds to the type field of the Request.
- The authenticator ends the authentication phase with a Success or Failure packet.

An authenticator authenticates the peer using a sequence of methods. A common example of this is an Identity request followed by an EAP authentication method, such as the legacy MD5-Challenge, which is analogous to the PPP CHAP protocol described in section 3.3.1, with MD5 (see section 2.1.3.2) as the specified algorithm. The basic steps, listed in Figure 3.4, are as follows.

[8]However, an initial Identity Request is not required, and may be bypassed in cases where the identity is presumed (leased lines, dedicated dial-ups, etc.).

The link between the peer and the authenticator is established (0). The initial method is started. The authenticator sends a packet to query the identity of the peer (1). The peer obtains the user identity and answers the authenticator (2). If the initial method completes unsuccessfully (3a), then the authenticator sends a Failure packet. If it completes successfully (3b), then the authenticator sends a Request packet for an authentication method. If the authentication type is acceptable (4b), the peer then sends a Response packet containing a type field matching the Request. If it is unacceptable (4a), then the peer sends a Response packet containing a 'Not Acknowledged' (NAK) type field plus the desired authentication method, and the authenticator may restart from step 3, changing to the peer's desired method.

The Notification Type is optionally used to convey a displayable message of an imperative nature to the peer (5). The peer sends a Response packet in reply to the Notification message (6). The sequence of authentication methods proceeds until either an authentication method fails (7a) or the final authentication method completes successfully (7b).

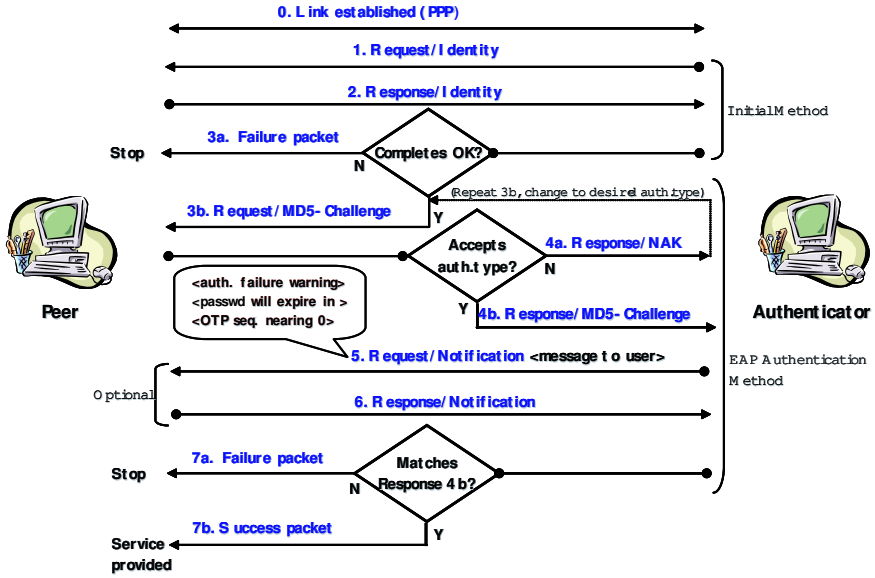


Figure 3.4: **PPP EAP-MD5 typical steps**

**Main Advantages of PPP EAP:** The PPP EAP protocol can support multiple authentication mechanisms without having to pre-negotiate a particular mechanism during the LCP Phase. When acting as an authenticator, certain devices (e.g. a NAS) do not necessarily have to understand each request type and may be able to simply act as a pass through agent for a 'backend' server on a host. The device only needs to look for the success/failure code to terminate the authentication phase.

**Main Disadvantages of PPP EAP:** PPP EAP requires the addition of a new authentication type to the LCP, and thus PPP implementations will need to be modified to use it. It also departs from the previous PPP authentication model of negotiating a specific authentication mechanism during LCP.

### 3.3.3 One-Time Password (OTP)

The One-Time Password (OTP) authentication system [28, 75, 133] provides authentication for system access (login), network access, and other applications requiring authentication. It is secure against passive attacks based on replaying captured reusable passwords. Such an attack can be performed by eavesdropping on network connections to obtain authentication information, such as the login identifiers (IDs) and passwords of legitimate users. Once this information is captured, it can be used at a later time to try to gain access to the system. OTP systems are designed to counter this type of attack, called a 'replay attack'.

The security of the OTP system is based on the non-invertibility of a secure hash function (see section 2.1.3.2). Such a function must be tractable to compute in the forward direction, but computationally infeasible to invert. The OTP authentication system uses a secret pass-phrase to generate a sequence of one-time (single use) passwords. With this system, the user's secret pass-phrase never needs to cross the network at any time, such as during authentication or

during pass-phrase changes. Thus, it is not vulnerable to replay attacks.

There are two entities in the OTP operation. The *generator*, which may be located at the remote entity requesting network access, must produce the appropriate one-time password from the user's secret pass-phrase and from information provided in the challenge from the server. The *server*, which may be located at the access network, must send a challenge that includes the appropriate generation parameters to the generator, must verify the one-time password received back from the generator, must store the last valid one-time password it received, and must store the corresponding one-time password sequence number. The server must also facilitate the changing of the user's secret pass-phrase in a secure manner.

In order to produce a one-time password, the generator passes the user's secret pass-phrase, along with a seed received from the server as part of the challenge, through multiple iterations of a secure hash function to produce a one-time password. After each successful authentication, the number of secure hash function iterations is reduced by one. Thus, a unique sequence of passwords is generated. The server verifies the one-time password received from the generator by computing the secure hash function once and comparing the result with the previously accepted one-time password[9].

**Main Advantage of OTP:** The OTP method protects the authentication system against passive eavesdropping and replay attacks. Added security is provided by the property that no secret information need be stored on any system, including the access network server being protected.

**Main Disadvantages of OTP:** The OTP system does not prevent a network eavesdropper from gaining access to private information. It also does not

[9]The server system has a database containing, for each user, the one-time password from the last successful authentication or the first OTP of a newly initialised sequence [75].

provide protection against either 'social engineering' or active attacks, such as Internet (TCP) session hijacking [75]. The use of IPsec (see section 3.6.5) is recommended to protect against TCP [157] session hijacking.

### 3.3.4 Generic Token Card (GTC)

Generic Token Card (GTC) [27, 28] is an EAP authentication method which was specifically defined for use with hardware authentication tokens that can generate dynamic user credentials. The request message contains a displayable message, which is shown in some way to the token holder. The token holder then enters the displayed information into the authentication token, which provides a response of some kind. This token response is then entered into the peer device, which uses it to construct a response message sent back to the authenticator. The EAP GTC method is intended for use with authentication tokens supporting challenge/response authentication, and must not be used to provide support for cleartext passwords in the absence of a protected tunnel with server authentication.

**Main Advantage of GTC:** The OTP and Generic Token Card methods provide protection against *dictionary* attacks[10]. Since the purpose of the OTP and Generic Token Card methods is to authenticate 'something the user has', neither method rests solely on a password, and so neither method is vulnerable to a dictionary attack, although passwords or PINs may be used to protect access to an authentication token.

**Main Disadvantages of GTC:** Both the OTP and Generic Token Card EAP methods provide one-way authentication, but do not support key generation

[10] A *dictionary* attack refers to the general technique of trying to guess a secret by running through a list of likely possibilities, often a list of words from a dictionary. It contrasts to a *brute-force* attack, in which all possibilities are tried (see http://en.wikipedia.org/wiki/Dictionary_attack).

[28]. As a result, the OTP and Generic Token Card methods, when used by themselves, are only appropriate for use on networks where physical security can be assumed. These methods should not be used on wireless networks, or over the Internet, unless the EAP conversation is protected. This can be accomplished using technologies such as TLS (see section 3.6.3) or IPsec (see section 3.6.5).

### 3.3.5 Addressing Legacy One-Way Authentication

There are a number of legacy one-way authentication methods, and we have reviewed some of them immediately above. However, because of the rapid proliferation of Internet remote access technologies, wireless devices and next generation service offerings, more secure and, most importantly, more flexible authentication mechanisms (i.e. mechanisms that are independent of underlying access technologies) are necessary. Since existing legacy one-way authentication solutions, e.g. CHAP carried by PPP authentication, possess a number of security deficiencies and are dependent on the access technologies, without such a new approach network providers will need either new transport mechanisms or extensions to existing legacy authentication mechanisms whenever a new access technology is introduced.

One approach to solving this problem would be to modify legacy authentication methods so as to enable key derivation, or to incorporate key material derived during the initial tunnel authentication. Nonetheless, since the motivation for continued use of legacy authentication technologies is to minimise the deployment of new technology, there does not seem any compelling logic to follow such an approach. This is because, in situations where deployment of a modified legacy method would be feasible, it would also normally be feasible to implement a wide range of alternatives. This could include the possible deployment of a new method supporting mutual authentication and key derivation, e.g. the re-use of solutions implemented in mobile systems in the Internet envi-

ronment [145], or the deployment of alternative technologies such as public key based authentication.

In the remainder of this chapter we summarise further existing authentication protocols relevant to this thesis, including the current EAP framework (now adapted for wired IEEE 802 networks, wireless LAN, and the Internet), mobile telecommunications methods, tunnelled authentication schemes, and public key based procedures. In Chapter 4 we describe both the problem space and a number of scenarios where existing authentication mechanisms are not sufficient. Finally, we argue that new, more secure and more flexible, authentication protocols are required.

## 3.4 EAP Architecture

The Extensible Authentication Protocol (EAP) is an authentication framework which supports multiple encapsulated authentication schemes, called EAP methods. An *EAP method* is thus an entity authentication algorithm carried by the EAP protocol, which provides one-way or mutual authentication between the communicating parties, and may also derive keying material. A 2007 Internet Draft [18] provides a framework for the generation, transport and use of keying material generated by EAP methods; it also specifies the EAP key hierarchy. A complete specification of the EAP architecture is given in RFC 3748 [13].

EAP typically runs directly over data link layer protocols, such as the PPP or IEEE 802, without requiring IP. EAP was designed for use in network access authentication, where IP layer connectivity may not be available. EAP may be used on dedicated links and switched circuits, and wired as well as wireless links.

In the following sections the current EAP architecture is summarised.

### 3.4.1 EAP Development

Whilst EAP was originally developed for use with PPP (section 3.3.2), it is now in use with a variety of lower layer protocols. In line with RFC 3748 [13], EAP was adapted for use on IEEE 802 wired media [84], IEEE wireless LANs [85, 171], and over the Internet [13]. It is important to observe that some legacy EAP protocols (e.g. PPP EAP-MD5) are susceptible to dictionary and brute-force attacks; do not provide confidentiality; do not support server authentication as required to prevent spoofing by rogue servers (gateways), and do not support the generation of keys in a way suitable for 802.11–2007 [85]. However, the current EAP framework allows the use of EAP methods which address these weaknesses.

### 3.4.2 EAP Basic Features

One of the main features of the EAP architecture is its flexibility [13]. EAP allows the protocol participants to select a specific authentication mechanism, typically after the authenticator requests more information in order to determine the authentication method to be used. Rather than requiring the authenticator to be updated to support each new authentication method, EAP permits the use of a backend authentication server, which implements some or all of the possible authentication methods, with the authenticator acting as a forwarding agent for some or all methods and peers.

EAP authentication is initiated by the server (authenticator), whereas many authentication protocols are initiated by the client (peer). As a result, it may be necessary for an authentication algorithm to add one or two additional messages (at most one round trip) in order to run over EAP.

EAP is a 'lock step' protocol, so that, other than the initial request, a new request cannot be sent prior to receiving a valid response[11]. As a result, EAP cannot efficiently transport bulk data.

### 3.4.3 EAP Exchange

The EAP authentication exchange proceeds as follows [13]:

- The authenticator sends a Request to authenticate the peer. The Request has a Type field to indicate what is being requested. Examples of Request Types include Identity, MD5-challenge, etc. Typically, the authenticator will send an initial Identity Request.
- The peer sends a Response packet in reply to a valid Request. The Response packet contains a Type field, which corresponds to the Type field of the Request.
- The authenticator sends an additional Request packet, and the peer replies with a Response. The sequence of Requests and Responses continues as long as needed.
- The conversation continues until either the authenticator cannot authenticate the peer (if unacceptable Responses have been received to one or more Requests), in which case the authenticator implementation transmits an EAP Failure, or the authenticator determines that successful authentication has occurred, in which case the authenticator transmits an EAP Success.

[11]The authenticator is responsible for retransmitting requests. After a suitable number of retransmissions, the authenticator ends the EAP conversation.

### 3.4.4 EAP Layers

EAP implementations consist of the following layers [13]:

- **Lower layer**. The lower layer is responsible for transmitting and receiving EAP frames between the peer and authenticator. EAP has been run over a variety of lower layers, including PPP [168], wired IEEE 802 LANs [84], IEEE 802.11 wireless LANs [85], and TCP [157].
- **EAP layer**. The EAP layer receives and transmits EAP packets via the lower layer, implements duplicate detection and retransmission, and delivers and receives EAP messages to and from the EAP peer and authenticator layers.
- **EAP peer and authenticator layers**. The EAP layer demultiplexes incoming EAP packets to the EAP peer and authenticator layers. Typically, an EAP implementation on a given host will support either peer or authenticator functionality, but it is possible for a host to act as both an EAP peer and an authenticator.
- **EAP method layers**. EAP methods implement the authentication algorithms and receive and transmit EAP messages via the EAP peer and authenticator layers. Since fragmentation support is not provided by EAP itself, this is the responsibility of EAP methods.

### 3.4.5 EAP Advantages and Disadvantages

The advantages and disadvantages of EAP can be summarised as follows:

**Advantages of EAP:** EAP can support multiple authentication mechanisms without having to pre-negotiate a particular such mechanism. Network Access

Server (NAS) devices (e.g., a switch or access point) do not have to understand every authentication method, and may act as a forwarding agent for a backend authentication server. Separation of the authenticator from the backend authentication server simplifies credential management and policy decision making.

**Disadvantages of EAP:** When used with PPP as the lower layer protocol, EAP requires the addition of a new authentication Type to the PPP LCP, and thus PPP implementations need to be modified to carry EAP. EAP also strays from the previous PPP authentication model in which a specific authentication mechanism is negotiated during LCP. Similarly, switch or access point implementations need to support IEEE 802.1X [84] in order to use EAP. Where the authenticator is separate from the backend authentication server, this complicates the security analysis and key distribution.

## 3.5 Mobile Authentication Methods

In this section a number of existing mobile telecommunications methods relevant to this thesis are reviewed, including GSM (section 3.5.1), GPRS (section 3.5.2), 3G/UMTS/AKA (section 3.5.3), 3G/GAA (section 3.5.4), and CDMA2000 (section 3.5.5). This will enable us to assess the suitability of the security solutions implemented in this domain for possible application in network (Internet) remote access.

### 3.5.1 Global System for Mobile Communications (GSM)

In this section an outline of the GSM[12] system security features is given, with a focus on the air interface protocol. An overview of the GSM system is presented, including a description of how the GSM security scheme operates (section 3.5.1.1), its main objectives (section 3.5.1.2) and the services that it provides (section 3.5.1.3).

#### 3.5.1.1 GSM System Overview

Figure 3.5 shows the GSM system components, which are described below. Further details of the GSM system can be found, for example, in [179].

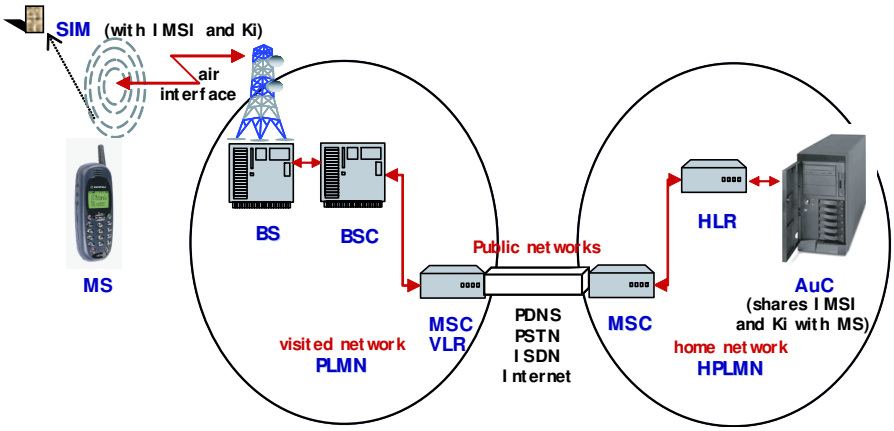


Figure 3.5: **GSM system overview**

- **Mobile Station (MS)**: this is made up of a Mobile Equipment (ME or 'mobile telephone') with its GSM Subscriber Identity Module (SIM). Through the SIM, each MS has a contractual relationship with a network,

[12]GSM was formerly an acronym for Groupe Spéciale Mobile (founded in 1982). It is now an acronym for Global System for Mobile communications (http://www.gsmworld.com). The GSM protocols have been standardised by the European Telecommunications Standards Institute (ETSI, http://www.etsi.org/).

called the home network, but may be allowed to roam in other visited networks when outside the home network coverage area.

- **International Mobile Subscriber Identity (IMSI) and Authentication Key ($K_i$)**: at the time the customer starts a subscription, the home network assigns the customer a unique and permanent identifier, the IMSI, together with a unique 128-bit secret key ($K_i$). Each customer's $K_i$ is also stored in an Authentication Centre (AuC) in the home network. The key $K_i$ plays two roles in GSM: *authentication*, in which the MS proves it possesses $K_i$, and *encryption*, which is performed with the use of a cipher key derived from $K_i$.

- **GSM Subscriber Identity Module (SIM)**: this is a smart card that must be inserted into the ME for service access. The IMSI and the authentication key $K_i$ of the MS are 'securely stored' in the SIM. In practice, the SIM is issued to the customer at the time the subscription is first taken out, and the customer never has access to the key $K_i$.

- **Public Land Mobile Network (PLMN) or Visited Network (VN)**: this is a network that is currently providing service to an MS, and the MS is said to be 'visiting' this network. An MS is registered with the PLMN which it is currently visiting. A PLMN contains, among others components, a collection of Base Stations (BSs) and a Visited Location Register (VLR).

- **Base Station (BS)**: this is a Base Transceiver Station belonging to a PLMN serving the MS. Base stations form a patchwork of radio cells over a given geographic coverage area. Base Stations are connected to base station controllers (BSCs).

- **Base Station Controller (BSC)**: this is a node controlling a number of BSs, coordinating handovers and performing BS co-ordination not related to switching. The BSC to BS link is, in many cases, a point to point mi-

crowave link. BSCs are also connected to mobile switching centres (MSCs) via fixed or microware links. MSCs are connected to public networks (e.g. PSTN, PDNS, ISDN and the Internet).

- **Visited Location Register (VLR)**: this is used to record information about all MSs 'visiting' a specific PLMN.
- **Home PLMN (HPLMN) or Home Network (HN)**: each MS has a home PLMN with which shares an IMSI and a secret key $K_i$. The HPLMN and the visited PLMN have a bilateral agreement, under which the visited PLMN trusts the HPLMN to pay for the services that the visited PLMN provides to the MS. Each HPLMN maintains a Home Location Register (HLR) and operates an AuC to support its MSs.
- **Home Location Register (HLR)**: this is used to record the most recent known location of all MSs belonging to a specific HPLMN.
- **Authentication Centre (AuC)**: this is used by a HPLMN to generate random challenges (RAND) and to store secret key information ($K_i$) relating to each of its MSs. The AuC can be integrated with other network functions, e.g. with the HLR.
- **Air Interface**: this is a synonym for the radio path between the BS and the MS. The MS 'visits' a PLMN by communicating with the serving BS across an air interface and receiving an entry in the VLR maintained by that PLMN.

#### 3.5.1.2 GSM Security Objectives

The main objectives of the security provisions built into the GSM system are "to make the system as secure as the public switched telephone network" (i.e. no more vulnerable to eavesdropping than fixed phones) [30], and to prevent

phone cloning[13]. Use of an air interface as the transmission medium gives rise to a number of potential threats because of the potential for eavesdropping. As stated in [145], "it was soon apparent in the threat analysis that the weakest part of the system was the radio path, as this can be easily intercepted". In fact, there was no attempt to provide security on the fixed network part of GSM. It should be noted that the GSM security system was designed with the following three constraints in mind [179]:

- To ensure that the level of confidentiality provided is not so high that it could cause export problems for the GSM system;
- GSM was not required to be resistant to 'active attacks' in which the attacker interferes with the operation of the system, perhaps masquerading as a system entity; and
- The trust between operators necessary for the operation of the security system should be minimised.

#### 3.5.1.3 GSM Security Services

In this section the three GSM air interface security services relevant here are reviewed, i.e. subscriber identity confidentiality, subscriber identity authentication and data confidentiality. Figure 3.6 illustrates the operation of these GSM security services. Further details of GSM security can be found, for example, in [135, 145, 179].

*Subscriber identity confidentiality* is achieved through the use of temporary identities. Apart from at initial registration, a user is not identified employing his permanent identity, i.e. his International Mobile Subscriber Identity (IMSI), but instead uses a temporary identity known as the Temporary Mobile

[13]Phone cloning occurs when someone with a scanner can eavesdrop on the communication between the mobile phone and the BS, and then make calls on that mobile phone's account [166, p113].

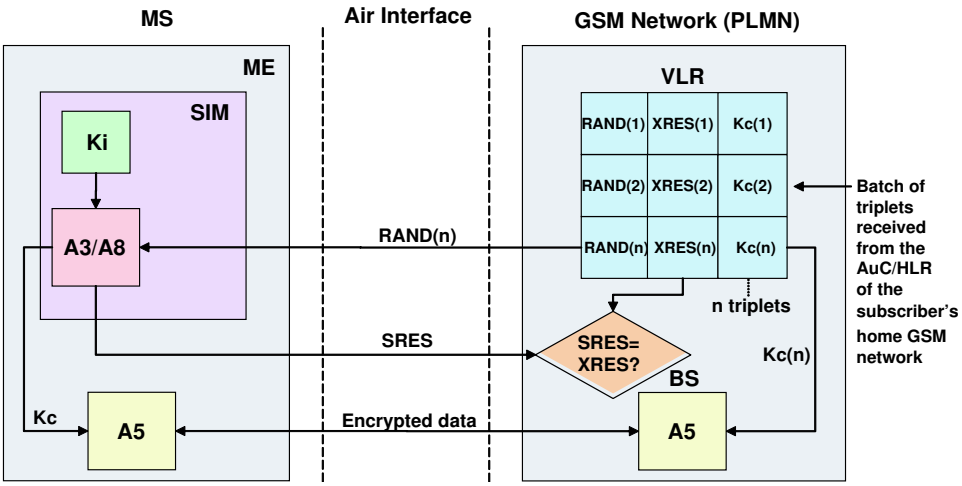


Figure 3.6: **Authentication and confidentiality for GSM**

Subscriber Identity (TMSI). The TMSI is only valid in a given location area, and thus it is always used together with the respective Location Area Identifier (LAI). The relationship between the TMSI and the IMSI is maintained by the VLR. To avoid user traceability, which may lead to the compromise of subscriber identity confidentiality, TMSIs are changed regularly by the visited network (VN) in an 'unlinkable' way. This unlinkability is supported by the fact that they are transmitted to the MS via an encrypted radio channel.

*Subscriber identity authentication* is used to authenticate the MS to the VN. This service is based on use of the secret key $K_i$, shared between the user's SIM and the AuC of the subscriber's HN. For each subscriber, and whenever necessary, the subscriber's HN selects one or more random challenge values *RAND*. Each *RAND* is input to a home network-specific MAC algorithm A3, along with the key $K_i$ for that subscriber, and the output is known as *XRES*. A set of pre-calculated (*RAND*, *XRES*) pairs are then supplied to the VN. Whenever the VN wishes to authenticate the MS, it sends it one of the *RAND* values. The MS inputs the *RAND* along with $K_i$ to algorithm A3, and sends back the output, known as *SRES*. The VN then compares the received *SRES* with the stored *XRES*, and if they agree the MS is deemed authentic.

As stated in [82], a variety of "subscriber related data is transferred over GSM/GPRS networks and needs to be protected". This data includes:

- Signalling information elements related to the user, such as the International Mobile Equipment Identity (IMEI), the IMSI, and the calling subscriber directory number;
- User information, such as short messages, transferred in a connectionless packet mode over a signalling channel, and voice and non-voice communications on traffic channels over the air interface.

In order to provide *data confidentiality* between the MS and the VN, a 64-bit encryption key $K_c$ is also produced at the same time as $XRES/SRES$ is generated[14], again as a function of $RAND$ and $K_i$, using a key generation algorithm A8. The key $K_c$ is passed from the AuC to the VN with the $RAND$ and $XRES$ values, as part of what is known as an 'authentication triplet' ($RAND$, $XRES$, $K_c$), and used as input to a stream cipher algorithm A5 to encrypt user and signalling data sent between the MS and the VN.

### 3.5.2 General Packet Radio Service (GPRS)

The General Packet Radio Service (GPRS) has been standardised by ETSI. In this section an overview of the security features of GPRS is provided. These services are similar to those provided by GSM, as described in section 3.5.1. We focus on the same three security features that we considered in section 3.5.1, i.e. subscriber identity confidentiality (section 3.5.2.1), subscriber identity authentication (section 3.5.2.2), and data confidentiality (section 3.5.2.3).

[14]In the network, the values of $K_c$ are calculated in the AuC/HLR simultaneously with the values for XRES. In the mobile, the current $K_c$ is stored in the mobile station until it is updated as part of the next authentication procedure.

#### 3.5.2.1 Subscriber Identity Confidentiality

As stated in [82], "the purpose of this function is to avoid an intruder... [identifying] a subscriber on the radio path". As mentioned in section 3.5.1.3, in GSM this is achieved by protecting the subscriber's IMSI using a temporary identity called a TMSI. A new TMSI is allocated as part of every location update. GPRS networks use a similar method based on a Temporary Logical Link Identity (TLLI) and a Routing Area Identity (RAI). Like the TMSI, the TLLI only has a meaning in a given Routing Area (RA), and so the TLLI is accompanied by the RAI to avoid ambiguity. The Serving GPRS Support Node (SGSN) maintains the relationship between TLLIs and IMSIs, in a similar way to the VLR in GSM.

#### 3.5.2.2 Subscriber Identity Authentication

According to [82], "the network can trigger this function for several reasons, including a subscriber applying for a change of a subscriber related information element in the VLR or HLR, a subscriber accessing a service (e.g. setting up a mobile originated or terminated call), or a cipher key mismatch".

The GPRS authentication procedure is handled in the same way as in GSM (see section 3.5.1.3), the main difference being that the procedures are executed in the SGSN, which requests the (*XRES, RAND*) pairs from the HLR/AuC.

#### 3.5.2.3 Data Confidentiality

GPRS data confidentiality is provided using an encryption method and a key establishment process directly analogous to those used by GSM. The main difference is that, in GPRS, encryption is applied at the logical link control layer and reaches further into the core network. Also, different encryption algorithms

are used.

### 3.5.3 Universal Mobile Telecommunications System (UMTS)

The Universal Mobile Telecommunication System (UMTS) has been developed by the Third Generation Partnership Project (3GPP)[15]. Third generation (3G) is the term used to describe the new generation of mobile services, currently being rolled out worldwide, which provide better quality voice and high-speed Internet and multimedia services.

In this section an overview of the 3GPP security architecture (section 3.5.3.1) and UMTS network access security services (section 3.5.3.2) is given.

#### 3.5.3.1 Third Generation (3GPP) Security

3G radio access link security employs a system developed from the GSM security scheme. 3GPP, that developed the 3G/UMTS standards, has adopted the security features from GSM that have proved to be robust, and tries to maximise architectural compatibility with GSM in order to ease inter-working and handover. 3G security [8] also tries to correct the problems identified in GSM by addressing security weaknesses and by adding new features. The 3G security scheme provides the following security features:

- mutual authentication and key agreement between MS and network;
- encryption of user traffic and signalling data over the air interface; and
- integrity protection of signalling data sent over the air interface.

The GSM security features retained and enhanced in 3GPP are, according to Walker and Wright [179]:

[15] http://www.3gpp.org/

- use of a smart card as a subscriber identity module (in the form of a UMTS SIM or USIM);
- authentication of the MS to the network;
- encryption of user traffic and signalling data sent over the air interface; and
- user identity confidentiality over the air interface.

The new security features required for 3GPP include mandatory integrity protection for critical signalling commands (e.g. for the start encryption command), which provides enhanced protection against false BS attacks (see [179]) by allowing the MS to check the authenticity of certain signalling messages. This feature also extends the influence of MS authentication when encryption is not applied, by allowing the VN to check the authenticity of certain signalling messages.

#### 3.5.3.2 UMTS Network Access Security: Authentication and Key Agreement (AKA)

UMTS network access security provides users with secure access to UMTS services, protecting in particular the UMTS radio access network (UTRAN). In this section, the four UMTS network access security features relevant here are summarised, i.e. entity authentication, signalling integrity, user traffic confidentiality, and user identity confidentiality. Further details of UMTS security can be found, for example, in [8, 26, 179].

**Entity Authentication** UMTS *mutual entity authentication* involves the Mobile Station (MS), the visited network (VN), and the home network (HN); the VN verifies the subscriber's identity by means of a challenge-response mechanism, while the MS checks that the VN has been authorised by the HN. A

128-bit secret key $K$ is shared by the Universal Subscriber Identity Module (USIM) and the HN AuC. An *authentication vector* is produced by the AuC from $K$ and a sequence number, and sent on demand to the VN. The authentication vector contains a random number $RAND$, a network authentication token $AUTN$, an expected result $XRES$, a temporary integrity key $IK$, and a temporary cipher key $CK$.

Whenever the VN wishes to authenticate the MS, it sends it the next unused ($RAND$, $AUTN$) pair. The MS verifies $AUTN$, using $K$ and the copy of the sequence number that it maintains. If this process is successful, the USIM sends $RES$, computed as a function of $K$ and $RAND$ using the UMTS message authentication function $f2$ [8], back to the VN. The USIM also inputs $K$ and $RAND$ to the UMTS key generating functions $f3$ and $f4$ [8] to obtain, respectively, $CK$ and $IK$. The VN then compares the received $RES$ with the stored $XRES$, and if they agree the MS is deemed authentic; $CK$ and $IK$ can then be used for connection security.

**Signalling Integrity** The UMTS mutual authentication and key agreement process provides enhanced protection against false BS attacks by allowing the MS to authenticate the VN. 3G authentication provides authentication of MS to VN and VN to MS, establishes a cipher key ($CK$) and an integrity key ($IK$), and gives assurance to the MS that the keys have not been used before. *Signalling data integrity and origin authentication* is provided by computing an integrity check using the 128-bit key $IK$, shared by the MS and the VN.

A new sequence number ($SQN$) is generated in the AuC and used as input to compute $AUTN$, and this latter is attached to the authentication vector (or 'quintet') to address the threat of 'quintet' re-use. The USIM verifies the freshness of a received $AUTN$ by checking that the sequence number $SQN$ used to compute $AUTN$ exceeds the most recently received such number. A MAC is also attached to show that the 'quintet' really came from the HN and to

integrity protect the attached *AUTN*.

**User Traffic Confidentiality** *User traffic confidentiality* is provided by encrypting traffic using the 128-bit key *CK*. The user traffic confidentiality feature extends to the Radio Network Controller (RNC).

Encryption of user traffic and signalling data sent over the air interface is performed using a block cipher called KASUMI (see section 2.1.3.2), which had an open design process, and takes a longer cipher key length (128 bits) than the GSM encryption algorithm. As stated above, the encryption terminates at the RNC, a 3G entity similar to the GSM BSC. The BS-RNC links, that may use microwave and thus be prone to interception, are encrypted. KASUMI is also used for the integrity protection of commands (critical signalling) between MS and RNC [179]. The 3G specifications introduce protection of network signalling information (including the authenticator vectors or 'quintets') transmitted between and within networks; if these networks were successfully attacked, then obtaining cleartext 'quintets' would enable an attacker to masquerade as valid MS. Further, to prevent false messages being introduced into the network, it is vital that the origin of such commands is authenticated.

**User Identity Confidentiality** Finally, UMTS provides *user identity confidentiality* through the use of temporary identities. Apart from at initial registration, a user is not identified employing his permanent identity, i.e. his IMSI, but instead uses a temporary identity known as the TMSI[16]. To avoid user traceability, which may lead to the compromise of user identity confidentiality, temporary identities are changed regularly in an 'unlinkable' way. In addition, it is required that any signalling or user data that might reveal the user identity is encrypted when sent across the UTRAN.

[16]It is TMSI in the circuit switched domain, and P-TMSI in the packet switched domain.

### 3.5.4 Generic Authentication Architecture (GAA)

The Generic Authentication Architecture (GAA) [10] has also been developed under the auspices of 3GPP (see section 3.5.3.1). In this section, an overview of GAA is provided (section 3.5.4.1), and the operation of the GAA scheme is described (section 3.5.4.2), including the mechanisms it uses to issue authentication credentials, using either shared secret (section 3.5.4.3) or digitally signed certificates (section 3.5.4.4). The potential advantages of GAA are then discussed (section 3.5.4.5). A complete specification of the GAA system is given in a 2007 3GPP Technical Report [10].

#### 3.5.4.1 GAA Overview

GAA is a generic architecture for peer authentication, which can, a priori, serve for any application, enabling cellular operators to extend the 3G authentication framework to support other (non-cellular) services. In other words, as stated by Laitinen et al. [124], "GAA is a general framework that allows the cellular authentication infrastructure used in authorising subscribers' access to the cellular network to be used in authorising access to new services". These services can be provided either by cellular network operators, or by third parties that have a business agreement with them.

According to Laitinen et al. [124], in GAA the mobile device and the service provider are automatically furnished with *fresh* credentials — an identifier and a shared key — after which they can authenticate each other. Credential provision, which requires a cellular authentication infrastructure, is performed over IP. Moreover, the mobile device possessing those credentials can be dynamically supplied with a subscriber certificate, and thus become part of a Public Key Infrastructure (see section 2.1.3.3).

Figure 3.7 shows the GAA system components, which are described below.

Further details of the GAA system can be found, for example, in [5, 6, 7, 10, 124].

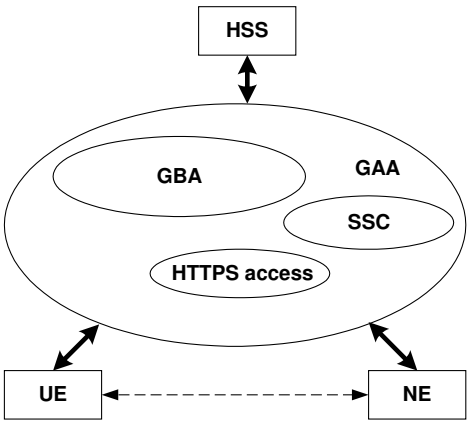


Figure 3.7: **GAA system overview**

- **User Equipment (UE)**: this is made up of a client device e.g. a mobile telephone with its subscriber's smart card (e.g. a USIM). Through the smart card, each UE has a contractual relationship with a network, called the home network, but may be allowed to roam in other visited networks when outside the home network coverage area. The GAA credentials are used between the UE and a network application server. As stated in section 4.2.4 of [7], the UE is required to support the HTTP Digest AKA protocol [138]. In addition, as discussed in section 4.3.3 of [6], the UE may have the capability to generate public and private key pairs (see section 2.1.3.3), protect the use of the private key (e.g. with a PIN), and store this key in non-volatile memory.

- **Home Subscriber System (HSS)**: each UE has an associated HSS, in which all the subscriber's security settings are stored. The HSS and the visited network have a bilateral agreement, under which the visited network trusts the HSS to pay for the services that the visited network provides to the UE. The general rule is that the UE always interacts

with its home network, and the resulting GAA credentials can be used with any network application server that has a relationship with the HSS. Each HSS operates an AuC to support its UEs.

- **Generic Bootstrapping Architecture (GBA)**: this is an application independent mechanism based on 3GPP AKA (see sections 3.5.3.1 and 3.5.3.2), that is used to provide a UE and a network application server with a common shared secret. This shared secret can be then used to authenticate communications between the UE and the application server. The mobile subscriber authentication procedure in GBA makes use of the HTTP Digest AKA protocol [138]. A complete description of the GBA mechanism is given in a 2007 3GPP Technical Specification (TS) [7].

- **Public Key Infrastructure (PKI) Portal**: this network element issues digitally signed public key certificates for UEs and operator CAs (see section 2.1.3.3). In both cases, certification requests and responses are protected by shared key material that has been previously established between the UE and a GAA functional element called the bootstrapping server function.

- **Support for Subscriber Certificates (SSC)**: this mechanism dynamically issues digitally signed certificates to mobile subscribers. Once a mobile subscriber has a key pair and has obtained a certificate for it, she can use them to produce digital signatures, as well as to authenticate herself to a network application server. In order to obtain a digital certificate, a UE sends a certificate request to a PKI portal of its HSS. This PKI portal, which plays the role of the application server, authenticates the UE's request. A complete description of the SSC mechanism is given in a 2007 3GPP TS [6].

- **Access to network application functions using HTTP over TLS (HTTPS)**: the HTTPS protocol [159] may be used in a variety of services to secure the application layer session between the UE and an application

server. When this occurs, a mechanism which makes use of a 'reverse proxy', called an authentication proxy (AP — described in section 6.5 of [10]) may be employed. A complete description of the authentication schemes that can be used with HTTPS is given in a 2006 3GPP TS [5].

- **Network Element (NE)**: a number of GAA functionalities are implemented in NEs, which may belong either to the visited network or to the HSS. As described in section 4.2 of [7], the set of GAA functionalities that are hosted in a NE includes, for instance, the bootstrapping server function (BSF), the network application function (NAF), the subscriber locator function (SLF), the Diameter proxy (Zn-Proxy), and the PKI portal. These GAA functionalities are discussed below.

#### 3.5.4.2 Operation of the GAA Scheme

In this section, a summary of the operation of the GAA scheme is provided. The GAA system [10] involves three main building blocks: GBA [7], SSC [6], and HTTPS access [5], which were briefly described in the previous section. GAA supports two types of authentication for mobile applications. One is based on a secret shared between the communicating entities, while the other is based on digitally signed public key certificates (see section 2.1.3.3). Figure 3.8 shows the two types of authentication mechanisms supported by GAA, which are described immediately below.

#### 3.5.4.3 GAA Authentication Mechanism Via Shared Secret

As previously stated, GBA [7] provides a mechanism based on 3GPP AKA (see sections 3.5.3.1 and 3.5.3.2) to install a shared secret between a UE and an application server. Figure 3.9 [7] illustrates the operation of this GBA bootstrapping network model, including the entities involved and the interfaces between them.

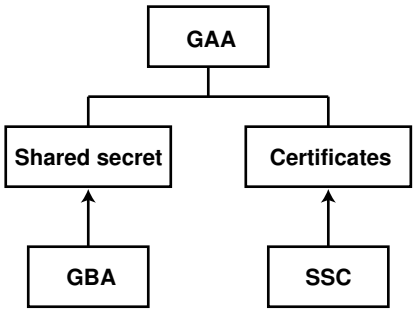


Figure 3.8: **GAA authentication mechanisms**

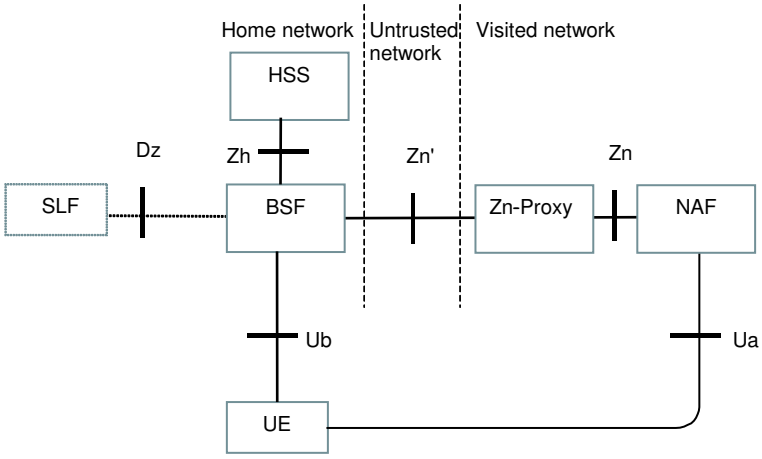


Figure 3.9: **GBA bootstrapping network model**

The GBA mechanism re-uses the 3GPP AKA scheme, introducing a new NE that implements the bootstrapping server function (BSF). The BSF has an interface (*Zh*) with the HSS, with which it performs the *credential fetching protocol*. This protocol is based on a Diameter application protocol (see section 3.9.2) given in a 2007 3GPP TS [2] and is used to fetch the required authentication information (i.e. authentication 'quintets' and GBA user security settings) from the home AuC in the HSS. The UE runs 3GPP AKA with the HSS via the BSF. The UE has an interface (*Ub*) with the BSF, across which the

*bootstrapping protocol* is executed, which is based on HTTP Digest AKA [138]. This protocol is used to support mutual authentication and key establishment. A shared session key is then established in the BSF and UE, derived from the ($CK$, $IK$) key pair established by this bootstrapping protocol.

Another NE, namely the network application server, implements the network application function (NAF). The NAF fetches the session key from the BSF, together with subscriber profile information (e.g. user security settings), via an interface ($Zn$) using the *key distribution protocol*. This is also based on a Diameter application protocol (see section 3.9.2). In the case where the UE has contacted a NAF that is operated in a visited network, this visited NAF uses a Diameter proxy (Zn-Proxy, described in section 4.2.2 of [7]) in its network to communicate, via an interface ($Zn'$), with the subscriber's home BSF.

The NAF and the UE will then share a secret key that can be used for application security, in particular for mutual authentication at the start of an application layer session. The use of GAA credentials between the UE and the NAF occurs via an interface ($Ua$) using the *application protocol*, which is secured using the keying material previously agreed via the interface ($Ub$) between the UE and the BSF. A variety of application protocols can be supported. For example, as stated in [124], 3GPP has provided GAA use profiles [1, 5], including for the HTTP Digest [68] and pre-shared key TLS [60] protocols.

Finally, the optional interface ($Dz$) between BSF and SLF is used to retrieve the address of the HSS, which maintains the user subscription.

#### 3.5.4.4 GAA Authentication Mechanism Based On Certificates

As described in the previous section, the SSC dynamically issues digitally signed public key certificates to mobile subscribers. Figure 3.10 [6] illustrates the operation of SSC, including the entities involved and the interfaces between them.

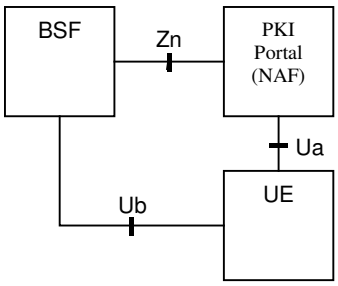


Figure 3.10: **SSC certificate issuing model**

To obtain a certificate, a UE sends a certificate request to a PKI portal of its HSS, which must authenticate that request. As stated before, this PKI portal plays the role of the application server, which implements the NAF. The UE must support an interface (*Ua*) with the PKI portal, using the *certification enrolment protocol*. This interface is protected using the shared keys previously established during the bootstrapping procedure.

The certificate enrolment process, i.e. the issuing of a certificate to a subscriber and the corresponding communication session between a UE and a PKI portal, requires authentication of the communicating entities. There are two options for this authentication process: use of a pre-shared secret or use of asymmetric cryptography and certificates. The latter is only used when a new certificate is requested from the PKI portal, and another valid certificate is already loaded in the UE. The former method requires a shared secret to be established between the PKI portal and the UE. If this shared secret is not pre-configured, the GBA mechanism [7] can be used to obtain it, through two interfaces (*Ub*, *Zn*) established, respectively, between the BSF and UE, and the BSF and the PKI portal. The BSF supports this by providing not only the authentication process, but the PKI portal specific user security settings.

Issuing a certificate to a mobile subscriber, which is described in [6], means

that the UE is loaded with a certificate corresponding to its (public, private) key pair. Once the certificate is in place, it can be used to authenticate the UE. The key pair and the corresponding digitally signed certificate can also be used for integrity protection or (less likely) confidentiality, but these are out of the scope of GAA.

#### 3.5.4.5 Potential advantages of GAA

According to Laitinen et al. [124], the GAA framework brings a number of potential advantages for *end users*. For instance, it is easy to add new services to the GAA architecture, since the creation of usernames and passwords during sign up is not necessary. Also, if the GAA customer makes use of multiple services, there is no need to maintain multiple passwords. It is also easy for the end user to switch mobile devices, as the access rights follow the smart card (e.g. a USIM) from which the GAA credentials are bootstrapped.

As stated in [124], the ability to authenticate mobile subscribers using the GAA architecture also creates a huge pool of potential customers for *service providers*, which do not need to supply their users with credentials. In addition, the GAA scheme provides a potentially strong authentication method, significantly improving on username/password methods. The GAA architecture also provides a potentially competitive advantage over other service providers, as it can offer, for example, integrated billing.

Finally, 3GPP GAA offers new business models for *cellular operators*, with which they can exploit existing assets, i.e. their subscriber base and roaming agreements.

### 3.5.5 Code Division Multiple Access 2000 (CDMA2000)

The Code Division Multiple Access 2000 (CDMA2000) wireless communications system is being developed by the Third Generation Partnership Project Two (3GPP2)[17]. CDMA2000, also known as IMT-CDMA Multi-Carrier or IS-2000, is the main path for CDMA operators to move from second generation (2G) to second-and-a-half (2.5G) and third generation (3G) cellular networks. 3GPP2 has created a set of standards that define the new air interface and specify radio access and core network changes that enhance network capacity, improve speed and bandwidth to mobile terminals, and will eventually allow end-to-end IP services.

CDMA2000 uses an identification and authentication system based on a combination of Mobile Identification Number (*MIN*) and Electronic Serial Number (*ESN*). It is intended that this scheme, when combined with CDMA2000 spread spectrum techniques, should make it very difficult for unauthorised users to intercept and decipher air interface traffic.

In this section, overviews of the evolution (section 3.5.5.1), security requirements (section 3.5.5.2), and security services (section 3.5.5.3) of this 3G mobile communication system are given.

#### 3.5.5.1 CDMA2000 Evolution

The International Telecommunication Union (ITU[18]), working with industry bodies from around the world, defines and approves technical requirements and standards as well as the use of spectrum for 3G systems under the International Mobile Telecommunication-2000 (IMT-2000) program. IMT-2000 is thus the global standard for 3G wireless communications, defined by a set of interdependent ITU recommendations, e.g. ITU-R M.1457 [106].

[17] http://www.3gpp2.org/
[18] http://www.itu.int/

The ITU requires that IMT-2000 (3G) networks deliver improved system capacity and spectrum efficiency over the 2G systems, and support data services at minimum transmission rates of 144 kbps in mobile (outdoor) and 2 Mbps in fixed (indoor) environments. Based on these requirements, in 1999 ITU approved five radio interfaces for IMT-2000 standards as a part of the ITU-R M.1457 recommendation [106]. CDMA2000 is one of the five standardised interfaces.

The CDMA2000 radio transmission technologies proposal meets the IMT-2000 requirements, while maintaining backward compatibility to what the industry terms 'cdmaOne', which is a complete family of standards. CDMA2000 is thus a technology for the evolution of cdmaOne/IS-95 to 3G services, which will provide enhanced services to CDMAOne subscribers, as well as forward and backward compatibility capabilities in terminals [106, 172].

CDMA2000 radio transmission technologies are being deployed in several phases. The first release, CDMA2000 1x, supports an average of 144 kbps packet data in a mobile environment. The second release of 1x, called 1x-EV-DO, will support data rates up to 2 Mbps on a dedicated data carrier. Finally, 1x-EV-DV will support even higher peak rates, simultaneous voice and high-speed data, as well as improved Quality of Service mechanisms.

Despite the existence of several releases, CDMA2000 1x is fully standardised under the auspices of 3GPP2 [172], and therefore all CDMA2000 1x networks which adhere to the standard are interoperable.

#### 3.5.5.2 CDMA2000 Security Requirements

As stated in the 3GPP2 vision document [172], security is an essential requirement for CDMA2000, which needs to be addressed not only for the air interface, but also for end-to-end service provisioning. The security features provided by

CDMA2000 must be flexible in order to give a level of security appropriate to the service/application being offered.

Consequently, the CDMA2000 specifications support a variety of wireless services using both voice and data. The security requirements of the CDMA2000 system protect service providers against fraud, and protect the privacy of system users. The following CDMA2000 security requirements are based on those given in the 3GPP2 vision document [172].

**Security in IP based networks.** Since the number of Internet applications implemented in mobile stations is expected to grow, and given that, in these cases, signalling data and user data may be sent via the same communications channel, IP security issues are critical. Therefore, threats to the Internet infrastructure will also become threats to future mobile environments, and thus security mechanisms similar to those provided in the wired Internet will be necessary.

**Scalable security architecture across all devices/spaces.** The need to ensure trust and confidentiality is independent of whether a device or system is connected to a wired network, a WLAN, wide-area cellular network, or any sort of hybrid network, or is simply a stand-alone device.

**Access security.** Future user authentication may include local authentication between a user and a terminal based on biometrics. These capabilities may complement the traditional user authentication methods.

**Seamless roaming across heterogeneous networks.** Seamless roaming between heterogeneous networks (e.g. wireless to wired) has typically been possible only if all the networks are controlled by the same entity. In a future world of heterogeneous networks, where seamless roaming between different types of access networks is possible, the trust and privacy equation becomes even more complex. This complexity requires the development of a scalable security architecture that can enable secure seamless

roaming in a true heterogeneous network.

**Support for certificate-based security.** Future applications (for example, m-commerce and m-transactions) will require certificate-based security. The MS must be able to support subscriber-based and server-based certificates.

**Content rights protection.** It is expected that a variety of multimedia content will be widely available to mobile devices, where the content copyright must be securely protected. Methods of protecting digital content will need to be developed.

**End-to-end security.** End-to-end application layer security (authentication, confidentiality and integrity) may be required independently of the underlying network architecture. Therefore it is expected that the network architecture and the underlying transport mechanisms will be transparent to application layer security mechanisms deployed to support end-to-end security.

**Security for short range interfaces.** Several short-range interface technologies, e.g. Bluetooth, are already available. In the near future, many applications may use these interfaces to provide connectivity between a mobile station and various external devices (e.g. display screen, external speakers, or pen interface). Therefore, security over those interfaces should be designed to give similar security levels to those provided for cellular technology.

**Robustness against potential attacks.** Network spamming has caused many problems in the Internet. Because of radio bandwidth limitations and airtime cost, this problem is magnified for a wireless network. DoS attacks and packet spoofing are becoming common both on the Internet and in mobile systems. The next generation 3GPP2 All-IP core network will try to provide an effective solution to minimise the threat posed by such

attacks.

#### 3.5.5.3 CDMA2000 1x Security Services

As described by Wingert and Naidu [183], CDMA2000 1x network security protocols rely on a 64-bit authentication key (*A-Key*) and the *ESN* of the mobile station. A random binary string called *RANDSSD*, which is generated in the HLR/AuC, also plays a role in the authentication procedures. The *A-Key* is programmed into the MS and stored in the home network AuC.

In addition to authentication, the *A-Key* is used to generate sub-keys for voice privacy and message encryption. CDMA2000 1x uses the standardised Cellular Authentication and Voice Encryption (CAVE) algorithm to generate a 128-bit sub-key called the 'Shared Secret Data' (*SSD*). The *A-Key*, the *ESN*, and the network-supplied *RANDSSD* are used as inputs to CAVE to generate the *SSD*. The *SSD* has two parts: *SSD_A* (64 bits), for authentication, and *SSD_B* (64 bits), used to generate keys to encrypt voice and signalling messages. The *SSD* can be shared with roaming service providers to allow local authentication. A fresh *SSD* can be generated when an MS returns to the HN or roams to a different system.

We next describe how CDMA 2000 1x implements three major mobile security features: authentication, data protection, and anonymity.

**Entity Authentication** As stated by Wingert and Naidu [183], in CDMA2000 1x networks the MS uses the *SSD_A* and the broadcast random number (*RAND*[19]) as inputs to the CAVE algorithm to generate an 18-bit authentication signature (*AUTH_SIGNATURE*), a type of MAC, and sends it to the base station. This signature is then used by the BS to verify that the subscriber is legitimate. Both

[19]The broadcast *RAND*, generated in the MSC, should not be confused with the *RANDSSD* from the HLR.

Global Challenge (where all MSs are challenged with the same random number) and Unique Challenge (where a specific *RAND* is used for each requesting MS) procedures are available to the operators for authentication. The Global Challenge method allows very rapid authentication. Also, both the MS and the network track the Call History Count (a 6-bit counter). This provides a way to detect cloning, as the operator is alerted if there is a mismatch.

The *A-Key* is re-programmable, but if it is changed both the MS and the network AuC must be updated. *A-Keys* may be programmed by any of the following: the factory; the dealer at the point of sale; subscribers via telephone; or via the air interface, i.e. so called over the air service provisioning (OTASP). OTASP transactions utilise a Diffie-Hellman key agreement algorithm (see section 2.1.3.3). The *A-Key* in the MS can be changed via OTASP, yielding an easy way to quickly cut off service to a cloned MS or initiate new services to a legitimate subscriber. As stated in [183], security of the *A-Key* is the most important component of the CDMA2000 security system.

**Voice, Signalling, and Data Confidentiality** As described in [183], the MS uses the *SSD_B* and the CAVE algorithm to generate a Private Long Code Mask (derived from an intermediate value called the Voice Privacy Mask), a Cellular Message Encryption Algorithm (CMEA) key (64 bits), and a Data Key (32 bits). The Private Long Code Mask is utilised in both the MS and the network to change the characteristics of a Long code. This modified Long code is used for voice scrambling, which adds an extra level of confidentiality over the CDMA2000 air interface. The Private Long Code Mask is not used to encrypt information; it simply replaces the well-known value used in the encoding of a CDMA2000 signal with a private value known only to the MS and the network. It is extremely difficult to eavesdrop on conversations without knowing the Private Long Code Mask.

Additionally, the MS and the network use the CMEA key with the Enhanced

CMEA (ECMEA) algorithm to encrypt signalling messages sent over the air interface and to decrypt the information received. A separate data key, and an encryption algorithm called ORYX [183], are used by the MS and the network to encrypt and decrypt data traffic on the CDMA2000 channels.

**Anonymity** CDMA2000 systems support the assignment of a TMSI to a MS, which is used in communications to and from a particular MS over the air interface. This feature is handled in the same way as in UMTS (see section 3.5.3.2), making it more difficult to correlate a mobile user's transmission to a user identity, i.e. avoiding user traceability, which may lead to the compromise of user identity confidentiality.

**CDMA2000 Further Releases** Further releases of 3G CDMA2000 technologies add more security protocols, including the use of 128-bit privacy and authentication keys. For CDMA2000 networks, as described by Wingert and Naidu [183], new algorithms such as SHA-1 (see section 2.1.3.2) are used for integrity protection, and AES (see section 2.1.3.2) for message encryption. The AKA protocol (see section 3.5.3.2) will be used for all releases following CDMA2000 release C. The AKA protocol will also be used in WCDMA-MAP networks, along with the Kasumi algorithm (see section 2.1.3.2) for encryption and message integrity.

## 3.6 Cryptographic Tunnelling and Key Generation

This section summarises a number of existing tunnelling and key generation schemes relevant to this thesis, including ISAKMP (section 3.6.1), IKE (section 3.6.2), TLS (section 3.6.3), WTLS (section 3.6.4), IPsec (section 3.6.5), EAP

Key Derivation for Multiple Applications (section 3.6.6), and EAP-PSK (section 3.6.7). These are reviewed to assess whether they are suitable for application in network (Internet) remote access.

### 3.6.1 Internet Security Association and Key Management Protocol (ISAKMP)

The Internet Security Association and Key Management Protocol (ISAKMP) [49, 131] defines procedures and packet formats to establish, negotiate, modify and delete Security Associations (SAs). SAs contain all the information required for execution of various network security services, such as IP layer services (including header authentication and payload encapsulation), transport or application layer services, or self-protection of negotiation traffic.

ISAKMP defines payloads for exchanging key generation and authentication data. These formats provide a consistent framework for transferring key and authentication data, in a way that is independent of the key generation technique, encryption algorithm and authentication mechanism.

ISAKMP is kept distinct from specific key exchange protocols in order to cleanly separate the details of security association management from the details of key exchange. There are many different key exchange protocols, each with different security properties. However, a common framework is required for agreeing to the format of SA attributes, and for negotiating, modifying, and deleting SAs. As described in RFCs 2408 [131] and 4306 [49], ISAKMP serves as this common framework.

### 3.6.2 Internet Key Exchange (IKE)

IKE is the default authentication and key exchange protocol used for creating IPsec security associations (see section 3.6.5). IKE operates in two phases: IKE Phase I (described in section 3.6.2.1) establishes an ISAKMP security association (section 3.6.1), which is then used to secure IPsec SA negotiation in IKE Phase II (section 3.6.2.2). More details on IKE can be found in RFCs 2409 [76] and 4109 [79]. Certain issues involving IKE, such as the resolution of certain known security defects, are addressed by IKE version 2 (IKEv2 — described in section 3.8.1), which has received a considerable amount of expert review.

#### 3.6.2.1 IKE Phase I: Session Key derivation for the ISAKMP SA

In IKE Phase I, an ISAKMP security association (section 3.6.1) can be established in two main ways. IKE additionally offers four authentication modes, for each of which the session key derivation technique is different. The pseudo-random functions employed for key derivation are negotiated in both IKE Phase I and IKE Phase II.

#### 3.6.2.2 IKE Phase II: Session Key derivation for the IPsec SA

In IKE Phase II, an IPsec security association (see section 3.6.5) is derived from the keying material previously computed for the ISAKMP SA (IKE Phase I). The precise nature of the keying material derived from IKE Phase II depends on whether or not perfect forward secrecy is required (see section 2.1.3.3).

### 3.6.3 Transport Layer Security Protocol (TLS)

The Transport Layer Security (TLS) protocol [44] provides security functions for data sent over the Internet. It achieves this by securing data traffic sent over

a *reliable transport* protocol, i.e. a transport protocol which includes a non-cryptographic message integrity check to protect against accidental, as opposed to deliberate, errors in transmission (e.g. TCP [157]). Within the protocol hierarchy, TLS is located underneath the application layer and on top of the transport layer. It provides entity authentication, data authentication, and data confidentiality, allowing client/server applications to communicate in a way that is designed to prevent eavesdropping, tampering, or message forgery. The TLS protocol provides both unilateral and mutual authentication, with session key establishment based on the use of public key certificates.

TLS can be used in order to create an authenticated tunnel to secure the communications of any application, not just between a web browser and server. The IETF TLS working group[20] is working on a second, enhanced version of the current TLS 1.0 specification.

#### 3.6.3.1 TLS and SSL

TLS is the IETF standardised version of the Secure Sockets Layer (SSL) protocol published by Netscape [69]. Since the description of TLS differs only in minor ways from the SSL specification, in this thesis we subsequently always refer to TLS.

#### 3.6.3.2 TLS Subprotocols

In accordance with RFC 2246 [44], the TLS protocol consists of two main subprotocols. The TLS Record subprotocol provides protection of the application data exchanged between two entities. The security association, including the session key required for the TLS Record subprotocol, is provided by the TLS Handshake subprotocol, which provides authentication and session key estab-

[20] http://www.ietf.org/html.charters/tls-charter.html

lishment.

**TLS Record subprotocol** At the lowest level, running over some reliable transport protocol (e.g. TCP [157]), is the TLS Record subprotocol, which provides two connection security properties:

- **The connection is private**. The data is encrypted using a symmetric encryption algorithm. The secret encryption keys are generated uniquely for each connection, and are based on a secret negotiated by another the TLS Handshake subprotocol.
- **The connection is reliable**. Message transport includes a message integrity check using a MAC.

**TLS Handshake subprotocol** The TLS Record subprotocol is used for encapsulation of various higher level protocols. One such encapsulated protocol, the TLS Handshake subprotocol, allows the server and client to authenticate each other and to negotiate an encryption algorithm and cryptographic keys before the application protocol transmits or receives its first byte of data; i.e., the TLS Handshake subprotocol is responsible for negotiating a session. Many connections can be instantiated using the same session through a resumption feature in the TLS Handshake subprotocol.

The TLS Handshake subprotocol provides three basic security properties:

- The peer's identity can be authenticated using *asymmetric cryptography* (see section 2.1.3.3).
- The negotiation of a shared secret is *secure*, i.e. the negotiated secret is unavailable to eavesdroppers, including active attackers.
- The negotiation is *reliable*, i.e. no attacker can modify the negotiation

communication without being detected by the legitimate parties to the communication.

#### 3.6.3.3 TLS Advantage and Goals

One advantage of TLS is that it is application protocol independent. Higher level protocols can execute on top of the TLS Protocol transparently.

The goals of the TLS Protocol are:

- **Cryptographic security**: TLS is used to establish a secure connection between two parties.
- **Interoperability**: Independent programmers are able to develop applications using TLS that will then be able to successfully exchange cryptographic parameters.
- **Extensibility**: TLS seeks to provide a framework into which new asymmetric and symmetric encryption methods can be incorporated as necessary.
- **Relative efficiency**: Cryptographic operations tend to be highly CPU intensive, particularly public key operations. For this reason, the TLS protocol has incorporated an optional session caching scheme to reduce the number of connections that need to be established from scratch.

### 3.6.4 Wireless Transport Layer Security Protocol (WTLS)

The Wireless Application Protocol (WAP) is a protocol stack for wireless communication networks, specified by the WAP Forum[21], which has become part of the Open Mobile Alliance (OMA). WAP is essentially a wireless equivalent of the Internet protocol stack.

[21] www.wapforum.org

Wireless Transport Layer Security (WTLS) [181] is the security layer for WAP applications. A complete specification of the WTLS protocol is given in [181]. Based on TLS, WTLS was developed to address certain limitations of mobile devices — such as limited processing power and memory capacity, and low bandwidth — and to provide adequate authentication, data integrity, and privacy protection mechanisms.

Because mobile networks do not provide end-to-end security, TLS had to be modified to address the special needs of wireless users. Designed to support datagrams in a high latency, low bandwidth environment, WTLS provides an optimised handshake through dynamic key refreshing, which allows encryption keys to be regularly updated during a secure session.

### 3.6.5 IPsec

The IP security (IPsec) protocol [118] is designed to provide interoperable, high quality security for IPv4 and IPv6 data flows. A brief description of the IPsec security framework is now provided. A complete specification of the IPsec protocol is given in RFC 4301 [118].

#### 3.6.5.1 IPsec Security Services

IPsec provides security services at the IP layer by enabling a system to select the required security protocols, determine the algorithms to be used to provide the services, and put in place any cryptographic keys required to provide the requested services[22].

The set of security services offered includes: access control, connectionless integrity, data origin authentication, protection against replays, confidentiality,

[22]IPsec can be used to protect one or more 'paths' between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.

and limited traffic flow confidentiality. These services are provided at the IP layer, offering protection for IP and/or upper layer protocols.

#### 3.6.5.2 ESP and AH

As stated in RFC 4301 [118], IPsec comprises two traffic security protocols: the *Encapsulating Security Payload* (*ESP*) [117] provides confidentiality and origin authentication functions to a data packet, and the *Authentication header* (*AH*) [116] provides origin authentication functions to a data packet. Both AH and ESP are vehicles for access control, based on the distribution of cryptographic keys and the management of traffic flows relative to these security protocols.

#### 3.6.5.3 IPsec SA

The concept of a 'Security Association' (SA) is fundamental to IPsec. Both AH and ESP make use of SAs, and a major function of IKE (see section 3.6.2) is the establishment and maintenance of security associations. A SA is 'a simplex connection that affords security services to the traffic carried by it' [118].

Security services are provided to the entities sharing an SA by the use of AH, or ESP, but not both[23]. A security association is uniquely identified by a triple consisting of a Security Parameter Index (SPI), an IP Destination Address, and a security protocol (AH or ESP) identifier.

#### 3.6.5.4 Tunnel and Transport Modes of Operation

Both the AH and the ESP protocols have two distinct modes of operation: *Tunnel mode* completely encapsulates the original packet within another IP

[23]If both AH and ESP protection is applied to a traffic stream, then two (or more) SAs need to be created. Moreover, to secure typical, bi-directional communication between two hosts, or between two security gateways, two SAs (one in each direction) are required.

header, and *Transport mode* keeps the original header and does not add the extra IP header.

**Tunnel Mode (DHCP IPsec)** In tunnel mode, the AH and ESP protocols are applied to tunnelled IP packets. Tunnel mode creates a second IP header in the packet, and uses both the local and remote security gateway addresses as source and destination IP addresses. Also, tunnel mode allows an instance of IP to run immediately above the IPsec layer. RFC 3456 [152] explores the requirements for host configuration in IPsec tunnel mode, and describes how the Dynamic Host Configuration Protocol (DHCP [46, 47]) may be used to support this configuration.

**Transport Mode** In transport mode, the AH and ESP protocols provide protection primarily for upper layer protocols. Transport mode does not add a second IP header and does not permit an instance of the IP protocol to be implemented above it in the protocol hierarchy. Instead, tunnel mode allows other tunnelling applications (e.g. L2TP tunnel [174]) to be run over an IPsec transport mode connection.

#### 3.6.5.5 Tunnel Mode SA

As described in RFC 4301 [118], a tunnel mode SA is essentially an SA applied to an IP tunnel. If either end of a security association is a security gateway, then the SA needs to be in tunnel mode. For a tunnel mode SA, the protocol which has an associated SA possesses an 'outer' IP header that specifies the IPsec processing destination, plus an 'inner' IP header that specifies the ultimate destination for the packet. The security protocol header appears after the outer IP header, and before the inner IP header. If the AH protocol is employed in tunnel mode, portions of the outer IP header are afforded protection, as well as

all of the tunnelled IP packet. If the ESP protocol is employed, the protection is afforded only to the tunnelled packet, not to the outer header.

### 3.6.6 EAP Key Derivation for Multiple Applications

Some EAP methods generate keying material shared by the EAP peers (see section 3.4); these keys can be used, for instance, with IEEE 802.11 [85] encryption. As described in [13], an EAP method typically produces a Master Session Key (MSK), which is sent by the EAP server to the authenticator. The authenticator then uses the MSK to derive Transient Session Keys (TSKs), which are used to protect the actual communication path. In addition, an EAP method may internally use some keys, known as Transient EAP Keys (TEKs), to protect its communication path. A complete specification of the generation, hierarchy, transport and use of EAP keying material is given in a 2007 Internet Draft [18].

The EAP protocol (see section 3.4) also defines an Extended Master Session Key (EMSK), which may be used to derive keys for multiple applications, such as protecting EAP messages, distributing credentials for re-authentication, or handoff mechanisms involving multiple WLAN access points [71]. In this case, it is desirable that such keys are cryptographically separate, i.e. knowledge of one key does not give any information about other keys. Cryptographic separation between different applications requires that the derivation of TSKs is coordinated.

In a 2003 Internet Draft [165], Salowey and Eronen proposed a mechanism to derive cryptographically separate keys for multiple applications independent of the EAP method in use. The Salowey-Eronen mechanism specifies a way of coordinating these key derivations using a key derivation function, which takes as input the EMSK described above, an application key label, and optional application data, and returns a multiple application master session key (AMSK). These AMSKs are then used to derive TSKs, which are used to actually protect

the data, e.g. to encrypt it.

### 3.6.7 EAP-PSK

Bersani and Tschöfenig [25] specified the EAP-Pre-Shared Key (EAP-PSK) protocol, an EAP method for mutual authentication and session key derivation, which uses a 16-byte pre-shared key (PSK) as its long term credential and relies on a single cryptographic primitive, i.e. AES-128 (see section 2.1.3.2). EAP-PSK was inspired by the EAP-Archie proposal [178], which is now abandoned. A complete specification of the EAP-PSK protocol is given in RFC 4764 [25]. As described in the EAP-PSK draft, a *pre-shared key* means a secret key (see section 2.1.3.2) which is derived by some prior mechanism and shared between the parties before the protocol using it takes place. It is simply a bit sequence of a given length, each bit of which has been chosen uniformly and independently at random.

When mutual authentication is successful, EAP-PSK provides a protected communications channel for the authenticated parties; it is designed for authentication over insecure networks, such as IEEE 802.11 [85].

EAP-PSK assumes that the PSK is only shared between the EAP peer and the EAP server. The PSK is used to derive two 16-byte subkeys, called the Authentication Key (AK) and the Key-Derivation Key (KDK). The AK is used to mutually authenticate the EAP peer and the EAP server, and the KDK is used to derive session keys shared by the EAP peer and the EAP server (namely, the TEK, MSK and EMSK).

EAP-PSK is made up of three protocols: a key setup protocol to derive the AK and KDK from the PSK, an authenticated key exchange protocol to mutually authenticate the communicating parties and derive session keys, and a protected channel protocol for the mutually authenticated parties to use for

data communications.

## 3.7 Compound Tunnelled Authentication Protocols

One of the main motivations behind introducing two-step (tunnelled) authentication protocols as allowed by EAP was to support the use of legacy authentication protocols and existing authentication key management infrastructures. Since its deployment, a number of weaknesses in EAP have become apparent. These include the lack of user identity confidentiality, integrity protection for the EAP negotiation, and a standardised mechanism for key exchange [149].

One of the main purposes of recent work in certain IETF working groups has been to fix these perceived weaknesses of EAP, while still retaining the primary benefit of EAP encapsulation: namely a standard interface between the inner client authentication protocol and the outer authentication protocol allowing support for multiple existing remote authentication protocols. This section summarises a number of recently proposed compound tunnelled authentication schemes relevant to this thesis, including XAUTH (section 3.7.1), PIC (section 3.7.2), PEAP (section 3.7.3), EAP-TTLS (section 3.7.4), PANA (section 3.7.5), PANATLS (section 3.7.6), and SeNAA (section 3.7.7). We present them here in order that we can subsequently consider their further application for network (Internet) remote access.

### 3.7.1 Extended Authentication within ISAKMP/Oakley (XAUTH)

The IKE protocol (see section 3.6.2) allows a device to set up a secure session by means of a bidirectional authentication method using either pre-shared keys (see

section 3.6.7) or digital certificates. However, IKE does not provide a method to exploit legacy authentication methods. The Extended Authentication scheme within ISAKMP/Oakley (XAUTH) [24] is a method for using existing unidirectional authentication mechanisms such as RADIUS (section 3.9.1), SecurID [111], and OTP (section 3.3.3) within IPsec's ISAKMP (section 3.6.1) protocol. A complete specification of the XAUTH technique is given in a 2001 Internet Draft [24].

The purpose of XAUTH is not to replace or enhance the existing authentication mechanisms described in IKE, but rather to allow them to be extended using legacy authentication mechanisms. As stated in [24], the XAUTH technique allows the IPsec ISAKMP/Oakley [76] protocol to support additional authentication mechanisms such as two-factor authentication, challenge/response and other remote access unidirectional authentication methods.

The XAUTH protocol is designed in such a way that extended authentication may be accomplished using any mode of operation for IKE phase I (i.e. Main Mode or Aggressive Mode) as well as any authentication method supported by IKE. This protocol may also be easily extended to support new modes or authentication methods.

### 3.7.2 Pre-IKE Credential (PIC) Provisioning Protocol

The Pre-IKE Credential (PIC) Provisioning Protocol [16] is a means of bootstrapping IPsec authentication via an 'Authentication Server' (AS) and user authentication mechanisms. A complete specification of the PIC protocol is given in a 2002 Internet Draft [16]. As described in the PIC draft, the client machine communicates with the AS using a key exchange protocol, where only the server is authenticated. The session keys derived as a result of this process are used to protect the user authentication protocol conducted between the client and the 'backend authentication server'. Once the user is authenticated,

the client machine obtains credentials from the AS that can be used later to authenticate the client.

PIC embeds EAP messages (see section 3.4) in ISAKMP payloads (see section 3.6.1) to support multiple forms of user authentication. If this user authentication succeeds, the client machine can request and obtain credentials from the AS[24]. The credentials are intended to be used by the client to perform regular IKE authentication with an IPsec-enabled security gateway.

The PIC protocol is defined between the Client and the AS. The PIC draft [16] describes the four main stages of the proposed PIC protocol as follows:

1. An optional round of messages provides partial protection of the AS against DoS attacks, by verifying that the initiator of the exchange is reachable at the purported source IP address.

2. The protocol establishes a one-way authenticated channel from the client to the AS, in which only the server is authenticated.

3. User authentication is performed over this secured channel. User authentication information is transported using EAP tunnelled within ISAKMP.

4. The AS sends the client a credential which can be used in subsequent IKE exchanges. This credential can be thought of as a certificate, or as a private key generated or stored by the AS and accompanied by a corresponding certificate. It may also be a secret key, or other information for deriving such a key.

In stage 4 the created ISAKMP tunnel is used for the secure provisioning of credentials for successfully authenticated users.

[24]The term 'credentials' is used here to mean both digital certificates and shared secret keys.

### 3.7.3 Protected EAP Protocol (PEAP)

Protected EAP (PEAP) [149] provides wrapping of the EAP protocol (see section 3.4) within TLS (see section 3.6.3). It claims to provide user anonymity and built-in support for key exchange. A complete specification of the PEAP protocol is given in a 2004 Internet Draft [149]. The relationship among the EAP peer (client), the front-end authenticator, known as the 'network access server' (NAS) in PEAP, and an authentication agent, known as the 'backend authentication server' in PEAP, is depicted in Figure 3.11.

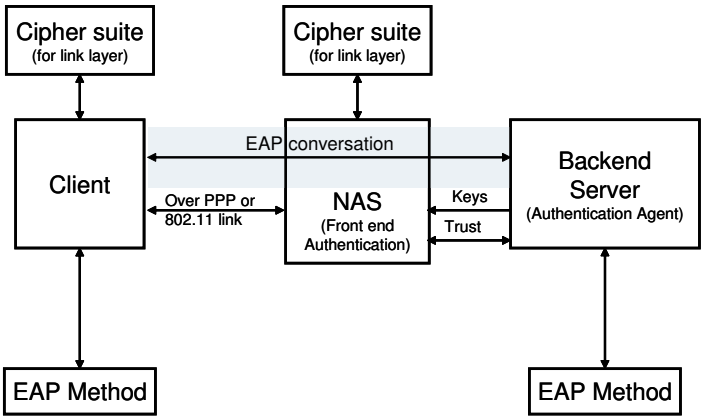


Figure 3.11: **Relationship among EAP client, backend authentication server, and NAS in PEAP**

As shown in Figure 3.11, the EAP conversation 'passes through' the NAS on its way between the client and the backend authentication server. While the authentication conversation is between the EAP client and the backend authentication server, the NAS and the backend authentication server need to establish trust for the conversation to proceed. I.e. in the case where the NAS and EAP server reside on separate machines, they both need to establish trust in each other beforehand; this is required to prevent spoofing by rogue servers (gateways).

The client and the backend server first set up a TLS channel over EAP. The client authentication protocol between the client and the backend server is encrypted and integrity protected within this TLS channel. As a result, the NAS does not have knowledge of the TLS master secret established between the client and the backend authentication server, and cannot decrypt the PEAP conversation.

The backend server derives master session keys from the TLS master secret using a one-way function, and conveys them to the NAS; the NAS can then use these session keys to protect subsequent link layer communications between it and the client. The PEAP draft [149] does not discuss the format of the attributes used to communicate the master session keys from the backend authentication server to the NAS. AAA carrier protocols such as RADIUS (see section 3.9.1) can be used for this purpose.

### 3.7.4 EAP Tunnelled TLS Authentication Protocol (EAP-TTLS)

The EAP Tunnelled TLS Authentication Protocol (EAP-TTLS) [70] claims to allow legacy password-based authentication protocols to be used with existing authentication databases, while protecting the security of these legacy protocols against eavesdropping, MitM (see section 3.2.3) and other cryptographic attacks. A complete specification of the EAP-TTLS protocol is given in a 2004 Internet Draft [70].

EAP-TTLS also allows the client and the backend server to establish keying material for use in the data connection between the client and the front-end authenticator. The keying material is established implicitly between the client and the backend server based on the TLS handshake (see section 3.6.3). EAP-TTLS derives sessions keys by applying a pseudo-random function to the TLS

master secrets and other input. The backend server distributes derived session keys to the front-end authenticator using the AAA protocol (see section 3.9). The client derives the same keys in parallel.

### 3.7.5 Protocol for Carrying Authentication for Network Access (PANA)

This section briefly introduces the draft PANA protocol [65], a link layer agnostic transport for EAP to enable client-to-network access authentication. Chapter 6 describes the PANA protocol in more detail. A complete specification of the PANA protocol is given in a 2005 Internet Draft [65].

PANA is designed for use between a PANA Client (PaC) and a PANA Authentication Agent (PAA) situated in the access network, where the PAA may optionally be a client of an AAA infrastructure (see section 3.9). A complete specification of the interworking of PANA with IETF AAA protocols (e.g. the Diameter EAP protocol — see section 3.9.3) is given in a 2005 Internet Draft [125]. This specification contains, for instance, a table with a PANA-Diameter message mapping (see section 5 of [125]).

PANA can carry any authentication mechanism that can be specified as an EAP method (see section 3.4), and can be used on any link that supports IP. The PANA protocol specification is designed to provide the client-to-network access authentication component within an overall secure network access framework, which would also need to include protocols and mechanisms for service provisioning, access control as a result of initial authentication, and accounting.

The payload of a PANA message consists of a (possibly empty) sequence of Attribute Value Pairs (AVPs), e.g. a Cookie AVP, used for making an initial handshake robust against 'blind resource consumption DoS attacks' (see section 3.2.3), a MAC AVP, protecting the integrity of a PANA message, or an EAP-

Payload AVP, which transports an EAP payload. PANA uses UDP [156] as its transport layer protocol, and sequence numbers to provide ordered delivery of EAP packets. A summary of the PANA header format is shown in Figure 3.12.

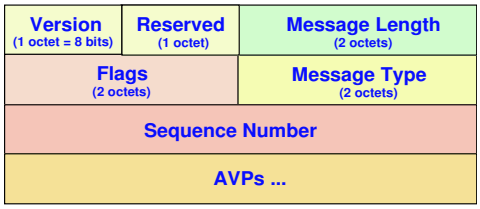


Figure 3.12: **PANA header format**

Two important features of PANA, namely the security association (SA) and the re-authentication procedure, are now described. Once the EAP method has completed, a session key is shared by the PaC and the PAA. The session key is provided to the PaC as part of the EAP key exchange process, and the PAA can obtain the session key from the EAP server via the AAA infrastructure (if used). PANA SA establishment based on the EAP session key is required where no physical or link layer security is available.

The re-authentication procedure extends the current PANA session lifetime by re-executing the EAP method. Re-authentication of an on-going PANA session must maintain the existing sequence numbers. In an instance of the re-authentication procedure, if there is an existing PANA SA, both PANA-Auth-Request/Answer messages are protected with a MAC AVP.

The whole of this thesis is based on one particular draft of the PANA specification [65]. Working on one particular draft has been necessary because it is a work in progress and changes relatively frequently.

### 3.7.6 PANA over TLS (PANATLS)

The Protocol for carrying Authentication for Network Access over Transport Layer Security (PANATLS) 2002 Internet Draft [143] specifies a method to carry authentication information over a TLS protected channel (see section 3.6.3) between a PaC and a PAA, both of which are on the same subnet [184]. PANATLS uses the TLS protocol to provide a secure means of exchanging authentication information.

The purpose of the PANATLS method is not only to provide a mechanism for carrying the authentication parameters, but also to address certain outstanding issues, e.g. re-authentication, security threats, etc. In particular, the security features provided by TLS are important for giving confidentiality and/or integrity protection for the entire authentication protocol exchange, including confidentiality for the identity of the client as well as the authentication result (e.g. EAP-Success/Failure). Such protection is not provided by other authentication protocols such as EAP (see section 3.4).

PANATLS is designed to carry any authentication protocol information, including EAP messages. It is also possible to use a TLS certificate for authenticating a PaC, without using any other authentication protocol. PANATLS supports the combination of multiple types of authentication to authenticate a PaC. For example, it is possible to use a TLS client certificate to authenticate the IP address of the PaC, and then to use EAP to authenticate the user of the PaC.

It is possible to launch MiTM attacks against PANATLS, typically with the objective of falsifying the authentication process (see sections 3.2.3 and 6.2.1.2). In order to prevent such attacks, it is necessary to create a binding between the security association established between the PAA and the PaC (i.e. the TLS session) and any state that is established based on the information carried

inside or outside of TLS. Therefore MitM attacks against one (or both) of the communicating entities can be prevented by PANATLS, because this protocol supports the creation of cryptographic binding between the PaC device identifier and the TLS session (see section 4.8.1 of [143]), and the EAP authentication session and the TLS session (see section 4.8.2 of [143]).

### 3.7.7 Secure Network Access Authentication (SeNAA)

The Secure Network Access Authentication (SeNAA) 2002 Internet Draft [67] describes how the reliable SeNAA protocol over UDP [156] can be used to carry TLS protocol exchanges (see section 3.6.3) inside a TLS payload. SeNAA messages are formatted in the same way as Diameter messages (see section 3.9.2), and they contain AVPs. In the SeNAA messages, almost all AVPs (with a few exceptions, e.g. the Session-Id AVP) are carried in a TLS payload and protected by the TLS Record subprotocol (see section 3.6.3.2). SeNAA provides secure transport for EAP (see section 3.4) when executed between a PaC and a PAA, by carrying the EAP protocol exchanges inside the TLS payload.

SeNAA mutual authentication is divided into two phases. In phase 1, which carries a TLS handshake (see section 3.6.3.2), the network is authenticated. Access network authentication is based on access network certificates. Phase 2 carries the EAP protocol which is used to authenticate the user. User authentication is bound to the device identifier, which is used to control access to the network.

SeNAA does not assume a secure channel between the PaC and the PAA. To provide such a channel, the SeNAA protocol makes use of the TLS protocol to negotiate a local security association between PaC and PAA, where TLS provides authentication, privacy, integrity, and replay protection. It is used to protect a number of SeNAA AVPs and EAP packets exchanged between PaC and PAA. AVPs that need protection are fed to the TLS Record subprotocol (see

section 3.6.3.2) and the resulting encrypted and compressed data is sent within a TLS-Payload AVP. The EAP protocol is carried inside an EAP-Payload AVP [59]. After a successful TLS handshake, the SeNAA protocol exchanges are protected using a checksum stored in the Msg-Checksum AVP. The AVP is protected using the TLS Record subprotocol.

As stated in [67], TLS is also used by SeNAA for re-authentication between a PaC and a PAA. Local re-authentication, in which a PaC authenticates to a PAA, occurs as part of phase 1, and is handled using the TLS session resumption feature (see section 3.6.3.2). TLS supports mutual authentication and can optionally be used instead of EAP for user authentication. In all cases TLS is used for access network authentication.

SeNAA messages carry information such as the PaC device identifier, that are integrity protected, as described in [184]. If the PAA supports a Diameter and/or RADIUS AAA backend (section 3.9), signalling between PaC and PAA can easily be extended to the backend.

## 3.8 Public Key Authentication for Network Access

The cryptographic techniques used to provide security features for the network access procedures can be either secret key (symmetric) or public key (asymmetric) techniques. Whereas use of the former class of schemes requires the involvement of the home network during the initial authentication process between a user and visited network, the latter allows for architectures that avoid on-line involvement of the home network, since authentication may then be based on certificates. Nevertheless, asymmetric techniques typically require a PKI to support key distribution, and use of this PKI may require on-line certifi-

cate status checking. While symmetric techniques are used almost exclusively today, it seems likely that asymmetric techniques will gain greater importance in future ubiquitous mobility access networks because of their greater flexibility. For further information on public key based network access, see Schwiderski-Grosche and Knospe [167].

This section summarises a number of public key authentication methods for network access relevant to this thesis, including IKEv2 (section 3.8.1) and public key based EAP methods (section 3.8.2). As previously, these methods are described here so that we can subsequently assess their suitability for further application.

### 3.8.1 Internet Key Exchange version 2 (IKEv2)

Internet Key Exchange version 2 (IKEv2) [49] is a component of the IP Security Protocol (see section 3.6.5) that is used for mutual authentication and to establish and maintain SAs. A complete specification of the IKEv2 protocol is given in RFC 4306 [49]. For further information on IKEv2 and its design rationale, see Perlman [154]. IKEv2 consists of two phases:

1. An authentication and key exchange protocol, which establishes an IKE-SA,

2. Messages and payloads which allow negotiation of parameters (e.g. algorithms, traffic selectors) in order to establish IPsec SAs (i.e. Child-SAs).

In the context of the IKE-SA, four cryptographic algorithms are negotiated: an encryption algorithm, an integrity protection algorithm, a Diffie-Hellman group (see section 2.1.3.3), and a pseudo-random function. The pseudo-random function is applied in the construction of keying material for the cryptographic algorithms used in both the IKE-SA and the CHILD-SAs (see section 2.13 of

[49]). In addition, IKEv2 also includes certain payloads and messages which allow configuration parameters to be exchanged for remote access scenarios.

IKEv2 is designed to address certain issues with IKEv1 (see section 3.6.2), as described in Appendix A of [49]. Of particular importance here are the reduced number of initial exchanges, support of legacy authentication, decreased latency of the initial exchange, optional DoS protection capability, and the resolution of certain known security defects. IKEv2 is a protocol that has received a considerable amount of expert review, and whose design benefits from the experience gained from IKEv1.

IKEv2 also provides authentication and key exchange capabilities which allow an entity to use symmetric as well as asymmetric cryptographic techniques, in addition to *legacy authentication*[25] support, within a single protocol. Such flexibility seems likely to be important for heterogeneous network access supporting ubiquitous mobility.

### 3.8.2 Public Key Based EAP Methods

As discussed previously, the EAP protocol supports the use of a number of different authentication mechanisms, known as EAP methods. This section discusses two EAP methods that are based on public key techniques, namely the EAP-TLS (section 3.8.2.1) and EAP-Double-TLS (section 3.8.2.2) authentication protocols.

[25] *Legacy authentication*, described in section 3.3, involves methods that are not strong enough to be used in networks where attackers can easily eavesdrop and spoof on the link (e.g. EAP-MD5 [27] over wireless links). They also may not be able to produce enough keying material. Use of legacy methods can be made more robust by carrying them over a secure channel (see also [49, 65]).

#### 3.8.2.1 EAP-TLS

The PPP EAP Transport Layer Security Authentication Protocol (EAP-TLS) [17] allows use of the protected cipher suite negotiation, mutual authentication and key management capabilities of the TLS protocol (see section 3.6.3). EAP-TLS requires that the peer and the EAP server be authenticated using asymmetric cryptographic techniques, the key management for which is based on X.509 [80] certificates. A complete specification of the EAP-TLS protocol is given in RFC 2716 [17].

As stated in RFC 2716 [17], as a result of the EAP-TLS conversation, the EAP endpoints mutually authenticate, negotiate a cipher suite, and derive a session key. The EAP-TLS conversation typically begins with the authenticator and the peer negotiating the use of EAP. Next, the authenticator sends an EAP-Request/Identity packet to the peer (step 1), and the peer responds with an EAP-Response/Identity packet to the authenticator, containing the peer's user ID (step 2). From this point forward, while nominally the EAP conversation occurs between the authenticator and the peer, the authenticator may act as a pass through device, with the EAP packets received from the peer being encapsulated for transmission to a RADIUS/Diameter (see section 3.9) or other backend security server (i.e. the EAP server).

Once it has received the peer's identifier, the EAP server responds with an EAP TLS/Start packet (step 4). The peer answers with a TLS client_hello handshake message (step 5), and the EAP server responds in turn with a TLS server_hello handshake message (step 6). At this point, the peer has authenticated the EAP server (server authentication). The next message (step 7) contains, among other things, a client_key_exchange message, which completes the establishment of a shared master secret between the peer and the EAP server. If the EAP server sent a certificate_request message in the preceding EAP-Request packet, then the peer must send, in addition, certificate and certificate_verify

handshake messages. The former contains a certificate for the peer's public signature verification key, while the latter contains the peer's signed authentication response to the EAP server.

After receiving this packet, the EAP server verifies the peer's certificate and digital signature, if requested (client authentication). If the peer authenticates successfully, the EAP server sends a response containing a finished_handshake message (step 8). If the EAP server is correctly authenticated, the peer must send an EAP-Response packet of EAP-Type equal EAP-TLS, and no data (step 9). The EAP server must then respond with an EAP-Success message (step 10).

#### 3.8.2.2 EAP-Double-TLS

EAP-Double-TLS [22] is an EAP protocol that extends EAP-TLS. A complete specification of the EAP-Double-TLS protocol is given in a 2006 Internet Draft [22]. In EAP-TLS, a full TLS (see section 3.6.3) handshake is used to mutually authenticate a peer and server and to share a secret key. EAP-Double-TLS extends this authentication negotiation by using a secure connection established by the TLS Pre-Shared Key (PSK — see section 3.6.7) handshake, to exchange additional information between peer and server. The secure connection established using the TLS PSK handshake is used to allow the server and the peer to securely exchange their identifiers, and to update security attributes for later sessions in order to ensure *perfect forward secrecy* (see section 2.1.3.3). A more detailed description of the TLS PSK handshake may be found in section 3.3.1 of [22].

EAP-Double-TLS allows the peer and server to establish keying material for use in subsequent data exchanges. The keying material is established implicitly between the peer and server as a result of the TLS Pre-Shared Key handshake. The TLS shared-key mechanism is designed for use as a 'resumed session' (i.e.

a secure connection may be terminated and *resumed*[26] later by the peer and server) using a pre-installed secret key. RFC 4279 [60] details the use of secret keys (see section 2.1.3.2) shared in advance among communicating parties in the TLS protocol. The secure connection established by the resumed handshake may then be used to allow the server to authenticate the peer using certificate authentication infrastructures (see section 2.1.3.3), PSK (see section 3.6.7), or smart cards.

Finally, EAP-Double-TLS allows anonymous exchanges and provides identity privacy protection against eavesdropping, MitM (see section 3.2.3) and other cryptographic attacks.

## 3.9 AAA Backend Infrastructure

A number of requirements apply to an authentication, authorisation and accounting (AAA) backend infrastructure. The AAA protocol evaluation criteria for network access, summarised in RFC 2989 [15], include the following:

**Scalability.** The AAA protocol must be capable of supporting millions of users and tens of thousands of simultaneous requests. Also the AAA architecture and protocol must be capable of supporting tens of thousands of devices, AAA servers, proxies and brokers.

**Mutual Authentication.** The AAA protocol must support mutual authentication between the AAA client and server.

**Transmission Level Security.** The AAA protocol requires authentication, integrity protection and confidentiality at the transmission layer. This

[26] TLS allows the peer and the server to resume sessions. When a TLS session is resumed, it must be resumed using the same cipher suite with which it was originally negotiated. The peer and the server may thus decide to resume a previous session instead of negotiating security parameters for a new session [44].

security model is also referred to as hop-by-hop security, whereas end-to-end security is established between two communicating peers.

**Data Object Confidentiality.** The AAA protocol requires confidentiality at the object level, where an object consists of one or more attributes.

**Data Object Integrity.** The AAA protocol requires authentication and integrity protection at the object level. Object level authentication must be persistent across one or more intermediate AAA entities (e.g. proxies, brokers, etc.), so that any AAA entity in a proxy chain may verify the integrity and authenticity of a data object.

**Certificate Transport.** The AAA protocol must be capable of transporting public key certificates.

In this section two AAA backend infrastructure protocols are discussed, namely RADIUS (section 3.9.1) and Diameter (section 3.9.2). The Diameter EAP application (see section 3.9.3) is also described.

### 3.9.1 RADIUS

The Remote Authentication Dial In User Service (RADIUS) [161] protocol carries authentication, authorisation, and configuration information between a network access server and a remote authentication server. RADIUS runs over the UDP protocol [156] . Historically, the RADIUS protocol has been used to provide AAA backend services for dial-up PPP [168] and terminal server access. A complete specification of the RADIUS protocol is given in RFC 2865 [161]. A number of RADIUS key features are listed below:

**Client/Server Model.** A network access server operates as a client of RADIUS. A RADIUS server receives user connection requests, authenticates the user, and then returns all the configuration information necessary for

the client to deliver service to the user. A RADIUS server can act as a proxy client to other RADIUS servers.

**Network Security.** Transactions between the client and RADIUS server are authenticated through the use of shared secret. Any user passwords input to the client device are sent to the RADIUS server in encrypted form.

**Flexible Authentication Mechanisms.** RADIUS servers can support a variety of methods to authenticate a user. When provided with a user name and password, a RADIUS server can support PPP PAP or CHAP, UNIX login, and other authentication mechanisms.

**Extensible Protocol.** RADIUS transactions are comprised of variable length attributes. New attribute values can be added without invalidating existing implementations of the protocol.

RADIUS protocol operation is described below, and summarised in Figure 3.13.

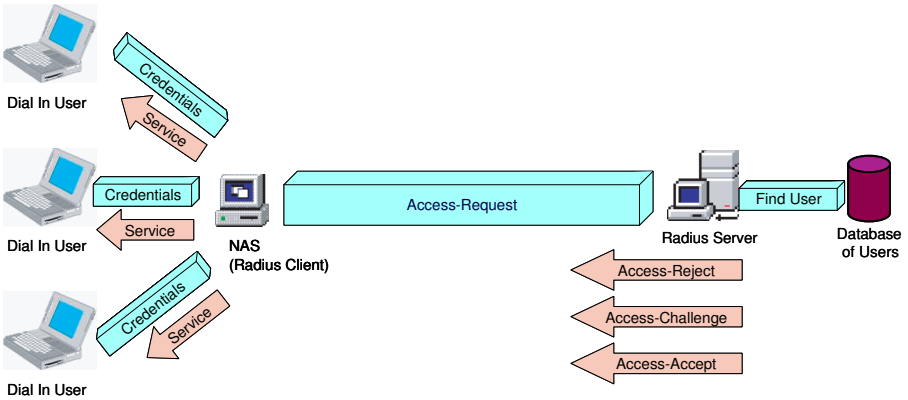


Figure 3.13: **RADIUS operation**

The Access-Request contains such Attributes as the user's name, the hash of the user's password, the ID of the client and the ID of the Port which the user is accessing. The user entry in the RADIUS server database contains a

list of requirements to be met if access for the user is to be permitted. If any condition is not met, the RADIUS server sends an 'Access-Reject' response, indicating that this user request is invalid. If all conditions are met and the RADIUS server wishes to issue a challenge, to which the user must respond, the RADIUS server sends an 'Access-Challenge' response, Finally, if all conditions are met and the RADIUS server is completely satisfied, the list of configuration values for the user are placed into an 'Access-Accept' response. These values include the type of service (e.g. SLIP, PPP, or Login User) and all necessary values to deliver the desired service.

As discussed in RFC 2865 [161], RADIUS is today a widely implemented and used means of authenticating and authorising dial-up and tunnelled network users. In addition, a number of significant extensions to RADIUS have been published in RFC 2869 [160]. Nevertheless, as discussed in RFC 3588 [34], the continued growth of the Internet and the introduction of new access technologies, including wireless, DSL (see section 3.2.1), Mobile IP [153], and Ethernet[27] based LANs, and the increasing complexity and density of routers and network access servers, will put new demands on AAA protocols, making the RADIUS protocol increasingly unsuitable for use in such networks.

The potential future problems with RADIUS have led to the development of Diameter [34]. The basic RADIUS model is retained by the Diameter protocol. However, Diameter, described immediately below, addresses the known flaws in the RADIUS Protocol so that AAA services can be provided to new access technologies.

### 3.9.2 Diameter

As mentioned in section 3.9.1, the Diameter protocol [34] was not designed from the ground up. Instead, the basic RADIUS model was retained, and known flaws

[27] http://www.ieee802.org/3/

in the RADIUS protocol were addressed. Diameter does not share a common protocol data unit with RADIUS, but does borrow sufficiently from the protocol to ease migration. The Diameter protocol was thus heavily inspired and builds upon the tradition of the RADIUS protocol.

The basic concept behind Diameter is to provide a base protocol that can be extended in order to provide AAA services for use with new access technologies (e.g. Roaming Protocols and Mobile-IP) in large scale systems with provisions for congestion control [34]. Such flexibility seems likely to be important for heterogeneous network access supporting ubiquitous mobility. A complete specification of the Diameter protocol is given in RFC 3588 [34].

Diameter runs over reliable transport mechanisms (TCP and SCTP, as defined in [19]; see section 3.6.3), and provides the following facilities:

- delivery of AVPs;
- capabilities negotiation;
- error notification;
- extensibility, through addition of new commands and AVPs; and
- basic services necessary for applications, such as handling of user sessions or accounting.

All data delivered by the protocol must be in the form of an AVP. Some of these AVP values are used by the Diameter protocol itself (e.g. the User-Name AVP), while others deliver data associated with particular applications that employ Diameter. AVPs may be added arbitrarily to Diameter messages, so long as the required AVPs are included. AVPs are used by the base Diameter protocol to support the following features:

- Transport of user authentication information, to enable the Diameter

server to authenticate the user.

- Transport of service specific authorisation information between Diameter client and servers, allowing the peers to decide whether a user's access request should be granted.
- Exchanging resource usage information to be used for accounting purposes, capacity planning, etc.
- Relaying, proxying and redirecting of Diameter messages through a server hierarchy.

The Diameter base protocol provides the minimum requirements needed for a AAA protocol, as listed in RFC 2989 [15]. The base protocol may be used by itself for accounting purposes only, or it may be used with a Diameter application, such as Mobile IPv4 [33], or network access [35]. It is also possible for the base protocol to be extended for use in new applications, via the addition of new commands or AVPs. Currently the focus of Diameter is network access and accounting applications.

In the Diameter base protocol, any node can initiate a request. In that sense, Diameter is a peer-to-peer protocol. A Diameter client is a device at the edge of the network that performs access control, such as a network access server or a foreign agent. A Diameter client generates Diameter messages to request authentication, authorisation, and accounting services for the user. A Diameter agent is a node that does not authenticate and/or authorise messages locally; agents include proxies, redirects and relay agents. A Diameter server performs authentication and/or authorisation of the user. A Diameter node may act as an agent for certain requests while acting as a server for others.

As stated in RFC 3588 [34], the current Diameter specification is made up of a base specification [34], a Transport Profile [19], and two applications: Mobile IPv4 [33], and the Diameter Network Access Server (NAS) application [35].

The Transport Profile document [19] discusses transport layer issues that arise with AAA protocols, and gives recommendations on how to overcome them[28]. The Diameter Mobile IPv4 document [33] specifies a Diameter application that allows a Diameter server to authenticate, authorise and collect accounting information for Mobile IPv4 services rendered to a mobile node. The Mobile IPv4 (see section 4.2.4) protocol allows a mobile node to change its point of attachment to the Internet while maintaining its fixed home address. Finally, the NAS document [35] defines a Diameter application that allows a Diameter server to be used in a PPP/SLIP dial-up and terminal server access environment; provisions are made in this application for servers that need to perform protocol conversion between Diameter and RADIUS.

### 3.9.3 Diameter EAP Application

As described in section 3.4, EAP [13] is an authentication framework which supports multiple authentication mechanisms. EAP may be used on dedicated links as well as switched circuits, and wired as well as wireless links.

The Diameter EAP application [59] carries EAP packets between a network access server working as an EAP authenticator and a backend authentication server. The Diameter EAP application is based on the Diameter NAS application and is intended for similar environments. A complete specification of the Diameter EAP application is given in RFC 4072 [59].

In the Diameter EAP application, authentication occurs between the EAP client and its home Diameter server. This end-to-end authentication process reduces the possibility for attacks on the authentication procedure (e.g. replay and MitM attacks). End-to-end authentication also provides a possibility for mutual authentication, which is not possible with PAP and CHAP (described in section 3.3.1) in a roaming PPP environment.

[28] This document also defines the Diameter failover algorithm and state machine.

Diameter EAP defines new Command-Codes and new AVPs, and can work with RADIUS EAP support (see RFC 3579 [14]). The use of EAP in Diameter involves the following steps (see Figure 3.14).

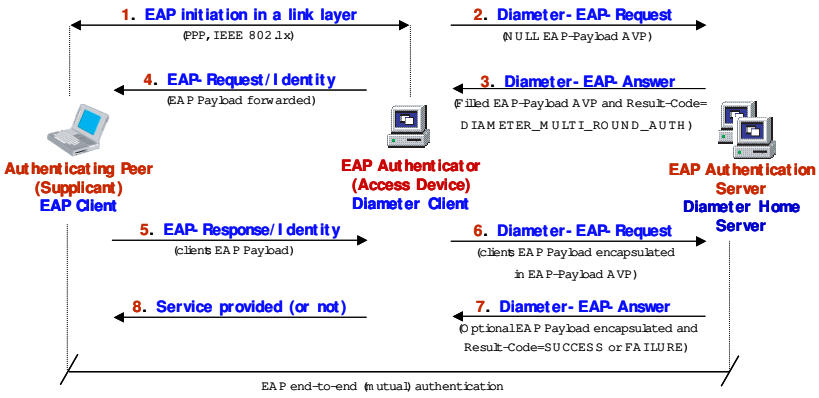


Figure 3.14: **Using EAP in Diameter**

1. The EAP conversation between the EAP Client (Authenticating Peer) and the Access Device (EAP Authenticator) begins with the initiation of EAP within a link layer, such as PPP [168] or 802.11 [85].

2. The Access Device (Diameter Client) will typically send to the Diameter Home Server a Diameter-EAP-Request (DER) message with a NULL EAP-Payload AVP, signifying an EAP-Start.

3. If the Diameter Home Server supports EAP, it must respond with a Diameter-EAP-Answer (DEA) message. The initial DEA message contains an EAP-Payload AVP (that encapsulates an EAP packet) and usually the Result-Code AVP is set to DIAMETER_MULTI_ROUND_AUTH, signifying that a subsequent request is expected. The initial DEA in a multi-round exchange normally encapsulates an EAP-Request/Identity payload, requesting the EAP Client to identify itself.

4. The Access Device forwards the EAP-Request/Identity payload to the EAP Client.

5. The EAP Client returns to the Access Device an EAP-Response/Identity packet, containing the EAP Client's identity (or user's identity).

6. Upon receipt of the EAP Client's EAP-Response, the Access Device will issue a second DER message to the Diameter Home Server, with the client's EAP packet encapsulated within the EAP-Payload AVP.

7. The conversation continues until the Diameter Server sends a DEA with a Result-Code AVP indicating SUCCESS or FAILURE, and an optional EAP-Payload (of type EAP-Success or EAP-Failure). If a response is received with the Result-Code AVP set to DIAMETER_COMMAND_UNSUPPORTED, it is an indication that the Diameter Home Server does not support EAP.

8. The Result-Code AVP is used by the Access Device to determine whether or not service is to be provided to the EAP Client.

## 3.10 Liberty Alliance Project

The Liberty Alliance Project represents a broad spectrum of organisations that have united to establish open technical specifications to support a vast range of *network identity*-based interactions. The *network identity* of a user is the global set of attributes composed from an individual's various accounts (see section 1.3 of [56]). A complete specification of the Liberty architecture is available at the Liberty Alliance Project web site[29].

The Liberty architecture specifies a *single sign-on* (SSO) solution, where an initial authentication of the user to an *Identity Provider* (IdP) can be reused for further authentication, via a network identity infrastructure, to a number of *Service Providers* (SPs). SPs are organisations offering Web-based services to

[29] http://www.projectliberty.org/specs/

users (e.g. Internet portals, retailers, transportation providers, financial institutions, entertainment companies, not-for-profit organisations, and government agencies). IdPs are SPs offering identity services; they may offer a range of business incentives to encourage other SPs to affiliate with them. As described in [56], establishing such SP affiliations creates *circles of trust*, i.e. a collaboration of businesses and IdPs having business relationships based on Liberty and operational agreements. Each circle of trust may contain multiple SPs and (in the simplest case) one IdP.

For example, in an enterprise circle of trust, the IdP is a company managing employee network identities across the enterprise. Another example is a consumer circle of trust, where a user's bank has established business relationships with a number of other SPs, allowing the user to use her bank-based network identity with them[30].

In this section, the key objectives (section 3.10.1), the main requirements (section 3.10.2) and the operation (section 3.10.3) of the Liberty scheme are described. After that, an overview of the Liberty SSO architecture is provided (section 3.10.4), and the three major components of the Liberty basic structure are then discussed, including the identity federation framework (section 3.10.5), the identity web services framework (section 3.10.6), and the identity service interface specifications (section 3.10.7). Finally, the security mechanisms incorporated in Liberty-enabled implementations are summarised (section 3.10.8).

### 3.10.1 Liberty Objectives

According to section 1.3.1 of [56], the key Liberty objectives are to:

- enable consumers to protect the security and privacy of their network

[30] Although these scenarios are enabled by SPs and IdPs deploying Liberty-enabled products in their infrastructure, they do not require users to use anything other than one of today's common Web browsers.

identity information;

- enable businesses to manage customer relationships without third party involvement;
- provide an open SSO standard including decentralised authentication and authorisation from multiple providers; and
- create a network identity infrastructure supporting all network access devices.

### 3.10.2 Liberty Requirements

The following Liberty engineering requirements are based on those given in the Liberty Architecture Overview specifications [56, 173].

**Interoperability.** Potential Liberty clients include a broad range of presently deployed Web browsers, Web-enabled client access devices, and newly designed Web-enabled browsers or clients with specific Liberty-enabled features.

**Openness.** Liberty must provide the widest possible support for operating systems, programming languages, and network infrastructures, and must facilitate multi-vendor interoperability between Liberty clients and services.

**Identity federation.** SPs and IdPs must notify the user regarding identity federation and defederation, in addition to notifying each other about user identity defederation. Each IdP also notifies appropriate SPs of user account terminations at the IdP. Each SP or IdP gives each of its users a list of the user's federated identifiers at the IdP or SP. An SP may also request an anonymous, temporary identifier for a user.

**Authentication.** The IdP's authenticated identifier must be given to the user before she presents her credentials to the IdP. Protection of information

exchanged between IdPs, SPs, and User Agents, and mutual authentication between IdPs and SPs, must be provided[31]. SPs must have the capability to cause the IdP to re-authenticate the user. An IdP, at the discretion of the SP, is allowed to authenticate the user via an IdP other than itself.

**Use of pseudonyms.** Liberty-enabled implementations must be able to support the use of pseudonyms that are unique on a per-identity-federation basis across all IdPs and SPs.

**Anonymity.** An SP may request that an IdP supplies a temporary pseudonym that will preserve the anonymity of a user. This identifier may be used to obtain information for or about the user, without requiring the user to consent to a long term relationship with the SP.

**Global logout.** Liberty-enabled implementations must be able to notify all affected SPs when a user logs out at the IdP.

**Service discovery.** The Liberty architecture must provide a mechanism for SPs to query the discovery service for the relevant providers of services to a particular user.

**Service registration.** The Liberty scheme must provide a mechanism for SPs to register/deregister a list of services that it provides for a specific user with the discovery service.

**Support for gathering consent.** The Liberty scheme must provide a mechanism for a relying SP to request that the invoking SP directs a user to the relying SP in order to request the user for consent. It is also required that the SP utilises a 'Liberty Enhanced Client Profile (LECP) communications channel' (see section 3.2.1 of [173]) to ask the user for consent and to obtain the user's response.

[31] A variety of authentication methods are supported by Liberty.

**Support for anonymous service.** The Liberty architecture must provide a mechanism for an SP to make anonymous attribute requests and receive anonymous attribute responses (i.e. the ability to share attributes without disclosing the identity of the user to the requestor or SP), in addition to a mechanism to prevent correlation of pseudonyms in service tokens with user identifiers.

**Support for usage indications.** The Liberty scheme must provide a mechanism for an SP to associate the user's intended usage with the requested attributes, in an attribute request to a relying SP. A mechanism is also required for an SP to associate the agreed upon intended usage indications with the attribute response, in addition to a mechanism for an SP to return a list of acceptable usage indications.

### 3.10.3 Operation of the Liberty Scheme

As stated in [56], Liberty is composed of three architectural components, the operation of which are described below and summarised in Figure 3.15.

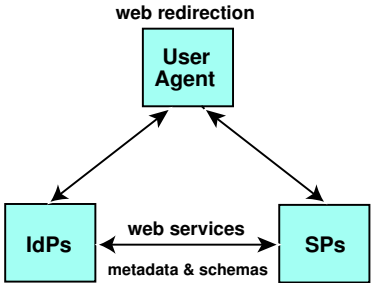


Figure 3.15: **Liberty operation**

#### 3.10.3.1 Web Redirection

The web redirection component enables Liberty-enabled entities to provide identity management services to users. There are two options for web redirection, namely HTTP-redirect and Form-POST (see section 4.1 of [56]). Both options create a communication channel between IdPs and SPs via the User Agent. Note that the term *User Agent* is used to mean software running on the consumer host acting on the user's behalf, e.g. http client software or a web browser.

#### 3.10.3.2 Web Services

The web services component consists of a set of protocol profiles which enable Liberty-enabled entities to directly communicate. In the Liberty context, a *protocol profile* means a combination of message content specification and message transport mechanisms, which includes possible mappings of protocol messages exchanged by IdPs and SPs to particular means of communications (e.g. HTTP and SOAP, described immediately below). Protocol profiles are grouped into categories according to the protocol message intent, as follows (see section 3 of [37]):

- Single Sign-On and Federation;
- Name Registration;
- Federation Termination Notification;
- Single Logout;
- Identity Provider Introduction;
- Name Identifier Mapping;
- Name Identifier Encryption.

A number of Liberty protocol interaction steps occur directly between system entities (in addition to other steps occurring via web redirection), and are based on Remote-Procedure-Call-like (RPC-like) protocol messages conveyed via the SOAP protocol [29, 74]. SOAP is a widely implemented specification for RPC-like interactions and message communications, which uses the Extensible Markup Language[32] (XML) and Hypertext Transfer (HTTP) [62] protocols (see section 4.2 of [56]).

#### 3.10.3.3 Metadata and Schemas

This Liberty architectural component consists of a set of metadata and formats used by Liberty-enabled sites to communicate a variety of provider-specific (and other) information. Metadata and schemas are generic terms referring to a variety of subclasses of information and associated formats exchanged between SPs and IdPs. The Liberty subclasses of exchanged information are: *Account/Identity*, *Authentication Context*, and *Provider Metadata* (see section 4.3 of [56]).

### 3.10.4 Liberty Architecture

As stated by Tourzan and Koga [173], the Liberty architecture consists of a multi-level layered specification set, based on open standards including the Security Assertion Markup Language (SAML) [128] and SOAP (see section 3.10.3.2). The Liberty architecture has three major components:

- The *Liberty Identity Federation Framework* (ID-FF), which specifies core protocols, schemata and profiles that allow implementers to create a standardised identity federation network.
- The *Liberty Identity Web Services Framework* (ID-WSF), which consists

[32] http://www.w3.org/XML/

of a set of schemata, protocols and profiles used to provide identity services, such as identity service discovery and invocation.

- The *Liberty Identity Service Interface Specifications* (ID-SIS), which utilise the ID-WSF and ID-FF to provide services that depend on network identity, such as contacts, presence detection or wallet services.

We discuss these three Liberty architecture components in more detail in the following sections.

### 3.10.5 Liberty Identity Federation Framework (ID-FF)

The main goal of the Liberty Identity Federation Framework (ID-FF) is to establish a basic structure to support a range of network identity based interactions, and give business:

- a basis for new revenue opportunities, building upon existing relationships with consumers and partners; and
- a framework that gives consumers choice, convenience and control when using any Internet-connected device.

In this federated commerce scenario, a user's online identity, personal profile, personalised online configurations, buying habits, and shopping preferences are administered by the user and securely shared with organisations of the user's choosing. A federated network identity model can then ensure that critical private information is only used by appropriate parties.

The first step to realising a federated identity infrastructure is the establishment of a standardised, multi-vendor, Web-based single sign-on with federated identifiers, based on today's commonly deployed technologies. A general specifi-

cation of the Liberty ID-FF structure, which offers an approach for implementing such a scheme, is given by Wason et al. [56].

As stated in section 2 of [56], Liberty ID-FF has two main facets: identity federation and single sign-on, which are described immediately below.

#### 3.10.5.1 Liberty Identity Federation

When a user first uses an IdP to login to an SP, she must be given the opportunity to federate any existing SP local identity with the identity she has at the IdP. *Identity federation* then involves linking distinct SP and IdP user accounts, and associating an opaque user handle with the two local user identities. This account linkage underlies and enables other facets of Liberty ID-FF.

Identity federation must only take place given prior agreement between IdPs and SPs. It should also be predicated upon notifying the user, obtaining the user's consent, and recording both the notification and consent in an auditable fashion.

After federation, the IdP and the SP share a pair of unlinkable pseudonyms (opaque user handles) for the user, one for each direction. They do not need to know one another's local identity for the user.

#### 3.10.5.2 Liberty Single Sign-On

*Single sign-on* allows a user to sign on once with an IdP in a federated group of SPs (or, from a provider's point of view, with a member of a *circle of trust*) and, after that, use other websites from the group without signing on again. Single sign-on is enabled once a user's IdP and SP identities have been federated.

From a user's perspective, single sign-on is realised when the user logs into an IdP, and uses multiple affiliated SPs without having to sign on again, as shown

in Figure 3.16. This process requires federation of the user's local identifiers between the applicable IdPs and SPs.

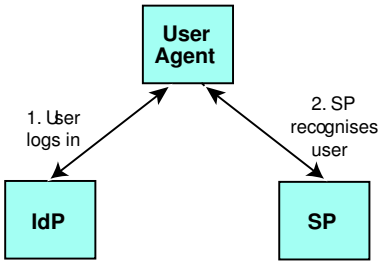


Figure 3.16: **Liberty user logs in at IdP and is recognised by SP**

#### 3.10.5.3 Single Sign-On and Federation Protocol

The Liberty single sign-on and identity federation processes are supported by the Single Sign-On and Federation Protocol, as specified by Cantor and Kemp [36]. This protocol enables both identity federation (see section 4.4.1 of [56]) and single sign-on (see section 4.4.2 of [56]) in a single overall flow. A variety of profiles implementing this overall protocol flow (see section 4.4.3 of [56]) are defined by Cantor, Kemp and Champagne [37].

### 3.10.6 Liberty Identity Web Services Framework (ID-WSF)

The Liberty ID-FF framework previously described requires the use of federated network identifiers. The Liberty ID-WSF builds upon this foundation and provides a framework for identity-based web services in a federated network identity environment. A general specification of the Liberty ID-WSF framework is given by Tourzan and Koga [173].

As stated by Tourzan and Koga [173], the Liberty ID-WSF defines a SOAP based invocation framework (see section 3.10.3.2), which allows identity Web

services to be discovered and invoked. Figure 3.17 [173] illustrates the Liberty entities involved in a possible scenario for identity Web service invocation.

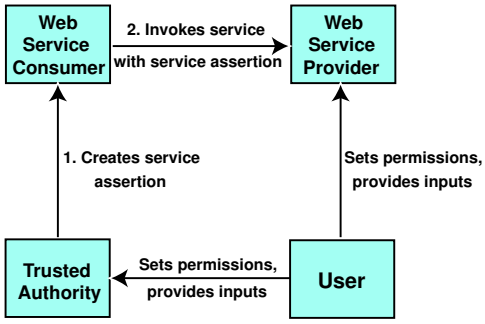


Figure 3.17: **Identity Web service invocation**

As shown in the figure above, once a service has been discovered and sufficient authorisation data has been received from a *Trusted Authority*, the invoking entity, called the *Web Service Consumer*, may invoke the service at the hosting/relying entity, called the *Web Service Provider*.

In order to convey the fact that a Liberty entity has the right to access a resource, the ID-WSF framework defines extensions so that service invocation authorisation data may be generated by a Trusted Authority, and then issued to the invoking entity. The Web Service Provider can make access control decisions using this authorisation data, based on its business practices and the preferences of the resource owner.

The Trusted Authority is, in most cases, an *IdP/Discovery Service*, which defines mechanisms for describing and discovering identity web services. An *identity web service* is a type of web service whose operations are indexed by identities (see section 1.1 of [23]). An *identity* will typically have one or more discovery services on the network, which allow other entities to discover its identity services.

A *discovery service* is thus a type of identity service that allows requesters

to discover resource offerings associated with a given identity (see section 5 of [23]). A *resource offering* is the association of a resource and a service instance that provides access to that resource (see section 4 of [23]). A *service instance* is a running web service at a distinct protocol endpoint (see section 3 of [23]).

The Liberty ID-WSF framework also defines an Interaction Service protocol [12]. This protocol provides schemas and profiles to enable an entity to interact with the owner of a resource that is exposed by a Web Service Provider. An example of a use of the Interaction Service would be to query the user for permissions in a web services context.

Finally, the Liberty Alliance has defined a Personal Profile service [115] for use with the Liberty ID-WSF framework. This service is designed to enable account creation in a web services context. The Personal Profile service provides a schema for requests for personal information, which allows a Web Service Consumer to gather the information necessary to create an account or provide personalised services.

### 3.10.7 Liberty Identity Service Interface Specifications (ID-SIS)

The Liberty ID-SIS component provides a collection of identity Web service specifications and corresponding implementation guidelines, which can be summarised as follows:

**Personal Profile (PP).** The ID-SIS-PP service specification [115] describes a Web service that provides a user's basic profile information (such as her name or contact details). A user might typically have two ID-SIS-PP service instances, one for a work identity and another for a private identity. The implementation guidelines that supplement the ID-SIS-PP specification are given in [114].

**Employee Profile (EP).** The ID-SIS-EP service specification [113] describes a Web service that provides an employee's basic profile information. The implementation guidelines that supplement the ID-SIS-EP specification are given in [112].

**Contact Book (CB).** The ID-SIS-CB service specification [55] describes a Web service offering contact information. The implementation guidelines that supplement the ID-SIS-CB specification, which offer the ability to manage a contact directory, are given in [50].

**Geolocation (GL).** The ID-SIS-GL service specification [52] describes a Web service offering geolocation information associated with a user, including position, speed and direction[33]. The implementation guidelines that supplement the ID-SIS-GL specification are given in [73].

**Presence (PRES).** The ID-SIS-PRES service specification [54] describes a Web service offering presence information associated with a user. The implementation guidelines that supplement the ID-SIS-PRES specification, which focus on guidelines for Presence Service Providers and Presence Service Clients, are given in [53].

### 3.10.8 Liberty Security Mechanisms

Table 3.1 [173] summarises the security mechanisms which, as described below, are incorporated in Liberty implementations at two different layers: channel security and message security. It also summarises the security-oriented processing requirements placed on Liberty implementations. A complete specification of the Liberty security mechanisms is given by Ellison, Hirsch and Madsen [58]. Separate security mechanism SAML [128] profiles, defining in particular the use

[33]ID-SIS-GL may also provide geolocation information in a more human readable format (e.g. street, city, region, and country).

of Liberty *security tokens*[34], are also given by Ellison, Hirsch and Madsen [57]. A complementary specification of the Liberty ID-WSF authentication service, which permits a Liberty-enabled User Agent (LUA — see section 3.10.3.1) to authenticate with an IdP and obtain a security token, is given by Hodges and Aarts [78].

Table 3.1: **Liberty security mechanisms**

| **Security Mechanism** | **Channel Security** | **Message Security** |
|---|---|---|
| Confidentiality. | Required. | Optional. |
| Per-message data integrity. | Not applicable. | Required. |
| Transaction integrity. | Not applicable. | Required. |
| Data origin authentication. | Not applicable. | Required. |
| Non-repudiation. | Not applicable. | Required. |

#### 3.10.8.1 Channel Security

*Channel security* covers how communications between IdPs, SPs, and User Agents are to be protected. Liberty implementations must use either TLS or SSL (see section 3.6.3) for channel security, or another communication security protocol with similar security characteristics, e.g. IPsec (see section 3.6.5). Critical issues for Liberty channel security include the following:

- SPs are required to authenticate IdPs using IdP server-side certificates. IdPs have the option to require authentication of SPs using SP client-side certificates.
- Each SP must specify a list of authorised IdPs, and each IdP is required to be equipped with a list of authorised SPs. Thus any SP-IdP pair must be mutually authorised before they engage in Liberty interactions. Such authorisation occurs in addition to the authentication process.

[34]The possession of a security token entitles the Liberty user to invoke services of the IdP, such as the Single-Sign-On service.

- The authenticated identifier of an IdP must be presented to a user before the user presents personal authentication data (or credentials) to that IdP.

#### 3.10.8.2 Message Security

*Message security* covers security mechanisms applied to the Liberty protocol messages (such as requests and assertions) passed between IdPs, SPs, and User Agents. These messages are exchanged across the communication channels whose security characteristics were discussed above. Critical issues for Liberty message security include the following:

- Liberty protocol messages must be digitally signed and verified, providing data integrity, data origin authentication, and a basis for non-repudiation (see section 2.1.1.4).
- IdPs and SPs must use cryptographic key pairs that are distinct from the key pairs used for TLS or SSL channel protection, and that are suitable for long-term signatures.
- In transactions between SPs and IdPs, message requests must be protected against replay, and received responses must be checked for correct correspondence with issued requests. Time-based assurance of freshness may also be employed in Liberty message exchanges. These security techniques provide transaction integrity.
- To become members of a Liberty circle of trust, providers must establish bilateral agreements on selecting CAs, obtaining X.509 credentials (see section 2.1.3.3), establishing and managing trusted public keys, and managing the life cycles of corresponding credentials.

# Part II

# Internet Remote Access Authentication

# Chapter 4

# Internet Authentication Problem Domain & Scenarios

## Contents

The aim of this chapter is to describe the problem domain which forms the main focus of this thesis. In addition we describe a variety of different scenarios related to Internet remote access authentication; these scenarios serve to further illustrate this problem domain.

Section 4.1 establishes the Internet remote access problem domain; this discussion covers a number of different issues, including remote dynamic service provider selection, tunnelled authentication procedures for carrying EAP, EAP encapsulated authentication methods, transport schemes for EAP, and current ad hoc solutions for Internet remote access.

Section 4.2 identifies a variety of different authentication scenarios for Internet remote access; the first two are categorised in terms of which layer in the protocol stack security services are provided. Next we depict a scenario covering the case where no security services are provided at the lower layers of the protocol hierarchy. We then describe further scenarios involving respectively mobile IP, personal area networks, and limited free access.

## 4.1 Problem Domain

Internet remote access networks which are not physically secured against unauthorised use are typically set up so that roaming entities are obliged to go through an authentication process. In some scenarios, an IP-based device is required to authenticate itself to the network prior to being authorised to use it. As stated in RFC 4058 [184], this authentication procedure usually requires a protocol that can support a variety of authentication methods, dynamic service provider selection, and roaming clients.

The Internet remote access authentication process thus needs a protocol between the remote entity and the network capable of transporting multiple authentication methods, e.g. CHAP (see section 3.3.1) and TLS (see section 3.6.3). In the light of the abundance of access technologies, e.g. GSM (section 3.5.1), IEEE 802.11 [85], and Bluetooth[1], it is important that the authentication protocol is able to carry a range of different authentication methods regardless of the underlying access technology.

In the absence of a link layer authentication mechanism that can satisfy these needs, current architectures fill the gap by using a number of methods which are far from ideal, both architecturally and from a security perspective. Operators typically adopt one of the following three approaches: use of non-standard ad hoc solutions at layers above the link layer, insertion of additional protocol layers for authentication, or misuse of an existing protocol in ways that were not intended by its designer.

Examples of such approaches include: inserting an additional layer between the link layer and the network layer for client authentication (e.g. PPPoE [129]), overloading another network layer protocol to achieve this goal (e.g. Mobile IPv4 [153], with a Registration-required flag), and even defining ad hoc application

[1] http://www.bluetooth.com/

layer authentication mechanisms (e.g. http redirects with web-based login). As stated in RFC 4058 [184], "in the absence of physical security (and sometimes in addition to it) a higher layer (L2+) access authentication mechanism is needed". In these and other cases, a network layer authentication protocol may provide a cleaner solution to the authentication problem.

This section establishes the problem domain for Internet remote access authentication; this discussion covers a number of different issues, including remote dynamic service provider selection (section 4.1.1), tunnelled authentication for carrying EAP (section 4.1.2), EAP encapsulated authentication methods (section 4.1.3), transport schemes for EAP (section 4.1.4), and current ad hoc solutions for Internet remote access (section 4.1.5).

### 4.1.1 Remote Dynamic Service Provider Selection

As stated in RFC 4058 [184], an important aspect of an authentication protocol for Internet remote access is the ability to provide dynamic service provider selection to the remote entities. Regardless of their network access provider (NAP), remote entities should be able to select an Internet access provider (ISP) of their choice. Separation of the NAP from the ISP, and the creation of a single NAP granting service for remote entities from multiple ISPs, are made possible by this characteristic [144].

### 4.1.2 Tunnelled Authentication for Carrying EAP

Support for a variety of authentication methods, including those providing dynamic service provider selection and support for roaming clients, can be achieved by using tunnelled authentication mechanisms (see sections 3.2.2 and 3.7) and even new arrangements. This thesis focuses on scenarios in which tunnelled authentication protocols that can carry EAP [13] are used. This is because EAP is

very flexible and can encapsulate arbitrary authentication methods (see section 3.4).

### 4.1.3 EAP Encapsulated Authentication Methods

Although most of the tunnelled authentication mechanisms proposed in IETF documents advocate the use of EAP, they do not discuss which authentication methods would be most suitable to be carried by EAP in particular practical applications. For example, if a remote entity wishes to access real time media applications available via the Internet through an access network, delay is not a good feature. So, in such situations, it would be important to reduce the number of round trips needed for the authentication protocol carried by EAP. Indeed, in such a case it may be important to consider if it is really necessary to use a challenge-response protocol for authentication. As stated in [127], a lightweight authentication method based on a one-time-pad, hash chains (see sections 2.1.3.2 and 3.3.3) or some kind of cryptographic random number sequence may sometimes be more suitable in such circumstances.

By using EAP to encapsulate (or carry) authentication methods, it is thus possible to create new authentication solutions at the application layer. This can be achieved, for instance, through the EAP encapsulation of authentication protocols arising from the mobile telecommunications sphere, or of public key based authentication protocols.

### 4.1.4 Transport Schemes for EAP

The use of EAP requires the provision of a transport scheme between the remote entity and the Internet access network (see section 3.4). Among the current access technologies, only IEEE 802 defines how to carry EAP at the link layer [84]. Other link layer technologies require the implementer to make a choice

between using PPP [168] or PPPoE [129] as a link layer agnostic way of carrying EAP, given that PPP-based authentication can provide some of the required functionality.

However, just using PPP for authentication is not a good choice, since inserting this additional layer between the link layer and network layer has undesirable properties. According to RFC 4058 [184],"using PPP just for remote entity authentication incurs additional messaging during the connection setup and extra per-packet processing. It also forces the network topology to a point-to-point model".

Defining a network layer transport for EAP, such as the proposed tunnelled authentication solutions (see sections 3.6 and 3.7), or other possible arrangements, provides a cleaner answer to the problem. As stated in RFC 4058 [184], such solutions provide support for a variety of authentication methods, dynamic service provider selection and roaming clients. In addition, it is also possible to define a link layer agnostic carrier for the EAP protocol, without having to incur the additional costs and limitations of inserting another layer in the stack, as in the case of PPP.

### 4.1.5 Ad Hoc Solutions for Internet Remote Access

For the time being, while a network layer authentication solution (see section 4.1.4) has not been approved as a standard, implementers are forced to design their own ad hoc solutions to the Internet remote authentication problem. One such solution is the application layer authentication method implemented using http redirects and web-based login[2]. In this method, once the link is established, user traffic is re-directed to a web server, which in turn generates a web-based

[2]This solution can, for example, be used for web mail access; however this is not exactly an Internet remote access example, but instead an instance of access to an Internet service. A better example would be the use of http redirects with web-based login to DSL networks (see section 3.2.1) that use DHCP (see section 3.6.5.4) as a configuration method, as described in section 4.2.3.

login, forcing users to provide the authentication information. In addition to being a non-standard solution, this method has well-known vulnerabilities, and therefore must only be considered as a stop-gap solution.

Another approach to providing network access authentication is based on overloading an existing network layer protocol. The Mobile IPv4 [153] protocol has a built-in authentication mechanism which works in this fashion. Nevertheless, such a solution has very limited applicability as a link layer agnostic method, since it relies on use of the Mobile IPv4 protocol.

## 4.2 Scenarios

The authentication scenarios for Internet remote access identified in this section were adapted from the handling scenarios described in Appendix B of RFC 4058 [184]. The first two, described in sections 4.2.1 and 4.2.2, are categorised in terms of which layer in the protocol stack the security services are provided. Next, in section 4.2.3, we give a scenario covering the case where no security services are provided at the lower layers. We then describe three further scenarios involving respectively mobile IP (section 4.2.4), personal area networks (section 4.2.5), and limited free access (section 4.2.6).

### 4.2.1 Tunnelled Authentication with Physical Security

In Internet access networks where a certain level of security is provided at physical layer, authenticating the remote entity is still important, since the physical layer does not provide information on the remote entity. Instead, if physical layer security is provided, then per-packet authentication (or *message authentication*, as described in section 2.1.2) and encryption do not necessarily need to be provided at higher layers. To illustrate this, we cite DSL networks (as

described in section 3.2.1) implemented on top of point-to-point phone lines. In this type of network, tunnelled authentication can be used both for entity authentication and as a hook to Internet remote access control.

There are a number of possible use scenarios for DSL networks with respect to remote entity configuration and authentication. In DSL networks in which PPP is used for both configuration and authentication, and even IP encapsulation, the providers may not need to migrate to more sophisticated authentication methods such as tunnelled solutions. This is because they can take advantage of the built-in PPP authentication method, without incurring additional costs, and without the limitations associated with inserting another layer in the protocol stack.

By contrast, some DSL networks use configuration methods other than PPP, e.g. DHCP (see section 3.6.5.4) or static IP configuration. Such networks use either an ad hoc network access authentication method, such as http-redirect with web-based login (see section 4.1.5), or no authentication method at all. In this case a new Internet remote access authentication procedure is needed. Thus a tunnelled authentication mechanism that can carry EAP, and/or even a public key based solution or a solution arising from the mobile communication sphere, can be used to meet this requirement.

### 4.2.2 Tunnelled Authentication with Link Security

In a number of situations, link layers might only be protected by security mechanisms outside the scope of an authentication protocol. In such cases, a higher layer authentication protocol carrying EAP can be used to regulate Internet access for remote entities. One example of such a scenario is provided by web-based login (see section 4.1.5) in current Wi-Fi[3] networks. Although in this kind of WLAN it is possible to enable Wired Equivalent Privacy (WEP) secu-

[3] http://www.weca.net/

rity to perform message authentication, WEP fails to meet its security design goal (see, for example, Walker [177]). In particular, the WEP encryption procedure is a fundamentally unsound construction. For further details see, for example, Fluhrer, Mantin and Shamir [63], who present several weaknesses in the mode of operation of the stream cipher RC4 (see section 2.1.3.2) used by WEP[4].

To provide entity authentication, the Wi-Fi Protected Access protocol implements 802.1X [84] and EAP (see section 3.4). Together, these schemes provide a framework for more robust remote entity authentication. This framework uses a central backend (AAA) authentication server, such as RADIUS (see section 3.9.1), to authenticate each remote entity wishing to gain network access.

A different approach can be found in the third generation standards for mobile telecommunication systems, which include W-CDMA UMTS (section 3.5.3 describes this 3GPP[5] standard) and the American CDMA2000 IS-2000 scheme (section 3.5.5 describes this 3GPP2[6] standard). These mobile network standards require remote entity authentication with the radio network (BS/MSC/VLR), before providing connectivity to the mobile access network. Following completion of the ad hoc authentication process specific to the access technology, link layer protection is provided. In particular, CDMA2000 networks offer two types of access service, namely Simple IP and Mobile IP, which we next briefly describe.

**Simple IP:** The *Simple IP* access service requires the remote entity to provide authentication credentials via PPP [168]. A RADIUS [161] based AAA backend infrastructure is used to verify the credentials provided by the remote entity, before network access is provided. Currently CDMA2000 networks in-

[4]IEEE 802.11–2007 [85] specifies the use of the Temporal Key Integrity Protocol to eliminate the known WEP shortcomings.

[5]http://www.3gpp.org

[6]http://www.cdg.org/technology/3g.asp

clude PPP as part of the protocol stack between the Mobile Node (MN), which corresponds to the remote entity, and the Packet Data Serving Node (PDSN), which corresponds to the Access Router.

Consequently, CDMA2000 networks rely on PPP functionality to authenticate a remote user to the access network. However, it is possible that future releases of the standard may not use PPP, but instead may adopt a simple framing scheme, such as High-level Data Link Control (HDLC [104]). In such a scenario, network remote access authentication can be performed over CDMA2000 using a Simple IP service, e.g. using a tunnelled authentication mechanism carrying EAP, where EAP encapsulates an appropriate lightweight authentication method.

**Mobile IP:** When the CDMA2000 MN chooses the *Mobile IP* [153] access service, authentication is performed by the Foreign Agent (FA) in the PDSN, which interacts with an AAA RADIUS [161] server (see Figure 4.1). The MN credentials and the Network Access Identifier (NAI), i.e. the MN identifier, are included in the Mobile IPv4 [153] Registration-Request message, issued by the MN to the FA via the radio network. The FA then uses the NAI and the MN credentials contained in the MN-AAA authentication extension in the RADIUS Access-Request message. After a successful response message from the RADIUS server (Access-Accept), the registration-request message is forwarded to the Home Agent (HA)[7].

This model merges an IP mobility scheme with network remote access authentication. The problem with this authentication model is that it can only be used in IPv4 networks in which every client implements mobile node functionality. A more flexible approach would be to separate network remote access and Mobile IPv4. Such an approach, i.e. a tunnelled authentication mechanism car-

[7]The HA may be assigned by the visited access provider network or by the home IP network.

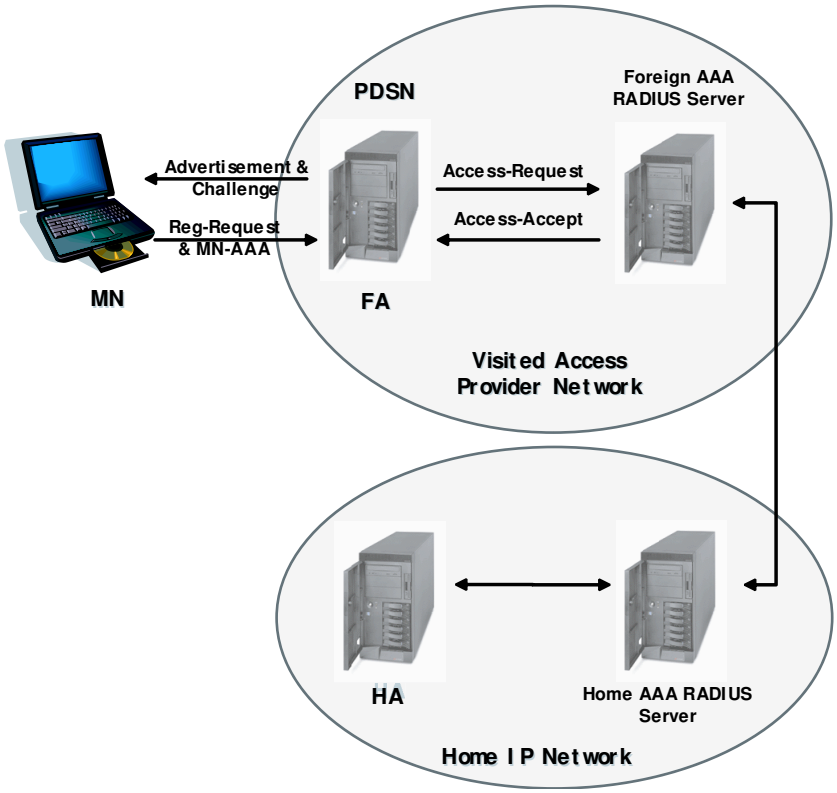


Figure 4.1: **AAA infrastructure for Mobile IPv4 service in a CDMA2000 network**

rying EAP, would be used to authenticate the user for network remote access, and the Mobile IPv4 messages would be sent after the authentication process has completed.

Since EAP is used, such an approach would be more flexible, since it enables different authentication schemes to be supported rather than relying on just the HMAC-MD5 MAC scheme (see section 2.1.3.2), which is the default MAC algorithm for the Mobile IPv4 (MN-AAA) authentication extension [153].

The IP mobility solution for IPv6 networks, described in [110], is slightly different from that proposed for IPv4 networks [153]. When Mobile IPv6 is deployed in CDMA2000 networks, the FA would no longer exist and, for this reason, the IPv4 scheme would no longer work. In such a scenario, the MN would

have to authenticate using another mechanism, and a tunnelled authentication mechanism carrying EAP is a possible solution.

In short, in order to achieve a more flexible model, authentication for network remote access, and authentication/authorisation for enabling IP Mobility, should be separated. This can be accomplished by using a tunnelled authentication mechanism carrying EAP for network remote access, with EAP encapsulating an appropriate authentication method, while allowing Mobile IP implementations to adhere to RFCs 3344 [153] and 3775 [110].

### 4.2.3 Absence of Lower Layer Security

There are scenarios where neither physical nor link layer access security is available on the network. One possible reason for such a scenario is a lack of adequate authentication capabilities in the link layer protocol in use. Link layer technologies generally provide a data encryption service, but inadequate authentication method support. As we have discussed previously, it is desirable to be able to support arbitrary authentication methods, without being limited to those that are specific to the underlying technology. Another cause of missing lower layer authentication is the difficulty of deployment.

Assuring physical security or enabling link layer security might not be practical in a number of scenarios. In the absence of such lower layer security and entity authentication schemes, not only are providers unable to control the use of their networks, but users will potentially also feel insecure while exchanging sensitive information.

In order to support authentication functionality in such systems, many providers today use a higher layer authentication scheme, e.g. http-redirect, commonly known as web-based login (see section 4.1.5). While this method partially solves the problem by allowing only authorised users to access the net-

work, it does not support lower layer security mechanisms, including per-packet (message) authentication and encryption. Moreover, it is a non-standard ad hoc solution that provides only limited authentication method support.

In such scenarios, a standard network layer solution, such as a tunnelled authentication mechanism carrying EAP, would be appropriate, since it would provide link layer agnostic Internet remote access authentication. In fact, a tunnelled authentication mechanism carrying EAP can support a variety of authentication schemes, and is also capable of enabling lower layer security. This solution may be appropriate if it can specify authentication methods that can derive and distribute keys for the authentication, integrity and confidentiality of data traffic, either at the link or the network layer.

For example, if the link layer does not support the desired authentication method but supports encryption, a tunnelled authentication mechanism can be used to bootstrap the latter. On the other hand, if the link layer neither supports the desired authentication method nor encryption, a tunnelled authentication mechanism carrying EAP can be used to bootstrap higher layer security protocols, such as IKE (see section 3.6.2) and IPsec (see section 3.6.5).

Thus use of a tunnelled authentication mechanism carrying EAP can result in a secured network environment, even in cases where the underlying layers do not have built-in security features. Also, assuming EAP will support a variety of authentication schemes, providers will have the advantage of using a single framework across multiple environments. Such flexibility seems likely to be important for heterogeneous network access supporting ubiquitous mobility.

### 4.2.4 Mobile IP

As described in section 4.2.2, Mobile IPv4 defines its own authentication extensions to authenticate and authorise mobile nodes at both foreign agents and

home agents. One of the possible modes of Mobile IPv4 involves the mobile node using a co-located care-of address, and therefore does not rely on any mobility management functionality of the foreign agent on the remote access network.

In this case, a mobile node can send its registration-request message directly to the home agent. Even in the co-located care-of address case, the protocol can require mobile nodes to register with a foreign agent by setting the Registration-Required bit in the agent advertisements. The problem here is that this forces mobile nodes to send their registration-request messages via a foreign agent, even though they would not interact with that agent.

This type of Mobile IP registration is used for performing network remote access authentication. As discussed previously, another problem with this remote authentication model is that it can only be used in IPv4 networks where every client implements mobile node functionality. Even for IPv4 clients, a more flexible approach would be to replace this protocol-specific authentication method by a common authentication protocol, such as a tunnelled authentication mechanism carrying EAP.

A solution of this latter type can be used with any client, regardless of Mobile IPv4 support; it can support various authentication methods, and can also be used with IPv6 clients. Mobile IPv6 [110] does not define a foreign agent in the access networks, or provide any protocol support for access authentication.

### 4.2.5 Personal Area Networks

As defined by Ohba et al. [144], "a personal area network (PAN) is the interconnection of devices within the range of an individual person". For example, connecting a cellular phone, a PDA, and a laptop via short range wireless links would form a PAN.

Devices in a PAN can directly communicate with each other; moreover, they can potentially access the Internet if any one of them is specifically designated as a mobile router and provides gateway functionality. Just like a remote access network, a PAN will typically also require authentication and authorisation prior to granting access to its clients. A mobile router can terminate the link layer from a PAN node, and act as the first-hop router for it.

Additionally, a mobile router can also perform access control as an authentication agent. PAN nodes might be using a variety of different link layer technologies to connect to a mobile router. Therefore, to simplify the task of the router, it is desirable to use authentication methods that are independent of the underlying link, e.g. those relying on a link layer agnostic authentication protocol, such as a tunnelled mechanism carrying EAP.

Another characteristic of a PAN is its small scale. In most cases, a PAN will consist of only a handful of nodes; thus the authentication process does not necessarily require a managed backend AAA infrastructure for credential verification. Locally stored information can be used in a tunnelled authentication deployment carrying EAP, without relying on a AAA backend.

The 3GPP architecture allows separation of a mobile termination (MT) device, such as a cellular phone, and a piece of termination equipment (TE), such as a laptop [182]. A TE can be connected to the Internet via a MT by establishing a PPP connection. One or more TEs can be connected to a MT to form a PAN. The current architecture does not allow direct communication between the TEs (if more than one are connected to the MT) without having to go through the cellular interface of the MT.

This architecture will benefit from using shared links (e.g. using Ethernet) between the TE and the MT. Shared links would allow TEs to communicate directly with one another, without having to send data through the power-limited MT, or over the expensive air interface. A tunnelled authentication mechanism

carrying EAP can be used for authenticating PAN nodes when shared links are used between TEs and the MT.

### 4.2.6 Limited Free Access

As stated by Ohba et al. [144], certain networks might allow clients to access a limited part of the network topology without any explicit authentication and authorisation. For example, in an airport network, information such as flight arrival and departure gate numbers, and information about airport shops and restaurants, are offered as free services by the airlines or airport authorities for their passengers. In order to access such information, users can simply plug their devices into the network, without performing any authentication.

The network will typically only offer link layer connectivity and limited network layer access to users. Access to further services or sites, using such local networks, requires authentication and authorisation. If users want such services, the access network can detect that attempt and initiate a user authentication procedure. This also allows the network to initiate authentication whenever appropriate. Once a user has successfully performed the authentication procedure, it will be allowed to go beyond the free access zone.

A tunnelled authentication mechanism carrying EAP can be an enabler to such limited free access scenarios, and can also offer a flexible access control framework for public hot-spot networks.

# Chapter 5

# Internet Remote Access Requirements

## Contents

---

The aim of this chapter is to provide a sound basis for the assessment of candidate entity authentication protocols against Internet remote access requirements. We define two main requirement sets, namely security requirements and implementation requirements.

Firstly, to establish the security requirements, we analyse and compare potential risks associated with entity authentication protocols, examining a number of aspects of entity authentication security for Internet remote access (section 5.1). Secondly, to obtain the implementation requirements, we analyse and compare features such as complexity, flexibility and performance (section 5.2).

The result of this critical analysis is then used to derive the security and implementation services and properties required of new entity authentication schemes for Internet access. These requirements are used to define and limit the scope of this thesis (section 5.3).

## 5.1 Security Requirements

The provision of a secure network access service requires the implementation of access control based on the mutual authentication and authorisation of clients and access networks. Initial and subsequent client-to-network authentication methods provide parameters that are needed to police the traffic flow through the enforcement points[1]. This thesis focuses on authentication protocols that carry such parameters between the client and the access network.

In this section, we analyse and compare potential risks associated with the entity authentication protocols considered in this thesis, examining a number of aspects of security for Internet remote access, including: client authentication (section 5.1.1), key establishment (section 5.1.2), use of EAP methods (section 5.1.3), mutual entity authentication (section 5.1.4), key freshness (section 5.1.5), re-authentication (section 5.1.6), authorisation, access control and accounting (section 5.1.7), AAA Backend infrastructure (section 5.1.8), absence of a secure channel (section 5.1.9), Denial-of-Service attacks (section 5.1.10), and client identity confidentiality (section 5.1.11). Some of the security requirements described here were adapted from RFC 4058 [184].

### 5.1.1 Client Authentication

New Internet remote access authentication schemes, such as those proposed in this thesis, must enable authentication of the client, i.e. the remote device, to the access network. This involves the client providing the credentials (see section 2.2.1) necessary to prove its identity. A client identifier can be authenticated by verifying the credentials supplied by one of the users of the device, or by the device itself.

[1] An enforcement point is a node on the access network where per-packet enforcement policies (i.e. filters) are applied on the inbound and outbound traffic of client devices.

Once network access is granted to the device, methods that may be used by the device to control which users can access the network are outside the scope of this thesis. After a successful client authentication procedure for remote network access, the methods that might be used to provide message authentication (section 2.1.1.2), integrity (section 2.1.1.3), and replay protection (sections 2.2.4 and 2.2.5) for data traffic, are also outside the scope of this thesis. That is, we focus here purely on the authentication and key establishment processes, and not on subsequent use made of the authenticated channel and/or keys that may have been established.

### 5.1.2 Key Establishment

As discussed in section 2.2.4, a key establishment facility enables network remote access authentication schemes to be linked to an integrity service, in order to provide ongoing data origin authentication and integrity. To achieve this, the entity authentication protocol needs to be integrated with a key establishment mechanism, such that a by-product of successful authentication is a session key, appropriate for use with an integrity mechanism used to protect subsequently exchanged data.

The following example shows the importance of providing this feature in Internet remote access authentication schemes. Certain types of service theft are possible when the device identifier of the remote client is not protected during or after an authentication protocol exchange; see, for example, [151]. Internet remote access methods should thus have the capability to exchange device identifiers securely between the authentication client and the access network, in cases where the network is vulnerable to MitM attacks (see section 3.2.3). One way of solving this problem (see, for example, [18]) requires cryptographic key generation to take place at both the remote client and in the access network.

### 5.1.3 Use of EAP Methods

Since the EAP protocol (see section 3.4) is very flexible and can encapsulate arbitrary authentication methods (section 4.1.3), it is clearly a protocol that satisfies many of the requirements for a variety of authentication scenarios. Therefore we subsequently assume that Internet remote access authentication schemes will make use of a tunnelled authentication mechanism carrying EAP (see section 4.1.2).

In networks which are not physically secured against unauthorised use (see section 4.2.3), link-layer or network-layer encryption mechanisms, such as IPsec (see section 3.6.5), can be used to provide such security. However, these mechanisms require the presence of keying material at the authentication client (see section 2.1.3.3).

Many EAP methods are capable of generating initial keying material, but this material cannot be directly used with IPsec. This is because it lacks the properties of an IPsec SA (see section 3.6.5.3), which includes secure cipher suite negotiation, mutual proof of possession of keying material (see section 2.2.5), and freshness of transient session keys (see section 2.2.5). However, these initial EAP keys can be used with an IPsec key management protocol, such as IKE (see section 3.6.2), to generate the required security associations. A separate 'secure association protocol', such as ISAKMP (see section 3.6.1), is required to generate an IPsec SA using the EAP keys.

### 5.1.4 Mutual Entity Authentication

The authentication client and the network may be able to perform mutual authentication in some Internet remote access schemes. Indeed, just providing the capability for the network to authenticate the client may not always be sufficient. Nevertheless, a mutual authentication capability is not always required.

For example, clients might not need to authenticate the access network when physical security is available to enable the client to implicitly authenticate the network (e.g. dial-up networks).

Moreover, as described in section 2.2.5, although mutual authentication is very commonly seen as the necessary precursor to the establishment of a secure connection in any environment, there do exist examples of cases where mutual authentication is not necessary, and, indeed, may impose unnecessary overheads on session establishment. Hence, and following [136], we claim that (mutual) entity authentication is not always an essential precursor for the establishment of secure communications. In some cases, the most important issue is to ensure that the properties of (implicit) key authentication and key freshness are provided for any established session keys. These session keys can be used to protect the integrity of security-sensitive data exchanged during the session, thereby preventing MitM attacks.

### 5.1.5 Key Freshness

As stated in section 2.2.5, a further property, useful in many applications, is key freshness. The absence of key freshness would enable an interceptor to force the verifier to keep re-using an 'old' session key, which might have been compromised. It would therefore seem reasonable to make key freshness a requirement for any key establishment processes within an authentication protocol designed for use in the Internet remote access environment.

### 5.1.6 Re-Authentication

As described in section 2.2.4, authentication protocols provide assurance regarding the identity of an entity *only* at a given instant in time. Thus the authenticity of the entity can be ascertained just for the instance of the au-

thentication exchange. If continuity of such an assurance is required, use of additional techniques is necessary. For example, authentication can be repeated periodically.

New entity authentication schemes, such as those defined in this thesis, should thus be capable of supporting both periodic and on-demand re-authentication. Moreover, both the remote client and the access network should be able to initiate the initial authentication and the re-authentication processes.

### 5.1.7 Authorisation, Access Control, and Accounting

After a device has been authenticated by Internet remote access methods, it will be authorised for network access. That is, the core requirement of Internet remote access schemes is to verify if the client device has the authorisation to send and receive IP packets. It may also be possible to provide finer granularity authorisation, such as authorisation for use of individual network services (e.g. use of http or ssh services).

While a backend authorisation infrastructure, e.g. RADIUS (see section 3.9.1) or Diameter (see section 3.9.2), might provide the necessary authorisation information to the access network, explicit support for authorisation functionality is outside the scope of this thesis. Therefore, in assessing possible new authentication schemes for Internet remote access, we do not consider the possible need for the access network to provide service authorisation information to the authenticated client device.

Client remote access authentication should be followed by access control, to make sure only authenticated and authorised clients can send and receive IP packets via the access network. Access control would typically involve implementing access control lists on the enforcement points. Although Internet remote access schemes identify clients that are authorised to access the net-

work, providing access control functionality in the network is outside the scope of this thesis.

Finally, issues associated with the transfer and management of accounting data are also outside the scope of this thesis.

### 5.1.8 AAA Backend

Internet remote access protocols, such as those proposed in this thesis, must not make any assumptions regarding the backend authentication mechanisms. An access network may interact with backend AAA infrastructures, such as RADIUS (see section 3.9.1) or Diameter (see section 3.9.2), but it is not a requirement. If the access network does not rely on a specific AAA protocol, e.g. RADIUS or Diameter, it can use a proprietary backend system, or rely on locally stored information.

The interaction between the access network and the backend authentication entities is outside the main scope of this thesis.

### 5.1.9 Secure Channel

Authentication schemes for Internet remote access must not assume the presence of a secure channel between the remote client and the access network. They need to be able to provide a secure authentication service in networks which are not protected against packet eavesdropping and spoofing. They should provide protection against replay attacks on both the client device and the access network.

Addressing this requirement partially relies on the mandatory use of EAP methods (see section 5.1.3). Use of EAP methods that provide mutual authentication and key derivation/distribution is essential to satisfy this requirement.

EAP does not rely on the presence of a secure channel, and supports a variety of authentication methods that can be used in such environments.

In addition, entity authentication protocols for Internet remote access should not contain vulnerabilities that can be exploited when they are used over insecure channels. Following RFC 4058 [184], they may provide a secure channel by deploying a two-phase authentication process. The first phase can be used for the creation of a secure channel, and the second phase for client and access network authentication.

### 5.1.10 Denial-of-Service Attacks

Authentication schemes proposed for Internet remote access need to be robust against Denial-of-Service attacks, in particular against 'blind resource consumption DoS attacks' (see section 3.2.3). Such attacks could swamp the access network, causing it to expend all available resources, and prevent network access by legitimate clients.

### 5.1.11 Client Identity Confidentiality

Some remote clients might prefer to hide their identity from visited access networks for privacy reasons. Providing identity confidentiality for remote clients is a potentially valuable feature, that it would be desirable for new authentication schemes proposed for Internet remote access to provide (at least as an option).

## 5.2 Implementation Requirements

In this section, we analyse and compare implementation features of entity authentication protocols applicable to this thesis, including: client identifiers (sec-

tion 5.2.1), IP address assignment (section 5.2.2), EAP lower layer requirements (section 5.2.3), flexibility (section 5.2.4), performance (section 5.2.5), complexity (section 5.2.6), and IP version independence (section 5.2.7). Some of the implementation requirements described in this section were adapted from RFC 4058 [184].

### 5.2.1 Client Identifiers

Any authentication scheme proposed for use in an Internet remote access environment should support a variety of client identifier types (e.g. username, Network Access Identifier, etc.), as well as a variety of remote device identifier types (e.g. IP address, link-layer address, port number of a switch, etc.).

An access network needs to be able to create a binding between the client identifier and the associated device identifier upon successful entity authentication. This can be achieved as a result of the authentication method communicating the client identifier and the device identifier to the access network during the protocol exchange. In order to prevent unauthorised access, the device identifier can be cryptographically protected; this case is described in RFC 4016 [151]. In this case, the keying material required by the cryptographic methods needs to be indexed by the device identifier.

The binding between the client identifier and the associated device identifier is typically used for access control and accounting in the access network (see section 5.1.7).

### 5.2.2 IP Address Assignment

Assigning an IP address to the client of an authentication scheme for Internet remote access is outside the scope of this thesis. We simply note here that the

authentication client needs to configure an IP address before running the entity authentication method.

### 5.2.3 EAP Lower Layer Requirements

EAP imposes many requirements on the underlying transport protocol that must be satisfied if EAP is to operate correctly. RFC3748 [13] describes the generic transport requirements to be satisfied by Internet remote access schemes making use of EAP.

### 5.2.4 Flexibility

Entity authentication schemes for Internet remote access need to support client devices with multiple interfaces, and access networks with multiple routers on multi-access links. In other words, they should not assume that the client device has only one network interface, that the access network has only one first hop router, or that the remote client device is using a point-to-point link.

### 5.2.5 Performance

An Internet remote access method must efficiently handle the authentication process in order to gain network access with minimum latency. For example, it might minimise protocol signalling by creating local security associations.

### 5.2.6 Complexity

Following the example shown in section 4.1.3, if a remote entity wishes to access real time media applications available on the Internet through an access network, delay is an undesirable feature. Hence, in such situations, it would be highly

desirable if the number of round trips needed by the authentication protocol could be minimised.

By using EAP to carry lightweight authentication methods, it is possible to create authentication solutions with low complexity at the application layer. This can be achieved, for instance, through the EAP encapsulation of lightweight authentication protocols arising from the mobile telecommunications sphere.

### 5.2.7 IP Version Independence

It is desirable that authentication schemes for Internet remote access can work with both IPv4 and IPv6.

## 5.3 Services and Properties of New Authentication Protocols

In this section, the result of the critical analyses made in sections 5.1 and 5.2 is used to deduce the security (section 5.3.1) and implementation (section 5.3.2) services and properties required of new entity authentication protocols for Internet remote access.

### 5.3.1 Security Services and Properties

The security services and properties required of new authentication schemes for Internet access are as follows:

- entity authentication service for remote network access, verifying the client supplied credentials;

- key establishment services with the key freshness property;
- use of a tunnelled authentication mechanism carrying EAP;
- mutual entity authentication services between the remote client and the access network;
- use of periodic and on-demand re-authentication techniques;
- possible interaction between the access network and backend AAA infrastructures (without the need to rely on a specific AAA protocol);
- absence of vulnerabilities that can be exploited over insecure channels (protecting against replay attacks, eavesdropping and spoofing on both the client device and the access network);
- robustness against DoS attacks, especially against 'blind resource consumption DoS attacks' (see section 3.2.3); and
- identity confidentiality service for remote clients.

### 5.3.2 Implementation Services and Properties

The implementation services and properties required of new authentication schemes for Internet access are as follows:

- support for a variety of identifier types for both authentication clients and remote devices, including the ability to create a cryptographic binding between the client identifier and the associated device identifier (upon successful entity authentication protocol exchange);
- satisfaction of the EAP generic transport requirements;
- flexibility, by offering support to client devices with multiple interfaces, and access networks with multiple routers on multi-access links;

- performance, by efficiently handling the authentication process, in order to gain network access with minimum latency;
- low complexity, with the goal of reducing the delay by using EAP encapsulation of lightweight authentication protocols; and
- IP version independence.

# Chapter 6

# PANA as the Target Transportation Environment

## Contents

The aim of this chapter is to justify the selection of the Protocol for carrying Authentication for Network Access (PANA) as the target environment for transporting the new Internet entity authentication schemes subsequently proposed in this thesis. This Chapter describes the PANA protocol in more detail (section 6.1), before explaining the reasons for choosing it as the transportation environment (section 6.2).

## 6.1 PANA Framework

IP based remote hosts that connect to the Internet via an access network will typically need to provide their credentials and be authenticated before being authorised to access the network. There is currently no generic network layer protocol to be used to authenticate a user device requesting network access. The IETF PANA protocol [65] is intended to address this issue.

As stated in section 3.7.5, PANA carries any authentication mechanism that can be specified as an EAP method (see section 3.4), and can be used on any link that supports IP. The PANA protocol specification provides the client-to-network access authentication component of an overall secure network access framework.

The aim of this section is to give a detailed description of the PANA framework. Firstly we identify the goals of PANA and give an overview of the IETF draft PANA protocol (section 6.1.1); we also list the terms used frequently in PANA documents (section 6.1.2). Secondly, a brief description of the payload of a PANA message, consisting of a series of Attribute Value Pairs (AVPs), is provided (section 6.1.3). After that, we identify five distinct phases of a PANA session, and describe them (section 6.1.4). We then summarise the PANA security association establishment process (section 6.1.5).

### 6.1.1 PANA Goals and Overview

The draft PANA protocol [65] provides a link layer agnostic and IP compatible transport for EAP (see section 3.4), that allows a remote host to be authenticated for network access. That is, PANA is a link layer agnostic transport for EAP that enables client-to-network access authentication between a user device (PANA Client or PaC) and a device at the network access point (PANA Authentication Agent or PAA), where the network access device may optionally

be a client of an AAA infrastructure (see sections 3.7.5 and 3.9). A summary of the PANA protocol is given in Figure 6.1.

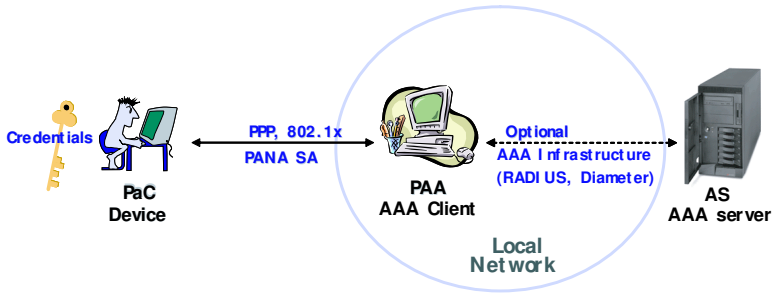


Figure 6.1: **PANA protocol overview**

The scope of the PANA draft [65] is thus the design of a link-layer agnostic transport for network access authentication methods, where the EAP protocol provides such authentication methods. In other words, PANA will carry EAP, which can carry a variety of authentication methods. By virtue of enabling transport of EAP above IP, any authentication method that can be carried as an EAP method is made available to PANA, and hence to any link-layer technology. As described in RFC 3748 [13], there is a clear division of labour between PANA (an EAP lower layer), EAP, and EAP methods.

PANA is an UDP-based [156] protocol. It has its own retransmission mechanism to reliably deliver messages. As stated in [65], 'the PANA protocol messaging consists of a series of requests and responses, some of which may be initiated by either end. Each message can carry zero or more AVPs as payload. The main payload of PANA is EAP which performs authentication'. PANA helps the PaC and the PAA to establish an EAP session.

A variety of access network scenarios can arise. For example, security services may or may not be provided at lower layers in the protocol hierarchy, and a variety of different client IP configuration and authentication methods might be deployed. The IETF draft 'PANA Framework' [109] defines a general frame-

work describing how these various deployment choices are handled by PANA and the access network architectures.

Appendix A of RFC 4058 [184] contains the problem statement that led to the development of PANA, whilst appendix B identifies a variety of environments and scenarios for PANA. Potential security threats for network-layer access authentication protocol are detailed in RFC 4016 [151]. The requirements for the PANA protocol are defined in RFC 4058 [184]. Some of these requirements are imposed by the chosen payload, i.e. EAP (see section 3.4).

### 6.1.2 PANA Terminology

Terminology frequently used when discussing the PANA protocol is as follows [65]:

**PANA Client (PaC).** The client side of the PANA protocol that resides in the access device (e.g. laptop, PDA, etc.). It is responsible for providing the credentials in order to prove its identity for the purposes of network access authorisation. The PaC and the EAP peer (see section 3.4) are assumed to be located in the same access device.

**Device Identifier (DI).** The identifier used by the network to control and police the network access of a remote device. Depending on the access technology, this identifier may contain an address that is carried in protocol headers (e.g. an IP or a link-layer address), or a local identifier that is made available by the local protocol stack of a connected device (e.g. a PPP interface id).

**PANA Authentication Agent (PAA).** The PANA protocol entity in the access network responsible for verifying the credentials provided by a PaC, and also for authorising network access to the client device, as identified by a DI. The PAA and the EAP authenticator (and optionally the EAP

server) are assumed to be located in the same node. The authentication and authorisation procedure can, according to the EAP model (see section 3.4), also be offloaded to the backend AAA infrastructure (see section 3.9).

**PANA Session.** A PANA session begins with a handshake between the PaC and the PAA, and terminates as a result of an authentication or liveness test failure, a message delivery failure after the number of retransmissions reaches a maximum value, session lifetime (see below) expiration, or an explicit termination message. A fixed session identifier (see below) is maintained throughout a session. A session cannot be shared across multiple network interfaces. Only one DI can be bound to a PANA session.

**Session Lifetime.** A time period associated with a PANA session, which limits its lifetime. For an established PANA session, the session lifetime is bound to the lifetime of the current authorisation given to the PaC. The session lifetime can be updated by a new round of EAP authentication, as long as this occurs before the session expires.

**Session Identifier.** The session identifier (*Session-Id*) is used to uniquely identify a PANA session between a PAA and a PaC. It includes an identifier for the PAA, and therefore it cannot be shared across multiple PAAs. It is included in a PANA message to bind the message to a specific PANA session. This bidirectional identifier is allocated by the PAA following the handshake, and is freed for re-use when the session terminates.

**PANA Security Association (PANA SA).** A PANA security association between a PaC and a PAA is made up of stored cryptographic keying material and associated context. The security association is used to protect bidirectional PANA signalling traffic between the PaC and the PAA.

**Network Access Server (NAS).** A network device that provides access to the network.

**Authentication Server (AS).** An entity that authenticates the PaC. It may

be co-located with the PAA, or it may be part of the AAA backend infrastructure (see section 3.9).

**Enforcement Point (EP).** A node on the access network at which per-packet enforcement policies (i.e. filters) are applied to the traffic of access devices. The PAA provides the unique DI to each client, together (optionally) with cryptographic keys to be used to support client-based filtering by the EP. The EP and the PAA may be co-located.

**Network Access Provider (NAP).** A service provider that provides physical and link-layer connectivity to an access network that it manages.

**AAA-Key.** A key derived by the EAP peer and EAP server and transported to the EAP authenticator. A complete specification of the framework for EAP key derivation, including the generation and use of EAP keys by EAP methods and AAA protocols, is given in the 2007 Internet Draft 'EAP Key Management Framework' [18].

### 6.1.3 PANA Payload (AVPs)

The payload of any PANA message consists of a number (possibly zero) of AVPs [65]. Possible PANA AVP types are as follows. A summary of the PANA header format is given in section 3.7.5.

**Cookie AVP:** contains a random value generated by the PAA and used for making PAA discovery robust against 'blind resource consumption DoS attacks' (see section 3.2.3). For further details, see section 6.1.4.1.

**Protection-Capability AVP:** contains the type of per-packet protection provided, based on a link-layer or a network-layer cryptographic mechanism enabled after the PANA authentication process.

**Device-Id AVP:** contains a device identifier for the PaC or the EP.

**EAP AVP:** contains an EAP payload.

**MAC AVP:** contains a Message Authentication Code (see section 2.1.3.2) that protects the integrity of a PANA message.

**Termination-Cause AVP:** contains the reason for session termination.

**Result-Code AVP:** contains information about the protocol execution results.

**Session-Id AVP:** contains the PANA session identifier value.

**Session-Lifetime AVP:** contains the duration of authorised access.

**Failed-AVP:** contains an offending AVP that caused a failure.

**Provider-Identifier AVP:** contains the identifier of a NAP or an Internet Service Provider (ISP).

**Provider-Name AVP:** contains the name of a NAP or an ISP.

**NAP-Information AVP, ISP-Information AVP:** contains the identifier of a NAP and an ISP, respectively.

**Key-Id AVP:** contains an AAA-Key identifier (see section 6.1.2).

**PPAC AVP:** Post-PANA-Address-Configuration AVP. Used to indicate the available/chosen IP address configuration methods that can be used by the PaC after successful PANA authentication.

**Nonce AVP:** contains a randomly chosen value (see sections 8.9 and 11.5 of [65]) that is used in PANA cryptographic key computations, e.g. as a PANA SA attribute; this AVP 'must be included in the first PANA-Auth-Request and PANA-Auth-Answer messages in the authentication and authorisation phase [described in section 6.1.4.2] when stateless PAA discovery is used' (see section 4.3 of [65]).

**Notification AVP:** contains a displayable message.

### 6.1.4 PANA Phases

PANA messages are sent between the PaC and PAA as part of a PANA session. A PANA session, illustrated in Figure 6.2, consists of five distinct phases [65]: the discovery and handshake phase (section 6.1.4.1), the authentication and authorisation phase (section 6.1.4.2), the access phase (section 6.1.4.3), the re-authentication phase (section 6.1.4.4), and the termination phase (section 6.1.4.5).

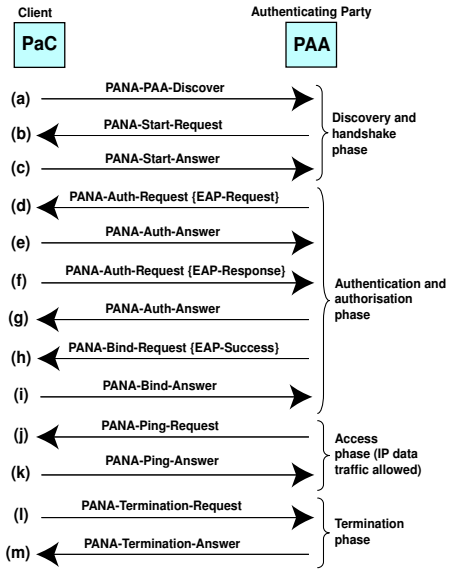


Figure 6.2: **Illustration of PANA messages in a session**

#### 6.1.4.1 Discovery and Handshake Phase

The *discovery and handshake* phase initiates a new PANA session. The PaC discovers the PAA(s) by either explicitly soliciting advertisements, or receiving unsolicited advertisements. The PaC's answer, sent in response to an advertisement, starts a new session. A complete specification of the discovery and handshake phase is given in section 4.3 of the PANA Internet Draft [65].

When a PaC attaches to a network, it may or may not know the IP address of the PAA. If it knows the PAA address, then it sends the PAA a PANA-PAA-Discover message, as shown in Figure 6.2 ($a$), and initiates the PANA exchange. If the PaC does not know the IP address of the PAA, it relies on dynamic discovery methods, such as 'multicast-based discovery' [65] to determine it. This involves the PaC sending a PANA-PAA-Discover message ($a$) to a scoped multicast address and UDP port. The multicast scope is configured such that the discovery messages only reach the designated PAA. Details of this scope configuration are given in RFC 2365 [134].

In both situations, the PAA responds with a PANA-Start-Request message ($b$). There may be more than one PAA in the access network, and thus the PaC may receive multiple PANA-Start-Requests. By default, the PaC chooses the PAA that sends the first PANA-Start-Request. The PaC then responds with a PANA-Start-Answer message ($c$), indicating it wishes to enter the authentication and authorisation phase.

A PANA-Start-Request message ($b$) may carry a Cookie AVP (see section 6.1.3), which contains a *cookie*, i.e. a random value generated by the PAA. This cookie is used to protect the PAA against 'blind resource consumption DoS attacks' (see section 3.2.3), launched by attackers bombarding the PAA with PANA-PAA-Discover messages ($a$).

If the PANA-Start-Request ($b$) contains a Cookie AVP, then the PANA-Start-Answer ($c$) must contain the cookie value copied from the request. When the PAA receives the PANA-Start-Answer ($c$), it checks whether the cookie it contains has the expected value (if no cookie is present then the received message is discarded). If the cookie is valid, the protocol enters the authentication and authorisation phase. Otherwise, it discards the received message.

A Protection-Capability AVP and a PPAC AVP (see section 6.1.3) may also be included in the PANA-Start-Request ($b$), in order to indicate the network

capabilities.

#### 6.1.4.2 Authentication and Authorisation Phase

The discovery and handshake phase is followed by the *authentication and authorisation* phase, which involves the transfer of EAP payloads between the PAA and PaC. The EAP payloads carry an EAP method (see section 3.4). A complete specification of the authentication and authorisation phase is given in section 4.4 of the PANA Internet Draft [65]. At the end of this phase, the PAA conveys the result of the authentication and authorisation process to the PaC. This phase may involve execution of two EAP sessions, one for the NAP and one for the ISP.

As shown in Figure 6.2, EAP-Request ($d$) and Response ($f$) messages are carried in PANA-Auth-Requests. PANA-Auth-Answer messages, i.e. ($e$) and ($g$), are typically used to acknowledge receipt of the requests. As an optimisation, a PANA-Auth-Answer may also carry the EAP-Response message.

PANA optionally allows execution of two separate authentication methods, one with the NAP and one with the ISP, within the same PANA session. When performed separately, the result of the first EAP authentication process is signalled via an exchange of PANA-FirstAuth-End-Request and PANA-FirstAuth-End-Answer messages, which distinguishes the execution of the first authentication method from the second. For further details on the NAP and ISP authentication processes, see section 4.8 of the PANA Internet Draft [65].

The result of the PANA protocol is sent to the PaC in a PANA-Bind-Request message ($h$). This message carries the final EAP authentication result (whether it is the second EAP result of separate NAP and ISP authentication exchanges, or the single EAP result) and the result of the PANA authentication procedure. The PANA-Bind-Request ($h$) is acknowledged with a PANA-Bind-Answer mes-

sage ($i$).

When an EAP method (see section 3.4) capable of deriving keys is used, and the keys are successfully derived in this phase, the PANA messages that carry the EAP-Success message and any subsequent message will also contain a MAC AVP (see section 6.1.3).

The PANA-Bind message exchange is also used to bind the device identifiers of the PaC and EP to the PANA SA (see section 6.1.5). The PANA-Bind-Request message may contain a Protection-Capability AVP (see section 6.1.3) to indicate that link-layer or network-layer encryption will be enabled after the authentication process[1].

#### 6.1.4.3 Access Phase

After a successful authentication process, the client device gains access to the network, and can thus send and receive IP data traffic through the EP. A complete specification of the *access* phase is given in section 4.5 of the PANA Internet Draft [65]. At any time during the access phase, as shown in Figure 6.2, the PaC and PAA may optionally ping each other to test the liveness of the PANA session, using PANA-Ping-Request ($j$) and PANA-Ping-Answer ($k$) messages, which carry a Session-Id AVP (see section 6.1.3). Both the PaC and the PAA are allowed to send a PANA-Ping-Request ($j$) to the communicating peer, and expect the peer to return a PANA-Ping-Answer ($k$).

When an appropriate PANA SA is available (see section 6.1.5), the PANA-Ping messages will be protected with a MAC AVP (see section 6.1.3).

[1] If the PaC does not support the protection capability indicated in this AVP, it sends a PANA-Error-Request message back to the PAA and terminates the PANA session.

#### 6.1.4.4 Re-authentication Phase

If successful, the authentication and authorisation phase determines the PANA session lifetime. However, as described in section 6.1.2, this session lifetime can be updated by conducting a new round of EAP authentication (see section 3.4) before the session expires. During the access phase, the PANA session can thus enter the *re-authentication* phase, in order to extend the current session lifetime by re-executing the EAP method. Once the re-authentication phase has successfully completed, the session re-enters the access phase; otherwise, the session is deleted. A complete specification of the re-authentication phase is given in section 4.6 of the PANA Internet Draft [65].

The (optional) re-authentication phase may be triggered by both the PaC and the PAA. The re-authentication procedure is summarised in Figure 6.3. In this figure, the name of each message is shown, followed by the sequence number in round brackets; square brackets are used to indicate the contents of the message.

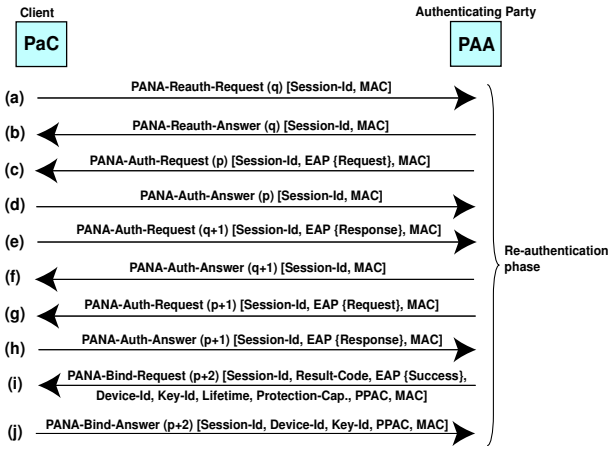


Figure 6.3: **Re-authentication phase initiated by the PaC**

When the PaC initiates the re-authentication phase, it sends a PANA-Reauth-Request message ($a$). This message contains a Session-Id AVP (see

section 6.1.3), which identifies the PANA session to the PAA. If the PAA has an established PANA session with a matching session identifier, it responds with a PANA-Reauth-Answer ($b$), followed by a PANA-Auth-Request ($c$) to re-execute the EAP method carried by PANA ($c$ to $j$); otherwise, it responds with a PANA-Error-Request message.

When the PAA initiates the re-authentication phase, it sends a PANA-Auth-Request ($c$), containing the session identifier, to the PaC. The PaC then enters the re-authentication phase by re-executing the EAP method carried by PANA ($c$ to $j$). The PAA must initiate the re-authentication phase before the current session lifetime expires.

As shown in Figure 6.3, if there is an established PANA SA (see section 6.1.5), all PANA-Reauth, PANA-Auth and PANA-Bind messages sent in the re-authentication phase will be protected with a MAC AVP (see section 6.1.3). Any subsequent EAP routine will be performed with the same ISP and NAP as were selected during the discovery and handshake phase. Re-authentication of an on-going PANA session must maintain the existing sequence numbers in the PANA header (see section 3.7.5). Also, the value of the 'S-flag' in the header of PANA messages (see section 3.7.5) needs to be inherited from the previous authentication/authorisation (or re-authentication) phase.

#### 6.1.4.5 Termination Phase

The PaC or PAA may choose to discontinue the access service at any time. The *termination* phase, a routine for explicitly terminating a PANA session, can thus be initiated either by the PaC (i.e. *disconnect indication*) or the PAA (i.e. *session revocation*). A complete specification of the termination phase is given in section 4.7 of the PANA Internet Draft [65]. The PANA-Termination-Request ($l$) and PANA-Termination-Answer ($m$) messages, shown in Figure 6.2, can be used for both disconnect indication and session revocation procedures.

The reason for termination is indicated in the Termination-Cause AVP (see section 6.1.3). If there is an established PANA SA (see section 6.1.5), all messages exchanged during this phase will be protected with a MAC AVP (see section 6.1.3). When the sender of the PANA-Termination-Request ($l$) receives a valid acknowledgment, all states maintained for the PANA session must be deleted immediately.

If the PaC or the PAA disconnects without engaging in termination messaging, it is expected that either the expiry of the session lifetime or failed liveness tests will clean up the session at the peer.

### 6.1.5 PANA Security Association

The PANA authentication protocol can be linked to an ongoing integrity service. In this case, in line with section 2.2.4, PANA can be integrated with a key establishment mechanism, such that a by-product of successful EAP entity authentication is a shared secret, i.e. a *fresh* (see section 2.2.5) and unique *session key*, appropriate for use with an integrity mechanism used to protect subsequently exchanged data. This assumes that the chosen EAP method allows *session key derivation* (see section 2.1.3.3). The session key is available for the PaC as part of the *authentication and key exchange* procedure (see section 2.1.2) of the selected EAP method. The PAA can obtain the session key from the EAP server via an AAA infrastructure (see sections 3.7.5 and 3.9), if one is being used. The Diameter Cryptographic Message Syntax (CMS) draft [32] describes how a session key can be securely carried (i.e. CMS protected or wrapped) between AAA servers.

Cryptographic protection of messages between the PaC and PAA is thus possible as soon as the EAP protocol (see section 3.4), in conjunction with the EAP encapsulated method (see section 4.1.3), exports a shared session key. This session key is used to create a *PANA security association* [65], which provides

per-message integrity protection and authentication services (see section 2.1.1). A complete specification of the PANA SA is given in section 5.3 of the PANA Internet Draft [65]. The establishment of a PANA SA is required in environments where no physical or link layer security is available (see section 4.2.3).

A PANA SA is created as an attribute of a PANA session, when EAP succeeds with the creation of an AAA-Key[2]. When two EAP sessions are performed in sequence, as in the case where separate NAP and ISP authentication processes (see section 6.1.4.2) are performed, it is possible that two AAA-Keys are derived. If this happens, then the PANA SA will be generated using both AAA-Keys.

When a new AAA-Key is generated in the re-authentication phase (see section 6.1.4.4), any key derived from the old AAA-Key needs to be updated using the new AAA-Key. In order to distinguish the new AAA-Key from previous keys, a Key-Id AVP (see section 6.1.3), which contains an AAA-Key identifier, is carried either in the PANA-Bind messages, as shown in Figure 6.3 ($i$) and ($j$), or in the PANA-FirstAuth-End messages (see section 6.1.4.2) at the end of the EAP method which was used to generate the AAA-Key.

PANA messages carrying a Key-Id AVP need to be protected with a MAC AVP (see section 6.1.3). The MAC AVP value field is computed using a new PANA_MAC_KEY value derived either from the new AAA-Key or from the new pair of AAA-Keys, in the case of separate NAP and ISP authentication processes (see section 6.1.4.2). More information on the computation of the MAC AVP value field can be found in section 5.4 of the PANA draft [65].

The PANA session lifetime is bounded by the authorisation lifetime granted by the authentication server (as for the AAA-Keys lifetime). The lifetime of the PANA SA is the same as the lifetime of the PANA session. The created PANA

[2]A PANA SA is not created when the PANA mechanism fails, or if no AAA-Key is produced by an EAP method.

SA is deleted when the corresponding PANA session is deleted.

## 6.2 Reasons for Choosing PANA

The aim of this section is to justify the selection of the PANA protocol as the target environment for transporting the new Internet authentication schemes proposed later in this thesis.

Firstly, we describe a variety of trust relationships and threat scenarios which affect the PANA method, by analysing and comparing potential risks associated with protocols used to carry authentication for network access (section 6.2.1). Secondly, the PANA security requirements arising from these threats will be established; we also identify the PANA implementation features (section 6.2.2).

The result of this critical analysis is then used to derive the services and properties required of the PANA protocol. These are assessed against the services and properties required of new entity authentication methods for Internet access, as listed in section 5.3. This assessment is used to validate the choice of PANA as the target transportation environment of the new authentication schemes proposed here (section 6.2.3).

### 6.2.1 PANA Threat Analysis

As stated in RFC 4016 [151], the PANA protocol will be used in network access environments where 'there is no a priori trust relationship or security association between the PaC and the PAA or EP'. In these environments, the link between the PaC and the PAA may be a shared medium. In addition, the PaCs may not trust each other, and any PaC (or any other entity with access to the shared medium) might pretend to be a PAA, spoof IP addresses, or launch a variety of other attacks. In the context of the above network access environments, there

are a variety of scenarios which affect the PANA threat model.

In this section, we examine two important aspects of the PANA threat model for network access environments, namely trust relationships (section 6.2.1.1) and threat scenarios (section 6.2.1.2).

#### 6.2.1.1 PANA Trust Relationships

The pairs of entities that must share a trust relationship before use of the PANA protocol are as follows [151]:

**PAA and AS.** When the PaC uses a domain other than its home domain for network access, then the PAA in the visited network needs to communicate with the home AS to verify the PaC credentials. Possible threats arising in the communication path between the PAA and AS are detailed in RFC 3579 [14]. To counter these threats, this traffic channel must be protected using a security association established between the PAA and AS.

**PAA and EP.** The PAA and EP must belong to the same domain. Where necessary, a security association can be established to protect the link between them.

**PaC and AS.** The PaC and AS must belong to the same domain and hence share a trust relationship. When the PaC uses a domain other than its home domain for network access, it provides its credentials to the PAA in the visited network. The information provided will therefore pass via the PaC-PAA and PAA-AS paths. For further information on the threats arising to data sent via the PAA-AS path, see RFC 3579 [14]. Section 6.2.1.2 describes the threats arising in the PaC-PAA path.

As described in RFC 4016 [151], it is possible that some of the PANA entities (e.g. the PAA, AS, and EP) are co-located. In those cases, it can be assumed

that there are no significant threats to their communications.

Pairs of entities that do not need to share a trust relationship prior to use of the PANA protocol are as follows [151]:

**PaC and PAA.** The PaC and PAA typically belong to different domains. They establish a security association during the authentication process.

**PaC and EP.** The authentication process may result in the establishment of a secret key shared by the PaC and PAA, which can also be used to secure the link between the PaC and EP.

**AS and EP.** The EP is not known outside of the access network, and therefore the AS and the EP do not need to share a security association.

#### 6.2.1.2 PANA Threat Scenarios

There are a variety of scenarios which need to be considered when developing the threat model for the PANA protocol. As cited in RFC 4016 [151], the threats to PANA can be grouped according to the stages through which the client goes in order to gain network access. In the following paragraphs, the threats related to the following stages are described:

- PAA discovery;
- the authentication procedure itself, which includes: false success or failure indications, MitM attacks, replay attacks, device identifier attacks, and device identifier confidentiality;
- the PaC leaving the network;
- service theft;
- PAA-EP communication; and
- other miscellaneous attacks.

**PAA Discovery** As described in section 6.1.4.1, the PAA is discovered by sending solicitations or receiving advertisements from the PaC. In this initial stage of the PANA protocol, the PaC has no assurance that the other end of the link is the PAA (see section 6.2.1.1), and an attacker can pretend to be a PAA by sending a spoofed advertisement. This threat is present mainly in environments where the PaC-PAA link is shared.

The advertisement may be used to include other information than the discovery of the PAA itself. This can, for instance, lead to a 'bidding down attack' (see section 6.1 of [151]), where an attacker sends a spoofed advertisement with capabilities indicating authentication methods less secure than those that the real PAA supports, thereby fooling the PaC into negotiating a method less secure than would otherwise be available. Of course, such an attack will only succeed if the fake PAA can break the weaker authentication method and the weaker method is accepted by the PaC. Moreover, the possibility of such an attack is essentially inevitable in any system allowing negotiation of the authentication method to be used.

**False Success or Failure Indications** As stated in section 3.7.5, PANA carries any authentication scheme that can be specified as an EAP method. EAP methods incorporate a message used to indicate success or failure (see section 3.4). By sending a false failure message, an attacker can prevent the client from accessing the network. By sending a false success message, an attacker can prematurely end the authentication exchange, denying service for the PaC.

This attack is possible if the success or failure indication is not protected by a security association between the PaC and the PAA. All PANA messages exchanged prior to completion of the key establishment process may be unprotected.

**Man-in-the-Middle Attacks** An attacker can claim to be the PAA to the real PaC, and also claim to be the PaC to the genuine PAA. As stated in section 3.2.3, this is called a Man-in-the-Middle attack, whereby the PaC is fooled into believing that it is communicating with the real PAA, which is also misled into believing that it is communicating with the genuine PaC (see also section 6.2.2 of [151]).

As stated in section 3.2.3, the use of tunnelled protocols in the first step, together with the use of legacy client authentication protocols in the second step, creates a vulnerability to an active MitM attack, which allows the attacker to impersonate the remote entity (see [21]). The attack becomes possible if the legacy client authentication protocol is used in multiple environments (e.g. with and without tunnel-protection). An instance of an active MitM attack, in which compound authentication methods are used, is described in [21]. In these attacks, the server first authenticates to the client. As the client has not yet proven its identity, the server acts as the MitM, tunnelling the identity of the genuine client to gain access to the network.

Asokan, Niemi and Nyberg [21] have shown that the problem can be fixed by either restricting the use of the legacy authentication protocol to a specific environment, or by implementing a cryptographic binding between the first step and the second step protocols. As detailed in RFC 4016 [151], this implies that PANA will be vulnerable to such attacks if compound methods are used without cryptographically binding them.

**Replay Attacks** As described by Parthasarathy [151], an attacker can replay the PANA messages that denote authentication failure or success at a later time, to create false failure or success indications. The attacker can also potentially replay other PANA protocol messages to deny service to the PaC.

**Device Identifier Attacks** When the PaC is authenticated, the PAA sends access control information to the EP, which is to be used for controlling the network (see section 2.1.1.5). As noted by Parthasarathy [151], this information 'typically contains the device identifier of the PaC, which is either obtained from the IP headers and MAC headers of the packets exchanged during the authentication process or carried explicitly in the PANA protocol field'. The attacker can thus gain unauthorised network access by taking the following steps [151].

- An attacker pretends to be a PAA and sends advertisements. The PaC is fooled and starts exchanging packets with the attacker.
- The attacker modifies the IP source address in the packet, adjusts the UDP [156] /TCP [157] checksum, and forwards the packet to the genuine PAA. It makes the same changes to the return packets.
- When the genuine PaC is authenticated, the attacker gains access to the network, as the packets sent to the PAA contain the IP and MAC addresses of the attacker.

**Device Identifier Confidentiality** Some clients might wish to hide their identities from visited access networks for privacy reasons. Although providing identity protection for clients is outside the scope of PANA, identity protection can be achieved by letting PANA carrying authentication methods that already have this capability.

**PaC Leaving the Network** When the PaC leaves the network, it can inform the PAA, so that the resources used by the client can be properly accounted for. As stated in [151], the PAA may also choose to revoke the PaC network access at any time it considers necessary. In this scenario, there are three possible threats.

- An attacker can pretend to be a PAA and revoke access to the PaC, causing a DoS attack on the PaC.
- An attacker can pretend to be a genuine PaC and transmit a disconnect message, again causing a DoS attack on the PaC.
- The PaC can leave the network without notifying the PAA or EP (e.g. if the network cable is unplugged). In this case, an attacker can pretend to be the PaC and can start using the network in place of the PaC.

**Service Theft** An attacker can gain unauthorised network access by stealing service from a legitimate client. Once the genuine PaC is authenticated, an EP will typically have filters in place to prevent unauthorised network access. These filters will be based on something carried in every packet, for example, the IP and MAC addresses. In this latter case, any received packets will be dropped unless they contain specific IP addresses matching the MAC addresses. The following are possible threats in this scenario:

- An attacker can spoof both the IP and MAC addresses of an authorised client to gain unauthorised access.
- The PaC can leave the network without notifying the PAA or EP (e.g. if the system crashes). In this case, an attacker can pretend to be the PaC and start using the network.

**PAA-EP Communication** When the PaC is authenticated, the PAA sends access control information to the EP which is to be used for controlling network access (see section 2.1.1.5). This information contains at least the device identifier of the PaC. If stronger protection is needed, the PAA will also communicate a shared secret known only to the PaC and PAA, to be used to set up a security association between the PaC and the EP. The following are possible threats:

- An attacker can eavesdrop on the information exchanged between the PAA and EP. The attacker can further use this information to spoof the genuine PaC and also to set up a security association for gaining network access.
- An attacker can pretend to be a PAA and send false information to an EP to gain network access.

These threats can be addressed by protecting the communications path between the PAA and the EP.

**Miscellaneous Attacks** As stated by Parthasarathy [151], the PaCs do not necessarily trust one another; any PaC can pretend to be a PAA, spoof IP addresses, and launch a range of other attacks. There are a variety of DoS attacks which affect the PAA and the backend AS. For instance, to launch a 'blind resource consumption DoS attack' (see section 3.2.3), an attacker can bombard the PAA with many PaC authentication requests. If the PAA and the AS are not co-located, the PAA may allocate local resources to store client state records, before it receives the AS response. If a sufficiently large number of requests are received, then this can exhaust the PAA memory resources. Also, depending on the method, an attacker can force the PAA or the AS to make computationally intensive computations, which can exhaust the available processing resources.

Another kind of attack, known as an 'IP address depletion attack' (see section 6.6 of [151]), is based on the fact that the PaC acquires an IP address before the PANA authentication process begins [184]. When this occurs, it opens up the possibility of DoS attacks in which attackers can exhaust the IP address space by acquiring multiple IP addresses, or deny IP address allocations to other entities by falsely responding to every duplicate address detection query.

The IP address depletion attack can be prevented by deploying a secure

address resolution scheme that does not depend on client authentication, such as the SEcure Neighbor Discovery (SEND) mechanism given in RFC 3971 [51].

### 6.2.2 PANA Security and Implementation Requirements

In this section, we establish the PANA security requirements (section 6.2.2.1); we also analyse and compare the PANA implementation features, in order to obtain implementation requirements for PANA (section 6.2.2.2).

#### 6.2.2.1 PANA Security Requirements

The PANA security requirements, arising from the threat analysis in section 6.2.1, can be summarised as follows [151].

- The PANA protocol must not assume that the PAA discovery process is protected (see 'PAA Discovery' in section 6.2.1.2).
- The PANA method must mutually authenticate the PaC and the PAA, and must be able to establish keys between them to protect message exchanges (see 'Success or Failure False Indications' in section 6.2.1.2).
- When compound authentication methods are carried by the PANA protocol, they must be cryptographically bound (see 'Man-in-the-Middle Attacks' in section 6.2.1.2).
- The PANA method must protect itself against replay attacks (see 'Replay Attack' in section 6.2.1.2).
- The PANA device identifier must be protected against spoofing in the PaC and PAA message exchanges (see 'Device Identifier Attack' in section 6.2.1.2).

- The PANA protocol must protect disconnect and revocation messages, and must not depend on the PaC sending a disconnect message (see 'PaC Leaving the Network' in section 6.2.1.2).
- The PANA method must securely bind the authenticated session to the client device identifier to prevent service theft; it must also establish a shared secret between the PaC and the PAA, which can be used to set up a security association between the PaC and the EP in order to protect against service theft (see 'Service Theft' in section 6.2.1.2).
- The communication between the PAA and EP must be protected against eavesdropping and spoofing attacks (see 'PAA-EP Communication' in section 6.2.1.2).

#### 6.2.2.2 PANA Implementation Requirements

We now analyse and compare the following PANA implementation features, which were adapted from the PANA requirements described in section 4 of RFC 4058 [184].

**Multiple identifiers.** PANA must support a variety of identifier types for authentication clients and remote devices, including the ability to create a cryptographic binding between the client identifier and the associated device identifier (upon successful PANA protocol exchange).

**IP Address Assignment.** The PaC must configure an IP address before entering the PANA authentication process (the PANA protocol will not make any assumptions about the mechanisms used for the PaC address configuration).

**EAP Lower Layer Requirements.** The EAP protocol [13] imposes many requirements on the underlying transport protocol that need to be satisfied by the PANA carrier for correct operation.

**Flexibility.** The PANA protocol will support PaCs with multiple network interfaces, and access networks with multiple routers (instead of only one first hop router,) on multi-access links (instead of point-to-point links).

**Disconnect Indication.** The PANA method cannot assume that the link is connection-oriented. This link may thus have a mechanism to provide disconnect indication, which is useful in helping the PAA to clean up resources when a client moves away from the network (e.g. to inform the enforcement points that the client is no longer connected).

**Location of PAA.** The PAA and the PaC will be exactly one IP hop away from each other. Bridging and tunnelling techniques can place two nodes exactly one IP hop away from each other, even if they are connected to separate physical links.

**Performance.** The PANA protocol design needs to efficiently handle the authentication process in order to gain network access with minimum latency; e.g. the protocol signalling may be minimised by creating local security associations.

**Complexity.** By using the EAP protocol to carry lightweight authentication methods, it is possible to make use of the PANA protocol to create new ongoing authentication solutions with low complexity at the application layer.

**IP Version Independence.** The PANA protocol will work with both the IPv4 and the IPv6 protocols.

### 6.2.3 PANA Services and Properties Assessment

In this section, the results of the critical analyses made in sections 6.2.1 and 6.2.2 are used to assess the security and implementation services and properties

possessed by the PANA protocol against the services and properties required of new authentication methods for Internet access (as defined in section 5.3).

Table 6.1: **PANA security services and properties assessment**

| **Security Requirements of New Authentication Methods** | **Security Services and Properties of the PANA Protocol** |
|---|---|
| Entity authentication service for remote network access. | Service provided by PANA (section 6.1.1). |
| Key establishment services with key freshness property. | Services possible in PANA (depending on the chosen EAP method; see section 6.1.5). |
| Use of a tunnelled authentication mechanism carrying EAP. | Property provided by PANA (section 6.1.1). |
| Mutual authentication services between the remote client and the access network. | Services provided by PANA (section 6.2.2.1). |
| Use of periodic and on-demand reauthentication techniques. | Service provided by PANA (section 6.1.4.4). |
| Possible interaction between the network and AAA infrastructures. | Property provided by PANA (section 6.1.5). |
| Absence of vulnerabilities that can be exploited over insecure channels. | Property possible in PANA (section 6.2.1.2). |
| Robustness against DoS attacks. | Property provided by PANA (section 6.1.4.1). |
| Identity confidentiality service for remote clients. | Service possible in PANA (depending on the chosen EAP method; see section 6.2.1.2). |

As shown in Tables 6.1 and 6.2, PANA has the potential to meet all the identified requirements for a transportation environment for new entity authentication schemes. This justifies the choice of PANA as the target environment for carrying the authentication techniques discussed in the remainder of this thesis.

Table 6.2: **PANA implementation service and properties assessment**

| **Implementation Requirements of New Authentication Methods** | **Implementation Services and Properties of the PANA Protocol** |
|---|---|
| Support for multiple client and device identifiers. | Service provided by PANA (section 6.2.2.2). |
| Satisfaction of the EAP transport requirements. | Property provided by PANA (section 6.2.2.2). |
| Flexibility. | Property provided by PANA (section 6.2.2.2). |
| Performance. | Property provided by PANA (section 6.2.2.2). |
| Low complexity. | Property provided by PANA (section 6.2.2.2). |
| IP version independence. | Property provided by PANA (section 6.2.2.2). |

# Part III

# Internet Authentication Protocols & Assessments

# Chapter 7

# PANA/GSM

## Contents



As described in section 1.2, this thesis proposes a series of new solutions for Internet remote access authentication, derived by adapting and reinforcing security techniques arising from a variety of different sources. The aim of this chapter is to present the first new authentication scheme, combining the GSM authentication mechanism (see section 3.5.1) with PANA (see section 3.7.5 and Chapter 6), which we call PANA/GSM.

## 7.1 Introduction

As described in section 4.1, Internet remote access networks which are not physically secured against unauthorised use are typically set up so that roaming entities are obliged to go through an *authentication process* (see section 2.2). In some *ubiquitous mobility scenarios* (see section 4.2), IP based remote hosts that connect to the Internet via an access network will typically need to provide their *credentials* (see section 2.2.2) and be authenticated before being authorised to access the network. For such a process we need an easy-to-use, strong, and scalable entity authentication infrastructure. According to Laitinen et al. [124], one of the most critical steps in setting up such an infrastructure is the provisioning of initial credentials to the user, which means, for example, registering username/password pairs, or distributing smart cards.

Entity authentication based on smart cards is more secure than reusing the same password at multiple sites, and more user-friendly than using a large collection of diverse passwords — each of which should be hard to guess (and hence hard to recall). As a result, passwords are typically either kept in an encrypted file protected by a single password, or (most usually) are written on a piece of paper kept somewhere near the PC monitor.

Credential provisioning is costly and takes time, which may be inconvenient for users. This motivates the idea of reusing already deployed user credentials for new Internet remote access services. In particular, cellular network operators already have an authentication infrastructure based on subscriber smart cards, for example in the form of GSM SIMs (see section 3.5.1.1). Therefore it seems potentially desirable to reuse this existing infrastructure for heterogeneous Internet remote access authentication.

As previously discussed, the IETF PANA protocol (see Chapter 6) is intended to be a flexible and scalable generic network layer protocol to be used to

authenticate a user device requesting Internet remote access. In addition, the GSM authentication infrastructure (see section 3.5.1) is by far the most widely deployed cellular network authentication system, with more than one billion users. Building on these two observations, we now present a new authentication scheme, combining GSM authentication mechanism with PANA, which we call PANA/GSM. This innovative proposal, previously described in [146], adapts the security techniques used in the GSM mobile telecommunication system to the PANA network remote access authentication framework, which interacts, via EAP (see section 3.4), with an AAA backend infrastructure (as described in sections 3.7.5 and 3.9) in a complete solution designed to support ubiquitous client mobility for Internet access.

The purpose (section 7.2) and the components used in the assembly (section 7.3) of the novel PANA/GSM scheme are first given. Second, the EAP-SIM mechanism (section 7.4), an EAP method (see section 3.4) published in RFC 4186 [77] and used as a component in the new PANA/GSM technique, is explained. The framework of the proposed new PANA/GSM protocol is then given (section 7.5). Next two important features of PANA/GSM, namely the security association and the re-authentication procedure (section 7.6), are described. Finally, the conclusions of the chapter are given (section 7.7).

The main novel contribution of this chapter lies in sections 7.3, 7.5, and 7.6. Whilst the EAP-SIM mechanism described in section 7.4 has been previously described (notably by Haverinen and Salowey [77]), the details of how it would operate when executed over PANA have not. This chapter does not contain a detailed security analysis of the new proposal — this issue is covered in Chapter 11.

## 7.2 PANA/GSM Objective

Currently there is no standard protocol for performing network access authentication above the link layer. Instead, a number of ad hoc and often inadequate solutions (as described in section 4.1.5) are being used to overcome the problem (itself described in section 4.1), in a variety of distinct scenarios (outlined in section 4.2).

The objective of the PANA/GSM protocol is thus to provide a network layer (see section 4.1.4), IP compatible, lightweight, attack-resistant (e.g. with respect to MitM and DoS attacks — see section 3.2.3), and relatively flexible authentication method, that allows a remote client to be authenticated in a heterogeneous Internet access environment supporting ubiquitous mobility. This authentication method must meet a number of detailed security and implementation requirements, as specified in Chapter 5.

## 7.3 PANA/GSM Protocol Hierarchy

In this section, an overview of the components used in the construction of the new PANA/GSM authentication scheme is given. The first component, as previously discussed, is the GSM SIM authentication mechanism. Section 3.5.1 gives an outline of the GSM system security features, with a focus on the air interface protocol, including the GSM SIM authentication procedure.

The second component used in the PANA/GSM protocol assembly is EAP (see section 3.4). The EAP protocol supports a variety of authentication schemes, giving network access providers the advantage of using a single framework across multiple environments. Such flexibility seems likely to be important for heterogeneous network access supporting ubiquitous mobility. Since EAP is very flexible and can encapsulate arbitrary authentication schemes (see section 4.1.3),

called EAP methods, it is clearly a protocol that satisfies many of the requirements for a variety of authentication scenarios (see sections 4.2 and 5.1.3).

Nevertheless, EAP itself does not specify any authentication method. It is only a transport mechanism, allowing concrete authentication methods for EAP, such as methods from the mobile telecommunications area, to be defined separately. In fact, the EAP-SIM protocol, an EAP method specified in RFC 4186 [77], describes a way of encapsulating the security parameters used by the GSM system within EAP. EAP-SIM also proposes enhancements to the GSM security procedures, in order to provide mutual authentication and session key agreement using the GSM SIM.

Although EAP-SIM re-uses a security solution implemented in a widely deployed mobile system (i.e. GSM) in a flexible authentication framework (i.e EAP), using EAP-SIM on its own for authentication is not a good choice. This is because it does not provide a complete authentication solution for ubiquitous client mobility for Internet access.

The effective use of EAP-SIM in this latter environment requires the provision of a transport scheme for authentication data between a remote entity seeking access to a network and another entity located in the access network (see section 4.1.4). More specifically, a transport scheme independent of the access network type is needed, to transfer user authentication information to the access network and, optionally, to the AAA backend infrastructure (see section 3.9). Defining a network layer transport for EAP-SIM, such as the proposed tunnelled authentication solutions (see section 3.7), provides a cleaner answer to the problem.

In Chapter 6, we justified the selection of PANA, a UDP-based protocol (see section 6.1.1), as the tunnelled network layer transportation environment. We describe in this chapter how to use PANA to support the use of EAP-SIM for Internet remote access authentication in ubiquitous mobility scenarios.

PANA is also able to interact with an AAA infrastructure supporting EAP, i.e. Diameter EAP (see sections 3.7.5 and 3.9.3). Consequently, PANA is our choice for the third component in the construction of our proposed technique, which thus combines GSM authentication with EAP-SIM and PANA interacting with Diameter EAP, into a scheme which we call PANA/GSM. A summary of the PANA/GSM protocol hierarchy is shown in Figure 7.1.

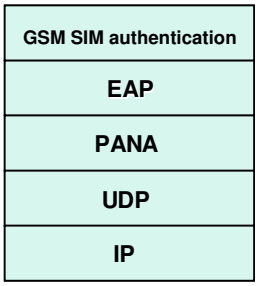


Figure 7.1: **PANA/GSM protocol hierarchy**

## 7.4 An EAP Mechanism for Carrying GSM

In this section, EAP-SIM, an EAP method which is used as a component of the PANA/GSM technique, is explained. RFC 4186 [77] describes this authentication and session key distribution mechanism, that uses the GSM SIM (see section 3.5.1). It involves a client acting on behalf of a user, an authenticating party, and an EAP server (see section 3.4). The EAP server, which typically belongs to the user's home Internet AAA network (see section 3.9), must be able to obtain 'authentication triplets' ($RAND$, $XRES$, $K_c$) from the subscriber's HN AuC (see section 3.5.1.3) in the GSM mobile network.

The EAP-SIM packet format and the use of attributes are specified in section 8 of [77]. Either the IMSI or the TMSI can be employed as part of the user identifier. Section 4.2 of [77] describes user identity management. EAP-SIM

includes optional identity privacy support (see section 4.2.1.2 of [77]), and an optional fast re-authentication procedure (see section 5 of [77]).

In EAP-SIM, a set of $n$ *RAND* challenges are used to generate 64-bit confidentiality keys $K_c$ ($n = 2$ or 3), which are combined to generate a 'more secure' key than can be obtained from individual GSM triplets. We label these $n$ confidentiality keys $K_{c,j}$ ($1 \leq j \leq n$). As described in section 3.5.1.3, each key $K_{c,j}$ is produced as a function of a challenge $RAND_j$ and the customer's unique 128-bit secret key $K_i$, using a key generation algorithm known as A8. In the GSM mobile network, the values of $K_{c,j}$ are calculated in the subscriber's HN AuC, before being sent to the EAP server within authentication triplets. In the EAP client, each $K_{c,j}$ is generated and stored in the GSM SIM until it is updated as part of the next authentication procedure, where:

$$K_{c,j} = A8(K_i, RAND_j). \tag{7.1}$$

The EAP client also inputs each of the $n$ challenges $RAND_j$ along with $K_i$ to a MAC algorithm known as A3 that is implemented in the GSM SIM, and obtains the resulting $n$ outputs, known as $SRES_j$, as follows:

$$SRES_j = A3(K_i, RAND_j). \tag{7.2}$$

In EAP-SIM authentication, a secret Master Key *MK* is derived by applying the hash function SHA-1 (see section 2.1.3.2) to the concatenation of the customer's identifier (*Identity* — written as $I$ in the equation below), the $n$ GSM keys $K_{c,j}$, a *nonce*, i.e. a randomly chosen value[1] (*NONCE_MT* — written as $N$ below) freshly generated by the EAP client, and other relevant context information $X$, i.e. the concatenation of the list of the supported EAP-SIM versions (*Version_List*) and the identifier of the EAP-SIM version in use (*Se-*

[1] *Nonces* are inputs to cryptographic functions; they contain pseudo random data used to guarantee liveliness during an exchange, and protect against replay attacks.

*lected_Version*). That is, the 160-bit key $MK$ is derived as follows, where here, as throughout, | denotes concatenation of data items, and $h$ denotes the SHA-1 hash function:

$$MK = h(I|K_{c,1}|\ldots|K_{c,n}|N|X). \tag{7.3}$$

As stated in RFC 4186 [77], the $MK$ value is then fed into a pseudo-random number generator algorithm, the details of which are specified in Change Notice 1 of FIPS 186-2 [141]. Figure 7.2 shows the main steps in a simplified version of this algorithm. In this algorithm, the parameter $b$ is set to 160, and $m$ is set to the number of 320-bit output values required. The values $XKEY$ and $XVAL$ are $b$ bits long (i.e. 160 bits), and $MK$ is used to set the initial value of the seed-key, $XKEY$. The function $G$ is constructed using SHA-1[2] (see section 2.1.3.2).

**Algorithm :** PSEUDO-RANDOM-GENERATOR$(MK, b, m)$

**comment:** Choose a new, secret value for the seed-key.
$XKEY \leftarrow MK$
$t \leftarrow$ 67452301 $EFCDAB89$ $98BADCFE$ 10325476 $C3D2E1F0$
**comment:** $t$ is the initial value for $H_0|H_1|H_2|H_3|H_4$ in $G$.
**for** $j \leftarrow 0$ **to** $m-1$
**do**
- **for** $i \leftarrow 0$ **to** 1
  - **do**
    - $XVAL \leftarrow XKEY \bmod 2^b$
    - $w_i \leftarrow G(t, XVAL)$
    - $XKEY = (1 + XKEY + w_i) \bmod 2^b$
- $x_j \leftarrow w_0|w_1$
- **return** $(x_j)$

Figure 7.2: **Pseudo-random number generator algorithm (FIPS 186-2)**

In line with section 3.6.6, this pseudo-random number algorithm produces separate Transient EAP Keys or *TEKs* for protecting EAP packets, a Master Session Key (*MSK*) for encryption of the traffic exchanged between the client and the network, and an Extended Master Session Key (*EMSK*) used to derive

[2]The function $G$ is very similar to SHA-1, but the message padding is different.

keys for multiple applications. EAP-SIM also requires the generation of two TEKs for its own purposes, i.e. the authentication key ($K_a$) to be used with the Message Authentication Code attribute (AT_MAC), and the encryption key ($K_e$) to be used with the data encryption attribute.

In the EAP-SIM full authentication procedure, the 320-bit random numbers $(x_0, x_1, \ldots, x_{m-1})$ output from the generator are concatenated and partitioned into suitable-sized bit strings, which are used as keys in the following order: $K_e$ (128 bits), $K_a$ (128 bits), *MSK* (64 bytes), and *EMSK* (64 bytes)[3].

In EAP-SIM fast re-authentication, the same pseudo-random number algorithm can be used to generate a new *MSK* and a new *EMSK*. In this case, the seed value ($XKEY'$) is calculated as given below, where $I$ denotes the next fast re-authentication user identifier, $c$ denotes the next counter value[4], $N$ denotes a freshly generated 16-byte nonce (known as *NONCE_S*), and $MK$ is the master key derived during the preceding full authentication:

$$XKEY' = h(I|c|N|MK). \tag{7.4}$$

The pseudo-random number generator described in Figure 7.2 is then run with the new seed value $XKEY'$, and the resulting 320-bit random numbers $(x_0, x_1, \ldots, x_{m-1})$ are concatenated and partitioned into two 64-byte strings, which are used as the new 64-byte *MSK* and the new 64-byte *EMSK*.

Finally, in order to provide mutual authentication, EAP-SIM enhances GSM authentication by accompanying the *RAND* challenges and other EAP-SIM messages with a MAC, generated using the HMAC-SHA-1 function (see section 2.1.3.2). The MAC is calculated over the whole EAP-SIM packet concatenated

[3] As stated in section 1.2 of [18], the MSK and the EMSK are individually at least 64 octets in length, where each octet or *byte*, as called in RFC 4186 [77], contains 8 bits.

[4] Both the peer and the EAP server maintain a copy of this counter, which is used to protect against replay attacks. The EAP server sends its counter value to the peer in the fast re-authentication request. The peer must verify that its counter value is less than or equal to the value sent by the EAP server.

with optional message-specific data, with the exception that the value field of the MAC attribute (AT_MAC) is set to zero when calculating the MAC value.

## 7.5 PANA/GSM Framework

In this section, the authentication framework for the new PANA/GSM scheme is described. The entities (section 7.5.1) involved in the PANA/GSM method are first given. After that, the PANA/GSM authentication scheme (section 7.5.2) is explained.

### 7.5.1 PANA/GSM Entities

The PANA/GSM method proposed here involves three entities, namely the PANA Client (also referred to here as the *PaC*, *client*, *user*, *customer* or *subscriber*), the PANA Authentication Agent (*PAA* or *authenticating party*) and the EAP server. The PaC is associated with a network device and a set of GSM credentials stored in a SIM; these credentials are used to authenticate the PaC identity for the purposes of network access. A possible implementation of the PaC would be an Internet access device (e.g. a laptop) with a PC card inserted in the PCMCIA[5] socket, where the PC card is itself equipped with a GSM-enabled SIM card.

The PAA verifies the GSM credentials provided by the PaC and grants network access. In the context of this chapter, the user's EAP server is assumed to be implemented on the AAA server (see section 3.9) and has an interface to the GSM network; that is, it operates as a *gateway* between the Internet AAA network and the GSM authentication infrastructure. The PAA is thus an AAA client that communicates with the user's EAP server through an AAA protocol

[5]Personal Computer Memory Card International Association (www.pcmcia.org/).

supporting EAP (i.e. Diameter EAP, described in section 3.9.3) and key wrap (e.g. Diameter CMS [32]), i.e. the use of a key-encrypting key to encrypt a content-encryption key. PANA/GSM also involves a further entity, namely the EP (see section 6.1.2), which applies per-packet enforcement policies (i.e. filters) to the traffic of the PaC's devices.

### 7.5.2 PANA/GSM Authentication Scheme

The aim of this section is to give a detailed description of the PANA/GSM scheme. Firstly we identify the distinct phases of a PANA/GSM session, and briefly describe them (section 7.5.2.1). Secondly, a complete description of the PANA/GSM message exchange is provided (section 7.5.2.2). We then summarise the calculation of the PANA/GSM-based MAC used during that exchange (section 7.5.2.3).

Figure 7.3 shows the PANA/GSM authentication procedure, which is further described below. In this figure, the name of each message is shown, followed by the contents of the message in round brackets; square brackets are used to denote optional fields.

#### 7.5.2.1 PANA/GSM Phases

The PANA/GSM authentication procedure has three main phases: (1) Discovery and Handshake, (2) Authentication and Authorisation, and (3) Access. In the *Discovery* phase, an IP address for the PAA is identified, and a PANA/GSM session is established between the PaC and the PAA, following the PANA model (see section 6.1.4.1). After this phase is complete, a session identifier (*Session-Id* — see section 6.1.2) is allocated by the PAA and included in all further messages; this identifier is freed when the PANA/GSM session terminates.

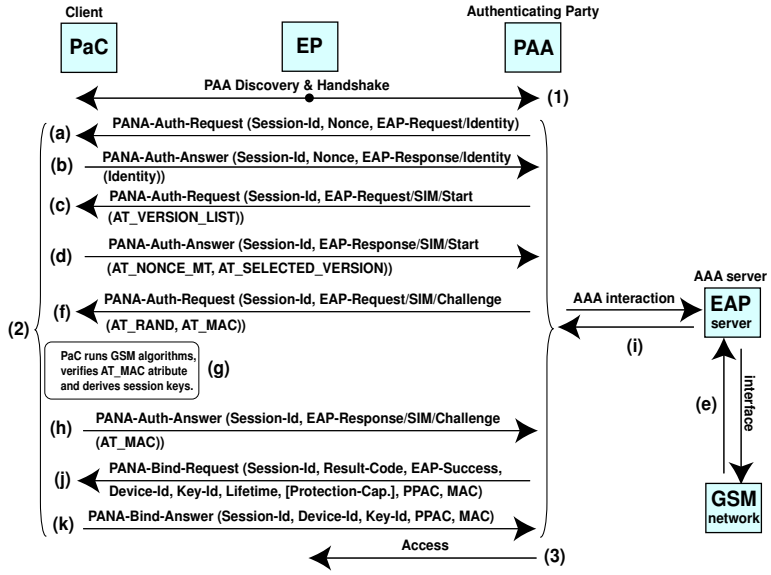


Figure 7.3: **PANA/GSM full authentication procedure**

In the *Authentication* phase, the main focus of this section and further explained below, EAP-SIM messages encapsulated in PANA/GSM messages are exchanged between the PaC and the PAA. In this phase, EAP Request and Response payloads are typically carried in PANA-Auth-Requests, and PANA-Auth-Answers are simply used to acknowledge receipt of the requests. However, taking advantage of an optimisation discussed in section 6.1.4.2 and adopted by PANA/GSM, in the context of this chapter a PANA-Auth-Answer will include an EAP-SIM Response payload.

As previously discussed, the PAA communicates with the EAP server through the AAA Diameter EAP protocol (see section 3.9.3). Hence, EAP-SIM packets encapsulated in Diameter-EAP messages are exchanged between the PAA, which is thus the *Diameter client*, and the EAP server, which is implemented on the *Diameter server*, following the process for using EAP in Diameter given in Figure 3.14. The PANA-Diameter message mapping, given in section 3.7.5, is also adopted here to allow the transport of EAP-SIM payloads between the PANA framework and the AAA Diameter infrastructure (see section 3.9.2). At

the end of the Authentication phase, a PANA SA is established, including the provision of a shared secret EAP-SIM session key *MSK* (see section 7.4); we call this the PANA/GSM SA.

During the *Access* phase, a separate protocol is used between the PAA and the EP to manage PaC network access control. After this phase, the established PANA/GSM session and the PANA/GSM SA are deleted, following the PANA draft standard (see section 6.1.4.5).

#### 7.5.2.2 PANA/GSM Message Exchange

During the *Authentication* phase, the first PANA-Auth-Request message (*a*) issued by the PAA carries a PANA-based *Nonce*, i.e. a randomly chosen value (see section 6.1.3), used in further PANA/GSM cryptographic key computations, and an EAP-Request/Identity payload, requesting the PaC to identify itself. The PaC responds (*b*) with a PANA-Auth-Answer, which also carries a PANA-based Nonce value, and an EAP-Response/Identity payload including the user's identifier (*Identity*).

The PAA then issues a Diameter-EAP-Request to the EAP server via an AAA interaction (see section 3.9.3), including the EAP-Response/Identity message in an EAP-Payload AVP, and the user's identifier value in a Diameter User-Name AVP (see section 3.9.2). The EAP server responds with a Diameter-EAP-Answer in a multi-round exchange, which includes a NULL EAP-Payload AVP and a Result-Code AVP set to DIAMETER_MULTI_ROUND_AUTH, signifying that a subsequent request is expected.

An EAP-Request/SIM/START packet, containing a list of the EAP-SIM versions supported by the PAA (*Version_List*), is now sent to the PaC in a PANA-Auth-Request (*c*). The PaC responds (*d*) with a message carrying the EAP-Response/SIM/Start payload, which includes *NONCE_MT*, a random

number chosen by the PaC, and the EAP-SIM version selected by the PaC (*Selected_Version*).

After receiving the EAP-Response/SIM/Start payload from the PAA via an AAA interaction, i.e. encapsulated in a Diameter-EAP-Request, the EAP server obtains the set of $n$ GSM triplets generated by the AuC within the home GSM network of the PaC ($e$). As specified in section 7.4, the EAP server can now derive the keying material, using as input into the hash function SHA-1 (see section 2.1.3.2) a combination of the values: the user's *Identity* ($I$), *NONCE_MT* ($N$), *Version_List*, and *Selected_Version* ($X$), together with the set of $n$ GSM keys $K_{c,j}$ obtained from the GSM triplets. The output will be the secret key *MK* (see equation 7.3). From *MK*, the EAP server is able to derive the keying material, including the *MSK*, which is used by PANA/GSM as the AAA-Key (see section 6.1.2), and $K_a$, which is used to calculate the MAC.

The EAP server then sends back to the PAA a Diameter-EAP-Answer in a multi-round exchange. This exchange includes an EAP-Request/SIM/Challenge payload, which contains the set of $n$ challenges $RAND_j$ obtained from the GSM triplets, and a MAC to protect the challenges. The MAC of this EAP-SIM payload is calculated by applying HMAC-SHA-1 (see section 2.1.3.2) to the concatenation of the *EAP_Packet* ($P$) and *NONCE_MT* ($N$), as shown in equation (7.5), where here, as throughout, $f_K(X)$ denotes an HMAC-SHA-1 MAC computed using the key $K$ and data $X$:

$$MAC = f_{K_a}(P|N). \tag{7.5}$$

The next PANA/GSM message ($f$) issued by the PAA encapsulates the received EAP-Request/SIM/Challenge payload detailed above. On receipt of this message, the PaC runs the GSM authentication algorithm inside a SIM to derive the keying material, as described in section 7.4. Each key $K_{c,j}$ is generated in the GSM SIM, as a function of the challenge $RAND_j$ and the customer's

unique secret key $K_i$, using key generation algorithm A8 (see equation 7.1). At the same time, $n$ values $SRES_j$ are generated, also as a function of $RAND_j$ and $K_i$, using MAC algorithm A3 (see equation 7.2). After that, the PaC derives the secret key $MK$ (see equation 7.3) and the resulting EAP-SIM session keys, including $MSK$ and $K_a$.

Next, the PaC calculates a copy of the MAC, as shown in equation 7.5, and verifies that the calculated MAC equals the received MAC ($g$). Since the $RAND_j$ challenges given to a PaC are accompanied by the AT_MAC, and since the PaC's $NONCE\_MT$ value contributes to AT_MAC, the PaC is able to verify that the EAP-SIM message is *fresh* (i.e. not a replay; see section 2.2.5) and that the sender possesses valid GSM triplets for the user.

If all the checks succeed, the PaC responds ($h$) with a PANA-Auth-Answer encapsulating the EAP-Response/SIM/Challenge payload, itself containing the AT_MAC attribute. This is calculated by applying HMAC-SHA-1, as given in equation (7.6), where $P$ is the $EAP\_Packet$, and the set of concatenated $SRES_j$ values are the PaC's responses to the $n$ received challenges $RAND_j$:

$$MAC = f_{K_a}(P|SRES_1|\ldots|SRES_n). \tag{7.6}$$

After receiving the EAP-Response/SIM/Challenge payload from the PAA via an AAA Diameter-EAP-Request, the EAP server verifies that the MAC value is correct. This involves using the EAP-SIM payload concatenated with the $n$ stored values of $XRES_j$, together with the key $K_a$ as inputs to HMAC-SHA-1. That is, checking involves recomputing the MAC according to equation (7.6), but using $XRES_j$ in place of $SRES_j$. The $n$ stored values $XRES_j$ are obtained from the same GSM triplets and in the same order as the $n$ previously sent $RAND_j$ challenges.

The EAP server then sends back ($i$) an AAA Diameter-EAP-Answer, which

includes a Result-Code AVP set to DIAMETER_SUCCESS. This message also includes an EAP-Payload AVP with a code field set to Success, which indicates that the authentication was successful. This EAP-Success packet carries derived AAA keying material, including an AAA-Key.

The PAA then encapsulates the PANA result code, the EAP-Success packet, and the PANA/GSM *session lifetime* (see section 6.1.2) in a PANA-Bind-Request message sent to the PaC ($j$), and receives back an acknowledgement through a PANA-Bind-Answer ($k$). On receipt of this message, the PAA issues a Diameter Accounting-Request (Start) to the EAP server, which indicates the start of the session, following the PANA-Diameter message mapping given in section 3.7.5.

PANA-Bind messages are protected by a PANA/GSM-based MAC AVP, calculated as described in the next section, and carry a Key-Id AVP (see section 6.1.3); this latter AVP contains an AAA-Key identifier that is assigned by the PAA and is unique within the PANA/GSM session.

Finally, PANA-Bind messages may also optionally contain a Protection-Capability AVP (see section 6.1.3), which is sent from the PAA to indicate that link-layer or network-layer encryption should be initiated after completion of PANA/GSM. PANA-Bind messages are also used for binding the device identifiers of the PaC and the PAA to the PANA/GSM SA established at the end of the authentication phase; this is achieved using a Device-Id AVP. PANA-Bind messages with a Result-Code AVP indicating successful authentication also include PPAC AVPs (see section 6.1.3), which help the PAA/PaC to negotiate the available/chosen IP address configuration method.

#### 7.5.2.3 PANA/GSM-based MAC

The PANA/GSM-based MAC ($M_{P_G}$) is calculated using HMAC-SHA-1, as given in equation (7.7), where $P_G$ denotes the PANA/GSM packet, and $K_p$ denotes the PANA_MAC_Key (see section 6.1.5):

$$M_{P_G} = f_{K_p}(P_G). \tag{7.7}$$

The EAP-SIM shared secret *MSK*, which is used to establish a PANA/GSM SA, is adopted as the AAA-Key, which is then used to generate a distinct PANA_MAC_Key $K_p$. However, two AAA-Keys may be produced as a result of separate NAP and ISP authentication processes (see section 6.1.4.2). In this case, the AAA-Key used in the $K_p$ generation procedure, which we call $K_{aaa}$, is derived as in equation (7.8), where $K_{nap}$ denotes the AAA-key produced by the NAP, and $K_{isp}$ denotes the AAA-key produced by the ISP:

$$K_{aaa} = K_{nap}|K_{isp}. \tag{7.8}$$

In this case, the PANA_MAC_Key $K_p$ is derived from a combination of PANA-based *Nonces* and the AAA-Key $K_{aaa}$. That is, $K_p$ is calculated by applying HMAC-SHA-1, as given in equation (7.9), using the key $K_{aaa}$ to the concatenation of two PANA-based *Nonces*, which we call $N_{pac}$ and $N_{paa}$, sent respectively by the PaC ($b$) and the PAA ($a$), and the PANA/GSM Session-Id AVP value ($S_{id}$):

$$K_p = f_{K_{aaa}}(N_{pac}|N_{paa}|S_{id}). \tag{7.9}$$

## 7.6 PANA/GSM SA and Re-Authentication

Two important features of PANA/GSM, namely the security association and the re-authentication procedure, are now described.

Once the EAP-SIM method has completed, a session key, i.e. the EAP-SIM *MSK*, which is used as the AAA-Key (as discussed in the previous section), is shared by the PaC and the PAA. This session key is provided to the PaC as part of the EAP key exchange process, and the PAA obtains the session key from the EAP server via the AAA infrastructure. PANA/GSM SA establishment based on the EAP session key is required where no physical or link layer security is available (see section 4.2.3).

The purpose of a re-authentication exchange is to allow for efficient re-keying, using the existing PANA/GSM security association, in situations where (depending on the security policy in force) full authentication is not required. Two types of re-authentication (or fast re-authentication) are supported by PANA/GSM. The first type enters the chosen EAP method, i.e. the EAP-SIM fast re-authentication process (see section 5 of [77]), during the authentication and authorisation phase, and in this case the initial discovery and handshake phase is omitted. The generation of a new session key, using an EAP-SIM fast re-authentication process, is described in section 7.4. The second type of re-authentication uses protected PANA/GSM messages exchanged directly during the access phase, without entering the authentication and authorisation phase, i.e. the PANA re-authentication phase (see section 6.1.4.4).

## 7.7 Conclusions

Authentication and key agreement are fundamental components of a secure procedure for heterogeneous network access supporting ubiquitous mobility. The

main challenges addressed here include the investigation and development of unified, secure and convenient authentication mechanisms that can be used in access networks of a wide range of types.

In this chapter, we have proposed the PANA/GSM protocol, providing an IP-compatible, lightweight, flexible and scalable method for authenticating a user to an access network. The protocol is based on PANA, a network-layer access authentication protocol carrier, which communicates, via Diameter EAP, with an AAA infrastructure interacting with an AuC in the GSM mobile network. PANA/GSM uses the EAP-SIM protocol, which encapsulates GSM parameters in EAP and provides enhancements such as stronger authentication and key agreement as well as mutual authentication.

The use of 'triplets' in PANA/GSM minimises the necessary trust relationship between operators, thereby increasing the likelihood of successful use. From the user perspective, the protocol works with a 'standard' GSM SIM card and requires only an appropriate Internet access device and a SIM card reader, which may or may not be integrated with the access device. The gains in performance arising from the two types of fast re-authentication, and the gains in security from the PANA/GSM SA, potentially make the PANA/GSM proposal attractive to GSM operators willing to offer their users heterogeneous Internet access in ubiquitous mobility networks.

This new Internet authentication scheme, designed to meet the requirements established in Chapter 5, is proposed here as a candidate for secure access procedure for heterogeneous network access supporting ubiquitous mobility (see section 1.1). In Chapter 11, the new scheme is submitted to a formal threat modelling process; it is also compared with the three further novel Internet entity authentication techniques proposed in Chapters 8, 9, and 10.

# Chapter 8

# PANA/UMTS

## Contents



As explained in Chapter 7, this thesis proposes a series of new solutions for Internet remote access authentication, derived by adapting and reinforcing security techniques arising from a variety of different sources. The aim

of this chapter is to present the second new authentication scheme, namely a means of combining the UMTS authentication and key agreement mechanism (see section 3.5.3.2) with PANA (see section 3.7.5 and Chapter 6), which we call PANA/UMTS.

## 8.1 Introduction

As described in section 7.1, in some ubiquitous mobility scenarios, IP based remote hosts that connect to the Internet via an access network will typically need to provide their credentials and be authenticated before being authorised to access the network. For such a process we need an easy-to-use, strong, and scalable entity authentication infrastructure. According to Laitinen et al. [124], one of the most critical steps in setting up such an infrastructure is the provisioning of initial credentials to the user, which means, for example, registering username/password pairs, or distributing smart cards. As stated before, authentication based on smart cards is more secure than reusing the same password at multiple sites, and more user-friendly than using a large collection of diverse passwords.

As outlined in section 7.1, credential provisioning is costly and takes time, which may be inconvenient to users. This motivates the idea of reusing already deployed user credentials for new Internet remote access services. In particular, cellular network operators already have an authentication infrastructure based on subscriber's smart cards, for example in the form of UMTS USIMs (see section 3.5.3.1). Therefore it seems potentially desirable to reuse this existing infrastructure for heterogeneous Internet remote access authentication.

As previously discussed, the IETF PANA protocol (see Chapter 6) is intended to be a flexible and scalable generic network layer protocol to be used to authenticate a user device requesting Internet remote access. In addition, the 3GPP UMTS AKA infrastructure (see section 3.5.3.2), currently being rolled out worldwide, is an internationally accepted standard for the new generation of mobile services, which provide both better quality voice and high-speed Internet and multimedia services (see section 3.5.3.2). Building on these two observations, we now present a new authentication scheme, combining the UMTS authentication and key agreement mechanism with PANA, which we call

PANA/UMTS. This innovative proposal, previously described in [148], adapts the security techniques used in the UMTS mobile telecommunication system to the PANA network remote access authentication structure, in a solution designed to support ubiquitous client mobility for Internet access.

The purpose (section 8.2) and the components used in the assembly (section 8.3) of the novel PANA/UMTS scheme are first given. Second, the EAP-AKA mechanism (section 8.4), an EAP method (see section 3.4) published in RFC 4187 [20] and used as a component of the new PANA/UMTS technique, is explained. The framework of the proposed new PANA/UMTS protocol is then given (section 8.5). Next, two important features of PANA/UMTS, namely the security association and the re-authentication procedure (section 8.6), are described. This is followed by a description of how the GAA architecture (see section 3.5.4) can be used to support an internal interface for PANA/UMTS (section 8.7). Finally, the conclusions of the chapter are given (section 8.8).

The main novel contribution of this chapter lies in sections 8.3, 8.5, 8.6, and 8.7. Whilst the EAP-AKA mechanism described in section 8.4 has been previously described (notably by Arkko and Haverinen [20]), the details of how it would operate when executed over PANA have not. This chapter does not contain a detailed security analysis of the new proposal — this issue is covered in Chapter 11.

## 8.2 PANA/UMTS Objective

Currently there is no standard protocol for performing network access authentication above the link layer. Instead, a number of ad hoc and often inadequate solutions (as described in section 4.1.5) are being used to overcome the problem (itself described in section 4.1), in a variety of distinct scenarios (outlined in section 4.2).

The objective of the PANA/UMTS protocol is thus to provide a network layer, IP compatible, lightweight, attack-resistant (e.g. with respect to MitM and DoS attacks — see section 3.2.3), and relatively flexible authentication method, that allows a remote client to be authenticated in a heterogeneous Internet access environment supporting ubiquitous mobility. This authentication method must meet a number of detailed security and implementation requirements, as specified in Chapter 5.

## 8.3 PANA/UMTS Protocol Hierarchy

In this section, an overview of the components used in the construction of the new PANA/UMTS authentication scheme is given. The first component, as previously discussed, is the UMTS USIM authentication and key agreement mechanism. Section 3.5.3 gives an outline of the UMTS system security features, with a focus on the air interface protocol, including the authentication and key agreement (AKA) scheme.

The second component used in the PANA/UMTS protocol assembly is EAP (see section 3.4). The EAP protocol, as previously discussed, supports a variety of authentication schemes, giving providers the advantage of using a single framework across multiple environments. Such flexibility seems likely to be important for heterogeneous network access supporting ubiquitous mobility. Since EAP is very flexible and can encapsulate arbitrary EAP methods, it is clearly a protocol that satisfies many of the requirements for a variety of authentication scenarios (see sections 4.2 and 5.1.3).

However, as previously described, EAP itself does not specify any authentication method. It is only a transport mechanism, allowing concrete authentication methods for EAP, such as methods from the mobile telecommunications area, to be defined separately. In fact, the EAP-AKA protocol, an EAP method

specified in RFC 4187 [20], describes a way of encapsulating the security parameters used by the UMTS AKA system within EAP, in order to provide mutual authentication and session key agreement using the UMTS USIM.

Although EAP-AKA re-uses a security solution implemented in a new generation of mobile system (i.e. UMTS AKA) in a flexible authentication framework (i.e EAP), using EAP-AKA on its own for authentication is not a good choice. This is because it does not provide a complete authentication solution for ubiquitous client mobility for Internet access.

The effective use of EAP-AKA in this latter environment requires the provision of a transport scheme for authentication data between a remote entity seeking access to a network and another entity located in the access network (see section 4.1.4). More specifically, a transport scheme independent of the access network type is needed, to transfer user authentication information to the access network and, optionally, to the AAA infrastructure (see section 3.9). Defining a network layer transport for EAP-AKA, such as the proposed tunnelled authentication solutions (see section 3.7), provides a cleaner answer to the problem.

In Chapter 6, we justified the selection of PANA, a UDP-based protocol (see section 6.1.1), as the tunnelled network layer transportation environment. We describe here how to use PANA to support the use of EAP-AKA for Internet remote access authentication. As stated previously, PANA is also able to interact with Diameter EAP (see sections 3.7.5 and 3.9.3). Consequently, PANA is our choice for the third component in the construction of our proposed technique, which thus combines UMTS authentication with EAP-AKA and PANA interacting with Diameter EAP, into a scheme which we call PANA/UMTS. A summary of the PANA/UMTS protocol hierarchy is shown in Figure 8.1.

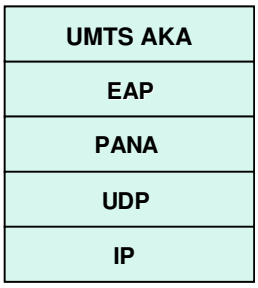


Figure 8.1: **PANA/UMTS protocol hierarchy**

## 8.4 An EAP Mechanism for Carrying UMTS

In this section, EAP-AKA, an EAP method which is used as a component of the PANA/UMTS technique, is explained. RFC 4187 [20] describes this authentication and session key distribution mechanism, that is based upon the UMTS AKA technique (see section 3.5.3). AKA is based on challenge-response mechanisms and symmetric cryptography (see section 2.1.3.2), and typically runs in a UMTS USIM. AKA can also be used in a CDMA2000 (Removable) User Identity Module ((R)UIM) [20], which is similar to a smart card, for all releases of CDMA2000 following release C (see section 3.5.5.3).

EAP-AKA involves a client acting on behalf of a user, an authenticating party, and an EAP server (see section 3.4). The EAP server, which typically belongs to the user's home Internet AAA network (see section 3.9), must be able to obtain authentication vectors from the subscriber's HN AuC (see section 3.5.3.2). The EAP-AKA packet format and the use of attributes are specified in section 8 of [20]. Either permanent identities, usually based on the IMSI, or temporary identities (pseudonyms), which are equivalent to the GSM TMSI, can be employed as part of the user identifier. Section 4.1 of [20] describes user identity management.

EAP-AKA typically uses two round trips to mutually authenticate the client and the network, and provide them with temporary shared secret keys. The pro-

tocol includes an exchange of EAP-Request/Response messages of types *Identity* and *AKA*. The message type *AKA* also has a subtype field[1] that admits the values: *Challenge*, *Authentication-Reject*, *Synchronization-Failure*, *Identity*, *Notification*, *Re-authentication*, and *Client-Error*.

In the EAP-AKA full authentication procedure, an identity request/response message pair is first exchanged. After obtaining the client identity, the EAP server is able to obtain an authentication vector from the subscriber's HN AuC in the UMTS mobile network. As stated in section 3.5.3.2, an authentication vector or 'quintet' (*RAND*, *AUTN*, *XRES*, *IK*, *CK*) is produced from a 128-bit secret key $K$, shared by the USIM and the HN AuC, and a sequence number. From the vector, the EAP server derives the EAP-AKA keying material, as further explained below.

Next, the EAP server sends an EAP-Request/AKA-Challenge message. This message contains a random challenge (*RAND*) and a network authentication token (*AUTN*), both obtained from the authentication vector, and a MAC attribute (AT_MAC). The message may also optionally contain encrypted data (AT_ENCR_DATA) for identity confidentiality and fast re-authentication support (see section 4.1 of [20]).

The client runs the AKA algorithm, usually inside a USIM (see section 3.5.3), and verifies *AUTN* and the MAC. If this is successful, the client has assurance that it is talking to a valid EAP server. It then derives *RES* and certain temporary keys as a function of $K$ and *RAND* (see section 3.5.3.2), and sends back the EAP-Response/AKA-Challenge, protected by another AT_MAC. The EAP server then checks the AT_MAC by comparing the received *RES* with the stored *XRES* from the authentication vector; the shared temporary keys can now be used.

As specified in section 7 of RFC 4187 [20], the EAP-AKA keying material

[1] The subtype-specific data is composed of parameters encapsulated in attributes.

is generated from a Master Key (*MK*), which is derived by applying the hash function SHA-1 (see section 2.1.3.2) to the concatenation of the customer's identifier (*Identity* — written as *I* below), the UMTS integrity key *IK*, and the UMTS confidentiality key *CK*. That is, the 160-bit key *MK* is derived as given in equation (8.1), where here, as throughout, | denotes concatenation of data items, and $h$ denotes the SHA-1 hash function:

$$MK = h(I|IK|CK). \tag{8.1}$$

According to RFC 4187 [20], the *MK* value is input to the pseudo-random number generator described in Figure 7.2, i.e. the same scheme as used in EAP/SIM (see section 7.4). The *MK* value is thus employed as the initial secret seed-key *XKEY*, and the derived material is used to generate the temporary keys. These temporary keys include the *MSK*, used for encryption of the traffic between the client and the network, the encryption key ($K_e$) to be used with AT_ENCR_DATA, the authentication key ($K_a$) to be used with AT_MAC, and the *EMSK*, itself used to derive keys for multiple applications (see section 3.6.6).

In the EAP-AKA full authentication procedure, the 320-bit random numbers $(x_0, x_1, \ldots, x_{m-1})$ output from the generator are concatenated and partitioned into suitable-sized bit strings, which are used as keys in the following order: $K_e$ (128 bits), $K_a$ (128 bits), *MSK* (64 bytes), and *EMSK* (64 bytes)[2].

Finally, EAP-AKA includes optional identity privacy support (see section 4.1.1.2 of [20]), and an optional fast re-authentication procedure (see section 5 of [20]). In EAP-AKA fast re-authentication, the pseudo-random algorithm described in Figure 7.2 can be used to generate a new *MSK* and a new *EMSK*. In this case, the seed value *XKEY'* is calculated as given in equation 7.4, where *I*

[2]As stated in section 1.2 of [18], the MSK and the EMSK are individually at least 64 octets in length, where each octet or *byte*, as it is called in RFC 4187 [20], contains 8 bits. In particular, EAP-AKA defines each *MSK* and *EMSK* to be 64 bytes in length.

denotes the next fast re-authentication user identifier, $c$ denotes the next counter value, $N$ denotes a freshly generated 16-byte nonce (known as *NONCE_S*), and $MK$ is the master key derived during the preceding full authentication. The pseudo-random number generator is then run with the new seed value $XKEY'$, and the resulting 320-bit random numbers $(x_0, x_1, \ldots, x_{m-1})$ are concatenated and partitioned into two 64-byte strings, which are used as the new 64-byte *MSK* and the new 64-byte *EMSK*.

## 8.5 PANA/UMTS Framework

In this section, the authentication framework for the new PANA/UMTS scheme is described. The entities involved in the PANA/UMTS method are first given (section 8.5.1). After that, the PANA/UMTS authentication scheme is explained (section 8.5.2).

### 8.5.1 PANA/UMTS Entities

The PANA/UMTS method proposed here involves three entities, namely the PaC (also referred to here as the *client*, *user*, *customer* or *subscriber*), the PAA (or *authenticating party*) and the EAP server. The PaC is associated with a network device and a set of UMTS credentials stored in a USIM; these credentials are used to prove the PaC identity for the purposes of network access. A possible implementation of the PaC would be an Internet access device (e.g. a laptop) with a PC card inserted in the PCMCIA socket (see section 7.5), where the PC card is itself equipped with a UMTS-enabled USIM card. There are other possible implementations, e.g. involving the use of a UMTS Mobile Equipment (ME, e.g. mobile phone) equipped with a USIM card and linked to a laptop (e.g. via cable, Bluetooth, infrared or WLAN)[3].

[3] An alternative described in [119] is to use USIM Toolkit commands, which enables the USIM to request the ME to open an infrared or Bluetooth channel with the user laptop.

The PAA authenticates the UMTS credentials provided by the PaC and grants network access. In the context of this chapter, the user's EAP server is assumed to be implemented on the AAA server (see section 3.9) and has an interface to the UMTS network; that is, it operates as a *gateway* between the Internet AAA network and the UMTS AKA infrastructure. The PAA is thus an AAA client that communicates with the user's EAP server through an AAA protocol supporting EAP (i.e. Diameter EAP, described in section 3.9.3) and key wrap (see section 7.5). PANA/UMTS also involves a further entity, namely the EP (see section 6.1.2), which applies per-packet enforcement policies (i.e. filters) to the traffic of the PaC's devices.

### 8.5.2 PANA/UMTS Authentication Scheme

The aim of this section is to give a detailed description of the PANA/UMTS scheme. Firstly we identify the distinct phases of a PANA/UMTS session, and briefly describe them (section 8.5.2.1). Secondly, a complete description of the PANA/UMTS message exchange is provided (section 8.5.2.2). We then summarise the calculation of the PANA/UMTS-based MAC used during that exchange (section 8.5.2.3).

Figure 8.2 shows the PANA/UMTS authentication procedure, which is further described below. In this figure, the name of each message is shown, followed by the contents of the message in round brackets. Square brackets are used to denote optional fields.

#### 8.5.2.1 PANA/UMTS Phases

The PANA/UMTS authentication procedure has three main phases: (1) Discovery and Handshake, (2) Authentication and Authorisation, and (3) Access. In the *Discovery* phase, an IP address for the PAA is identified, and

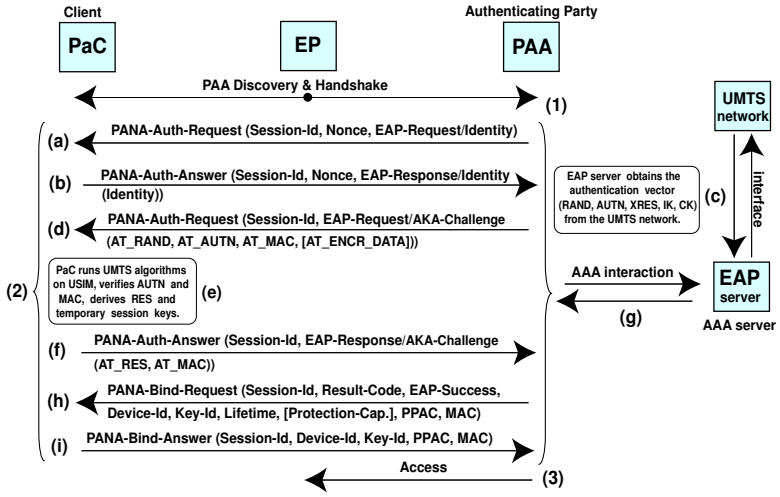


Figure 8.2: **PANA/UMTS full authentication procedure**

a PANA/UMTS session is established between the PaC and the PAA, following the PANA model (see section 6.1.4.1). After this phase is complete, a session identifier (*Session-Id* — see section 6.1.2) is allocated by the PAA and included in all further messages; this identifier is freed when the PANA/UMTS session terminates.

In the *Authentication* phase, the main focus of this section and further explained below, EAP-AKA messages encapsulated in PANA/UMTS messages are exchanged between the PaC and the PAA. In this phase, EAP-AKA Request payloads are carried in PANA-Auth-Requests. Moreover, taking advantage of an optimisation discussed in section 6.1.4.2 and adopted by PANA/UMTS, in the context of this chapter a PANA-Auth-Answer will include an EAP-AKA Response payload.

As previously discussed, the PAA communicates with the EAP server using the AAA Diameter EAP protocol (see section 3.9.3). Hence, EAP-AKA packets encapsulated in Diameter-EAP messages are exchanged between the PAA,

which is thus the *Diameter client*, and the EAP server, which is implemented on the *Diameter server*, following the process for using EAP in Diameter given in Figure 3.14. The PANA-Diameter message mapping, given in section 3.7.5, is also adopted here to allow the transport of EAP-AKA payloads between the PANA framework and the AAA Diameter infrastructure (see section 3.9.2). At the end of the Authentication phase, a PANA SA is established, including the provision of a shared secret EAP-AKA session key *MSK* (see section 8.4); we call this the PANA/UMTS SA.

During the *Access* phase, a separate protocol is used between the PAA and the EP to manage PaC network access control. After this phase, the established PANA/UMTS session and the PANA/UMTS SA are deleted, following the PANA draft standard (see section 6.1.4.5).

#### 8.5.2.2 PANA/UMTS Message Exchange

During the *Authentication* phase, the first PANA-Auth-Request message (*a*) issued by the PAA carries a PANA-based *Nonce*, i.e. a randomly chosen value (see section 6.1.3), used in further PANA/UMTS cryptographic key computations, and an EAP-Request/Identity payload, requesting the PaC to identify itself. The PaC responds (*b*) with a PANA-Auth-Answer, which also carries a PANA-based Nonce value, and an EAP-Response/Identity payload including the user identifier *Identity*.

The PAA then issues a Diameter-EAP-Request to the EAP server via an AAA interaction (see section 3.9.3), including the EAP-Response/Identity packet in an EAP-Payload AVP, and the user's identifier value in a Diameter User-Name AVP (see section 3.9.2). After receiving this message, the EAP server is able to obtain the user's authentication vector (*RAND*, *AUTN*, *XRES*, *IK*, *CK*) from the PaC's home UMTS network (*c*). Parts of this vector are subsequently used to derive certain temporary keys.

As specified in section 8.4, the EAP server can now derive the EAP-AKA keying material. This is achieved by inputting the concatenation of: the customer's *Identity* ($I$), the UMTS integrity key $IK$, and the UMTS confidentiality key $CK$ to the hash function SHA-1. The output is the secret key $MK$ (as given in equation 8.1). Using $MK$, the EAP server is able to derive the EAP-AKA keying material, including the $MSK$, which is used by PANA/UMTS as the AAA-Key (see section 6.1.2), and $K_a$, which is used to calculate the MAC.

The EAP server then sends back to the PAA a Diameter-EAP-Answer in a multi-round exchange, with a Result-Code AVP set to DIAMETER_MULTI_ROUND_AUTH, signifying that a subsequent request is expected. This exchange also includes an EAP-Request/AKA-Challenge payload, which contains the $RAND$ and $AUTN$ values obtained from the authentication vector, a MAC to protect the whole EAP packet, and an optional AT_ENCR_DATA field (see section 8.4). The MAC of this EAP-AKA payload is calculated by applying HMAC-SHA-1 (see section 2.1.3.2) to the EAP packet ($P$), as shown in equation (8.2), where here, as throughout, $f_Y(X)$ denotes an HMAC-SHA-1 MAC computed using the key $Y$ and data $X$:

$$MAC = f_{K_a}(P). \tag{8.2}$$

The next PANA/UMTS message ($d$) issued by the PAA encapsulates the received EAP-Request/AKA-Challenge payload detailed above. On receipt of this message, the PaC runs the UMTS AKA algorithm inside a USIM to derive the keying material and calculate $AUTN$, using the secret key $K$ and the copy of the sequence number that it maintains, as described in section 8.4. The PaC also computes, again using the USIM, the UMTS keys $CK$ and $IK$, which are obtained by applying the key generating functions $f3$ and $f4$ (see section

3.5.3.2) to $K$ and $RAND$, as given in equations (8.3) and (8.4):

$$CK = f3_K(RAND); \tag{8.3}$$

$$IK = f4_K(RAND). \tag{8.4}$$

The PaC then derives the secret key $MK$ (following equation 8.1) and the resulting EAP-AKA session keys, including $MSK$ and $K_a$. The authentication key $K_a$ is then used to calculate the MAC on the received EAP-Request/AKA-Challenge packet (see equation 8.2).

After computing a copy of $AUTN$ and MAC, the PaC checks that they are the same as the received values[4]. If the check succeeds, the PaC assumes that the received message is *fresh* (i.e. not a replay; see section 2.2.5) and that the sender possesses a valid authentication vector for the user (the EAP server is forbidden to reuse old authentication vectors). The PaC then derives $RES$ and the temporary keying material ($e$) for further use. The $RES$ value is computed in the USIM by applying the message authentication function $f2$ (see section 3.5.3.2) to $K$ and $RAND$, as shown in equation (8.5):

$$RES = f2_K(RAND). \tag{8.5}$$

If all the checks succeed, the PaC responds ($f$) with a PANA-Auth-Answer encapsulating the EAP-Response/AKA-Challenge payload, itself containing $RES$ and MAC. This MAC is computed as given in equation 8.2, i.e. using HMAC-SHA-1 (see section 2.1.3.2) on the EAP packet ($P$) with key $K_a$.

After receiving the EAP-Response/AKA-Challenge payload from the PAA via an AAA Diameter-EAP-Request, the EAP server verifies that the MAC

[4]If the $AUTN$ does not match, the PaC then sends back to the PAA an explicit error packet (EAP-Response/AKA-Authentication-Reject) inside a PANA-Auth-Answer message. If the MAC does not match, the PaC silently ignores the previous message and does not send any authentication results to the PAA.

is correct, and compares the received *RES* with the stored *XRES* from the authentication vector; if they agree, the PaC is deemed authentic.

The EAP server then sends back ($g$) an AAA Diameter-EAP-Answer, which includes a Result-Code AVP set to DIAMETER_SUCCESS. This message also includes an EAP-Payload AVP with a code field set to Success, which indicates that the authentication was successful. This EAP-Success packet carries derived AAA keying material, including an AAA-Key.

The PAA then encapsulates the PANA result code, the EAP-Success packet, and the PANA/UMTS *session lifetime* (see section 6.1.2) in a PANA-Bind-Request message sent to the PaC ($h$), and receives back an acknowledgement through a PANA-Bind-Answer ($i$). On receipt of this message, the PAA issues a Diameter Accounting-Request (Start) to the EAP server, which indicates the start of the session, following the PANA-Diameter message mapping given in section 3.7.5.

PANA-Bind messages are protected by a PANA/UMTS-based MAC AVP, the calculation of which is described in the following section, and carry a Key-Id AVP (see section 6.1.3); this latter AVP contains an AAA-Key identifier that is assigned by the PAA and is unique within the PANA/UMTS session.

Finally, PANA-Bind messages may also optionally contain a Protection-Capability AVP (see section 6.1.3), which is sent from the PAA to indicate that link-layer or network-layer encryption should be initiated after completion of PANA/UMTS. PANA-Bind messages are also used for binding the device identifiers of the PaC and the PAA to the PANA/UMTS SA established at the end of the authentication phase; this is achieved using a Device-Id AVP. PANA-Bind messages with a Result-Code AVP indicating successful authentication also include PPAC AVPs (see section 6.1.3), which help the PAA/PaC to negotiate the available/chosen IP address configuration method.

#### 8.5.2.3 PANA/UMTS-based MAC

The PANA/UMTS-based MAC ($M_{P_U}$) is calculated using HMAC-SHA-1, as given in equation (8.6), where $P_U$ denotes the PANA/UMTS packet, and $K_p$ denotes the PANA_MAC_Key (see section 6.1.5):

$$M_{P_U} = f_{K_p}(P_U). \tag{8.6}$$

The EAP-AKA shared secret *MSK*, which is used to establish a PANA/UMTS SA, is adopted as the AAA-Key, which is then used to generate the key $K_p$. However, as previously discussed, two AAA-Keys may be produced as a result of separate NAP and ISP authentication processes (see section 6.1.4.2). In this case, the two keys are concatenated to yield $K_{aaa}$, which is then used to compute $K_p$, as given in equation 7.8.

More specifically, and following equation (7.9), the PANA_MAC_Key $K_p$ is calculated by applying HMAC-SHA-1, using the key $K_{aaa}$ to the concatenation of the PANA-based Nonces $N_{pac}$ and $N_{paa}$, sent respectively by the PaC ($b$) and the PAA ($a$), with the PANA/UMTS Session-Id AVP value ($S_{id}$).

## 8.6 PANA/UMTS SA and Re-Authentication

Two important features of PANA/UMTS, namely the security association and the re-authentication procedure, are now described.

Once the EAP-AKA method has completed, a session key, i.e. the EAP-AKA *MSK*, which is used as the AAA-Key (as discussed in the previous section), is shared by the PaC and the PAA. This session key is provided to the PaC as part of the EAP key exchange process, and the PAA can obtain the session key from the EAP server via the AAA infrastructure. PANA/UMTS SA establishment

based on the EAP session key is required where no physical or link layer security is available (see section 4.2.3).

The purpose of a re-authentication exchange is to allow for efficient re-keying, using the existing PANA/UMTS security association, in situations where (depending on the security policy in force) full authentication is not required. Two types of re-authentication (or fast re-authentication) are supported by PANA/UMTS. The first type enters the chosen EAP method, i.e. the EAP-AKA fast re-authentication procedure (see section 5 of [20]), during the authentication and authorisation phase, and thus the initial discovery and handshake phase is omitted. The second type uses protected PANA/UMTS messages exchanged directly during the access phase, without entering the authentication and authorisation phase, i.e. the PANA re-authentication phase (see section 6.1.4.4).

## 8.7 PANA/UMTS with GAA Infrastructure

There is a problem that has not been addressed in this chapter. Section 8.5.1 states that, when using PANA/UMTS, the EAP server is assumed to be implemented on the AAA server (see section 3.9) and has an interface to the UMTS network (see section 3.5.3). It thus operates as a *gateway* between the Internet AAA network and the UMTS AKA infrastructure. However, the EAP server/UMTS network interface has not been defined. In this section, a possible solution, which incorporates part of the GAA infrastructure (see section 3.5.4) into the PANA/UMTS scheme, is proposed to address this problem.

Figure 8.3 shows a scheme in which the PANA/UMTS authentication protocol incorporates the GAA framework. The scheme is described immediately below.

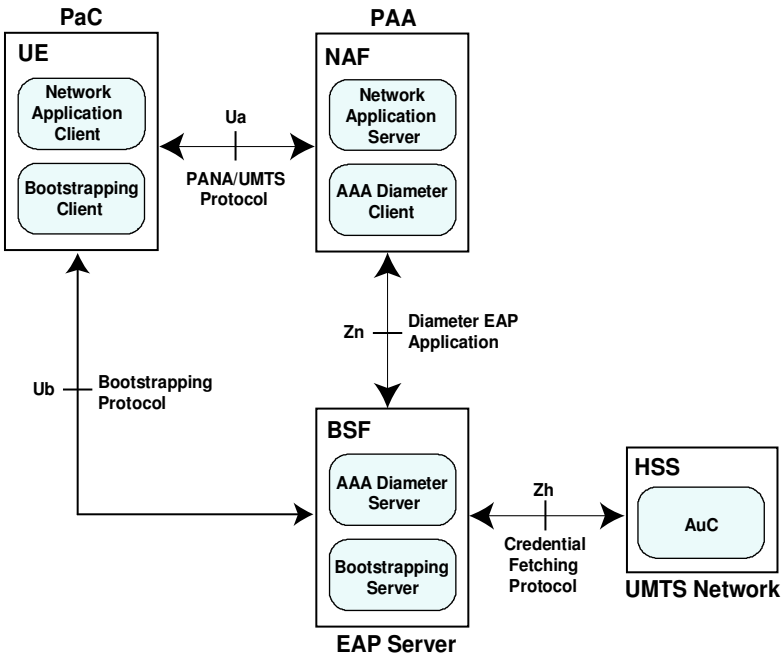


Figure 8.3: **PANA/UMTS incorporating the GAA framework**

### 8.7.1 PANA/UMTS with GAA Entities

As previously stated, the PaC is associated with a network device, which contains a set of UMTS credentials stored in a USIM. In the solution proposed below, this network device also hosts the set of GAA functionalities required from a UE. The PAA then authenticates the UMTS credentials provided by the PaC, via the PANA/UMTS protocol, and grants network access. The network access device which hosts the PAA is implemented on a client of an AAA Diameter infrastructure; it also hosts the set of functionalities required from a NAF.

This AAA Diameter client communicates, via the Diameter EAP application (see section 3.9.3), with the user's EAP server implemented on the AAA Diameter server. The network element which implements the EAP server also hosts the set of functionalities required from a BSF. As stated before, the EAP

server, which is implemented in the AAA server, has an interface (*Zh*) to the UMTS network, in particular to the home AuC in the HSS. This interface is described immediately below.

### 8.7.2 An Internet AAA and UMTS AKA Interface

As described in section 3.5.4.3, the BSF (i.e. the EAP server) has a bootstrapping interface (*Zh*) with the HSS (i.e. the UMTS network), with which it performs the *credential fetching protocol*. This protocol is based on a Diameter application protocol (see section 3.9.2) and is used to fetch the required authentication information, i.e. authentication vectors and GBA User Security Settings (GUSS[5]), from the home AuC in the HSS. Section 4 of [3] gives a complete description of the application logic of the interface (*Zh*) between BSF and HSS, while section 4.4.5 of [7] establishes the requirements for this interface. A summary of the protocol hierarchy of the interface (*Zh*) is shown in Figure 8.4 [3]. The Diameter Base protocol is described in section 3.9.2, and the Diameter application protocol is given in a 2007 3GPP TS [2].

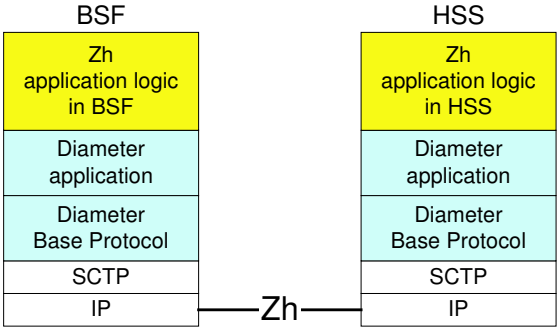


Figure 8.4: **Protocol hierarchy of the *Zh* interface**

According to a 2007 3GPP TS [3], the overall GAA bootstrapping procedure

[5]A *GUSS* includes an application, i.e. a service offered by the mobile network operator (or a third party) to the mobile subscriber, and a subscriber parameter set that contains two parts: an authentication part, which contains the list of needed user identifiers, and an authorisation part, which contains the user permission flags (see section 3.1 of [7]).

can be summarised in the following three steps:

1. A UE starts the bootstrapping procedure via the interface *Ub* with the BSF, with which it executes the *bootstrapping protocol*, by passing the user identifier (see section 16 of [4]). As stated in section 8.4, either permanent identities, usually based on the IMSI, or temporary identities (pseudonyms), which are equivalent to the TMSI, can be employed as part of the user identifier.

2. The BSF starts the credential fetching protocol using the bootstrapping (*Zh*) interface with the user's HSS, to request an authentication vector and GUSS corresponding to the user identifier provided. The HSS then supplies the BSF with the requested authentication vector and GUSS (if any)[6].

3. The BSF continues the bootstrapping protocol via the interface *Ub* with the UE (see section 4 of [1]).

In the solution proposed in this section, which incorporates part of the GAA framework into PANA/UMTS, steps 1 and 3 listed above, involving the interface *Ub*, can be omitted. This is because the PaC (i.e. the UE), as previously shown in item (*b*) of Figure 8.2, sends to the PAA (i.e. the NAF) a PANA-Auth-Answer encapsulating an EAP-Response/Identity payload, which includes the user identifier. By sending a Diameter-EAP-Request, the PAA then forwards this user identifier to the EAP server (i.e. the BSF) in a Diameter User-Name AVP (see sections 3.9.2 and 8.5.2.2).

Finally, by using the GAA bootstrapping (*Zh*) interface (step 2 above), the EAP server can now operate as a *gateway* between the Internet AAA network and the UMTS AKA infrastructure, performing the retrieval of authentication

[6] If more than one HSS is deployed within the UMTS network, the BSF may have to contact the SLF using the *Dz* interface, prior to sending the request to the HSS (see section 3.5.4.3)

vectors and the GUSS from the HSS. This provides a complete solution, addressing the problem raised above. In fact, the solution described here could be applied directly to EAP-AKA, since GAA *Zh* can be used to allow an EAP-AKA server to obtain authentication vectors from the HSS[7].

## 8.8 Conclusions

As previously discussed, authentication and key agreement are fundamental components of a secure procedure for heterogeneous network access supporting ubiquitous mobility. The main challenges addressed here include the investigation and development of unified, secure and convenient authentication mechanisms that can be used in access networks of a wide range of types.

In this chapter, we have proposed the new PANA/UMTS protocol, in order to provide an IP-compatible, lightweight, flexible and scalable method for authenticating a user to an access network. The protocol is based on PANA, a network-layer access authentication protocol carrier, which communicates, via EAP, with an AAA infrastructure interacting with an AuC in the UMTS mobile network. PANA/UMTS uses EAP-AKA, which allows use of the AKA infrastructure in network scenarios in which mobile devices are equipped with a USIM.

Use of UMTS authentication vectors minimises the necessary trust relationship between operators, thereby increasing the likelihood of successful use. From the user perspective, the protocol works with a 'standard' UMTS USIM (or even a 'standard' CDMA2000 (R)UIM) card and requires only an appropriate Internet access device and a USIM (or (R)UIM) card reader, which may or may not be integrated with the access device. The gains in performance arising from the

[7]Although the EAP-AKA server is able to obtain authentication vectors from the home AuC in the HSS, the communication between the EAP server and the HSS is outside the scope of the EAP-AKA specification (see section 8.4).

two types of fast re-authentication, the gains in security from the PANA/UMTS SA, and the gains in flexibility and scalability by incorporating part of the GAA architecture, potentially make the PANA/UMTS proposal attractive to UMTS operators willing to offer their users heterogeneous Internet access in ubiquitous mobility networks.

This new Internet authentication scheme, designed to meet the requirements established in Chapter 5, is proposed here as a candidate for secure access procedure for heterogeneous network access supporting ubiquitous mobility (see section 1.1). In Chapter 11, the new scheme is submitted to a formal threat modelling process; it is also compared with the three further novel Internet entity authentication techniques proposed in Chapters 7, 9, and 10.

# Chapter 9

# PANA/Liberty

## Contents

As explained in Chapter 7, this thesis proposes a series of new solutions for Internet remote access authentication, derived by adapting and reinforcing security techniques arising from a variety of different sources. The aim of this chapter is to present the third new authentication scheme, namely a means of combining the Liberty Alliance architecture (see section 3.10) and the 3GPP GAA security mechanisms (described in section 3.5.4), with PANA (see section 3.7.5 and Chapter 6), which we call PANA/Liberty.

## 9.1 Introduction

As described in section 7.1, in some ubiquitous mobility scenarios, IP based remote hosts that connect to the Internet via an access network will typically need to provide their credentials and be authenticated before being authorised to access the network, demanding an easy-to-use, strong, and scalable entity authentication infrastructure. The Liberty Alliance architecture (see section 3.10) offers an open Single Sign-On (SSO) standard, including decentralised authentication and authorisation from multiple providers. The 3GPP GAA framework (see section 3.5.4) offers the cellular authentication mechanisms to other mobile applications, providing to communicating entities either a shared secret based on 3GPP AKA (see section 3.5.3.2) or digitally signed public key certificates (see section 2.1.3.3). Although the Liberty protocols have been defined independently of GAA, according to Laitinen et al. [124] the two schemes can complement each other. Therefore, it seems potentially desirable to combine Liberty and GAA to help to build an authentication infrastructure for Internet remote access.

In Chapter 8 we presented the new PANA/UMTS scheme, which combines UMTS AKA (see section 3.5.3.2), an internationally accepted standard for the new generation of mobile services, with PANA (see Chapter 6), which is intended to be a flexible and scalable generic network layer protocol for authenticating a user device requesting Internet remote access. In particular, a possible variant of PANA/UMTS, proposed in section 8.7, brings to this scheme the potential advantages of incorporating part of the GAA framework (see section 3.5.4.5). GAA offers a standardised, generic way to reuse the cellular network authentication infrastructure deployed in subscriber smart cards for other mobile services. In addition, the Liberty architecture aims to provide an open SSO standard, and create a network identity infrastructure supporting all network access devices (see section 3.10.1).

Building on the above observations, we now present a new authentication scheme, combining the Liberty Alliance Project framework and 3GPP GAA security mechanisms with PANA, which we call PANA/Liberty. This innovative proposal first incorporates the security techniques used in the UMTS mobile telecommunication system and part of the GAA infrastructure into the PANA authentication structure; this scheme is then complemented by the Liberty SSO service, which can be used to extend this initial authentication to all Liberty-enabled Service Providers (SPs), in a solution designed to support ubiquitous client mobility for Internet access.

The purpose (section 9.2) and the components used in the assembly (section 9.3) of the novel PANA/Liberty scheme are first given. Second, a description of how two of the PANA/Liberty components, namely the Liberty Alliance framework (see section 3.10) and the GAA architecture (see section 3.5.4) can be used in combination (section 9.4), is provided. The framework of the proposed new PANA/Liberty protocol is then given (section 9.5). Next, two important features of PANA/Liberty, namely the security association and the re-authentication procedure (section 9.6), are described. This is followed by a description of other possible ways in which PANA can be integrated into Liberty (without the GAA framework), including a discussion of alternative schemes that may be used as the PANA inner authentication protocol instead of 3GPP AKA (section 9.7). Finally, the conclusions of the chapter are given (section 9.8).

The main novel contribution of this chapter lies in sections 9.3, 9.5, 9.6, and 9.7. Whilst the use of a combination of the Liberty Alliance architecture and the 3GPP GAA security mechanisms described in section 9.4 has been previously described (notably in the 3GPP TR 33.980 Specification [11]), the details of how they would operate when executed with PANA have not. This chapter does not contain a detailed security analysis of the new proposal — this issue is covered in Chapter 11.

## 9.2 PANA/Liberty Objective

Currently there is no standard protocol for performing network access authentication above the link layer. Instead, a number of ad hoc and often inadequate solutions (as described in section 4.1.5) are being used to overcome the problem (itself described in section 4.1), in a variety of distinct scenarios (outlined in section 4.2).

The objective of the PANA/Liberty protocol is thus to provide a network layer, IP compatible, lightweight, attack-resistant (e.g. with respect to MitM and DoS attacks — see section 3.2.3), and relatively flexible authentication method, that allows a remote client to be authenticated in a heterogeneous Internet access environment supporting ubiquitous mobility. This authentication method must meet a number of detailed security and implementation requirements, as specified in Chapter 5.

## 9.3 PANA/Liberty Protocol Hierarchy

In this section, an overview of the components used in the construction of the new PANA/Liberty authentication scheme is given. The first component, as previously discussed, is the UMTS USIM authentication and key agreement mechanism. Section 3.5.3 gives an outline of the UMTS system security features, including the authentication and key agreement (AKA) scheme.

The second component used in the PANA/Liberty protocol assembly is EAP (see section 3.4). As previously described, EAP is very flexible, gives providers the advantage of using a single framework across multiple environments, and can encapsulate arbitrary EAP methods. In particular, the EAP-AKA protocol, an EAP method specified in RFC 4187 [20], describes a way of encapsulating the security parameters used by the UMTS AKA system within EAP, in order

to provide mutual authentication and session key agreement using the UMTS USIM. However, the effective use of EAP-AKA for ubiquitous client mobility in an Internet access environment requires the provision of a transport scheme independent of the access network type.

The third component used in the construction of our proposed technique is PANA. In Chapter 6, we justified the selection of PANA, a UDP-based protocol (see section 6.1.1), as the tunnelled network layer transportation environment. In fact, the PANA/UMTS scheme, which we proposed in Chapter 8, uses PANA to support the use of EAP-AKA for Internet remote access authentication, combining UMTS authentication with EAP-AKA and PANA interacting with Diameter EAP (see sections 3.7.5 and 3.9.3). Therefore, part of the PANA/Liberty solution incorporates the components used in the PANA/UMTS protocol assembly.

In section 8.7, we described how to incorporate part of the GAA infrastructure (see section 3.5.4) into the PANA/UMTS scheme. Since GAA offers an easy-to-use, generic, and scalable way to reuse the cellular network authentication infrastructure deployed in subscriber smart cards to other mobile application users (such as PANA/UMTS customers), it is clearly an architecture that satisfies many of the requirements for a variety of authentication scenarios (see sections 4.2 and 5.1.3). As a result, GAA is our choice for the fourth component in the construction of the new PANA/Liberty authentication scheme.

We describe here how to use the Liberty SSO standard, which provides a network identity infrastructure supporting all network access devices, to extend the PANA/UMTS user authentication to all Liberty-enabled SPs. As stated previously, and also explained immediately below, Liberty is able to interact with 3GPP GAA. Consequently, Liberty is our choice for the fifth component in the construction of our proposed technique, which thus combines the Liberty and 3GPP GAA security mechanisms with PANA interacting with Diameter

EAP, into a scheme which we call PANA/Liberty.

A summary of the PANA/Liberty protocol hierarchy is shown in Figure 9.1. The Diameter Base protocol is described in section 3.9.2, and the Diameter application protocol is given in section 3.5.4.3. The 3GPP GAA *Zh* interface is explained in sections 3.5.4.3 and 8.7.2. The SSL/TLS protocols are described in section 3.6.3. Finally, the HTTP, XML, SOAP, and SAML standards from the Liberty protocol hierarchy are outlined in sections 3.10.3 and 3.10.4.

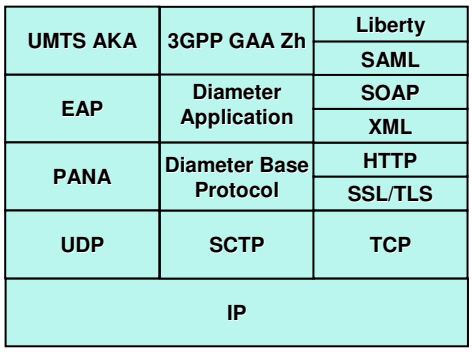


Figure 9.1: **PANA/Liberty protocol hierarchy**

## 9.4 Liberty with GAA Infrastructure

In this section, a description of how two of the PANA/Liberty components, namely the Liberty Alliance framework (see section 3.10) and the 3GPP GAA architecture (see section 3.5.4) can be used in combination, is provided. The 3GPP TR 33.980 Specification [11] describes the possible interworking methods which can be used between Liberty ID-FF (see section 3.10.5), Liberty ID-WSF (described in 3.10.6), and GAA GBA (given in section 3.5.4).

Firstly, we describe how the Liberty and GAA authentication schemes can complement each other (section 9.4.1). Secondly, a description of the archi-

tecture for a collocated NAF/IdP service is provided (section 9.4.2). Next, we summarise the federation and session concepts of Liberty with GAA (sections 9.4.3 and 9.4.4). We then illustrate three possible SSO scenarios involving Liberty with GAA (section 9.4.5).

### 9.4.1 Liberty with GAA Authentication

Both the Liberty and GAA systems separate the authentication procedure from the process of accessing services. The Liberty scheme reuses a single initial authentication of the user for successive authentication to other SPs. However, as noted by Laitinen et al. [124], the Liberty documents do not specify how the initial user authentication is done, or how to provision user credentials.

As the GAA security mechanisms provide the means to authenticate the user by reusing already deployed user credentials (for example, in the form of UMTS USIMs — see section 3.5.3.1), the Liberty and GAA authentication schemes can thus complement each other. GAA can be used first to authenticate the user, and the Liberty SSO service can then be used to extend this authentication to all Liberty-enabled SPs.

In this case, as described immediately below, the Liberty Identity Provider (IdP) would function as a network application server (see section 4.2.1.1 of [11]) which, as discussed in section 3.5.4.3, implements the GAA network application function (NAF).

### 9.4.2 Architecture for collocated NAF/IdP

According to the 3GPP TR 33.980 Specification [11], when an IdP is collocated with a NAF, the NAF/IdP host authenticates the User Equipment (UE) using the GAA credentials. There is only one reference point carrying both Liberty

and GAA related information, i.e. the reference point between a NAF/IdP host and a UE. The architecture for a collocated NAF/IdP service, further described below, is shown in Figure 9.2.

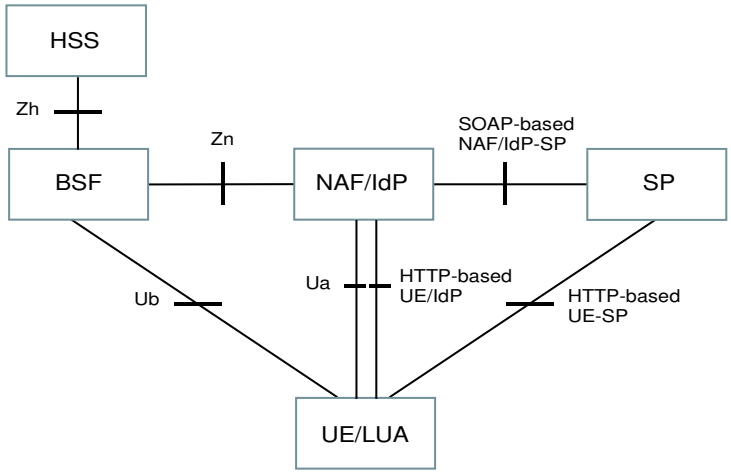


Figure 9.2: **Liberty ID-FF and GAA with a collocated NAF and IdP**

As discussed in section 3.5.4.3, the BSF has an interface (*Zh*) with the HSS, with which it performs the *credential fetching protocol*, used to fetch the required authentication information (i.e. authentication 'quintets' and GBA User Security Settings — GUSS, described in section 8.7.2) from the home AuC in the HSS. The UE runs 3GPP AKA (see section 3.5.3.2) with the HSS via the BSF. The UE has an interface (*Ub*) with the BSF, across which the *bootstrapping protocol* is executed. A shared session key derived from the (*CK*, *IK*) key pair is then established in the BSF and UE, using this bootstrapping protocol.

The NAF/IdP fetches the session key from the BSF, together with subscriber profile information (e.g. GUSS), via an interface (*Zn*) using the *key distribution protocol*. The NAF/IdP and the UE will then share a secret key that can be used for application security, in particular to mutually authenticate the UE and the NAF/IdP. The use of GAA credentials between the UE and the NAF/IdP occurs via an interface (*Ua*) using the *application protocol*, which is secured

using the keying material previously agreed via the interface (*Ub*) between the UE and the BSF (see section 3.5.4.3).

The protocols used to initiate the single sign-on and identity federation processes in the UE are defined in Liberty ID-FF (see section 3.10.5). The UE may implement a Liberty-enabled User Agent (LUA — see section 3.10.8). All Liberty ID-FF and ID-WSF specific tasks (see sections 3.10.5 and 3.10.6) are executed by the IdP implementation in the NAF; this procedure is transparent to the set of GAA functionalities implemented in a UE.

### 9.4.3 Federation in Liberty with GAA

As discussed in section 3.10.5, Liberty adopts the concept of *federating* user identities, which involves linking distinct SP and IdP user accounts, and associating an opaque user handle with the two user identities. This relationship between two entities requires a *mapping*. In order to map the GAA credentials and the Liberty information, the NAF/IdP maintains a *user table*. Following section 4.3.1 of [11], in the case of *non-anonymous* user access, the NAF/IdP has two options for the identifier used to label this table:

- the IP Multimedia Private Identity (defined in section 13.3 of [4]); or
- the User Identifier (UID), which may be an IP Multimedia Public Identity (defined in section 13.4 of [4]).

The table also stores the bootstrapping transaction identifier (B-TID[1]), the key lifetime data, the key generation time, and the opaque user handles. This table may also contain the NAF keying material, the GUSS, and further SP related data. The table logically separates temporary GBA related data (e.g. B-TID, key lifetime) from the Liberty IdP related and persistent data (e.g. SP

[1] A *B-TID* is used to bind the subscriber identity to the keying material in reference points *Ua*, *Ub* and *Zn* (see section 3.1 of [7]).

related data, the opaque user handle, and the GUSS). The temporary GBA data is deleted on the expiry of the key or the Liberty session. The IdP related data and the user identifier are permanent (see section 4.3.1 of [11]).

For *anonymous* user access, the B-TID is used as a temporary user identifier for the table. The federation lasts as long as the Liberty session, and the maximal length of the federation is the key lifetime. In this anonymous user case, the whole table is temporary. NAF/IdP can manage *defederation* by deleting the opaque user handles and SP related information from the table.

### 9.4.4 Liberty with GAA Session

The duration of a Liberty-GAA session depends on the key lifetime of the NAF keying material. The maximum Liberty session lifetime must be at most the remaining lifetime of the key. When the Liberty session expires, the temporary GBA related data is deleted from the user table. If a session is explicitly terminated (e.g. via Single Logout, described in section 3.10.3.2), then the temporary GBA related data is deleted in the NAF/IdP. For the next login, the UE would be required to execute the bootstrapping procedure again (discussed in section 3.5.4.3), since it no longer shares a key with the NAF/IdP.

If a new bootstrapping procedure has been executed since the last contact between UE and NAF/IdP, the new temporary GBA related data is inserted into the user table. If the *freshness* of the received keying material (see section 2.2.5) is not satisfactory, then the NAF/IdP sends a *bootstrapping renegotiation* request to the UE (as outlined in section 4.5.3 of [7]) and uses the new keying material for the Liberty session.

When a UE acting on behalf of a user initiates a Liberty-GAA session with the NAF/IdP, it contacts the NAF/IdP via the *Ua* reference point, and a *mutual authentication* process (described in section 5.4 of [5]) is started. Depending

on the entries in the table of the NAF/IdP, three possible approaches can be followed:

1. If the B-TID exists in the table and has not expired, the NAF/IdP has all the required data, and can thus start communication with the UE without communicating via the *Zn* reference point[2].

2. If the B-TID is not present in the table, and the GUSS received over *Zn* contains a user identity which already exists in the table, then the entry in the table is updated with the B-TID and related information.

3. If the B-TID is not present in the table, and the GUSS received over *Zn* contains a user identity which is not present in the table or no user identity is sent, then the IdP creates a new entry in the table.

### 9.4.5 Liberty with GAA Scenarios

In the 3GPP TR 33.980 Specification [11], three possible message flows for SSO scenarios are outlined:

**Liberty-GAA ID-FF with *AuthnResponse* transfer.** In this scenario, described in section 4.3.3 of [11], the UE is not Liberty-enabled. The protocol elements are taken from within the Liberty ID-FF component (see section 3.10.5), and are complemented by the GAA specific details from the secure access methods to NAF using HTTPS (see section 3.5.4.1).

**Liberty-GAA ID-FF with artifact transfer.** This scenario is similar to the previous one, with the extension that the SP is able to contact the IdP directly. The IdP supports an additional interface to the SP, to allow the SP to retrieve the authentication assertion (see section 4.3.4 of [11]).

[2]If the IdP decides that the remaining lifetime of the B-TID is too short, it may indicate to the UE that a bootstrapping renegotiation is required. In this situation, the procedure will be similar to case 2.

**Liberty-GAA ID-WSF authentication service.** In this scenario, the UE implements a LUA, which is a Liberty-enabled client that has (or knows how to obtain) knowledge about the IdP that the user wishes to use with the SP (see section 4.4.3.3 of [56]). The protocol elements are taken from within the Liberty ID-WSF component (see section 3.10.6), in particular from the Liberty ID-WSF authentication service, which permits a LUA to initially authenticate with an IdP and obtain a 'security token' (see section 3.10.8), and the interaction of the UE with the IdP involves two consecutive protocol runs. The active LUA client first contacts the NAF/IdP, before accessing the service provided by the SP (see section 4.3.5 of [11]). The SSO interactions of the UE with the IdP and the SP are specified in a single sign-on protocol profile (see section 3.10.3.2), called the *Liberty-Enabled Client and Proxy Profile* (detailed in section 3.2.4 of [37]).

## 9.5 PANA/Liberty Framework

In this section, the authentication framework for the new PANA/Liberty scheme is described. The entities involved in the PANA/Liberty method are first given (section 9.5.1). The PANA/Liberty authentication scheme is then explained (section 9.5.2).

### 9.5.1 PANA/Liberty Entities

The PANA/Liberty method proposed here involves five entities, namely the PaC (also referred to here as the *client*, *user*, *customer* or *subscriber*), the PAA (or *authenticating party*), the Liberty-enabled SP, the EAP server, and the home AuC in the HSS. These PANA/Liberty entities are shown in Figure 9.3.

The PaC is associated with a network device and a set of GAA credentials

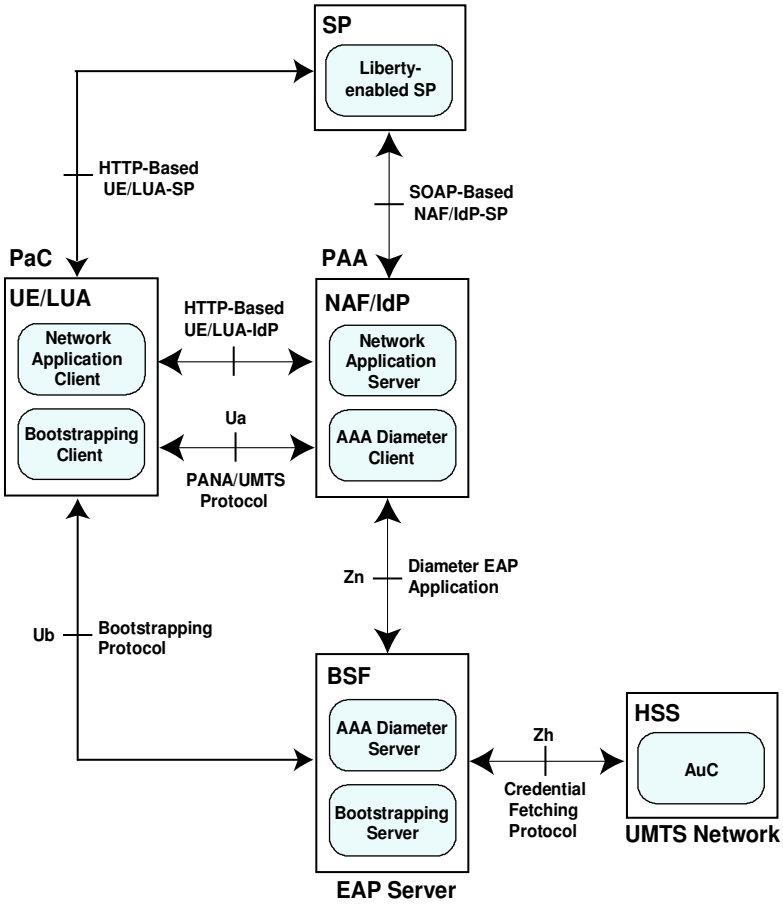


Figure 9.3: **Entities involved in the PANA/Liberty scheme**

stored in a USIM; these credentials are used to establish the PaC identity for the purposes of network access. In the solution proposed below, this network device hosts the set of GAA functionalities required from a UE; it also implements a LUA.

The PAA authenticates the GAA credentials provided by the PaC, using the PANA/UMTS protocol incorporating the GAA framework (see section 8.7), and grants network access. As stated before, the Liberty SSO service can be used to extend this authentication to all Liberty-enabled SPs. The network access device which hosts the PAA is implemented on a client of an AAA Diameter infrastructure (see section 3.9); it also hosts the set of functionalities required

from a GAA NAF and a Liberty IdP.

The PAA has an interface (*Zn*) with the user's EAP server (implemented on the AAA Diameter server), across which the Diameter EAP application (see section 3.9.3) is executed. The network element which implements the EAP server also hosts the set of functionalities required from a GAA BSF. As stated in section 3.5.4.3, the EAP server has an interface (*Zh*) to the UMTS network, in particular to the home AuC in the HSS, operating as a *gateway* between the Internet AAA network and the UMTS AKA infrastructure.

PANA/Liberty also involves a further entity, namely the EP (see section 6.1.2), which applies per-packet enforcement policies (i.e. filters) to the traffic of the PaC's devices.

### 9.5.2 PANA/Liberty Authentication Scheme

The aim of this section is to give a detailed description of the PANA/Liberty scheme. Figure 9.4 shows the PANA/Liberty authentication procedure, which has five main phases: (1) Discovery and Handshake, (2) Authentication and Authorisation, (3) Network Access, (4) Internet Single Sign-On, and (5) Service Access. These phases are explained immediately below.

#### 9.5.2.1 Discovery and Handshake Phase

In the *Discovery* phase, the user first contacts the NAF/IdP of her choosing, before accessing the service provided by the Internet SP by entering the corresponding Uniform Resource Locator (URL) into her web browser (see section 3.10.3). The UE/LUA acting on behalf of the user (i.e. the PaC) then initiates a Liberty-GAA session (see section 9.4.4) with the NAF/IdP (i.e. the PAA) via the *Ua* reference point. In addition, a PANA/Liberty session is established

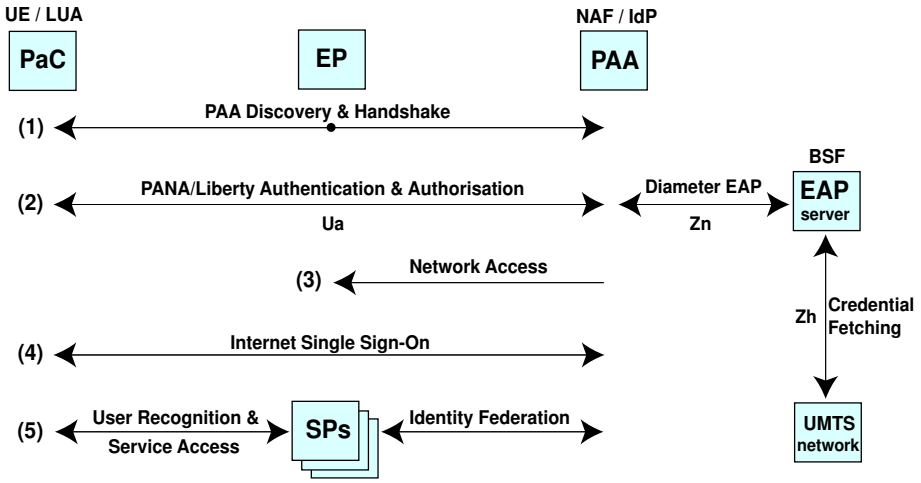


Figure 9.4: **PANA/Liberty authentication procedure**

between the PaC and the PAA, following the PANA model (see section 6.1.4.1).

After this phase is complete, a session identifier (*Session-Id* — see section 6.1.2) is allocated by the PAA and included in all further messages; this identifier is freed when the PANA/Liberty session terminates.

#### 9.5.2.2 Authentication and Authorisation Phase

In the PANA/Liberty *Authentication* phase, EAP-AKA messages encapsulated in PANA messages are exchanged between the PaC and the PAA, through the *Ua* interface, following the PANA/UMTS authentication phase incorporating the GAA framework (detailed in sections 8.5.2.2 and 8.7). The Liberty-GAA collocated NAF/IdP service (described in section 9.4.2), which maintains a *user table* for mapping the Liberty information and the GAA credentials (see section 9.4.3), complements this authentication procedure.

Depending on the entries in the NAF/IdP user table, a number of possible approaches can be followed (as given in section 9.4.4). If the B-TID exists in the table and has not expired, then the NAF/IdP has all the necessary data, and

can thus start communication with the UE/LUA without communicating via the *Zn* reference point. Otherwise, the NAF/IdP fetches the session key from the BSF (i.e. the EAP server), together with a GUSS corresponding to the user identifier, using the Diameter EAP protocol (see section 3.9.3).

In this case, as explained in section 8.7.2, the BSF starts the credential fetching protocol, using the *Zh* interface with the user's HSS, to request an authentication vector and GUSS. Once the HSS has supplied the BSF with the requested authentication vector and GUSS from the PaC's home UMTS network, the PAA and the PaC will thus share a secret key, which is used to mutually authenticate UE/LUA and NAF/IdP.

Moreover, when the UE/LUA authenticates with the NAF/IdP, it retrieves from it a Liberty 'security token' (see sections 3.10.8 and 9.4.5), carried as a 'child' payload of a SOAP header. This 'security token' entitles the user to invoke the Single-Sign-On service of the IdP. At the end of the Authentication phase, a PANA SA is established, including the provision of a shared secret EAP-AKA session key *MSK* (see section 8.4); we call this the PANA/Liberty SA.

Once the PAA has authenticated the GAA credentials provided by the PaC and authorises network access, the chosen NAF/IdP web page is displayed to the user. This web page typically offers links to multiple Internet SPs in the circle of trust maintained by NAF/IdP.

#### 9.5.2.3 Network Access Phase

During the *Network Access* phase, a separate protocol is used between the PAA and the EP to manage PaC network access control. At any time while the network access phase is 'live', the Internet Single Sign-On (and, after that, the Service Access) phase can be started by the user. I.e., at some later time,

the user can use her UE/LUA to visit one of the multiple affiliated (Liberty-enabled) SPs in the circle of trust maintained by NAF/IdP, in order to access specific Internet services. She can do this by choosing a SP link available in the NAF/IdP web page or simply by entering a URL.

At the end of this phase, the established PANA/Liberty session and the PANA/Liberty SA are deleted, following the PANA draft standard (given in section 6.1.4.5).

#### 9.5.2.4 Internet Single Sign-On Phase

In the *Internet Single Sign-On* phase, the interaction of the UE/LUA with the NAF/IdP involves two consecutive protocol runs, following the Liberty-GAA ID-WSF scenario given in section 9.4.5. The user first logs in to the NAF/IdP, which subsequently helps the user to be automatically authenticated (without having to sign on again) to the visited SP, by offering guarantees of the user network identity. The UE/LUA then invokes the single sign-on service of the NAF/IdP using the 'security token' (denoted here by <sec:Token>) previously provided.

In this phase, we assume that the UE/LUA has already been authenticated by the NAF/IdP. Thus, a valid PANA/Liberty session exists for the user at the identity provider. The UE/LUA receives the authentication assertion (i.e. the authentication and authorisation information) from the NAF/IdP to be used at the visited SP. This transfer of authentication assertion requires direct interaction between NAF/IdP and SP. As shown in Figure 9.3, the Liberty protocol used for this interaction is SOAP based (see section 3.10.3.2), with SAML assertions[3] (see section 3.10.4) carrying the assertion information. This interaction is outside the scope of the description here.

[3]The content of each SAML assertion is (partly) given by the results of the GBA operation (see section 3.5.4.3), including information such as protocol parameters (e.g. execution time) and user-specific parameters (e.g. taken from GUSS).

After this step, the UE/LUA presents the authentication assertion to the SP to obtain user recognition and to be given web service access. Figure 9.5 summarises the PANA/Liberty SSO phase, which is further described below. In this figure, the PANA/Liberty SSO message flow is shown, including the Liberty-specified HTTP headers in the messages sent and received by the UE/LUA, signifying to IdPs and SPs that it is 'Liberty-enabled'[4] (see section 9.4.5).

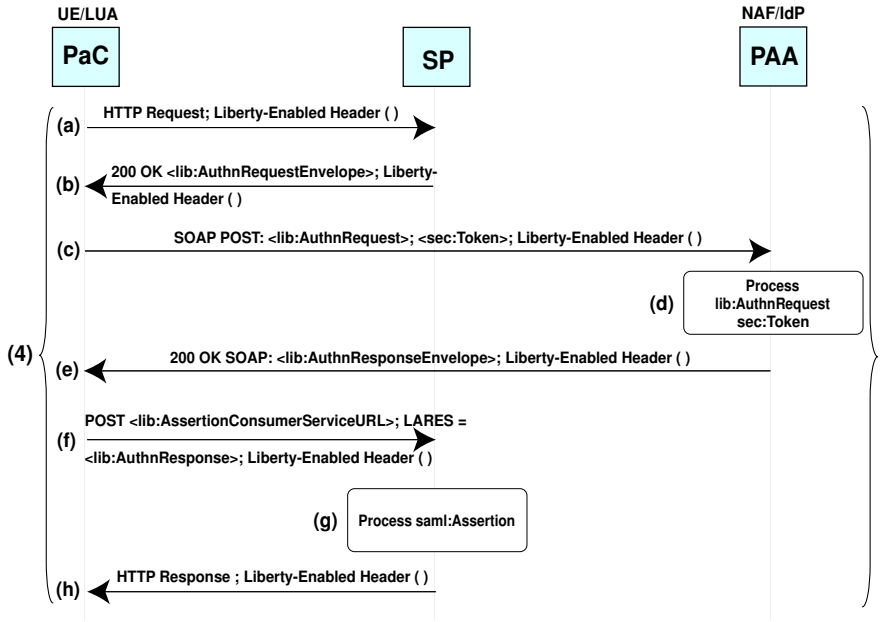


Figure 9.5: **PANA/Liberty SSO message flow**

During the PANA/Liberty SSO phase, the first HTTP Request message (*a*) is issued by the UE/LUA to access resource offerings (see section 3.10.6) available at the chosen SP. The visited SP responds (*b*) with an HTTP 200 OK answer, which carries an authentication request <lib:AuthnRequestEnvelope>[5] payload including an <lib:AuthnRequest> element and a list of the identity

[4] I.e., signifying to IdPs and SPs that the UE/LUA can support capabilities beyond those supported by common non-Liberty-enabled user agents.

[5] The contents of the payloads: <lib:AuthnRequestEnvelope>, <lib:AuthnRequest>, <lib:AuthnResponseEnvelope>, <lib:AssertionConsumerServiceURL>, and <lib:AuthnResponse> are defined in the Liberty SSO and Federation Protocol specification (see section 3.10.5.3).

providers that it recognises.

On receipt of this message, the UE/LUA chooses the appropriate NAF/IdP to use (i.e. the one the user has already been authenticated by) and then issues ($c$) an HTTP POST message that encapsulates the received <lib:AuthnRequest> element in the body and the <sec:Token> element in the <wsse:Security> header of a SOAP request packet. At this point, the identity provider's SSO service URL processes[6] the <lib:AuthnRequest> and <sec:Token> elements ($d$), confirms that the UE/LUA has already been authenticated by the NAF/IdP, and obtains consent from the user for federating (or not) the existing SP local identity with the identity she has at the NAF/IdP (see the section immediately below).

The NAF/IdP then sends back to the UE/LUA an HTTP 200 OK message ($e$). This message includes a SOAP response packet, encapsulating a <lib:AssertionConsumerServiceURL> and a <lib:AuthnResponse> element in a <lib:AuthnResponseEnvelope> payload. The next HTTP POST message ($f$) issued by the UE/LUA carries this latter <lib:AuthnResponse> value in a POST form field called 'LARES'; this message also includes the received <lib:AssertionConsumerServiceURL> value. This value contains the SP's assertion consumer service URL, consisting of the target of the POST form. This target must specify HTTPS (see section 3.5.4.1) as the URL scheme.

At this point, the SP processes the SAML assertion[7] (<saml:Assertion>) value ($g$) received in the <lib:AuthnResponse> element, to check its validity and how to respond to the UE/LUA's original request. The signature on the <saml:Assertion> value must be verified in order to gain assurance regarding the identity of the user. Finally, the SP sends back to the UE/LUA an HTTP response ($h$) that either allows or denies access to the web service resources that

[6]This is executed according to the rules defined in the Liberty SSO and identity federation protocol (see sections 3.10.5.3 and 3.10.8).

[7]The SP processing of this assertion must adhere to the rules defined in the SAML specification (see section 3.10.4).

were originally requested.

#### 9.5.2.5 Identity Federation Process

As stated in section 3.10.5.1, when a user first uses an IdP to login to an SP, he/she must be given the opportunity to federate any existing SP local identity with the identity she has at the IdP. *Identity federation* involves linking distinct SP and IdP user accounts, associating an opaque user handle with the two local user identities; this requires permission to be granted by the user.

After identity federation, the SP and the NAF/IdP share a pair of unlinkable pseudonyms (opaque user handles) for the user, one for each direction. The Liberty single sign-on and identity federation processes are supported by the Single Sign-On and Federation protocol (described in section 3.10.5.3).

#### 9.5.2.6 Service Access Phase

During the *Service Access* phase, a separate (HTTP-based) protocol is used between the UE/LUA and the SP to manage user web service access control. Once this phase is complete, the user is able to access web services provided by the SP.

At the end of this phase, the established Liberty session is deleted, following the Liberty Single Logout standard (given in section 3.10.3.2), which provides synchronised session logout functionality across all sessions that were authenticated by a particular IdP.

## 9.6 PANA/Liberty SA and Re-Authentication

Two important features of PANA/Liberty, namely the security association and the re-authentication procedure, are now described.

Once the EAP-AKA method has completed, a session key, i.e. the EAP-AKA *MSK*, which is used as the AAA-Key (as discussed in section 8.5.2.2), is shared by the PaC and the PAA. This session key is provided to the PaC as part of the EAP key exchange process, and the PAA can obtain the session key from the EAP server via the AAA infrastructure. PANA/Liberty SA establishment based on the EAP session key is required where no physical or link layer security is available (see section 4.2.3).

The purpose of a re-authentication exchange is to allow for efficient re-keying, using the existing PANA/Liberty security association, in situations where (depending on the security policy in force) full authentication is not required. Two types of re-authentication (or fast re-authentication) are supported by PANA/Liberty. The first type enters the chosen EAP method, i.e. the EAP-AKA fast re-authentication procedure (see section 5 of [20]), during the authentication and authorisation phase, and thus the initial discovery and handshake phase is omitted. The second type uses protected PANA/Liberty messages exchanged directly during the access phase, without entering the authentication and authorisation phase, i.e. the PANA re-authentication phase (see section 6.1.4.4).

## 9.7 Alternatives for PANA/Liberty Integration

Section 9.5 discussed just one way in which PANA could be integrated into Liberty. However, there are other possibilities. For instance, whilst we propose that in PANA/Liberty the PaC implements a LUA (as mentioned in section

9.5.1) following the Liberty-GAA ID-WSF SSO scenario (as described in section 9.5.2.4), there are other possible scenarios with distinct SSO message flows. For example, if the PaC is not Liberty-enabled, then the Liberty-GAA ID-FF with *AuthnResponse* transfer and the Liberty-GAA ID-FF with artifact transfer scenarios (as outlined in section 9.4.5) could be used. Moreover, it is possible to integrate PANA into Liberty without using the GAA framework.

In this section, we give a description of other possible ways in which PANA can be integrated into Liberty without making use of the GAA framework (as explained in section 9.7.1). In particular, a discussion of alternative schemes that may be used as the PANA inner authentication protocol instead of 3GPP AKA (as given in section 9.7.2) is provided.

### 9.7.1 PANA/Liberty without GAA Framework

As stated before, the PANA/Liberty scheme combines the Liberty Alliance architecture (see section 3.10) and the 3GPP GAA security mechanisms (described in section 3.5.4), with PANA (see Chapter 6). Because we include GAA in this scheme, we are limited to the two types of cellular authentication mechanisms supported by GAA, i.e. either a shared secret based on 3GPP AKA or digitally signed public key certificates (see section 3.5.4.2). However, it is possible to integrate PANA into Liberty without depending on the GAA architecture. Although this new scenario does not exploit the advantages of combining Liberty and GAA, it opens up the possibility of integrating PANA into Liberty using an alternative to 3GPP AKA as the PANA inner authentication protocol.

In order to combine the Liberty SSO advantages with PANA in this way, it is necessary to choose an inner authentication protocol to replace the 3GPP AKA, and incorporate it within an EAP method, which will be thus carried by PANA. Alternative schemes for this scenario are described immediately below.

### 9.7.2 Possibilities for PANA Inner Authentication

In this section, a number of alternative schemes that may be used as the PANA inner authentication protocol are discussed. As stated in section 3.7.5, PANA can carry any authentication mechanism that can be specified as an EAP method. In other words, the inner authentication protocols used by PANA must be initially encapsulated within EAP messages (given in section 3.4).

One alternative scheme that may be used here as the PANA inner protocol is an EAP method encapsulating the 3GPP2 CDMA2000 1x identification and authentication message exchanges (described in section 3.5.5); we might call this method EAP-CDMA. A potential advantage of using EAP-CDMA for this purpose is based on the fact that (as discussed in section 3.5.5.3) further releases of 3G CDMA2000 technologies add more security protocols, including the use of 128-bit privacy and authentication keys[8].

A second alternative scheme for this purpose consists of the EAP-PSK protocol, proposed by Bersani and Tschöfenig (see section 3.6.7) for authentication over insecure networks (e.g. IEEE 802.11 [85]). EAP-PSK is an EAP method for mutual authentication and session key derivation, which uses a 16-byte preshared key (PSK) as its long term credential. The PSK, which is used to derive two 16-byte subkeys called the authentication key and the key-derivation key, is assumed to be known only to the EAP peer and the EAP server. Therefore, EAP-PSK is a good candidate for use as the PANA inner authentication mechanism.

Another possibility involves using asymmetric methods, employing public/private key pairs, certificates, and PKIs (see section 2.1.3.3). In this case, a good candidate for use as a PANA inner authentication mechanism is the EAP-IKEv2 protocol, described by Tschöfenig, Kroeselberg, Ohba and Bersani

[8]Another potential advantage derives from the flexibility of EAP-CDMA, since (as discussed in section 8.4) the AKA protocol can also be used in a CDMA2000 (R)UIM for all releases of CDMA2000 following release C.

[175]. EAP-IKEv2 specifies a way of encapsulating the first phase of the IKEv2 protocol (see section 3.8.1), which supports both symmetric and asymmetric authentication, within EAP. The next chapter discusses this in more detail.

## 9.8 Conclusions

As previously discussed, authentication and key agreement are fundamental components of a secure procedure for heterogeneous network access supporting ubiquitous mobility. The main challenges addressed here include the investigation and development of unified, secure and convenient authentication mechanisms that can be used in access networks of a wide range of types.

In this chapter, we have proposed the new PANA/Liberty protocol, in order to provide an IP-compatible, lightweight, flexible and scalable method for authenticating a user to an access network, reusing the cellular network authentication infrastructure deployed in subscriber smart cards, and offering an open SSO standard service. The protocol is based on the PANA/UMTS scheme presented in Chapter 8, and incorporates the security techniques used in the UMTS mobile telecommunication system and the GAA infrastructure into the PANA authentication structure.

This scheme is complemented by the Liberty SSO service, which can be used to extend a PANA/UMTS initial authentication to all Liberty-enabled SPs, and create a network identity infrastructure supporting all network access devices. A description of other possible ways in which PANA can be integrated into Liberty (without the GAA framework) was provided, including a discussion of schemes that may be used as the PANA inner authentication protocol instead of 3GPP AKA.

The gains in performance arising from the two types of fast re-authentication,

the gains in security from the PANA/Liberty SA, and the gains in interoperability, flexibility and scalability by incorporating the Liberty Alliance framework and mechanisms of the GAA architecture, potentially make the PANA/Liberty proposal attractive to Liberty operators willing to offer their users heterogeneous Internet access in ubiquitous mobility networks.

This new Internet authentication scheme, designed to meet the established requirements (see Chapter 5), has been proposed here as a candidate for secure access procedure for heterogeneous network access supporting ubiquitous mobility (see section 1.1). In Chapter 11, the new scheme is submitted to a formal threat modelling process; it is also compared with the three further novel Internet entity authentication techniques proposed in Chapters 7, 8, and 10.

# Chapter 10

# PANA/IKEv2

## Contents



As explained in Chapter 7, this thesis proposes a series of new solutions for Internet remote access authentication, derived by adapting and reinforcing security techniques arising from a variety of different sources. The aim of this chapter is to present the fourth new authentication scheme, namely a means of combining the IKEv2 authentication mechanism (see section 3.8.1) with PANA (see section 3.7.5 and Chapter 6), which we call PANA/IKEv2.

## 10.1 Introduction

As described in section 7.1, in some ubiquitous mobility scenarios, IP based remote hosts that connect to the Internet via an access network will typically need to provide their credentials and be authenticated before being authorised to access the network. For such a process we need a flexible, strong, and scalable entity authentication infrastructure. In particular, the cryptography used in setting up such an infrastructure can be based on either secret key (or symmetric — see section 2.1.3.2) or public key (or asymmetric — see section 2.1.3.3) techniques. Whereas the former requires the involvement of the home network during the initial authentication process between a user and a visited network, the latter allows for architectures that avoid on-line involvement of the home network, since authentication may then be based on *public key certificates* (see section 2.1.3.3).

Nevertheless, asymmetric techniques typically require a Public Key Infrastructure to support key distribution, and use of this PKI may require on-line certificate status checking. While symmetric techniques are used almost exclusively in today's mobile networks, it seems likely that asymmetric techniques will gain greater importance in future ubiquitous mobility access networks because of their greater flexibility.

As previously discussed, the IETF PANA protocol (see Chapter 6) is intended to be a flexible and scalable generic network layer protocol, to be used to authenticate a user device requesting Internet remote access. In addition, the EAP-IKEv2 protocol, an EAP method currently specified as an IETF Internet draft [175][1], describes a way of encapsulating the first phase of the IKEv2 protocol (see section 3.8.1), which supports both symmetric and asymmetric au-

[1]The whole of this chapter is based on one particular draft of the EAP-IKEv2 specification [175]. Working on one particular draft has been necessary because it is a work in progress and changes relatively frequently. The latest version of this draft [176] was published as this thesis was being completed, and it would therefore be desirable to make any necessary changes in the schemes described in this chapter to reflect the changes to the EAP-IKEv2 text.

thentication, within EAP. Building on these two observations, we now present a new authentication scheme, combining the IKEv2 authentication and key exchange mechanism with EAP-IKEv2 and PANA, which we call PANA/IKEv2. This innovative proposal, previously described in [147], adapts the symmetric and asymmetric techniques used in the IKEv2 authentication mechanism to the PANA network remote access structure, in a solution designed to support ubiquitous client mobility for Internet access.

The main advantage of PANA/IKEv2 is that it does not define a new cryptographic protocol, but re-uses the well-analysed IKEv2 authentication exchanges within the EAP and PANA frameworks. As a result, it provides strong cryptographic properties as well as a high degree of flexibility. Of particular interest is the fact that PANA/IKEv2 provides an efficient 'shared secret key' based authentication method (see section 2.1.3.2). It also provides mutual authentication between the PaC and the PAA. This may be based either on symmetric methods using 'pre-shared keys' (see section 3.6.7), or on asymmetric methods, based on public/private key pairs, certificates, and PKIs (see section 2.1.3.3).

The purpose (section 10.2) and the components used in the assembly (section 10.3) of the novel PANA/IKEv2 scheme are first given. Second, the EAP-IKEv2 mechanism (section 10.4), an EAP method (see section 3.4) published in the 2006 Internet draft [175] and used as a component of the new PANA/IKEv2 technique, is explained. The framework of the proposed new PANA/IKEv2 protocol is then given (section 10.5). Next, two important features of PANA/IKEv2, namely the security association and the re-authentication procedure (section 10.6), are described. Finally, the conclusions of the chapter are given (section 10.7).

The main novel contribution of this chapter lies in sections 10.3, 10.5, and 10.6. Whilst the EAP-IKEv2 mechanism described in section 10.4 has been previously described (notably by Tschöfenig, Kroeselberg, Ohba and Bersani

[175]), the details of how it would operate when executed over PANA have not. This chapter does not contain a detailed security analysis of the new proposal — this issue is covered in Chapter 11.

## 10.2 PANA/IKEv2 Objective

Currently there is no standard protocol for performing network access authentication above the link layer. Instead, a number of ad hoc and often inadequate solutions (as described in section 4.1.5) are being used to overcome the problem (itself described in section 4.1), in a variety of distinct scenarios (outlined in section 4.2).

The objective of the PANA/IKEv2 protocol is thus to provide a network layer, IP compatible, lightweight, attack-resistant (e.g. with respect to MitM and DoS attacks — see section 3.2.3), and relatively flexible authentication method, that allows a remote client to be authenticated in a heterogeneous Internet access environment supporting ubiquitous mobility. This authentication method must meet a number of detailed security and implementation requirements, as specified in Chapter 5.

## 10.3 PANA/IKEv2 Protocol Hierarchy

In this section, an overview of the components used in the construction of the new PANA/IKEv2 authentication scheme is given. The first component, as previously discussed, is the IKEv2 mechanism. IKEv2 provides authentication and key exchange capabilities, and supports both symmetric and asymmetric cryptographic techniques. Section 3.8.1 gives an outline of the IKEv2 security features.

The second component used in the PANA/IKEv2 protocol assembly is EAP (see section 3.4). The EAP protocol, as previously discussed, supports a variety of authentication schemes, giving providers the advantage of using a single framework across multiple environments. Such flexibility seems likely to be important for heterogeneous network access supporting ubiquitous mobility. Since EAP is very flexible and can encapsulate arbitrary EAP methods, it is clearly a protocol that satisfies many of the requirements for a variety of authentication scenarios (see sections 4.2 and 5.1.3).

However, as previously described, EAP itself does not specify any authentication method. It is only a transport mechanism, allowing concrete authentication methods for EAP, such as public key based authentication schemes, to be defined separately. In fact, the EAP-IKEv2 protocol, an EAP method currently specified as an IETF Internet draft [175], specifies a way of encapsulating the security parameters used by the first phase of the IKEv2 mechanism within EAP, in order to provide mutual authentication and session key agreement.

Although EAP-IKEv2 re-uses a public key based protocol (i.e. IKEv2) in a flexible authentication framework (i.e EAP), using just EAP-IKEv2 for authentication is not a good choice. This is because it does not provide a complete authentication solution for ubiquitous client mobility for Internet access.

The effective use of EAP-IKEv2 in this latter environment requires the provision of a transport scheme for authentication data between a remote entity seeking access to a network and another entity located in the access network (see section 4.1.4). More specifically, a transport scheme independent of the access network type is needed, to transfer user authentication information to the access network and, optionally, to the AAA infrastructure (see section 3.9). Defining a network layer transport for EAP-IKEv2, such as the proposed tunnelled authentication solutions (see section 3.7), provides a cleaner answer to the problem.

In Chapter 6, we justified the selection of PANA, a UDP-based protocol (see section 6.1.1), as the tunnelled network layer transportation environment. We describe in this chapter how to use PANA to support the use of EAP-IKEv2 for Internet remote access authentication. As stated previously, PANA is also able to interact with Diameter EAP (see sections 3.7.5 and 3.9.3). Consequently, PANA is our choice for the third component in the construction of our proposed technique, which thus combines IKEv2 authentication with EAP-IKEv2 and PANA, into a scheme which we call PANA/IKEv2. A summary of the PANA/IKEv2 protocol hierarchy is shown in Figure 10.1.

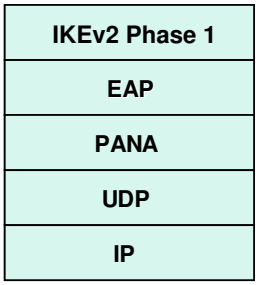


Figure 10.1: **PANA/IKEv2 protocol hierarchy**

## 10.4 An EAP Mechanism for Carrying IKEv2

The EAP-IKEv2 protocol [175] is an EAP mechanism (see section 3.4) for authentication and session key distribution that uses the IKEv2 protocol (see section 3.8.1). It offers the security benefits of IKEv2, which was defined for Internet key exchange, in all scenarios using EAP-based authentication, without establishing IPsec SAs (see section 3.6.5). IKEv2 provides authentication and key exchange capabilities, and supports both symmetric and asymmetric authentication within a single protocol. Such flexibility is likely to be important for an EAP method.

Figure 10.2 shows the EAP-IKEv2 message flow. In this figure, the name of each message is shown, followed by the contents of the message in round brackets. Square brackets are used to denote optional fields.

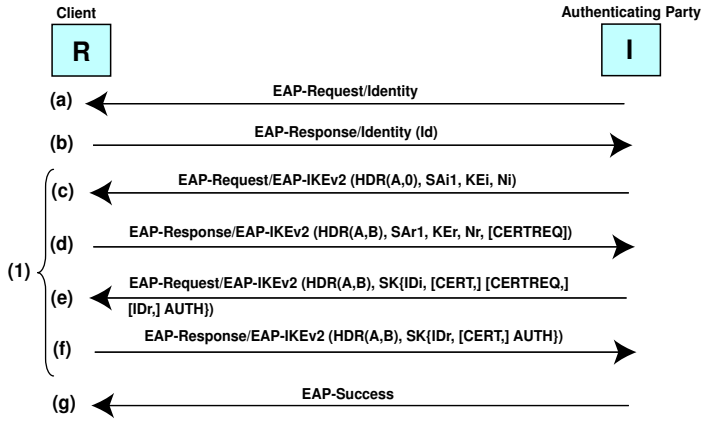


Figure 10.2: **EAP-IKEv2 message flow**

The EAP-IKEv2 message flow occurs between the Initiator ($I$) and the Responder ($R$). $R$ is also referred to here as the *Client* (acting on behalf of a user), whereas $I$ is referred to as the *Authenticating Party*. $I$ may be co-located with the *EAP server*, which is the network element that terminates the EAP protocol (see section 3.4). However, the EAP server is typically implemented on a separate AAA server in the user's home Internet AAA network, with whom $I$ communicates using an AAA protocol (see section 3.9).

The core EAP-IKEv2 exchange (1) consists of four messages (two round trips) only, where the first message pair $(c, d)$ negotiates cryptographic algorithms, exchanges nonces, and performs a Diffie-Hellman exchange (see section 2.1.3.3). The second message pair $(e, f)$ authenticates the previous messages, and exchanges the identities of $I$ and $R$, as well as public key certificates.

In EAP-IKEv2 full authentication, an identity request/response message pair $(a, b)$ is first exchanged. Next, $I$ sends $(c)$ an EAP-Request/EAP-IKEv2

message that contains an IKEv2 header ($HDR$[2]), a payload with the cryptographic suites supported by $I$ for the IKE-SA ($SA_{i1}$), a Diffie-Hellman value ($KE_i$), and a nonce ($N_i$). $R$ then responds with a message ($d$) that contains its choice of a cryptographic suite from among $I$'s offers ($SA_{r1}$), its value to complete the Diffie-Hellman key exchange ($KE_r$), and its nonce ($N_r$).

At this point, each party can generate the *SKEYSEED* value, from which the keying material for the IKE-SA is derived. The *SKEYSEED* value ($K_s$) is calculated by applying a negotiated IKEv2 pseudo random function (see section 3.8.1) to the concatenation of $N_i$ and $N_r$, using the Diffie-Hellman shared secret[3] ($g^{ir}$), as given in equation (10.1), where here, as throughout, $pf_K(X)$ denotes a pseudo random function $pf$ computed using the secret $K$ and data $X$ (see section 2.14 of [49]):

$$K_s = pf_{g^{ir}}(N_i|N_r). \tag{10.1}$$

The keying material derived from $K_s$ includes a temporary key called $K_d$. This is taken from the output of the pseudo random function $pf^*$, as shown in equation (10.2), where here, as throughout, ${pf^*}_K(X)$ denotes the pseudo random function $pf$ applied iteratively, as specified in section 2.13 of [49], using $K_s$ and the concatenation of $N_i$, $N_r$, and the SPIs chosen by $I$ and $R$, written as $S_i$ and $S_r$ below, as input (see section 2.14 of [49]):

$$K_d|\ldots = {pf^*}_{K_s}(N_i|N_r|S_i|S_r). \tag{10.2}$$

The temporary key $K_d$ is then used to create further EAP-IKEv2 keying

[2] *HDR* contains Security Parameter Indexes (SPIs), version numbers, and flags of various sorts. SPIs are values chosen by $I$ and $R$ to identify a unique IKE-SA (see section 3.8.1). *HDR(A,0)* means that $I$ assigned the SPI *'A'* and $R$ has not yet chosen its SPI, while *HDR(A,B)* means that $I$ chose the SPI *'A'* and $R$ chose the SPI *'B'*.

[3] $g$ denotes a Diffie-Hellman generator value agreed between the parties, which is used in conjunction with a prime $P$. The pair of exponents ($i$, $r$) denote random values chosen, respectively, by $I$ and $R$. The Diffie-Hellman values ($KE_i, KE_r$) exchanged via the message pair ($c, d$) are calculated as follows: $KE_i = g^i \bmod P$ and $KE_r = g^r \bmod P$. The value $(KE_i)^r$ is then calculated by $R$, and $(KE_r)^i$ is computed by $I$; both calculations yield the same value, i.e. $g^{ir} \bmod P$.

material, called *KEYMAT*. Since the required length of *KEYMAT* is greater than the length of the output of the pseudo random function $pf$, this function is also used iteratively here (see section 9 of [175]). That is, $K_m$, which denotes *KEYMAT*, is derived by iteratively applying $pf$, to the concatenation of $N_i$ and $N_r$, together with $K_d$, as shown in equation (10.3) (see section 2.17 of [49]):

$$K_m = pf^*{}_{K_d}(N_i|N_r). \tag{10.3}$$

The keying material *KEYMAT* is then exported as part of the EAP keying framework (see section 3.4) to derive further keys, including *MSK*, used to encrypt the traffic between the client and the network, and *EMSK*, used to derive keys for multiple applications (see section 3.6.6).

All but the IKEv2 headers of the messages that follow are encrypted and integrity protected, and this is indicated in Figure 10.2 using the notation $SK\{\dots\}$. The recipients must verify that all signatures and *MACs* (see section 2.1.3.2) are computed correctly, and that the identities $ID_i$ and $ID_r$ correspond to the keys used to generate the Authentication ($AUTH$) payload (see section 1.2 of [49]).

$I$ sends back ($e$) a message to assert its identity ($ID_i$), to prove knowledge of the secret corresponding to $ID_i$, and to integrity protect the contents of the first message using the $AUTH$ payload (see section 2.15 of [49]). It may also send its certificate ($CERT$) and a list of its 'trust anchors', i.e. the names of the CAs (see section 2.1.3.3) whose public keys it trusts ($CERTREQ$); the optional $ID_r$ payload enables $I$ to specify which of $R$'s identities it wants to talk to (e.g. when $R$ is hosting multiple users at the same IP address). $R$ then asserts its identity ($ID_r$), optionally sends one or more certificates ($CERT$), and authenticates its identity with $AUTH$ ($f$). The message flow finishes with an EAP-Success message ($g$).

Man-in-the-Middle attacks that apply to certain tunnelled authentication protocols (see section 3.2.3) are not applicable to EAP-IKEv2, as the extended authentication feature of IKEv2 is not supported by EAP-IKEv2 (see section 13.2 of [175]). Hence, the cryptographic binding requirement, described in section 3.2.3, is not applicable.

## 10.5 PANA/IKEv2 Framework

In this section, the authentication framework for the new PANA/IKEv2 scheme is described. The entities involved in the PANA/IKEv2 method are first given (section 10.5.1). After that, the PANA/IKEv2 authentication scheme is explained (section 10.5.2).

### 10.5.1 PANA/IKEv2 Entities

The PANA/IKEv2 mechanism proposed here involves three functional entities, namely the *PaC* (also referred to here as the *client*, *user*, *customer* or *subscriber*), the *PAA* (or *authenticating party*) and the *EAP server*. The *PaC* is associated with a network device and a set of credentials; these credentials are used to prove the PaC identity for the purposes of network access.

The PAA authenticates the IKEv2 credentials provided by the PaC and grants network access. In the context of this chapter, the user's EAP server is assumed to be implemented on the AAA server (see section 3.9). The PAA is thus an AAA client that communicates with the user's EAP server through an AAA protocol supporting EAP (i.e. Diameter EAP, described in section 3.9.3) and key wrap (see section 7.5). PANA/IKEv2 also involves a further entity, namely the EP (see section 6.1.2), which applies per-packet enforcement policies (i.e. filters) to the traffic of the PaC's devices.

### 10.5.2 PANA/IKEv2 Authentication Scheme

The aim of this section is to give a detailed description of the PANA/IKEv2 scheme. Firstly we identify the distinct phases of a PANA/IKEv2 session, and briefly describe them (section 10.5.2.1). Secondly, a complete description of the PANA/IKEv2 message exchange is provided (section 10.5.2.2). We then summarise the calculation of the PANA/IKEv2-based MAC used during that exchange (section 10.5.2.3).

Figure 10.3 shows the PANA/IKEv2 authentication procedure, which is further described below. In this figure, the name of each message is shown, followed by the contents of the message in round brackets. Square brackets are used to denote optional fields.

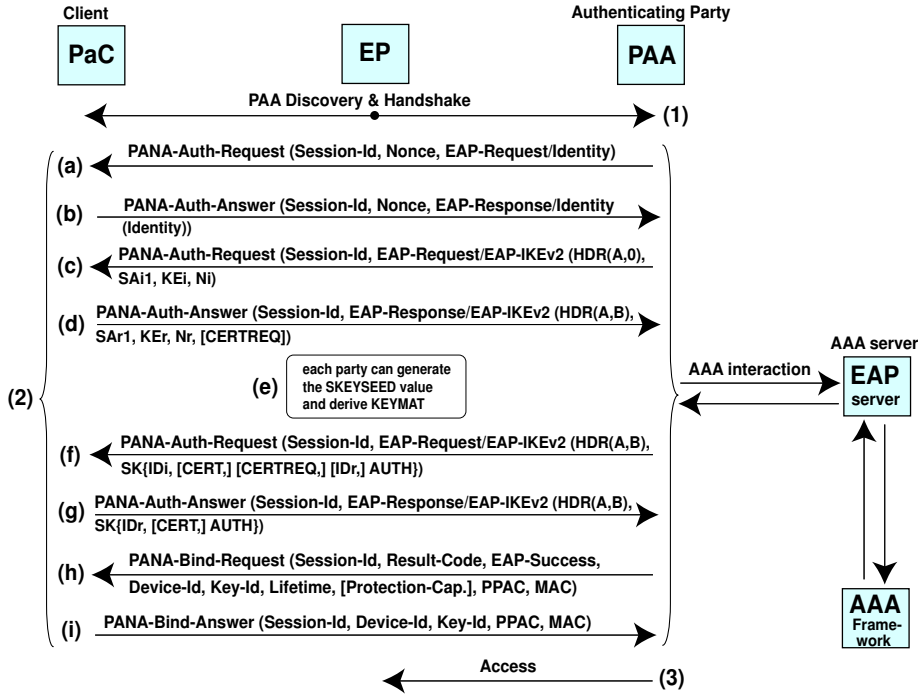


Figure 10.3: **PANA/IKEv2s full authentication procedure**

#### 10.5.2.1 PANA/IKEv2 Phases

The PANA/IKEv2 authentication procedure has three main phases: (1) Discovery and Handshake, (2) Authentication and Authorisation, and (3) Access. In the *Discovery* phase, an IP address for the PAA is identified, and a PANA/IKEv2 session is established between the PaC and the PAA, following the PANA model (see section 6.1.4.1). After this phase is complete, a session identifier (Session-Id — see section 6.1.2) is allocated by the PAA and included in all further messages; this identifier is freed when the PANA/IKEv2 session terminates.

In the *Authentication* phase, the main focus of this section and further explained below, EAP-IKEv2 messages encapsulated in PANA/IKEv2 messages are exchanged between the PaC and the PAA. In this phase, EAP-IKEv2 Request payloads are carried in PANA-Auth-Requests. Moreover, taking advantage of an optimisation discussed in section 6.1.4.2 and adopted by PANA/IKEv2, in the context of this chapter a PANA-Auth-Answer will include an EAP-IKEv2 Response payload.

As previously discussed, the PAA communicates with the EAP server using the AAA Diameter EAP protocol (see section 3.9.3). Hence, EAP-IKEv2 packets encapsulated in Diameter-EAP messages are exchanged between the PAA, which is thus the *Diameter client*, and the EAP server, which is implemented on the *Diameter server*, following the process for using EAP in Diameter given in Figure 3.14. The PANA-Diameter message mapping, given in section 3.7.5, is also adopted here to allow the transport of EAP-IKEv2 payloads between the PANA framework and the AAA Diameter infrastructure (see section 3.9.2). At the end of the Authentication phase, a PANA SA is established, including the provision of a shared secret EAP-IKEv2 session key *MSK* (see section 10.4); we call this the PANA/IKEv2 SA.

During the *Access* phase, a separate protocol is used between the PAA and the EP to manage PaC network access control. After this phase, the established PANA/IKEv2 session and the PANA/IKEv2 SA are deleted, following the PANA draft standard (see section 6.1.4.5).

#### 10.5.2.2 PANA/IKEv2 Message Exchange

During the *Authentication* phase, the first PANA-Auth-Request message (*a*) issued by the PAA encapsulates a PANA-based *Nonce*, i.e. a randomly chosen value (see section 6.1.3), used in further PANA/IKEv2 cryptographic key computations, and an EAP-Request/Identity payload, requesting the PaC to identify itself. The PaC responds (*b*) with a PANA-Auth-Answer, which also carries a PANA-based Nonce value, and an EAP-Response/Identity payload including the user's identifier *Identity*.

The PAA then issues a Diameter-EAP-Request to the EAP server via an AAA interaction (see section 3.9.3), including the EAP-Response/Identity packet in an EAP-Payload AVP, and the user's identifier value in a Diameter User-Name AVP (see section 3.9.2). The EAP server responds with a Diameter-EAP-Answer in a multi-round exchange, with a Result-Code AVP set to DIAMETER_MULTI_ROUND_AUTH, signifying that a subsequent request is expected. This exchange also includes an EAP-Request/EAP-IKEv2 packet, which contains *HDR*, $SA_{i1}$, $KE_i$, and also $N_i$, the random number chosen by the EAP server, encapsulated in an EAP-Payload AVP. This EAP payload is now sent to the PaC in a PANA-Auth-Request (*c*).

The next PANA-Auth-Answer message (*d*) issued by the PaC includes the EAP-Response/EAP-IKEv2 packet, containing $SA_{r1}$, $KE_r$, the random number $N_r$ chosen by the PaC, and *CERTREQ*, an optional list of the PaC trust anchors. The PAA then sends a Diameter-EAP-Request to the EAP server, encapsulating this latter EAP response packet in an EAP-Payload AVP.

At this point, as specified in section 10.4, each party can derive the keying material ($e$) for the IKE-SA, starting with the *SKEYSEED* value (see equation 10.1). From *SKEYSEED*, the EAP server then derives a variety of IKEv2 secret keys, including the temporary key $K_d$ (see equation 10.2), from which further EAP-IKEv2 keying material, called *KEYMAT*, is created (see equation 10.3). The *KEYMAT* is then exported as part of the EAP keying framework to derive further keys, including *MSK*, which is used by PANA/IKEv2 as the AAA-Key (see section 6.1.2). All but the *HDR* fields of the EAP payloads that follow are encrypted and integrity protected.

The EAP server then sends back to the PAA a Diameter-EAP-Answer in a multi-round exchange. This exchange includes an EAP-Request/EAP-IKEv2 packet with its identity $ID_i$, an $AUTH$ value, and the following optional payloads: *CERT*[4], *CERTREQ*, and $ID_r$, which enables the EAP server to specify which of PaC's identities it wants to talk to. As previously, in Figure 10.3 the notation $SK\{\dots\}$ indicates that the content between brackets is encrypted and integrity protected.

The next PANA/IKEv2 message ($f$) issued by the PAA encapsulates the received EAP-Request/EAP-IKEv2 payload detailed above. On receipt of this message, the PaC then sends a PANA-Auth-Answer message carrying an EAP-Response/EAP-IKEv2 packet to assert its identity ($ID_r$); this message also includes $AUTH$ and optionally *CERT* ($g$). After receiving this latter EAP response packet from the PAA via an AAA Diameter-EAP-Request, and if all the checks succeed, the EAP server then sends back an AAA Diameter-EAP-Answer, which includes a Result-Code AVP set to DIAMETER_SUCCESS. This message also includes an EAP-Payload AVP with a code field set to Success, which indicates that the authentication was successful. This EAP-Success packet carries derived AAA keying material, including an AAA-Key.

[4]If any CERT payloads are included, the first certificate provided will contain the public key required to verify the AUTH field. If symmetric cryptographic techniques are being used, the *CERT* and *CERTREQ* payloads are not required (see [49]).

The PAA encapsulates the PANA result code, the EAP-Success packet, and the PANA/IKEv2 *session lifetime* (see section 6.1.2) in a PANA-Bind-Request message sent to the PaC ($h$), and receives back an acknowledgement through a PANA-Bind-Answer ($i$). On receipt of this message, the PAA issues a Diameter Accounting-Request (Start) to the EAP server, which indicates the start of the session, following the PANA-Diameter message mapping given in section 3.7.5.

PANA-Bind messages are protected by a PANA/IKEv2-based MAC AVP, calculated as described in the next section, and carry a Key-Id AVP (see section 6.1.3); this latter AVP contains an AAA-Key identifier that is assigned by the PAA and is unique within the PANA/IKEv2 session.

Finally, PANA-Bind messages may also optionally contain a Protection-Capability AVP (see section 6.1.3), which is sent from the PAA to indicate that link-layer or network-layer encryption should be initiated after completion of PANA/IKEv2. PANA-Bind messages are also used for binding the device identifiers of the PaC and the PAA to the PANA/IKEv2 SA established at the end of the authentication phase; this is achieved using a Device-Id AVP. PANA-Bind messages with a Result-Code AVP indicating successful authentication also include PPAC AVPs (see section 6.1.3), which help the PAA/PaC to negotiate the available/chosen IP address configuration method.

#### 10.5.2.3 PANA/IKEv2-based MAC

The PANA/IKEv2-based MAC ($M_{P_I}$) is calculated using HMAC-SHA-1, as shown in equation (10.4), where $P_I$ denotes the PANA/IKEv2 packet, and $K_p$ denotes the PANA_MAC_Key (see section 6.1.5):

$$M_{P_I} = f_{K_p}(P_I). \tag{10.4}$$

The EAP-IKEv2 shared secret *MSK*, which is normally used to establish a PANA/IKEv2 SA, is adopted as the AAA-Key, which is then used as input to generate a distinct key $K_p$. However, as previously discussed, two AAA-Keys may be produced as a result of separate NAP and ISP authentication processes (see section 6.1.4.2). In this case, $K_{aaa}$, which denotes the AAA-Key used in the $K_p$ generation procedure, results from the concatenation of the two keys, as given in equation 7.8.

The PANA_MAC_Key $K_p$ is calculated by applying HMAC-SHA-1, using the key $K_{aaa}$ to the concatenation of the PANA-based Nonces $N_{pac}$ and $N_{paa}$, sent respectively by the PaC ($b$) and the PAA ($a$), and the PANA/IKEv2 Session-Id AVP value ($S_{id}$), as given in equation 7.9.

## 10.6 PANA/IKEv2 SA and Re-Authentication

Two important features of PANA/IKEv2, namely the security association and the re-authentication procedure, are now described.

As detailed in the previous section, the PANA/IKEv2 method generates the keying material *KEYMAT*. This keying material is used within the IKE-SA for protection of EAP-IKEv2 payloads (e.g. in the *AUTH* exchanges — see section 10.4). It is also used to derive additional session keys (i.e. the *MSK* and the *EMSK*) that are exported as part of the EAP keying framework (see section 3.6.6). Once the PANA/IKEv2 scheme has completed, these session keys are shared by the PaC and the PAA. The session keys are provided to the PaC as part of the EAP key exchange process, and the PAA can obtain the session keys from the EAP server via the AAA infrastructure. PANA/IKEv2 SA establishment based on these EAP session keys is required where no physical or link layer security is available (see section 4.2.3).

The purpose of a re-authentication exchange is to allow for efficient re-keying, using the existing PANA/IKEv2 security association, in situations where (depending on the security policy in force) full authentication is not required. Two types of re-authentication (or fast reconnection) are supported by PANA/IKEv2. The first type enters the chosen EAP method, i.e. the EAP-IKEv2 fast reconnection process (see section 11 of [175]), during the authentication and authorisation phase, and in this case the initial discovery and handshake phase is omitted. The second type uses protected PANA messages exchanged directly during the access phase, without entering the authentication and authorisation phase, i.e. the PANA re-authentication phase (see section 6.1.4.4).

## 10.7 Conclusions

As previously discussed, authentication and key agreement are fundamental components of a secure procedure for heterogeneous network access supporting ubiquitous mobility. The main challenges addressed here include the investigation and development of unified, secure and convenient authentication mechanisms that can be used in access networks of a wide range of types.

In this chapter, we have proposed the new PANA/IKEv2 protocol, in order to provide an IP-compatible, flexible and scalable method for authenticating a user to an access network using either symmetric or asymmetric techniques. The protocol is based on PANA, a network-layer access authentication protocol carrier, which communicates, via EAP, with an AAA infrastructure. PANA/IKEv2 uses EAP-IKEv2, which allows use of the IKEv2 infrastructure defined for Internet key exchange in any scenario using EAP-based authentication.

The gains in performance arising from the two types of fast reconnection, the increase in flexibility provided by the public key based authentication option, and the gains in security given by the PANA/IKEv2 SA, potentially make the

PANA/IKEv2 proposal attractive to all operators willing to offer their users heterogeneous Internet access in ubiquitous mobility networks.

This new Internet authentication scheme, designed to meet the requirements established in Chapter 5, is proposed here as a candidate for secure access procedure for heterogeneous network access supporting ubiquitous mobility (see section 1.1). In Chapter 11, the new scheme is submitted to a formal threat modelling process; it is also compared with the three further novel Internet entity authentication techniques proposed in Chapters 7, 8, and 9.

# Chapter 11

# Threat Modelling & Evaluation

## Contents

---

In this chapter we give a formal threat model, and use this model to conduct a comparative analysis of the four new Internet authentication techniques proposed in this thesis. The primary goal of this chapter is to discover which of them is the most secure, lightweight, flexible and scalable method for allowing a client to be authenticated in a heterogeneous Internet access environment supporting ubiquitous mobility.

## 11.1 Introduction

As explained in section 1.2, this thesis proposes, evaluates and compares new entity authentication protocols for Internet remote access. In this chapter we give a formal threat model, and use this model to conduct a comparative analysis of the four authentication techniques proposed in Chapters 7, 8, 9, and 10. These new techniques are designed to meet the security and implementation services and properties established in Chapter 5. These security requirements are used in conjunction with the Internet authentication problem domain established in Chapter 4 to define and limit the scope of this thesis.

Firstly, the four new authentication schemes are submitted to a formal threat modelling process (sections 11.2 to 11.6). Secondly, we make a comparative analysis of the protocols, to determine which of these techniques is the most secure, lightweight, flexible and scalable authentication method that allows a client to be authenticated in a heterogeneous Internet access environment (section 11.7). Finally, the conclusions of the chapter are given (section 11.8).

## 11.2 Threat Modelling

As stated in section 1.2, the security analysis of the proposed authentication protocols is performed using the threat modelling process described in Chapter 4 of Howard and LeBlanc [81, p69-124]. According to Howard and LeBlanc, 'a *threat* to a system is a potential event that will have an unwelcome consequence if it becomes an attack. A *vulnerability* is a weakness in a system, such as a coding bug or a design flaw. An *attack* occurs when an attacker has a *motive*, or reason to attack, and takes advantage of a vulnerability to threaten an *asset*' [81, p87]. An asset is also referred to as a *threat target*. A *threat model* is thus a security-based analysis that can be used to determine the highest level security

risks posed to an application, and how attacks can manifest themselves [81, p69].

The goal of using this security-based analysis here is to determine which threats to the new authentication techniques require mitigation and how to mitigate them, reducing via a formal process of threat modelling the overall risk to the protocols to an acceptable level. As discussed in [81], it is cheaper to find a security flaw in a protocol during the design stage and remedy the solution before coding starts. Figure 11.1 shows the threat modelling process taken from Howard and LeBlanc [81, p72], the steps in which are further described in the following sections.

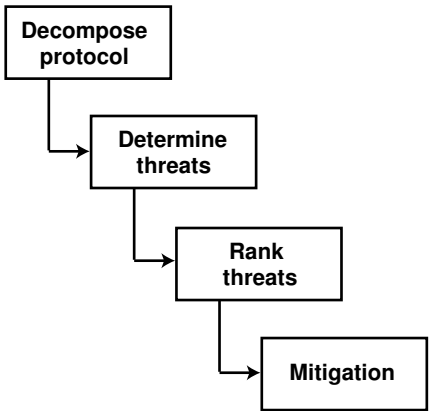


Figure 11.1: **Process of threat modelling**

## 11.3 Formally Decomposing the Protocols

As stated by Howard and LeBlanc [81, p74], the use of formal decomposition methods, such as data flow diagrams (DFDs), is a critical component prior to performing the threat analysis process for a protocol design. The leading principle for DFDs is that an application or a system can be decomposed into subsystems, and subsystems can be decomposed into still lower-level subsystems.

Figure 11.2 shows the key data flow diagram symbols used in this chapter.

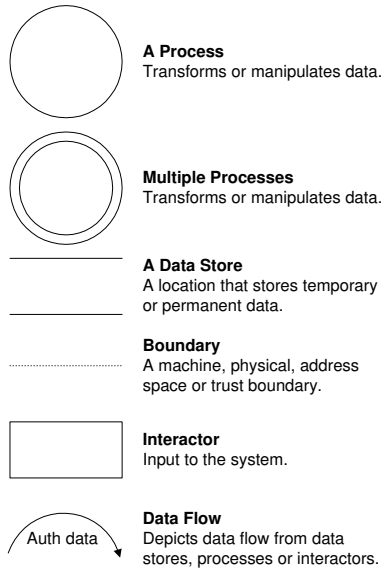


Figure 11.2: **Basic data flow diagram symbols**

The first phase of decomposition makes use of a high-level *context data flow* diagram, also referred to as level-0 (zero) DFD, which determines the scope of the authentication techniques being analysed, and helps us to understand the boundaries between trusted and untrusted components (see section 11.3.1). Once this phase is complete, we focus on greater protocol detail lower levels using *level-1* and *level-2* diagrams (see section 11.3.2). We discuss these decomposition phases in more detail in the following sections.

### 11.3.1 Context Data Flow Diagrams

The new authentication techniques proposed in Chapters 7, 8, 9, and 10 follow the *general entity authentication model*[1] discussed in section 2.3, and have their scope limited by the problem domain established in section 4.1. They all use

[1]This model states that, to meet the goals of an authentication protocol, the entities generate and exchange standardised messages.

the PANA protocol as a common target transport environment, as justified in Chapter 6. However, they make use of four different PANA inner authentication protocols, for each of which, with the exception of the solutions proposed in Chapters 8 and 9, the security mechanism is distinct.

This section defines the scope of the proposed authentication techniques being analysed and the boundaries between trusted and untrusted components by means of *context diagrams*. As stated by Howard and LeBlanc [81, p75], 'a context diagram has only one process and usually no data store'. A high-level context data flow diagram is presented in Figures 11.3, 11.4, 11.5, and 11.6 for each of the four entity authentication proposals.

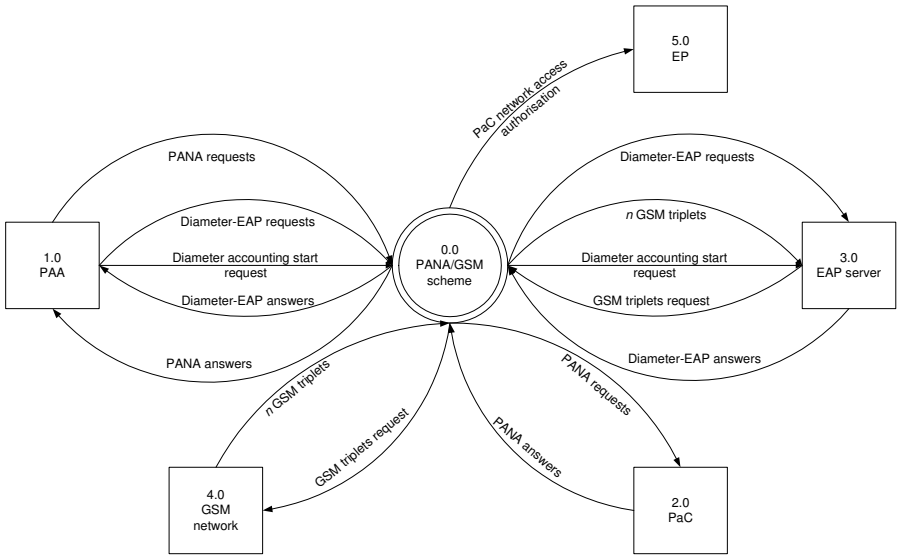


Figure 11.3: **PANA/GSM context diagram**

### 11.3.2 Level-1 and Level-2 Diagrams

Each of the context diagrams presented in Figures 11.3, 11.4, 11.5, and 11.6 comprises one or more processes, and needs to be decomposed accordingly. Figures 11.7, 11.8, 11.9, and 11.10 show the corresponding *level-1* data flow diagrams,

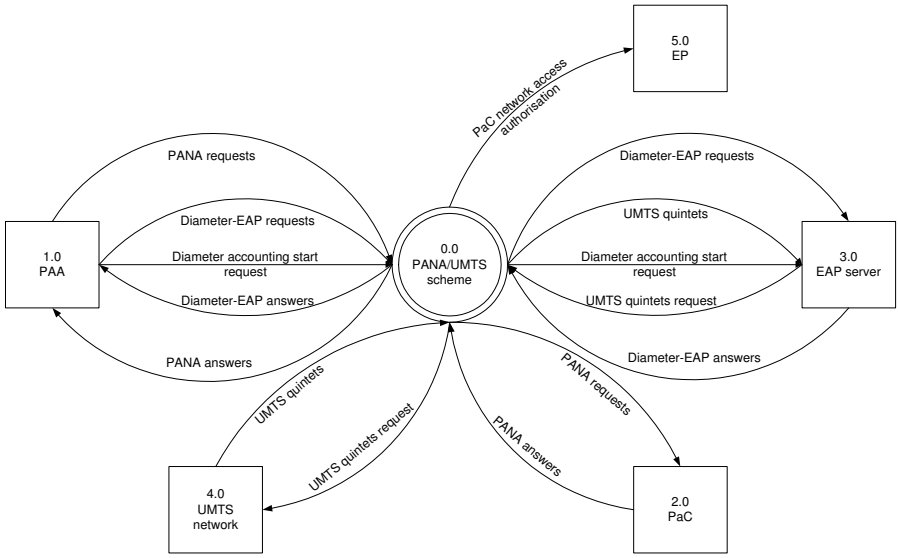


Figure 11.4: **PANA/UMTS context diagram**

which include depictions of general processes and data stores from the context of use of each of the four entity authentication proposals.

At this phase of the DFD decomposition method, it is important to define and limit the main focus of the formal threat model. Indeed, the reason for using DFDs in this security analysis is not to determine how everything works (e.g. as occurs in a typical application design), but rather to go sufficiently deep to achieve an understanding of the composition of the proposed authentication protocols in order to determine the threats that apply.

In Chapter 5, we established that this thesis focuses mainly on authentication and key establishment processes that carry parameters[2] between the client and the access network, and not on any subsequent uses of the authenticated channel and/or keys that may have been established (see section 5.1.1). In particular, it is important to note that the entity authentication schemes proposed in Chapters 8 and 9 make use of an identical underlying security mechanism[3].

[2]As stated in section 5.1, these parameters are needed to police the traffic flow through enforcement points.

[3]As described in section 9.3, part of the PANA/Liberty authentication solution incorpo-

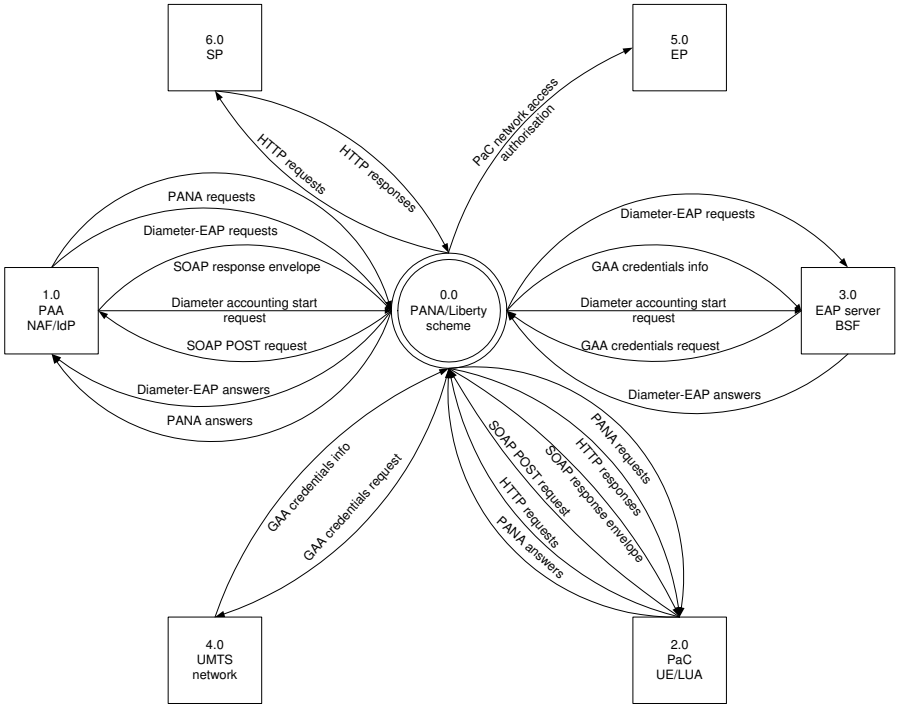


Figure 11.5: **PANA/Liberty context diagram**

In addition, the AAA interaction between the access network and the back-end authentication entities is outside the main scope of this thesis (see section 5.1.8). Also, in the proposed authentication techniques a separate protocol is used between the PAA and the EP to manage PaC network access control (as explained in sections 7.5.2.1, 8.5.2.1, 9.5.2.3, and 10.5.2.1).

Building on the above observations, and after analysing the level-1 DFDs of the four proposed protocols, we have deduced that the main focus of this formal threat model needs to be directed towards the data flows exchanged between the remote client (i.e. the 2.0 PaC interactor) and the visited access network (i.e. the 1.0 PAA interactor) boundaries, through the PANA/xxx[4] underlying authentication process, as shown in Figures 11.7, 11.8, 11.9, and 11.10.

rates the security components used in the PANA/UMTS protocol assembly.

[4]Where 'xxx' denotes GSM, UMTS, Liberty or IKEv2.

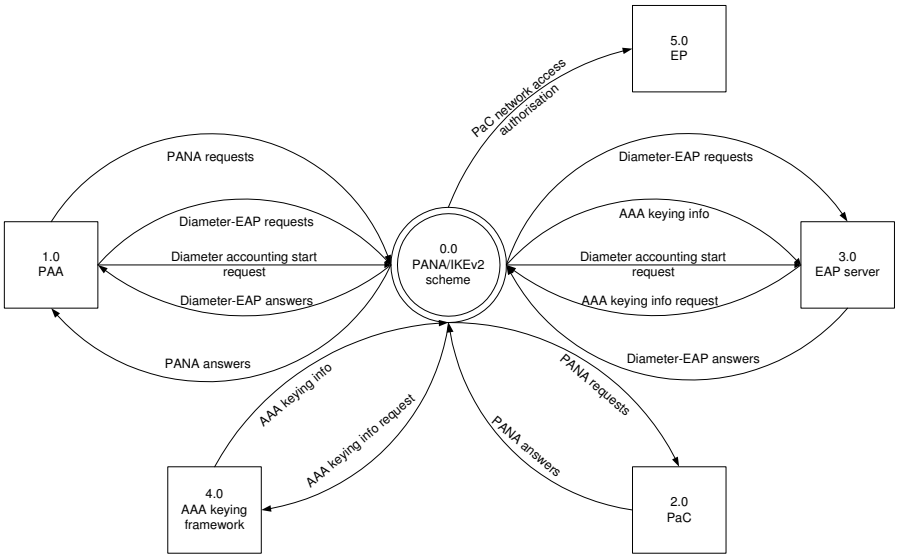


Figure 11.6: **PANA/IKEv2 context diagram**

The analysis of the level-1 DFDs has enabled us to decide on the focus of our analysis, as stated above. This is because the selected data flows incorporate the PaC-PAA exchanges of the PANA protocol which, as stated in Chapter 6, is the common target transport environment for the new authentication schemes. Moreover, this decision allows us to concentrate our threat analysis on the underlying client-to-network access authentication component of each of the proposed overall secure network access frameworks (see section 6.1). Hence, this choice enables us to avoid wasting time on threats that are outside the scope and beyond the control of the new proposals.

Having produced the level-1 DFDs, the next step should be to produce more detailed level-2 (or *child*) DFDs for each of the underlying PaC-PAA authentication exchanges. However, the detailed data flows for the PANA/GSM authentication process have already been shown in Figure 7.3; similarly, the flows for PANA/UMTS and PANA Liberty are given in Figure 8.2, and the flows for PANA/IKEv2 in Figure 10.3. As a result, we do not give the diagrams again here, but base our analysis on the diagrams given in previous chapters.

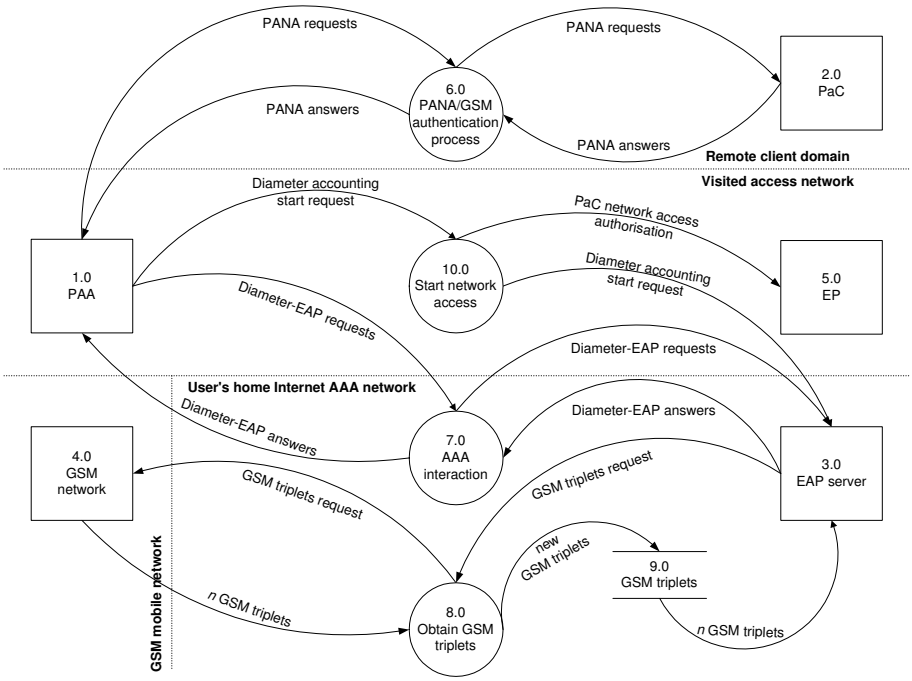


Figure 11.7: **PANA/GSM level-1 diagram**

## 11.4 Determining the Threats to the Protocols

According to Howard and LeBlanc [81, p83], after the decomposition process performed in the previous section, the next step is to take the identified *assets* and treat them as the *threat targets* in the threat model, investigating the components of the protocols and how data flows between them.

In this section, we perform two important parts of the threat analysis process for authentication protocols, namely analysing the threat categories (section 11.4.1) and the threat trees (section 11.4.2).

### 11.4.1 Threat Categories and STRIDE

Determining the threats to the proposed authentication techniques involves dividing the threats into well-defined *threat categories*. In this case, as suggested

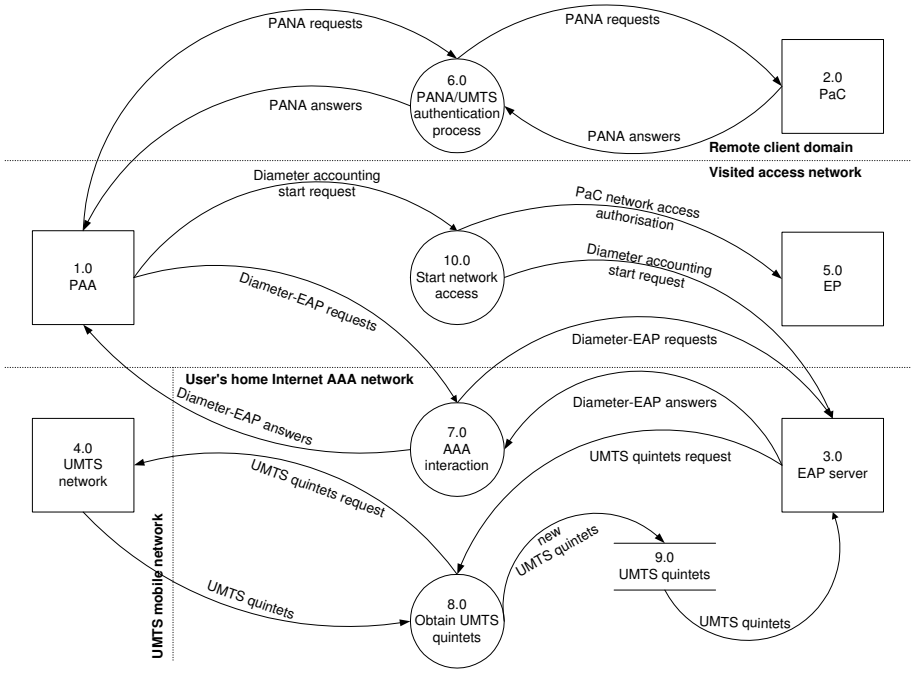


Figure 11.8: **PANA/UMTS level-1 diagram**

in [81, p83], we use the concept of *STRIDE*, an acronym derived from the six threat categories described below:

**Spoofing Identity** Spoofing threats allow an attacker to pose as another user or allow a rogue server to pose as a valid server. One example of user identity spoofing would involve illegally accessing and then using another user's authentication *credentials* (see section 2.2.2). Examples of server spoofing include 'DNS spoofing' and 'DNS cache poisoning' [81, p84]. *Entity authentication* services (as given in section 2.1.1.2) can be used to prevent the realisation of threats in this category.

**Tampering with Data** Data tampering involves malicious modification of data, e.g. unauthorised changes made in a database, or alteration of data as it flows between two machines over the Internet. *Data integrity* services (detailed

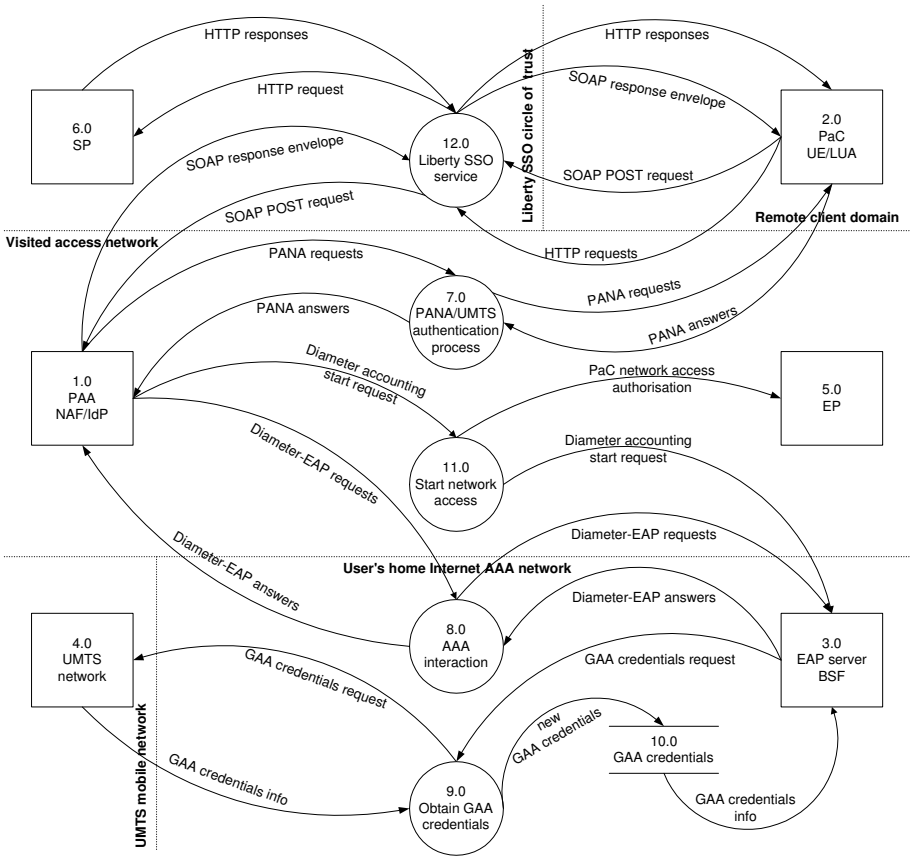


Figure 11.9: **PANA/Liberty level-1 diagram**

in section 2.1.1.3) can be used to mitigate such threats.

**Repudiation** Repudiation threats arise from users who deny performing an action which they have in fact carried out. An example of repudiation is a user performing an illegal operation in a system and, after that, denying her action. A *non-repudiation* service (see section 2.1.1.4) can provide a system with the ability to counter repudiation threats.

**Information Disclosure** Information disclosure threats involve the exposure of information to individuals who are not supposed to have access to it. Examples of such a threat include a user reading a file to which he/she has not

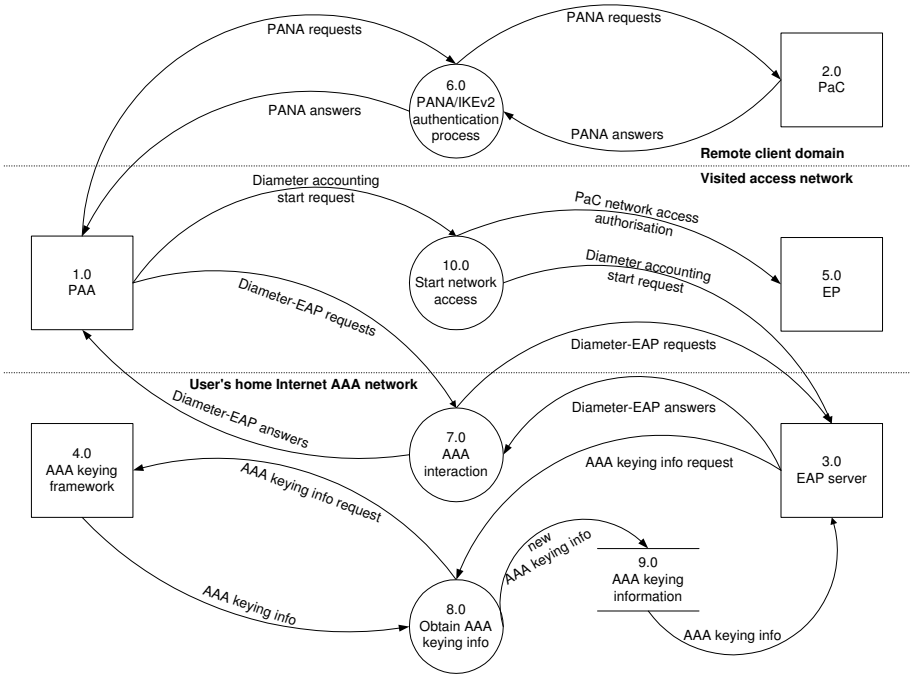


Figure 11.10: **PANA/IKEv2 level-1 diagram**

been granted access, and an eavesdropper reading data sent between two machines. *Confidentiality services* (explained in section 2.1.1.1) can protect against information being disclosed to entities not authorised to have that information.

**Denial-of-Service** DoS attacks deny service to valid users, e.g. by making a web server temporarily unavailable or unusable[5]. *Availability* services (described in section 2.1.1.6) can protect against certain types of DoS threats by improving system availability and reliability, helping to ensure that computer system assets are available to authorised parties when needed.

**Elevation of Privilege** In this type of threat, an unprivileged user gains privileged access to resources (e.g. a computing resource, communications re-

[5]Real-life examples include various Distributed-Denial-of-Service attacks (DDoS) that can be launched using publicly available attack tools, such as 'trinoo' and 'stacheldraht' (German for 'barbed wire') [81, p85].

source, or information resource). The effects of such an attack could include realisation of most if not all of the other types of threat. Elevation of privilege threats include those situations in which an attacker has effectively penetrated all system defences and become part of the trusted system itself. *Access control* services (see section 2.1.1.5) can protect against unauthorised access, helping to ensure that access to protected resources is controlled.

It is worth observing that threats in the six categories are often closely interrelated, and that realisation of a threat in one category can enable the realisation of a threat in a different category. For instance, information disclosure threats can lead to spoofing identity threats, if the user's credentials are not protected [81, p86].

Having examined the concept of STRIDE, we next attempt to determine the threats to the new authentication techniques by applying the STRIDE classification to *threat trees*.

### 11.4.2 Threat Trees

We first briefly describe the basic concepts underlying threat trees (section 11.4.2.1). We then examine threat trees for the proposed authentication techniques (sections 11.4.2.2 to 11.4.2.4).

#### 11.4.2.1 Introduction to Threat Trees

As stated in [81, p87], once a potential threat to a protocol component has been recognised, we can determine how that threat could manifest itself by using *threat trees*. The core idea behind this analysis is that the proposed authentication techniques are composed of threat targets identified in the decomposition process, and these targets could have vulnerabilities that compromise the sys-

tems when successfully attacked.

A threat tree describes a decision-making process used by an attacker in order to find a way to compromise a protocol component. The root node in the threat tree (represented graphically by the 'top box' in the threat tree diagram) corresponds to the ultimate threat and the STRIDE threat category to which it belongs. This root node is then linked to nodes representing possible means of realising the threat, where these subsidiary nodes are again represented as boxes in the threat tree diagram. We use dotted lines to link the boxes to indicate the least likely attack points, and solid lines for the most likely. Also, we place circles below the least likely nodes in the tree, indicating how the threat has been mitigated[6].

#### 11.4.2.2 PANA/GSM Threat Trees

We first consider the security threats to the PANA/GSM protocol (described in detail in Chapter 7), in order to create the corresponding threat trees.

**PAA Spoofing and Triplet Exposure** PANA/GSM provides mutual authentication via the EAP-SIM mechanisms. The PaC believes that the PAA is authentic because the network is able to calculate a correct AT_MAC value from the *RAND* challenges in the challenge request. The PAA believes that the PaC is genuine because the MAC computed from the *SRES* response values is correct. Moreover, PANA/GSM provides the means to validate a received EAP packet through its PANA message validity check scheme.

In order to be able to calculate a correct AT_MAC, as required to successfully impersonate a valid PAA to the PaC, it suffices to know the *RAND* and $K_c$ values from $n$ GSM triplets for the subscriber. Given physical access to the

[6] As discussed in [81, p91], these *mitigation circles* should be added later, after the threat modelling process.

subscriber's SIM card, it is easy to obtain any number of GSM triplets. Triplets can also be obtained by mounting an attack on the PaC platform via a virus or other malicious software. The PaC thus needs to be protected against triplet querying attacks by malicious software. Indeed, as discussed in section 7.5.2.2, 'the PaC is able to verify that the EAP-SIM message is *fresh* (i.e. not a replay; see section 2.2.5) and that the sender possesses valid GSM triplets for the user'.

In addition, if the same SIM credentials are also used for GSM traffic, the triplets could be revealed in the GSM network. Care should therefore be taken not to compromise the $K_c$ keys used in PANA/GSM to attackers when they are transmitted between entities, or handled outside a protected environment.

A threat tree summarising how a spoof network access device could impersonate a valid PAA to the PaC by illegally accessing and then using GSM triplets is given in Figure 11.11.

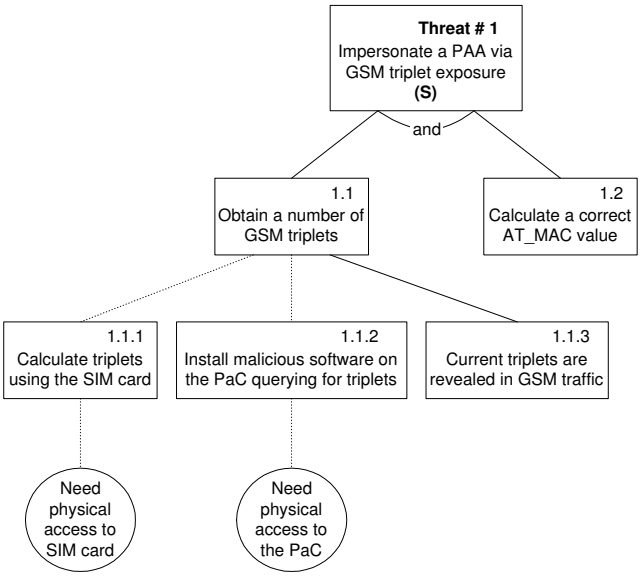


Figure 11.11: **Threat tree for PAA spoofing via GSM triplet exposure**

**User Identity Disclosure** As stated in section 6.2.1.2, some clients might wish to hide their identities from visited access networks for privacy reasons. PANA/GSM includes user identity confidentiality support, a GSM security service (see section 3.5.1.3) which protects the privacy of the user identifier against *passive* attacks (e.g. eavesdropping). However, the mechanism cannot be used on the first connection with a given PAA, since in this case the permanent user identifier needs to be sent in clear (as discussed in section 3.5.1.3). In this case, an *active* attacker that impersonates the access network may learn the subscriber's permanent identifier. However, the PaC can refuse to send the cleartext permanent user identifier to the PAA if it believes that the visited access network should be able to recognise its temporary identifier (or *pseudonym*).

If user identity confidentiality is required and the PaC and PAA cannot guarantee that the pseudonym will be maintained reliably, then an external security mechanism may be used to provide additional protection. Nevertheless, this kind of tunnelling mechanism can itself introduce new security vulnerabilities, as described in section 3.2.3.

A threat tree summarising how a malicious user could learn a permanent GSM user identifier by using passive or active attacks is given in Figure 11.12.

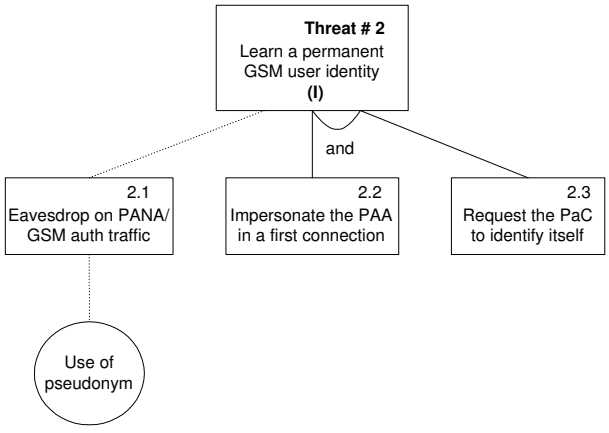


Figure 11.12: **Threat tree for a permanent GSM user identifier disclosure**

**Session Key Disclosure** PANA/GSM key derivation combines several GSM triplets in order to derive stronger keying material and AT_MAC values (as detailed in section 7.5.2.2). The actual strength of the resulting keys depends, among other things, on the operator-specific authentication algorithms, the strength of the key $K_i$, and the quality of the *RAND* challenges. At no point does PANA/GSM require the keys $K_c$ or the derived *SRES* values to be communicated.

A passive eavesdropper can learn $n$ different *RAND* values and the corresponding AT_MAC, and may be able to link this information to the user identity. An active attacker that impersonates a GSM subscriber could easily obtain $n$ different *RAND* values and the corresponding AT_MAC values from the EAP server for any user identity. However, as long as the cryptographic functions used are sufficiently robust, this should not enable the attacker to deduce the correct *SRES* and $K_c$ values.

A threat tree summarising how an attacker could attempt to disclose the correct PANA/GSM session keys is given in Figure 11.13.

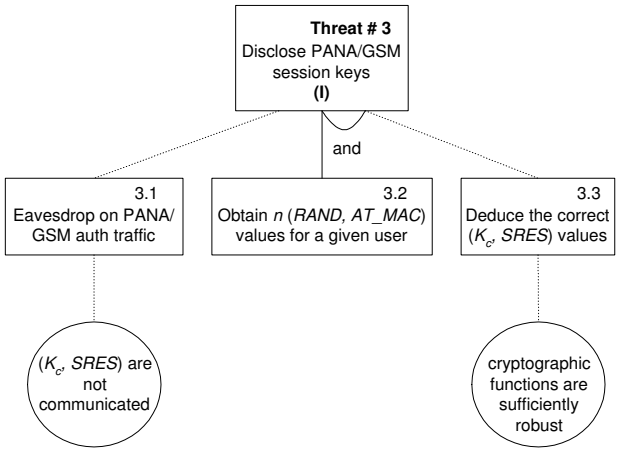


Figure 11.13: **Threat tree for a PANA/GSM session key disclosure**

**Man-in-the-Middle Attacks** As stated in section 6.2.1.2, an attacker can claim to be the PAA to the real PaC, and also claim to be the PaC to the genuine PAA. This is called a Man-in-the-Middle (MitM) attack, whereby the PaC is fooled into believing that it is communicating with the real PAA, which is also misled into believing that it is communicating with the genuine PaC.

Care has to be taken to avoid MitM attacks arising when tunnelling is used with PANA/GSM, e.g. when using PEAP (described in section 3.7.3), or when EAP-SIM (detailed in section 7.4) is part of a sequence of EAP methods. Such vulnerabilities can arise even when the individual authentication protocols used are in themselves secure. An example of such a MitM problem is described by Asokan, Niemi and Nyberg [21] (as discussed in sections 3.2.3 and 6.2.1.2).

When such attacks are successfully carried out, the attacker acts as an intermediary between a PaC victim and a legitimate PAA. This allows the attacker to authenticate successfully to the PAA, as well as to obtain access to the network. As a solution to the problem, Asokan, Niemi and Nyberg suggest cryptographically binding the session keys of the two phases, i.e. binding together the tunnel session key and the *MSK* derived from the EAP-SIM method. Even when tunnelling or an EAP sequence of methods are not used with PANA/GSM, user data need to be integrity protected on physically insecure networks to avoid MitM attacks and session hijacking.

A threat tree summarising how an attacker could attempt to launch a MitM attack against the PANA/GSM authentication scheme is given in Figure 11.14.

**Service Theft and Dictionary Attacks** As discussed in section 6.2.1.2, an attacker can gain unauthorised network access by stealing service from a legitimate client. Once the genuine PaC has been authenticated, an EP will typically have filters in place to prevent unauthorised network access. These

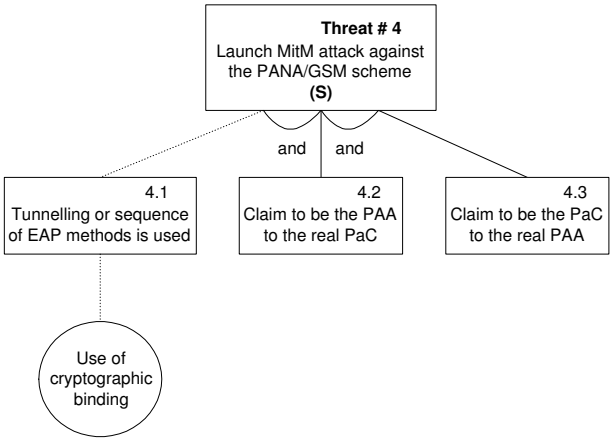


Figure 11.14: **Threat tree for MitM attacks against PANA/GSM**

filters will be based on something carried in every packet, for example, the IP and MAC addresses. In this latter case, any received packets will be dropped unless they contain specific IP addresses matching the MAC addresses.

PANA/GSM does not specify a mechanism for preventing service theft (described in section 6.2.1.2). Therefore an attacker can gain unauthorised access to the network by spoofing both the IP and MAC addresses of a legitimate PaC, and thereby steal service from another user. In a non-shared medium, service theft can be prevented by simple IP address and MAC address filters. In shared links, filters are not sufficient to prevent service theft as they can easily be spoofed (as described by Parthasarathy [151]). An Internet draft [150] describes how an IPsec SA (see section 3.6.5) can be established to secure the link between the PaC and the EP, which can be used to prevent service theft in the access network.

A threat tree summarising how an attacker could gain unauthorised access to the network by stealing service from another PANA/GSM user is given in Figure 11.15.

Because PANA/GSM is not a password-based protocol, it is not vulnerable

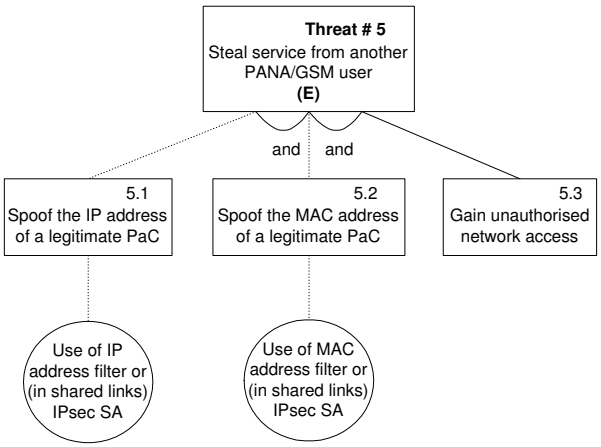


Figure 11.15: **Threat tree for service theft attacks against PANA/GSM**

to *dictionary* attacks (see section 3.3.4), assuming that the pre-shared secret is not derived from a weak password, name, or other low entropy source.

**Credential Reuse and Brute-Force Attacks** PANA/GSM cannot prevent attacks taking place within the GSM networks. If the SIM credentials used for PANA/GSM are also used in GSM, then it is possible to mount attacks via the GSM air interface. A passive attacker can eavesdrop on GSM traffic and obtain (*RAND*, *SRES*) pairs. The attacker can then use a *brute-force* attack[7] to obtain each of the 64-bit keys $K_c$ used to encrypt the GSM data. If the attacker can obtain $n$ 64-bit confidentiality keys $K_c$ ($n = 2$ or 3), he/she can then impersonate a valid network to a PANA/GSM client.

An active attacker can mount a 'false GSM base station (BS) attack', replaying previously seen *RAND* challenges to obtain *SRES* values (see [145] for further details). The attacker can then use a brute-force attack to obtain the keys $K_c$. If the attack is successful, then the attacker can impersonate a valid network or decrypt previously seen traffic. However, it should be noted that these attacks are not possible if the SIM credentials used in PANA/GSM are

[7] An attack in which all possibilities to guess a secret are tried.

not also used in the GSM network. It should also be noted that performing a brute-force search for a 64-bit key is a non-trivial task that could not be executed in real time; moreover, it is unlikely to be worth the effort of performing such a search just to steal network access.

A threat tree summarising how an attacker could attack PANA/GSM when using SIM credentials also used in the GSM network is given in Figure 11.16[8].

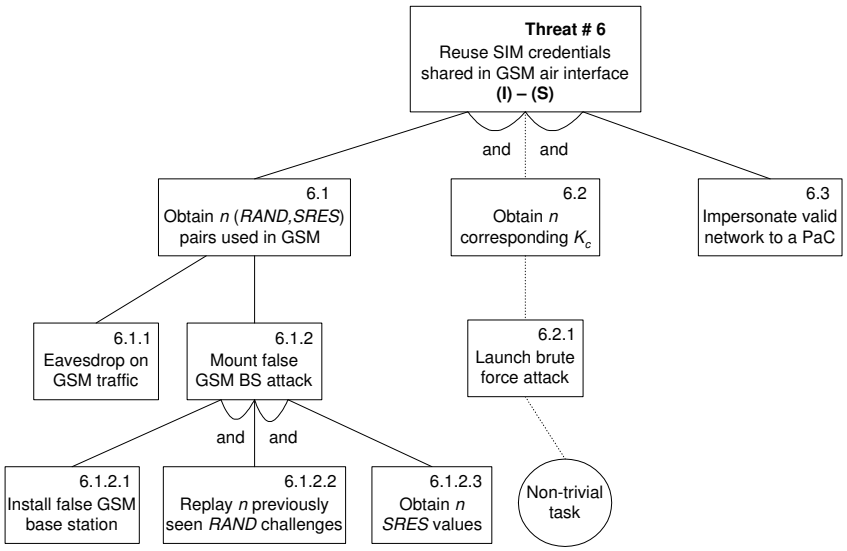


Figure 11.16: **Threat tree for SIM credential reuse and brute-force attacks**

**Tampering with Signalling Traffic** PANA/GSM signalling data could be modified as it flows between the PaC and the PAA. The protection of signalling message exchanges through the PANA/GSM SA prevents an opponent from acting as a MitM adversary, from session hijacking, from injecting packets, from replaying messages, and from modifying the content of the exchanged packets. Also, as with all PANA methods, in PANA/GSM an integrity object is defined, supporting data-origin authentication, replay protection using sequence num-

[8]As discussed in [81, p86], note that 'some threat types can interrelate'. It is common for information disclosure threats ($I$) to lead to spoofing threats ($S$). This effect can be seen by comparing subthreat 6.3 of this figure with the root node of the threat tree shown in Figure 11.11, which are almost identical.

bers and nonces, and integrity protection using a MAC function (see sections 6.1.4 and 7.5.2.1).

Moreover, certain EAP-SIM attributes are used to provide integrity, replay protection, and confidentiality for EAP-SIM payloads, except for the initial EAP/SIM/Start round trip (see section 7.4). However, in this latter case the protocol values are protected by a later PANA/GSM exchange.

A threat tree summarising how an attacker could attempt to tamper with the PANA/GSM signalling traffic is given in Figure 11.17.

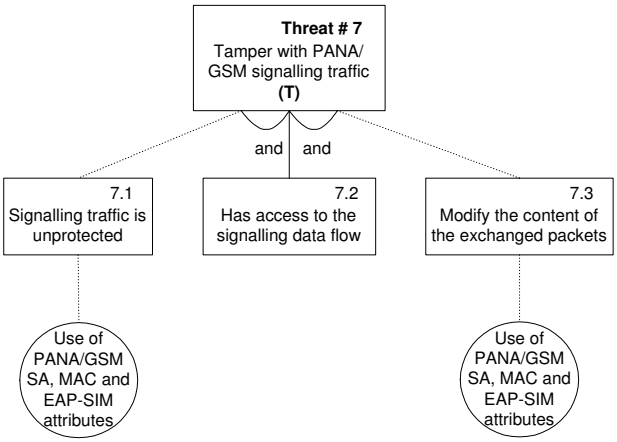


Figure 11.17: **Threat tree for PANA/GSM signalling traffic tampering**

**Bidding Down Attack** As described in section 6.1.4.1, the PAA is discovered by sending solicitations or receiving advertisements from the PaC. In this initial stage of the PANA/GSM protocol, the PaC has no assurance that the other end of the link is the PAA (see section 6.2.1.1), and an attacker can pretend to be a PAA by sending a spoofed advertisement. This threat primarily applies in environments where the PaC-PAA link is shared.

The advertisement may be used to include information other than the discovery of the PAA itself. This can, for instance, lead to a *bidding down* attack

(see section 6.2.1.2), where an attacker sends a spoofed advertisement with capabilities indicating authentication methods less secure than those that the real PAA supports, thereby fooling the PaC into negotiating a method less secure than would otherwise be available. Of course, such an attack will only succeed if the fake PAA can break the weaker authentication method and the weaker method is accepted by the PaC.

Moreover, the possibility of such an attack is essentially inevitable in any system allowing negotiation of the authentication method to be used. Hence, EAP method downgrading attacks might be possible, because PANA/GSM does not protect the EAP method negotiation, especially if the user employs the EAP-SIM identifier with other EAP methods. However, the specification of the EAP architecture (see section 3.4) describes how to avoid attacks that negotiate the least secure EAP method from among a set. If a peer needs to make use of different EAP authentication methods, then distinct identifiers should be employed, each of which identifies exactly one authentication method.

In any case, some protection against such an attack can be offered by repeating the list of supported EAP methods protected with the PANA/GSM SA. PANA/GSM does not support cipher suite negotiation, but includes an EAP-SIM version negotiation procedure[9] (see section 7.4). Of course, full protection against such an attack is provided if legitimate parties only accept the use of robust cryptographic techniques.

A threat tree summarising how an attacker could launch a bidding down attack against the PANA/GSM authentication scheme is given in Figure 11.18.

[9]The shared secret used to establish a PANA/GSM SA is derived from the secret key $MK$. As shown in equation 7.3, $MK$ is the output of the hash function SHA-1, which uses as input, among other values, the concatenation of the list of the supported EAP-SIM versions ($Version\_List$) and the identifier of the EAP-SIM version in use ($Selected\_Version$).

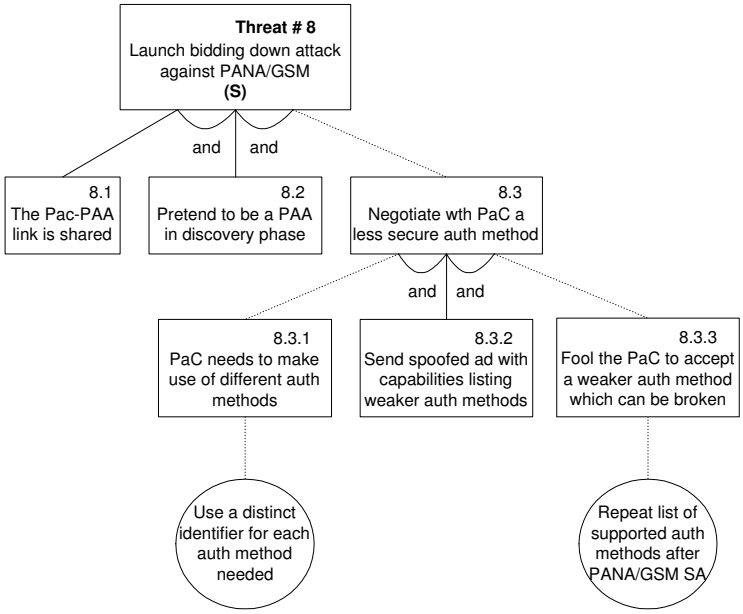


Figure 11.18: **Threat tree for bidding down attacks against PANA/GSM**

**Blind Resource Consumption DoS Attack** There are a variety of DoS attacks which can be launched against the PANA/GSM authentication process. For instance, to launch a 'blind resource consumption DoS attack' (described in section 3.2.3), an attacker can bombard the PAA with a large number of PaC authentication requests. If the PAA and the EAP server are not co-located, then the PAA may allocate local resources to store client state records before it receives the EAP server response. If a sufficiently large number of requests are received, then this could exhaust the PAA memory resources. Also, an attacker can force the PAA to make computationally intensive computations, which might exhaust the available processing resources.

PANA/GSM sequence numbers and cookies (as described in sections 3.2.3 and 6.1.4.1) provide protection against blind resource consumption DoS attacks. But PANA/GSM does not protect the EAP-SIM method exchange itself. Since, in particular, the PAA is not allowed to discard packets, and packets have to be

stored or forwarded to an AAA infrastructure, a risk of DoS attacks remains. Also PANA/GSM adopts the EAP-SIM mechanism, that is not a tunnelling method. Hence an adversary can both eavesdrop on the EAP-SIM payloads and inject arbitrary messages, which might confuse both the PaC and the PAA.

A threat tree summarising how an attacker could launch a blind resource consumption DoS attack against the PANA/GSM authentication scheme is given in Figure 11.19.

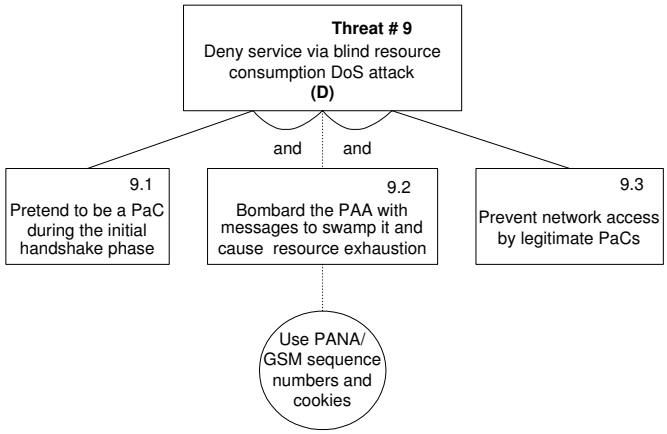


Figure 11.19: **Threat tree for blind resource consumption DoS attacks against PANA/GSM**

**DoS Attack using Termination Messages** The PaC or PAA may choose to discontinue the access service at any time[10]. Hence, as discussed in section 6.2.1.2, an attacker can pretend to be a PAA in a PANA/GSM exchange and revoke access to the PaC, causing a DoS attack on the PaC. An attacker can also pretend to be a genuine PaC and transmit a disconnect message, again causing a DoS attack on the PaC.

This kind of termination message causes state removal, a stop to the accounting procedure, and removes the installed packet filters. Thus such messages

[10] As explained in section 6.1.4.5, a routine for explicitly terminating a PANA session can be initiated either by the PaC (i.e. *disconnect indication*) or the PAA (i.e. *session revocation*).

need to be protected to prevent an adversary from deleting state information and thereby causing DoS attacks. If there is an established PANA/GSM SA (see section 7.6), all messages exchanged during the termination phase will be protected with a PANA/GSM-based MAC AVP, which neutralises this threat.

A threat tree summarising how an attacker could launch a DoS attack using PANA/GSM termination messages is given in Figure 11.20.

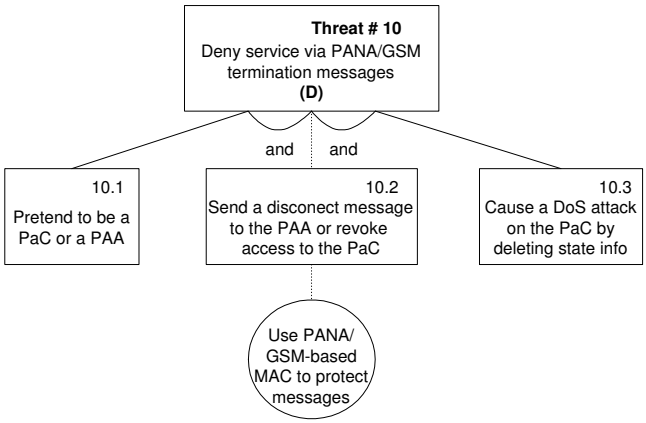


Figure 11.20: **Threat tree for DoS attacks using PANA/GSM termination messages**

**DoS Attack using False Success or Failure Indications** In physically insecure networks, an attacker might attempt to mount DoS attacks by sending false PANA/GSM success or failure indications. As discussed in section 6.2.1.2, by sending a false failure message, an attacker can prevent the client from accessing the network. By sending a false success message, an attacker can prematurely end the authentication exchange, denying service for the PaC. This attack is possible if the success or failure indication is not protected by a security association between the PaC and the PAA. All PANA/GSM messages exchanged prior to completion of the key establishment process may be unprotected.

Nevertheless, the attacker cannot force the PaC or the PAA to believe successful authentication has occurred when mutual authentication has failed or has not yet happened. In addition, any message whose sequence number is different to the expected value (e.g. a duplicate answer), and any message that fails to pass the MAC verification step, is immediately discarded by the receiver.

A threat tree summarising how an attacker could launch a DoS attack using false PANA/GSM success or failure indications is given in Figure 11.21.

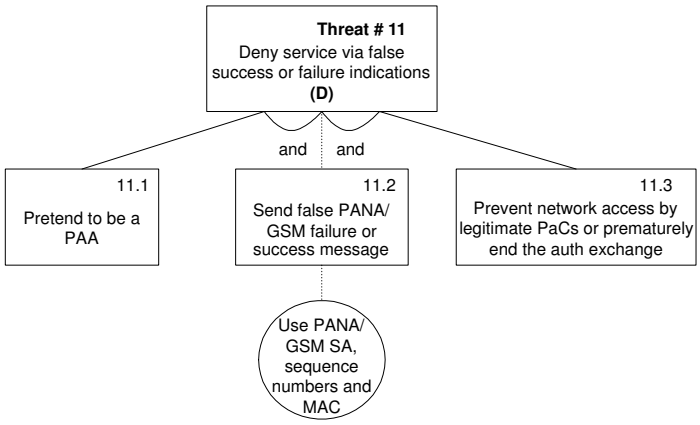


Figure 11.21: **Threat tree for DoS attacks using false PANA/GSM success or failure indications**

**IP Address Depletion Attack** Another kind of attack, known as an 'IP address depletion attack' (see section 6.2.1.2), arises from the fact that the PaC acquires an IP address before the PANA/GSM authentication process begins. When this occurs, it opens up the possibility of DoS attacks in which attackers can exhaust the IP address space by acquiring multiple IP addresses, or deny IP address allocations to other entities by falsely responding to every duplicate address detection query.

An IP address depletion attack can be prevented by deploying a secure address resolution scheme that does not depend on the client authentication pro-

cess, such as the SEND mechanism (see section 6.2.1.2).

A threat tree summarising how an attacker could launch an IP address depletion attack against the PANA/GSM authentication scheme is given in Figure 11.22.

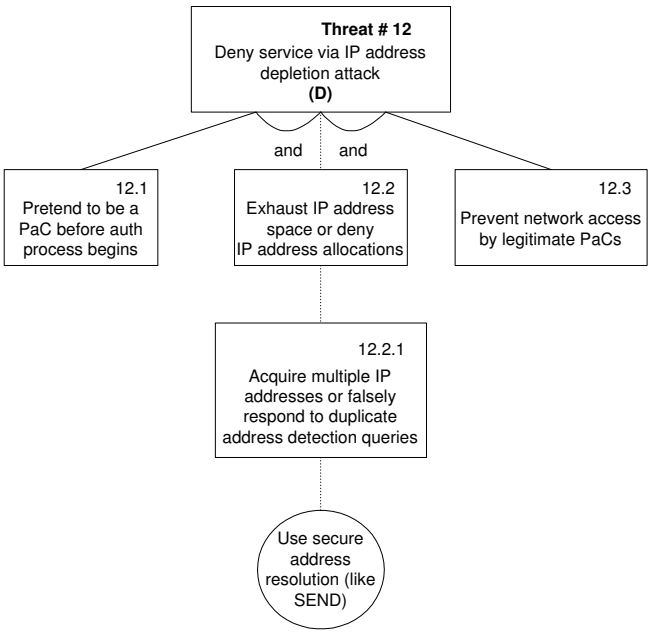


Figure 11.22: **Threat tree for IP address depletion attacks against PANA/GSM**

#### 11.4.2.3 PANA/UMTS and PANA/Liberty Threat Trees

As previously stated, the entity authentication schemes proposed in Chapters 8 and 9 make use of an identical underlying security mechanism (see section 11.3.2). Consequently, after the decomposition process performed in section 11.3, analogous threat targets are identified in both authentication techniques. This leads to the recognition of similar potential threats, which thus allows use of the same threat trees for both protocols.

We now consider the security threats to the PANA/UMTS and PANA/Liberty protocols (described in detail in Chapters 8 and 9), in order to create the corresponding threat trees.

**User Identity Disclosure** Both PANA/UMTS and PANA/Liberty include user identity confidentiality support, a UMTS security feature (see section 3.5.3.2) which protects the privacy of the user identifier against passive attacks. However, the mechanism cannot be used on the first connection with a given PAA, since in this case the permanent user identifier needs to be sent in clear (as discussed in section 3.5.3.2). Thus, an active attacker that impersonates the access network may learn the subscriber's permanent identifier. However, the PaC can refuse to send the cleartext permanent user identifier to the PAA if it believes that the visited access network should be able to recognise its pseudonym.

If user identity confidentiality is required, and the PaC and PAA cannot guarantee that the pseudonym will be maintained reliably, then an external security mechanism may be used to provide additional protection. Nevertheless, this kind of tunnelling mechanism can itself introduce new security vulnerabilities, as described in section 3.2.3.

A threat tree summarising how a malicious user could learn a permanent UMTS user identifier by using passive or active attacks is given in Figure 11.23.

**MitM Attacks** Care has to be taken to avoid MitM attacks arising when tunnelling is used with PANA/UMTS or PANA/Liberty, e.g. when using PEAP (described in section 3.7.3), or when EAP-AKA (detailed in section 8.4) is part of a sequence of EAP methods[11]. An example of such a MitM problem is discussed by Asokan, Niemi and Nyberg [21]. As a solution to the problem,

[11] As discussed in section 11.4.2.2, when such attacks are successfully carried out, the attacker acts as an intermediary between a PaC victim and a legitimate PAA. This allows the attacker to authenticate successfully to the PAA, as well as to obtain access to the network.

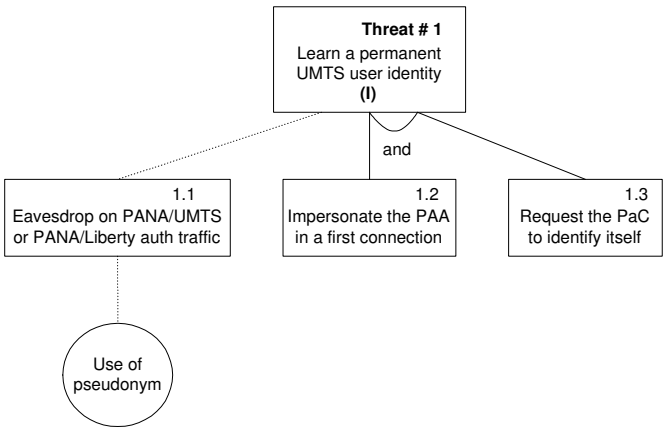


Figure 11.23: **Threat tree for a permanent UMTS user identifier disclosure**

Asokan, Niemi and Nyberg suggest cryptographically binding the session keys of the two phases, i.e. binding together the tunnel session key and the *MSK* derived from the EAP-AKA method. Even when tunnelling or an EAP sequence of methods are not used with PANA/UMTS or PANA/Liberty, user data need to be integrity protected on physically insecure networks to avoid MitM attacks and session hijacking.

A threat tree summarising how an attacker could attempt to launch a MitM attack against the PANA/UMTS or the PANA/Liberty authentication schemes is given in Figure 11.24.

**Service Theft Attacks** Both PANA/UMTS and PANA/Liberty do not prevent an attacker from gaining unauthorised access to the network by stealing service from another user (described in section 6.2.1.2). However, a summary of how to prevent service theft in the access network was given in section 11.4.2.2. Hence, the solutions adopted by PANA/GSM for shared and non-shared links can also be adopted by PANA/UMTS and PANA/Liberty.

A threat tree summarising how an attacker could gain unauthorised access to

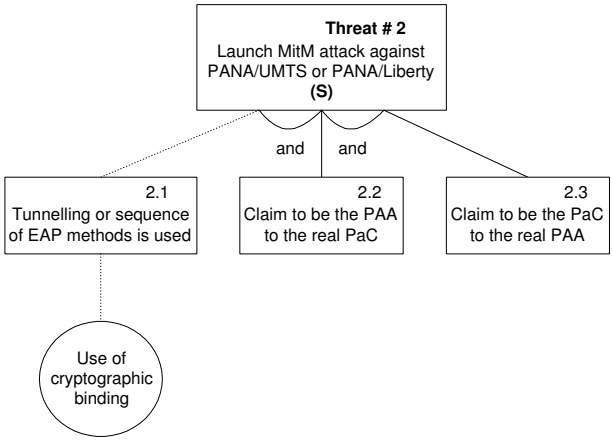


Figure 11.24: **Threat tree for MitM attacks against PANA/UMTS or PANA/Liberty**

the network by stealing service from another PANA/UMTS or PANA/Liberty user is given in Figure 11.25.

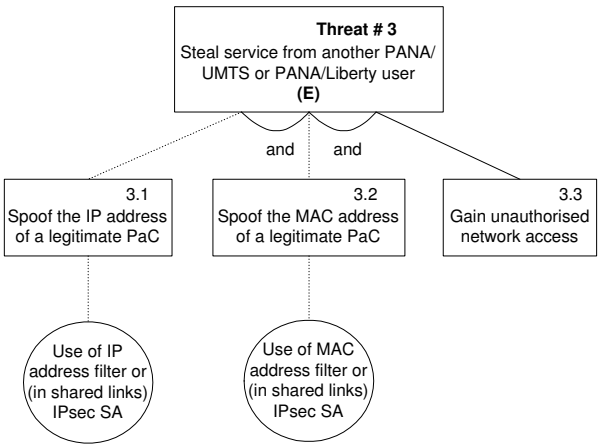


Figure 11.25: **Threat tree for service theft attacks against PANA/UMTS or PANA/Liberty**

**Tampering with Signalling Traffic** PANA/UMTS or PANA/Liberty signalling data could be modified as it flows between the PaC and the PAA. The protection of signalling traffic through PANA/UMTS or PANA/Liberty

SAs prevents an opponent from acting as a MitM adversary, from session hijacking, from injecting packets, from replaying messages, and from modifying the content of the exchanged packets. Also, as with all PANA methods, in both PANA/UMTS and PANA/Liberty an integrity object is defined, supporting data-origin authentication, replay protection using sequence numbers and nonces, and integrity protection using a MAC function (see sections 6.1.4 and 8.5.2.1).

Moreover, certain EAP-AKA attributes are used to provide integrity, confidentiality, and replay protection for EAP-AKA payloads exchanged in both the PANA/UMTS and PANA/Liberty schemes. In this case, integrity protection is based on a MAC (i.e. AT_MAC — see section 8.4). The messages may also optionally contain encrypted data (AT_ENCR_DATA) for identity confidentiality and fast re-authentication support (as discussed in section 8.4). On full authentication, replay protection for the EAP-AKA payload is provided by the underlying UMTS AKA scheme, which makes use of a random challenge (*RAND*) and a network authentication token (*AUTN*), both obtained from the authentication vector[12].

A threat tree summarising how an attacker could attempt to tamper with the PANA/UMTS or PANA/Liberty signalling traffic is given in Figure 11.26.

**Bidding Down Attack** EAP method *bidding down* attacks[13] might be possible, because PANA/UMTS and PANA/Liberty do not protect the EAP method negotiation[14], especially if the user employs the EAP-AKA identifier with other EAP methods. However, a summary of how to avoid attacks that negotiate the least secure EAP method from among a set was given in the previous sec-

[12]The authentication vector (*RAND*, *AUTN*, *XRES*, *IK*, *CK*) is produced from a 128-bit secret key $K$, shared by the USIM and the HN AuC, and a sequence number (see section 8.4).

[13]As discussed in the previous section, in a *bidding down* attack an attacker fools the PaC into negotiating an authentication method less secure than would otherwise be available.

[14]PANA/UMTS and PANA/Liberty actually do not support EAP-AKA protocol version negotiation or ciphersuite negotiation.

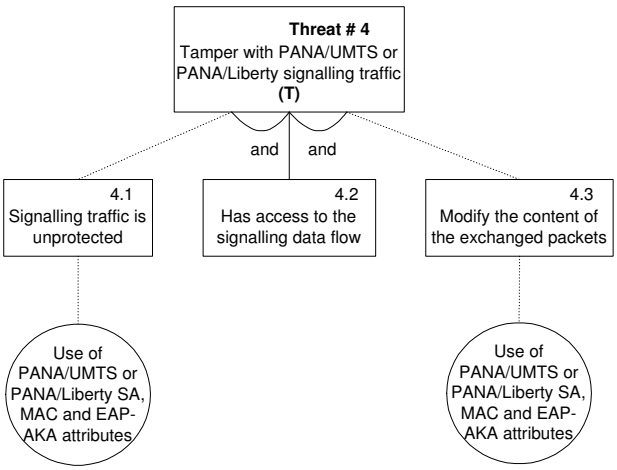


Figure 11.26: **Threat tree for PANA/UMTS or PANA/Liberty signalling traffic tampering**

tion. Hence, the solutions adopted by PANA/GSM can also be adopted by PANA/UMTS and PANA/Liberty.

A threat tree summarising how an attacker could launch a bidding down attack against the PANA/UMTS or the PANA/Liberty authentication schemes is given in Figure 11.27.

**Blind Resource Consumption DoS Attack** In order to launch a 'blind resource consumption DoS attack' (see section 3.2.3) against PANA/UMTS or PANA/Liberty, an attacker could make use of the same steps adopted against PANA/GSM (detailed in the previous section). Like PANA/GSM, PANA/UMTS and PANA/Liberty do not protect the EAP-AKA method exchange itself, and the EAP-AKA mechanism is not a tunnelling method. Hence an adversary can both eavesdrop on the EAP-AKA payloads and inject arbitrary messages which might confuse both the PaC and the PAA. A summary of how to provide protection against blind resource consumption DoS attacks by means of sequence numbers and cookies was given in the previous section. Hence, the solutions adopted by PANA/GSM can also be adopted by PANA/UMTS and

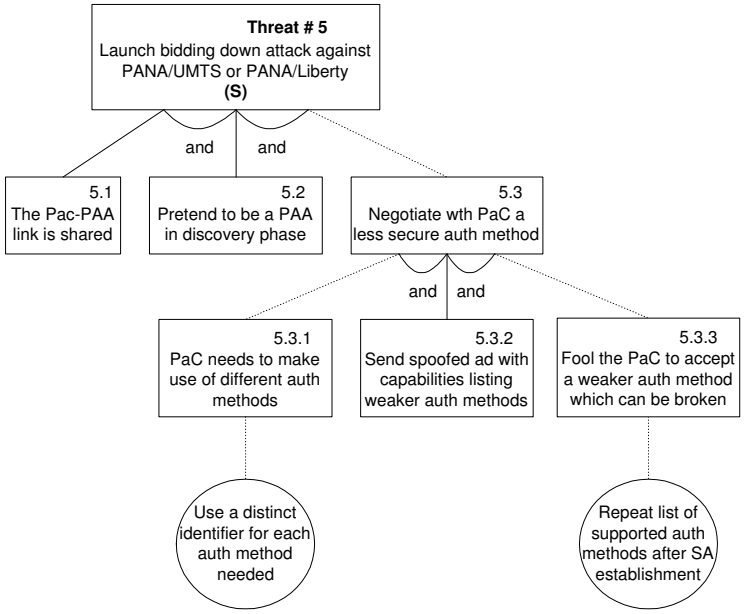


Figure 11.27: **Threat tree for bidding down attacks against PANA/UMTS or PANA/Liberty**

PANA/Liberty.

A threat tree summarising how an attacker could launch a blind resource consumption DoS attack against the PANA/UMTS or the PANA/Liberty authentication schemes is given in Figure 11.28.

**DoS Attack using Termination Messages** As discussed in section 6.2.1.2, an attacker can pretend to be a PAA in a PANA/UMTS or PANA/Liberty exchange and revoke access to the PaC, causing a DoS attack on the PaC. An attacker can also pretend to be a genuine PaC and transmit a disconnect message, again causing a DoS attack on the PaC.

This kind of termination message causes state removal, a stop to the accounting procedure, and removes the installed packet filters. Thus such messages need to be protected to prevent an adversary from deleting state information and thereby causing DoS attacks. If there is an established security association,

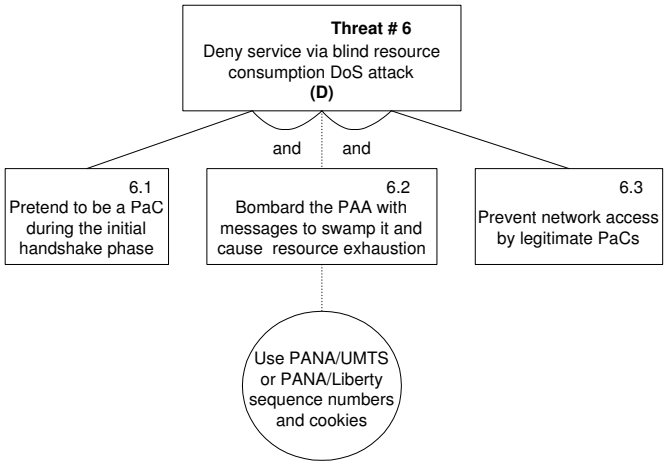


Figure 11.28: **Threat tree for blind resource consumption DoS attacks against PANA/UMTS or PANA/Liberty**

all messages exchanged during the termination phase will be protected with a PANA/UMTS-based or PANA/Liberty-based MAC AVP, which neutralises this threat.

A threat tree summarising how an attacker could launch a DoS attack using PANA/UMTS or PANA/Liberty termination messages is given in Figure 11.29.

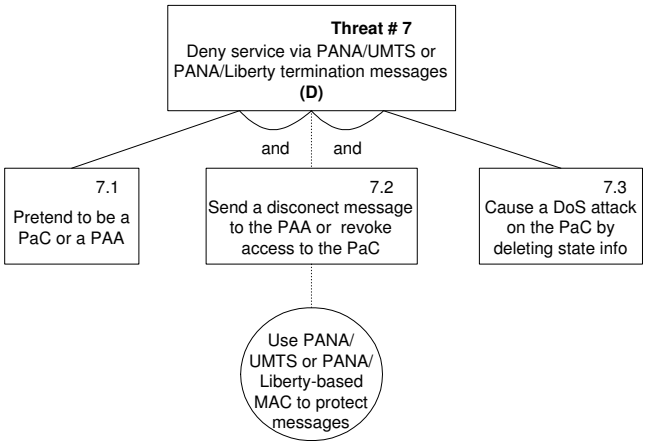


Figure 11.29: **Threat tree for DoS attacks using PANA/UMTS or PANA/Liberty termination messages**

**DoS Attack using False Success or Failure Indications** In physically insecure networks, an attacker might attempt to mount DoS attacks by sending false PANA/UMTS or PANA/Liberty success/failure indications. As discussed in the previous section, this attack is possible if the success or failure indication is not protected by a security association between the PaC and the PAA. All PANA/UMTS or PANA/Liberty messages exchanged prior to completion of the key establishment process may be unprotected.

Nevertheless, the attacker cannot force the PaC or the PAA to believe successful authentication has occurred when mutual authentication has failed or has not yet happened. In addition, any message whose sequence number is different to the expected value (e.g. a duplicate answer), and any message that fails to pass the MAC verification step, is immediately discarded by the receiver.

A threat tree summarising how an attacker could launch a DoS attack using false PANA/UMTS or PANA/Liberty success/failure indications is given in Figure 11.30.

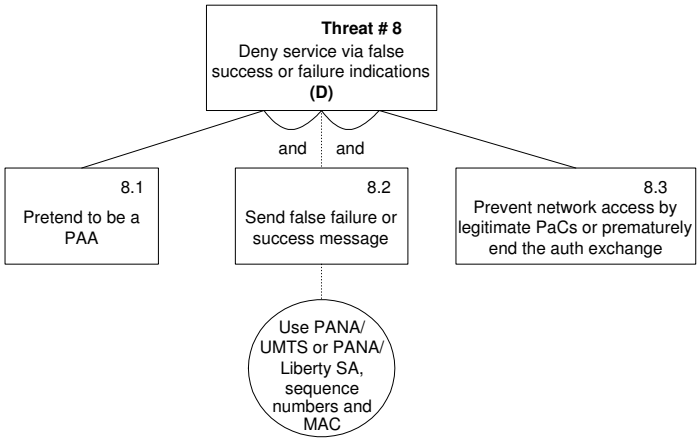


Figure 11.30: **Threat tree for DoS attacks via false PANA/UMTS or PANA/Liberty success/failure indications**

**IP Address Depletion Attack** An 'IP address depletion attack' (see section 6.2.1.2) arises from the fact that the PaC acquires an IP address before the PANA/UMTS or the PANA/Liberty authentication processes begin. As stated previously, this opens up the possibility of DoS attacks in which attackers can exhaust the IP address space by acquiring multiple IP addresses, or deny IP address allocations to other entities by falsely responding to every duplicate address detection query. An IP address depletion attack can be prevented by deploying a secure address resolution scheme that does not depend on the client authentication process, such as the SEND mechanism (see section 6.2.1.2).

A threat tree summarising how an attacker could launch an IP address depletion attack against the PANA/UMTS or PANA/Liberty authentication schemes is given in Figure 11.31.

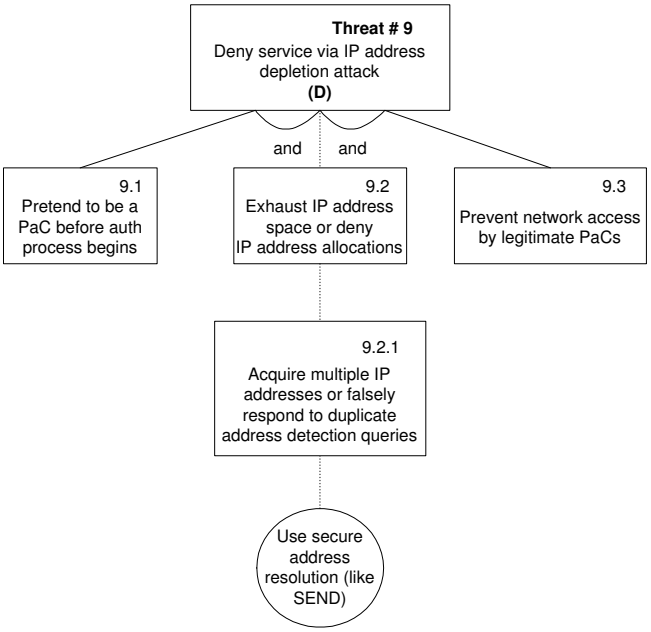


Figure 11.31: **Threat tree for IP address depletion attacks against PANA/UMTS or PANA/Liberty**

**Brute-Force and Dictionary Attacks** The effective key length in both PANA/UMTS and PANA/Liberty is 128 bits (see section 8.4), and there are no known computationally feasible brute-force attacks. Because PANA/UMTS and PANA/Liberty are not password-based protocols, they are not vulnerable to *dictionary* attacks (see section 3.3.4), assuming that the pre-shared secrets are not derived from a weak password, name, or other low entropy source.

A threat tree summarising how a malicious user could attempt to learn PANA/UMTS or PANA/Liberty keying material by using brute-force attacks is given in Figure 11.32.

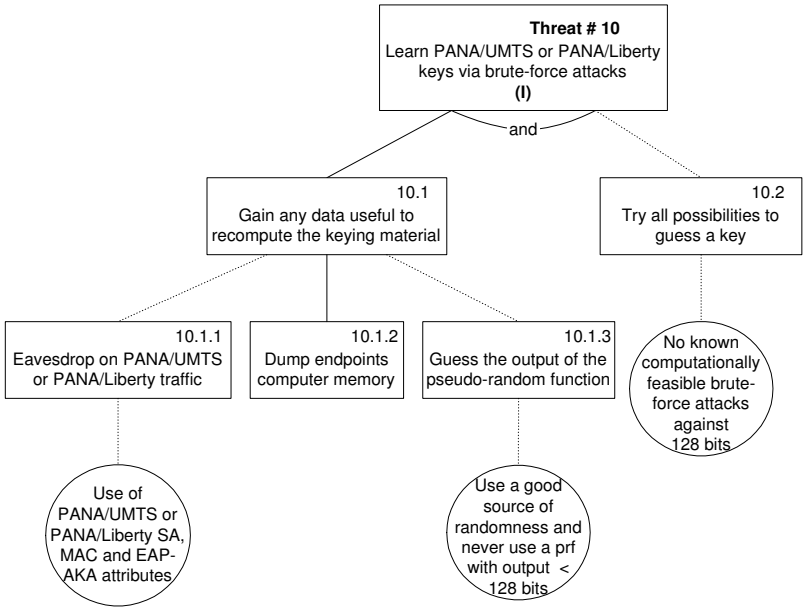


Figure 11.32: **Threat tree for brute-force attacks against the PANA/UMTS or PANA/Liberty keying material**

#### 11.4.2.4 PANA/IKEv2 Threat Trees

We now consider the security threats to the PANA/IKEv2 protocol (described in detail in Chapter 10), in order to create the corresponding threat trees.

**MitM Attacks** As discussed in section 11.4.2.2, when MitM attacks are successfully carried out, the attacker acts as an intermediary between a PaC victim and a legitimate PAA. Care has to be taken to avoid MitM attacks arising when tunnelling is used with PANA/IKEv2, e.g. when using PEAP (described in section 3.7.3), or when EAP-IKEv2 (detailed in section 10.4) is part of a sequence of EAP methods[15]. An example of such a MitM problem is discussed by Asokan, Niemi and Nyberg[16] [21]. As a solution to the problem, Asokan, Niemi and Nyberg suggest cryptographically binding the session keys of the two phases, i.e. binding together the tunnel session key $T$ (a typical example of $T$ is the TLS master key derived in the TLS handshake of PEAP) and the *KEYMAT* derived from the EAP-IKEv2 method, to generate an ultimate session key $K$.

There are two ways to achieve the necessary binding between *KEYMAT* and $K$. In the first method, the binding is established directly by taking *KEYMAT* in addition to $T$ as input to the computation of the session key $K$. This provides *implicit key authentication* of the PaC (see section 2.2.5). The second method is to make use of a cryptographic check value to verify that the PaC who is in possession of $T$ is also in possession of *KEYMAT*. This second type of binding provides *explicit key authentication* of the PaC (as described in section 2.2.5).

A threat tree summarising how an attacker could attempt to launch a MitM attack against the PANA/IKEv2 authentication scheme is given in Figure 11.33.

**User Identity Disclosure** As stated in section 6.2.1.2, some clients might wish to hide their identities from visited access networks for privacy reasons. In PANA/IKEv2, a large number of identities are involved due to multiple uses of identifiers for routing (i.e. authentication end point indication). The identifier

[15]Even when tunnelling or an EAP sequence of methods are not used with PANA/IKEv2, user data need to be integrity protected on physically insecure networks to avoid MitM attacks and session hijacking.

[16]The MitM attack described is taken into account in the design of IKEv2 (see section 3.8.1).

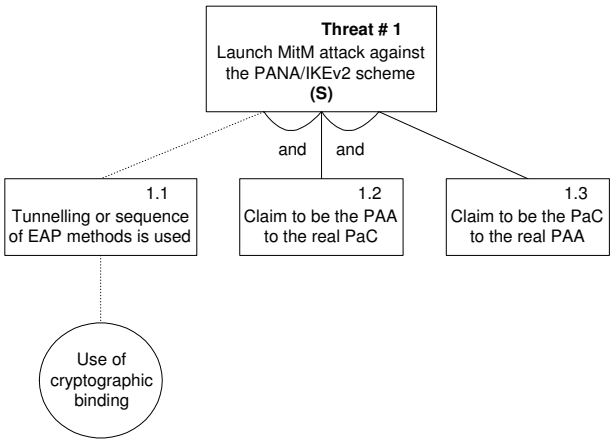


Figure 11.33: **Threat tree for MitM attacks against PANA/IKEv2**

types, their requirements for confidentiality and integrity protection, and their potential disclosure threats are as follows.

As shown in Figure 10.3, the identifier *Identity*, used in the first round trip of the PANA/IKEv2 authentication phase ($b$), indicates where the EAP messages terminate; it is not used to identify the PaC, and thus it does not allow the adversary to learn the identity of the PaC. The identifiers $ID_i$ and $ID_r$ are used respectively to identify the PAA ($f$) and PaC ($g$); $ID_i$ can be a fully-qualified domain name (FQDN), and $ID_r$ can be associated with a user identifier (e.g. an email address). Both identifiers are of importance for the PANA/IKEv2 Access phase (3), and are thus encrypted and integrity protected by PANA/IKEv2.

In summary, PANA/IKEv2 includes identity confidentiality and integrity protection support, which protects the privacy of the PaC and PAA identities against disclosure threats involving *passive* (e.g. eavesdropping) and *active* attackers (e.g. impersonation of the access network).

A threat tree summarising how a malicious user could attempt to learn a PANA/IKEv2 user identifier by using passive or active attacks is given in Figure 11.34.

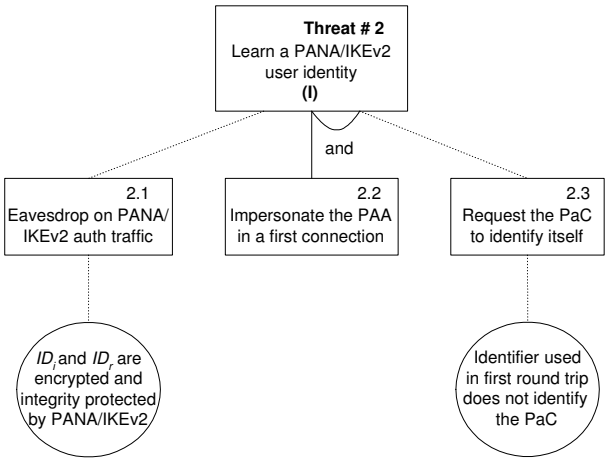


Figure 11.34: **Threat tree for a PANA/IKEv2 user identifier disclosure**

**Service Theft and Dictionary Attacks** PANA/IKEv2 does not prevent an attacker from gaining unauthorised access to the network by stealing service from another user (described in section 6.2.1.2). However, a summary of how to prevent service theft in the access network was given in section 11.4.2.2. The solutions adopted by PANA/GSM for shared and non-shared links can also be adopted by PANA/IKEv2.

A threat tree summarising how an attacker could gain unauthorised access to the network by stealing service from another PANA/IKEv2 user is given in Figure 11.35.

Because PANA/IKEv2 is not a password-based protocol, it is not vulnerable to *dictionary* attacks (see section 3.3.4), assuming that the pre-shared secret or the key used for digital signature are not derived from a weak password, name, or other low entropy source.

**Perfect Forward Secrecy, Brute-Force Attacks and Generation of Random Numbers** PANA/IKEv2 generates IKEv2 keying material using an ephemeral Diffie-Hellman exchange, in order to achieve the property of 'per-

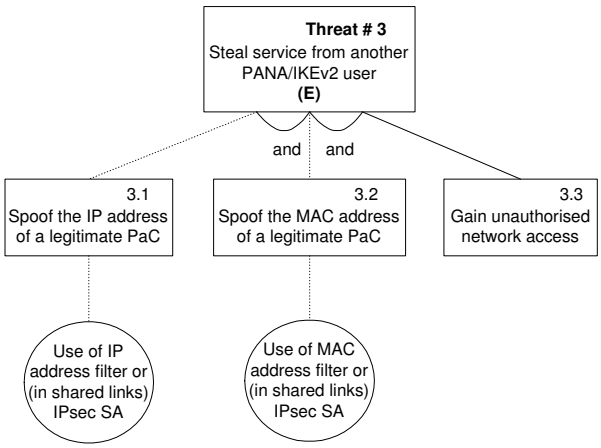


Figure 11.35: **Threat tree for service theft attacks against PANA/IKEv2**

fect forward secrecy' (see section 2.1.3.3). Support of this property requires that, when a connection is closed, each endpoint securely deletes not only the keys used by the connection but any data that could be used to recompute those keys.

The Diffie-Hellman exchange must be based on one of the groups defined in RFC 4306 [49] (see section 3.8.1), where all but the first of the groups (which is only present for historical reasons) offers security against any known computationally feasible *brute-force* attack. It is assumed that all Diffie-Hellman exponents are erased from computer memory after use.

In the context of the PANA/IKEv2 SA (see section 10.6), four cryptographic algorithms are negotiated: an encryption algorithm, an integrity protection algorithm, a Diffie-Hellman group, and a pseudo-random function (prf). The prf is used for the construction of keying material for all of the cryptographic algorithms used. The strength of all IKEv2 keys against brute-force attacks is limited by the size of the output of the negotiated prf. For this reason, a prf whose output is shorter than 128 bits (e.g. a CBC-MAC derived using a 64-bit block cipher) must never be used with the PANA/IKEv2 protocol. Finally, a

PANA/IKEv2 implementation also needs to use a good source of randomness to generate the random numbers (nonces) required in the protocol[17].

A threat tree summarising how a malicious user could attempt to learn PANA/IKEv2 keying material by using brute-force attacks is given in Figure 11.36.

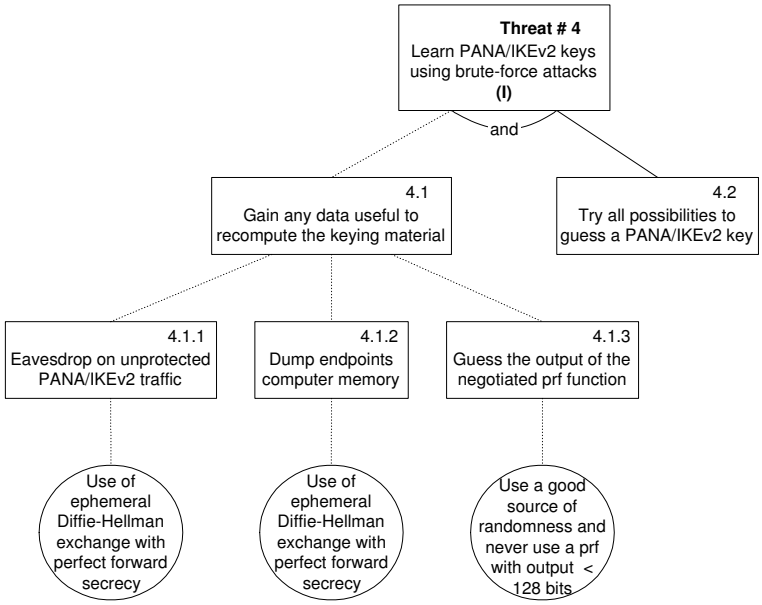


Figure 11.36: **Threat tree for brute-force attacks against the PANA/IKEv2 keying material**

**Tampering with Signalling Traffic** PANA/IKEv2 signalling data could be modified as it flows between the PaC and the PAA. The protection of signalling traffic through an PANA/IKEv2 SA prevents an opponent from acting as a MitM adversary, from session hijacking, from injecting packets, from replaying messages, and from modifying the content of the exchanged packets. Also, as with all PANA methods, in PANA/IKEv2 an integrity object is defined, supporting data-origin authentication, replay protection using sequence numbers

[17]See RFC 1750 [48] and ISO/IEC 18031 [105] for details on generating random numbers for security applications.

and nonces, and integrity protection using a MAC function (see sections 6.1.4 and 8.5.2.1).

Moreover, as discussed in section 10.4, in PANA/IKEv2 all but the IKEv2 headers of the messages that follow the Diffie-Hellman exchange are encrypted and integrity protected. The recipients must verify that all signatures and MACs are computed correctly, and that the identities $ID_i$ and $ID_r$ correspond to the keys used to generate the Authentication ($AUTH$) payload. The use of nonces guarantees liveliness during an exchange, and also protects against replay attacks.

A threat tree summarising how an attacker could attempt to tamper with the PANA/IKEv2 signalling traffic is given in Figure 11.37.

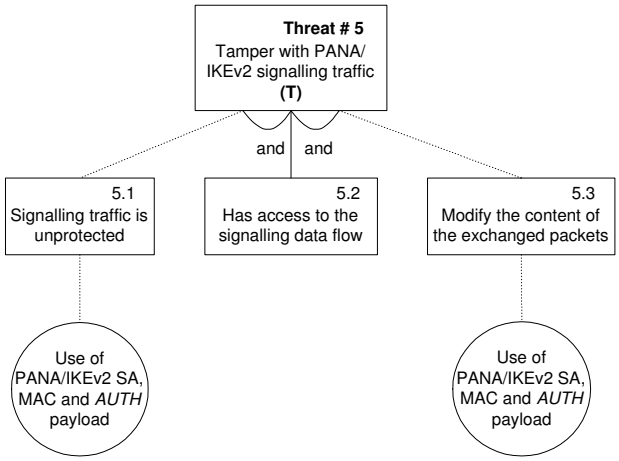


Figure 11.37: **Threat tree for PANA/IKEv2 signalling traffic tampering**

**Bidding Down Attack** EAP method *bidding down* attacks[18] might be possible, because PANA/IKEv2 does not protect the EAP method negotiation[19].

[18] As discussed in section 11.4.2.2, in a *bidding down* attack an attacker fools the PaC into negotiating an authentication method less secure than would otherwise be available.

[19] PANA/IKEv2 does not support EAP-IKEv2 protocol version negotiation, but supports cipher suite negotiation through IKEv2. In the context of the IKEv2 SA, four cryptographic algorithms are negotiated (see section 3.8.1).

However, a summary of how to avoid attacks that negotiate the least secure EAP method from among a set was given in section 11.4.2.2. The solutions adopted by PANA/GSM can also be adopted by PANA/IKEv2.

A threat tree summarising how an attacker could launch a bidding down attack against the PANA/IKEv2 authentication scheme is given in Figure 11.38.

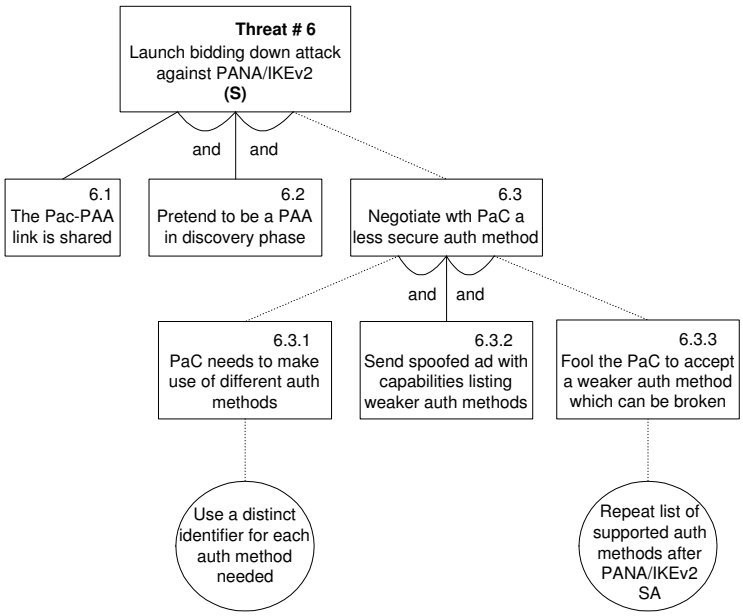


Figure 11.38: **Threat tree for bidding down attacks against PANA/IKEv2**

**Blind Resource Consumption DoS Attack** In order to launch a 'blind resource consumption DoS attack' (see section 3.2.3) against PANA/IKEv2, an attacker could make use of the same steps adopted against PANA/GSM (detailed in section 11.4.2.2). Like PANA/GSM, PANA/IKEv2 does not protect the EAP-IKEv2 method exchange itself, and the EAP-IKEv2 mechanism is not a tunnelling method. Hence an adversary can both eavesdrop on the EAP-IKEv2 payloads and inject arbitrary messages which might confuse both the

PaC and the PAA. A summary of how to provide protection against blind resource consumption DoS attacks through the use of sequence numbers and cookies was given in section 11.4.2.2. The solutions adopted by PANA/GSM can also be adopted by PANA/IKEv2.

A threat tree summarising how an attacker could launch a blind resource consumption DoS attack against the PANA/IKEv2 authentication scheme is given in Figure 11.39.

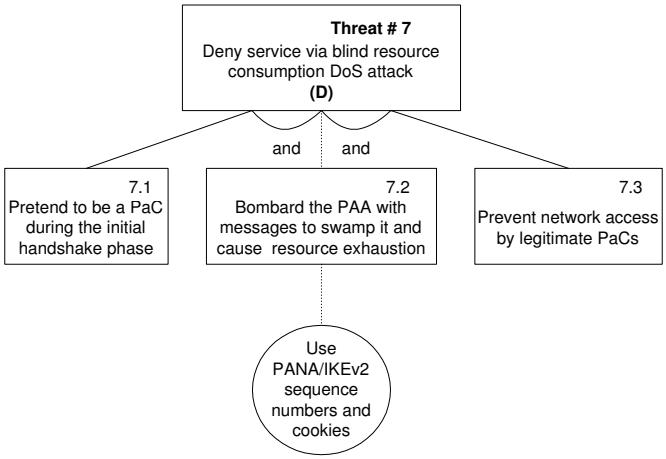


Figure 11.39: **Threat tree for blind resource consumption DoS attacks against PANA/IKEv2**

**DoS Attack using Termination Messages** As discussed in section 6.2.1.2, an attacker can pretend to be a PAA in a PANA/IKEv2 exchange and revoke access to the PaC, causing a DoS attack on the PaC. An attacker can also pretend to be a genuine PaC and transmit a disconnect message, again causing a DoS attack on the PaC.

This kind of termination message causes state removal, a stop to the accounting procedure, and removes the installed packet filters. Thus such messages need to be protected to prevent an adversary from deleting state information and thereby causing DoS attacks. If there is an established security association,

all messages exchanged during the termination phase will be protected with a PANA/IKEv2-based MAC AVP, which mitigates this threat.

A threat tree summarising how an attacker could launch a DoS attack using PANA/IKEv2 termination messages is given in Figure 11.40.

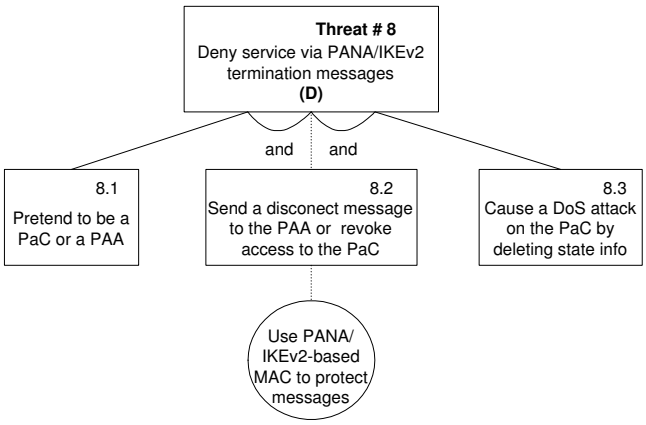


Figure 11.40: **Threat tree for DoS attacks using PANA/IKEv2 termination messages**

**DoS Attack using False Success or Failure Indications** In physically insecure networks, an attacker might attempt to mount DoS attacks by sending false PANA/IKEv2 success/failure indications. As discussed in section 11.4.2.2, this attack is possible if the success or failure indication is not protected by a security association between the PaC and the PAA. All PANA/IKEv2 messages exchanged prior to completion of the key establishment process may be unprotected.

Nevertheless, the attacker cannot force the PaC or the PAA to believe successful authentication has occurred when mutual authentication has failed or has not yet happened. In addition, any message whose sequence number is different to the expected value (e.g. a duplicate answer), and any message that fails to pass the MAC verification step, is immediately discarded by the receiver.

A threat tree summarising how an attacker could launch a DoS attack using false PANA/IKEv2 success/failure indications is given in Figure 11.41.

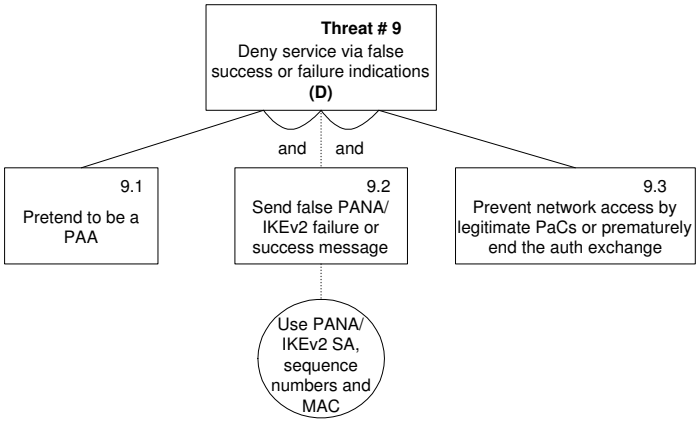


Figure 11.41: **Threat tree for DoS attacks via false PANA/IKEv2 success/failure indications**

**IP Address Depletion Attack** An 'IP address depletion attack' (see section 6.2.1.2) arises from the fact that the PaC acquires an IP address before the PANA/IKEv2 authentication processes begin. As previously described, this opens up the possibility of DoS attacks in which attackers can exhaust the IP address space by acquiring multiple IP addresses, or deny IP address allocations to other entities by falsely responding to every duplicate address detection query. An IP address depletion attack can be prevented by deploying a secure address resolution scheme that does not depend on the client authentication process, such as the SEND mechanism (see section 6.2.1.2).

A threat tree summarising how an attacker could launch an IP address depletion attack against the PANA/IKEv2 authentication scheme is given in Figure 11.42.

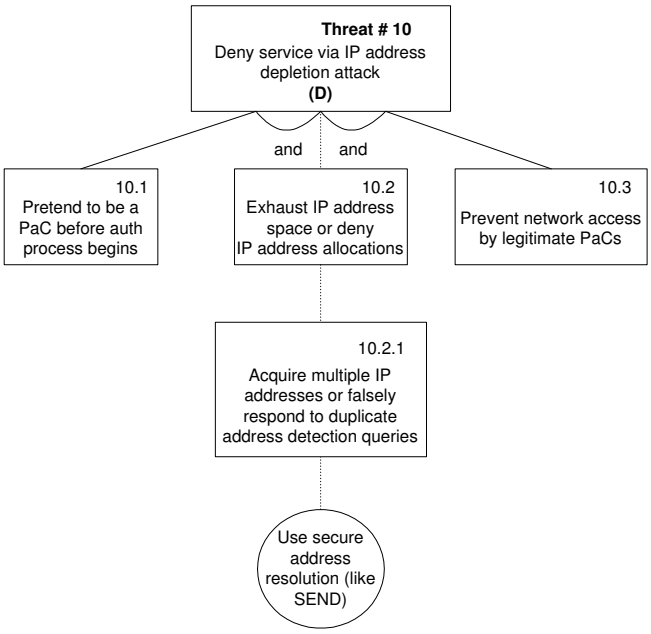


Figure 11.42: **Threat tree for IP address depletion attacks against PANA/IKEv2**

## 11.5 Ranking the Threats by Decreasing Risk

According to Howard and LeBlanc [81, p93], after creating the threat trees in the previous section, the next step is to use a threat ranking method, such as DREAD, to determine the security risk for each of the captured threats.

In this section, a description of the basic concepts underlying DREAD is first provided (section 11.5.1). We then use this threat ranking mechanism to calculate the overall security risk for each of the proposed authentication techniques (section 11.5.2).

### 11.5.1 DREAD Ranking Method

Ranking the threats involves calculating the risk that the threat causes to the proposed authentication techniques. As suggested in [81, p93], we use the con-

cept of *DREAD*, an acronym derived from the five terms (or *metrics*, rated on a scale of 1 to 10) described below:

**Damage Potential** This metric involves rating the extent of the actual damage which is possible with a particular threat. Typically, the worst score (with metric value 10) is for a threat that allows the attacker to circumvent all security restrictions, and then do virtually any damage he/she wishes.

**Reproducibility** This term is used to measure how easy it is to realise a threat, i.e. to use it to generate an exploit. Some security flaws (especially those existing in features installed by default) enable attacks that work every time, and thus have high reproducibility (10), whilst others result in attacks whose results are unpredictable and might work only sporadically.

**Exploitability** This metric assesses how much effort and expertise is required to mount an attack (e.g. if a novice programmer with a home computer can mount the attack, then it scores a 10)[20]; it also considers what degree of authentication and authorisation is required to attack the system. For instance, if an anonymous remote user can attack the system, then the exploitability metric is set to 10, whilst a local user exploit requiring strong credentials has a lower exploitability.

**Affected Users** This metric quantifies roughly what percentage of users would be impacted if the threat were exploited by an attack: 91–100 percent (equating to a metric of 10) down to 0–10 percent (with a metric of 1). Distinguishing between server and client attacks is important here, since if a server is the threat target then a larger number of clients will be affected. Thus, all else being

[20]However, an attack that can only be launched by a national government needing to invest millions of dollars probably scores 1.

equal, attacks affecting servers are assigned a higher metric value than attacks on clients.

**Discoverability** The analysis below assumes that a threat will always be discoverable — this is because the attacks are against protocol specifications, which are likely to be in the public domain, as opposed to source code, which may not be generally available. Hence, we label each threat with the highest rating (10) for this metric, relying on the other metrics to guide our threat ranking.

We determine a DREAD overall rating for each of the captured threats by *averaging* the five metric values listed above (i.e. adding the values and dividing the sum by five). Once we have calculated the risk rating of each threat, we then sort all the threats to each of the proposed authentication techniques in descending order (i.e. threats with a higher risk first and lower-risk threats last).

### 11.5.2 Using DREAD to Calculate Security Risk

We now use the DREAD threat ranking procedure, discussed in the previous section, to determine the security risk (sections 11.5.2.1 to 11.5.2.3) and sort the threats (section 11.5.2.4) applying to each of the entity authentication proposals.

#### 11.5.2.1 PANA/GSM Security Risk

Each of the PANA/GSM threat trees presented in Figures 11.11 to 11.22 corresponds to a threat which needs to have its security risk calculated. Tables 11.1 to 11.12 summarise the corresponding threat target and threat category, the related DREAD risk metrics, and the resulting overall risk rating.

Table 11.1: **PANA/GSM threat #1**

| **Threat description** | **PAA spoofing via GSM triplet exposure** |
|---|---|
| **Threat target**<br>**Threat category** | PaC (2.0)<br>Spoofing identity (S) |
| **Risk** | Damage potential: 9<br>Reproducibility: 3<br>Exploitability: 3<br>Affected users: 6<br>Discoverability: 10<br>Overall rating: 6.2 |
| **Comments** | The target in question is the PaC (2.0), as shown in subthreats 1.1.1 and 1.1.2 (given in Figure 11.11). Reproducibility and exploitability are low because the only realistic way to exploit this threat is via a GSM network vulnerability, if the same SIM credentials are also used for GSM traffic. |

Table 11.2: **PANA/GSM threat #2**

| **Threat description** | **Permanent GSM user identifier disclosure** |
|---|---|
| **Threat target**<br>**Threat category** | PANA answers (2.0—6.0—1.0)<br>Information disclosure (I) |
| **Risk** | Damage potential: 5<br>Reproducibility: 9<br>Exploitability: 8<br>Affected users: 1<br>Discoverability: 10<br>Overall rating: 6.6 |
| **Comments** | The most likely attack would be from a rogue user using a network protocol analyser (see subthreat 2.1 of Figure 11.12), which is cheaper (in terms of time, effort, and money) than adopting an active attack (see subthreats 2.2 and 2.3 of Figure 11.12). After that, the attacker might wait for the PaC to start a first connection with a given PAA to reveal its identity. |

Table 11.3: **PANA/GSM threat #3**

| **Threat description** | **PANA/GSM session key disclosure** |
|---|---|
| **Threat target**<br>**Threat category** | PANA requests (1.0—6.0—2.0) and answers (2.0—6.0—1.0)<br>Information disclosure (I) |
| **Risk** | Damage potential: 10<br>Reproducibility: 3<br>Exploitability: 3<br>Affected users: 10<br>Discoverability: 10<br>Overall rating: 7.2 |
| **Comments** | Reproducibility and exploitability are low as long as the cryptographic functions used are sufficiently robust (see subthreat 3.3 of Figure 11.13). |

Table 11.4: **PANA/GSM threat #4**

| **Threat description** | **MitM attacks against PANA/GSM** |
|---|---|
| **Threat target**<br>**Threat category** | PAA (1.0) and PaC (2.0)<br>Spoofing identity (S) |
| **Risk** | Damage potential: 7<br>Reproducibility: 10<br>Exploitability: 7<br>Affected users: 2<br>Discoverability: 10<br>Overall rating: 7.2 |
| **Comments** | Care has to be taken to avoid this threat arising when tunnelling is used with PANA/GSM, or when EAP-SIM is part of a sequence of EAP methods. Asokan, Niemi and Nyberg [21] suggest the use of a cryptographic binding (see subthreat 4.1 of Figure 11.14), thereby reducing both reproducibility and exploitability. |

Table 11.5: **PANA/GSM threat #5**

| **Threat description** | **Service theft attacks against PANA/GSM** |
|---|---|
| **Threat target**<br>**Threat category** | PANA/GSM authentication process (6.0)<br>Elevation of privilege (E) |
| **Risk** | Damage potential: 10<br>Reproducibility: 5<br>Exploitability: 5<br>Affected users: 1<br>Discoverability: 10<br>Overall rating: 6.2 |
| **Comments** | An unprivileged rogue user can gain network access because PANA/GSM does not specify a mechanism for preventing service theft. The use of IP/MAC address filters or an IPsec SA can be adopted for this purpose. |

Table 11.6: **PANA/GSM threat #6**

| **Threat description** | **SIM credential reuse and brute-force attacks** |
|---|---|
| **Threat target**<br>**Threat category** | PaC (2.0)<br>Information disclosure (I) and potentially spoofing identity (S) |
| **Risk** | Damage potential: 8<br>Reproducibility: 6<br>Exploitability: 4<br>Affected users: 2<br>Discoverability: 10<br>Overall rating: 6.0 |
| **Comments** | The target is the PaC (2.0), once the GSM network is beyond the control of our threat analysis. A brute-force search for a 64-bit key (see subthreat 6.2.1 of Figure 11.16) is a non-trivial task that could not be executed in real time. |

Table 11.7: **PANA/GSM threat #7**

| **Threat description** | **PANA/GSM signalling traffic tampering** |
|---|---|
| **Threat target**<br>**Threat category** | PANA requests (1.0—6.0—2.0) and answers (2.0—6.0—1.0)<br>Tampering with data (T) |
| **Risk** | Damage potential: 10<br>Reproducibility: 5<br>Exploitability: 5<br>Affected users: 8<br>Discoverability: 10<br>Overall rating: 7.6 |
| **Comments** | The attacks derived from this threat can be prevented by using a PANA/GSM SA, sequence numbers, nonces, MAC and EAP-SIM attributes. |

Table 11.8: **PANA/GSM threat #8**

| **Threat description** | **Bidding down attacks against PANA/GSM** |
|---|---|
| **Threat target**<br>**Threat category** | PaC (2.0)<br>Spoofing identity (S) |
| **Risk** | Damage potential: 5<br>Reproducibility: 10<br>Exploitability: 3<br>Affected users: 1<br>Discoverability: 10<br>Overall rating: 5.8 |
| **Comments** | Some protection against this threat can be offered by repeating the list of supported EAP methods protected with the PANA/GSM SA, in addition to the use of a distinct identifier for each authentication method needed. Full protection is provided if legitimate parties only accept the use of robust cryptographic techniques. |

Table 11.9: **PANA/GSM threat #9**

| **Threat description** | **Blind resource consumption DoS attacks against PANA/GSM** |
|---|---|
| **Threat target**<br>**Threat category** | PANA/GSM authentication process (6.0)<br>Denial of service (D) |
| **Risk** | Damage potential: 9<br>Reproducibility: 4<br>Exploitability: 4<br>Affected users: 8<br>Discoverability: 10<br>Overall rating: 7.0 |
| **Comments** | PANA/GSM sequence numbers and cookies provide protection against this threat. |

Table 11.10: **PANA/GSM threat #10**

| **Threat description** | **DoS attacks via PANA/GSM termination messages** |
|---|---|
| **Threat target**<br>**Threat category** | PANA requests (1.0—6.0—2.0) and answers (2.0—6.0—1.0)<br>Denial of service (D) |
| **Risk** | Damage potential: 7<br>Reproducibility: 4<br>Exploitability: 5<br>Affected users: 1<br>Discoverability: 10<br>Overall rating: 5.4 |
| **Comments** | All messages exchanged during the termination phase need to be protected with a PANA/GSM-based MAC, which neutralises this threat. |

Table 11.11: **PANA/GSM threat #11**

| **Threat description** | **DoS attacks using false PANA/GSM success or failure indications** |
|---|---|
| **Threat target**<br>**Threat category** | PANA requests (1.0—6.0—2.0) and answers (2.0—6.0—1.0)<br>Denial of service (D) |
| **Risk** | Damage potential: 7<br>Reproducibility: 4<br>Exploitability: 4<br>Affected users: 3<br>Discoverability: 10<br>Overall rating: 5.6 |
| **Comments** | We use PANA/GSM SA, sequence numbers and MAC to neutralise this threat. |

Table 11.12: **PANA/GSM threat #12**

| **Threat description** | **IP address depletion attacks against PANA/GSM** |
|---|---|
| **Threat target**<br>**Threat category** | PANA/GSM authentication process (6.0)<br>Denial of service (D) |
| **Risk** | Damage potential: 9<br>Reproducibility: 3<br>Exploitability: 3<br>Affected users: 9<br>Discoverability: 10<br>Overall rating: 6.8 |
| **Comments** | This threat can be prevented by deploying secure address resolution, such as the SEND mechanism (see section 6.2.1.2). |

#### 11.5.2.2 PANA/UMTS and PANA/Liberty Security Risk

Each of the PANA/UMTS and PANA/Liberty threat trees presented in Figures 11.23 to 11.32 corresponds to a threat which needs to have its security risk calculated. Tables 11.13 to 11.22 summarise the corresponding threat targets and threat categories, the related DREAD risk metrics, and the resulting overall risk ratings.

Table 11.13: **PANA/UMTS or PANA/Liberty threat #1**

| **Threat description** | **Permanent UMTS user identifier disclosure** |
|---|---|
| **Threat target**<br>**Threat category** | PANA answers (2.0—6.0—1.0)<br>Information disclosure (I) |
| **Risk** | Damage potential: 5<br>Reproducibility: 9<br>Exploitability: 8<br>Affected users: 1<br>Discoverability: 10<br>Overall rating: 6.6 |
| **Comments** | The most likely attack would be from a rogue user using a network protocol analyser (see subthreat 1.1 of Figure 11.23), which is cheaper (in terms of time, effort, and money) than adopting an active attack (see subthreats 1.2 and 1.3 of Figure 11.23). After that, the attacker might wait for the PaC to start a first connection with a given PAA to reveal its identity. |

#### 11.5.2.3 PANA/IKEv2 Security Risk

Each of the PANA/IKEv2 threat trees presented in Figures 11.33 to 11.42 corresponds to a threat which needs to have its security risk calculated. Tables 11.23 to 11.32 summarise the corresponding threat targets and threat categories, the related DREAD risk metrics, and the resulting overall risk ratings.

Table 11.14: **PANA/UMTS or PANA/Liberty threat #2**

| **Threat description** | **MitM attacks against PANA/UMTS or PANA/Liberty** |
|---|---|
| **Threat target** | PAA (1.0) and PaC (2.0) |
| **Threat category** | Spoofing identity (S) |
| **Risk** | Damage potential: 7<br>Reproducibility: 10<br>Exploitability: 7<br>Affected users: 2<br>Discoverability: 10<br>Overall rating: 7.2 |
| **Comments** | Care has to be taken to avoid this threat arising when tunnelling is used with PANA/UMTS or PANA/Liberty, or when EAP-AKA is part of a sequence of EAP methods. Asokan, Niemi and Nyberg [21] suggest the use of a cryptographic binding (see subthreat 2.1 of Figure 11.24), thereby reducing both reproducibility and exploitability. |

Table 11.15: **PANA/UMTS or PANA/Liberty threat #3**

| **Threat description** | **Service theft attacks against PANA/UMTS or PANA/Liberty** |
|---|---|
| **Threat target** | PANA/UMTS or PANA/Liberty authentication processes (6.0) |
| **Threat category** | Elevation of privilege (E) |
| **Risk** | Damage potential: 10<br>Reproducibility: 5<br>Exploitability: 5<br>Affected users: 1<br>Discoverability: 10<br>Overall rating: 6.2 |
| **Comments** | An unprivileged rogue user can gain network access because neither PANA/UMTS nor PANA/Liberty specifies a mechanism for preventing service theft. The use of IP/MAC address filters or an IPsec SA can be adopted for this purpose. |

Table 11.16: **PANA/UMTS or PANA/Liberty threat #4**

| **Threat description** | **PANA/UMTS or PANA/Liberty signalling traffic tampering** |
|---|---|
| **Threat target**<br>**Threat category** | PANA requests (1.0—6.0—2.0) and answers (2.0—6.0—1.0)<br>Tampering with data (T) |
| **Risk** | Damage potential: 10<br>Reproducibility: 1<br>Exploitability: 1<br>Affected users: 8<br>Discoverability: 10<br>Overall rating: 6.0 |
| **Comments** | The attacks derived from this threat can be prevented by using a PANA/UMTS or a PANA/Liberty SA, sequence numbers, nonces, MAC and EAP-AKA attributes (e.g. AT_ENCR_DATA, for encrypted data, and AT_AUTN, a network authentication token used for replay protection). |

Table 11.17: **PANA/UMTS or PANA/Liberty threat #5**

| **Threat description** | **Bidding down attacks against PANA/UMTS or PANA/Liberty** |
|---|---|
| **Threat target**<br>**Threat category** | PaC (2.0)<br>Spoofing identity (S) |
| **Risk** | Damage potential: 5<br>Reproducibility: 10<br>Exploitability: 3<br>Affected users: 1<br>Discoverability: 10<br>Overall rating: 5.8 |
| **Comments** | Some protection against this threat can be offered by repeating the list of supported EAP methods protected with the PANA/UMTS or PANA/Liberty SA, in addition to the use of a distinct identifier for each authentication method needed. Full protection is provided if legitimate parties only accept the use of robust cryptographic techniques. |

Table 11.18: **PANA/UMTS or PANA/Liberty threat #6**

| **Threat description** | **Blind resource consumption DoS attacks against PANA/UMTS or PANA/Liberty** |
|---|---|
| **Threat target** | PANA/UMTS or PANA/Liberty authentication processes (6.0) |
| **Threat category** | Denial of service (D) |
| **Risk** | Damage potential: 9<br>Reproducibility: 4<br>Exploitability: 4<br>Affected users: 8<br>Discoverability: 10<br>Overall rating: 7.0 |
| **Comments** | PANA/UMTS or PANA/Liberty sequence numbers and cookies provide protection against this threat. |

Table 11.19: **PANA/UMTS or PANA/Liberty threat #7**

| **Threat description** | **DoS attacks using PANA/UMTS or PANA/Liberty termination messages** |
|---|---|
| **Threat target** | PANA requests (1.0—6.0—2.0) and answers (2.0—6.0—1.0) |
| **Threat category** | Denial of service (D) |
| **Risk** | Damage potential: 7<br>Reproducibility: 4<br>Exploitability: 5<br>Affected users: 1<br>Discoverability: 10<br>Overall rating: 5.4 |
| **Comments** | All messages exchanged during the termination phase need to be protected with a PANA/UMTS-based or a PANA/Liberty-based MAC, which neutralises this threat. |

Table 11.20: **PANA/UMTS or PANA/Liberty threat #8**

| **Threat description** | **DoS attacks via false PANA/UMTS or PANA/Liberty success/failure indications** |
|---|---|
| **Threat target** | PANA requests (1.0—6.0—2.0) and answers (2.0—6.0—1.0) |
| **Threat category** | Denial of service (D) |
| **Risk** | Damage potential: 7<br>Reproducibility: 4<br>Exploitability: 4<br>Affected users: 3<br>Discoverability: 10<br>Overall rating: 5.6 |
| **Comments** | We use PANA/UMTS or PANA/Liberty SA, sequence numbers and MAC to neutralise this threat. |

Table 11.21: **PANA/UMTS or PANA/Liberty threat #9**

| **Threat description** | **IP address depletion attacks against PANA/UMTS or PANA/Liberty** |
|---|---|
| **Threat target** | PANA/UMTS or PANA/Liberty authentication processes (6.0) |
| **Threat category** | Denial of service (D) |
| **Risk** | Damage potential: 9<br>Reproducibility: 3<br>Exploitability: 3<br>Affected users: 9<br>Discoverability: 10<br>Overall rating: 6.8 |
| **Comments** | This threat can be prevented by deploying secure address resolution, such as the SEND mechanism (see section 6.2.1.2). |

Table 11.22: **PANA/UMTS or PANA/Liberty threat #10**

| **Threat description** | **Brute-force attacks against PANA/UMTS or PANA/Liberty keying material** |
|---|---|
| **Threat target** | PANA requests (1.0—6.0—2.0), answers (2.0—6.0—1.0), and authentication process (6.0) |
| **Threat category** | Information disclosure (I) |
| **Risk** | Damage potential: 8<br>Reproducibility: 2<br>Exploitability: 2<br>Affected users: 2<br>Discoverability: 10<br>Overall rating: 4.8 |
| **Comments** | The use of a PANA/UMTS or PANA/Liberty SA, MAC and EAP-AKA attributes, in addition to the use of a pseudo-random function whose output is greater than 128 bits, and also the use of a good source of randomness, contribute to reduce both reproducibility and exploitability. |

Table 11.23: **PANA/IKEv2 threat #1**

| **Threat description** | **MitM attacks against PANA/IKEv2** |
|---|---|
| **Threat target** | PAA (1.0) and PaC (2.0) |
| **Threat category** | Spoofing identity (S) |
| **Risk** | Damage potential: 7<br>Reproducibility: 10<br>Exploitability: 7<br>Affected users: 2<br>Discoverability: 10<br>Overall rating: 7.2 |
| **Comments** | Care has to be taken to avoid this threat arising when tunnelling is used with PANA/IKEv2, or when EAP-IKEv2 is part of a sequence of EAP methods. Asokan, Niemi and Nyberg [21] suggest the use of a cryptographic binding (see subthreat 1.1 of Figure 11.33), thereby reducing both reproducibility and exploitability. |

Table 11.24: **PANA/IKEv2 threat #2**

| **Threat description** | **PANA/IKEv2 user identifier disclosure** |
|---|---|
| **Threat target** | PANA answers (2.0—6.0—1.0) |
| **Threat category** | Information disclosure (I) |
| **Risk** | Damage potential: 1<br>Reproducibility: 1<br>Exploitability: 1<br>Affected users: 1<br>Discoverability: 10<br>Overall rating: 2.8 |
| **Comments** | The PaC and PAA identifiers are encrypted and integrity protected by PANA/IKEv2, which prevents eavesdropping (see subthreat 2.1 of Figure 11.34). In addition, the identifier used in the first round trip does not allow to learn the PaC's identity (see subthreats 2.2 and 2.3 of Figure 11.34). |

Table 11.25: **PANA/IKEv2 threat #3**

| **Threat description** | **Service theft attacks against PANA/IKEv2** |
|---|---|
| **Threat target** | PANA/IKEv2 authentication process (6.0) |
| **Threat category** | Elevation of privilege (E) |
| **Risk** | Damage potential: 10<br>Reproducibility: 5<br>Exploitability: 5<br>Affected users: 1<br>Discoverability: 10<br>Overall rating: 6.2 |
| **Comments** | An unprivileged rogue user can gain network access because PANA/IKEv2 does not specify a mechanism for preventing service theft. The use of IP/MAC address filters or an IPsec SA can be adopted for this purpose. |

Table 11.26: **PANA/IKEv2 threat #4**

| **Threat description** | **Brute-force attacks against PANA/IKEv2 keying material** |
|---|---|
| **Threat target** | PANA/IKEv2 requests (1.0—6.0—2.0), answers (2.0—6.0—1.0), and authentication process (6.0) |
| **Threat category** | Information disclosure (I) |
| **Risk** | Damage potential: 8<br>Reproducibility: 1<br>Exploitability: 1<br>Affected users: 2<br>Discoverability: 10<br>Overall rating: 4.4 |
| **Comments** | The use by PANA/IKEv2 of an ephemeral Diffie-Hellman exchange with the 'perfect forward secrecy' property (see section 2.1.3.3), in addition to the use of a prf whose output is equal or greater than 128 bits, and also the use of a good source of randomness, contribute to reduce both the reproducibility and exploitability values. |

Table 11.27: **PANA/IKEv2 threat #5**

| **Threat description** | **PANA/IKEv2 signalling traffic tampering** |
|---|---|
| **Threat target** | PANA requests (1.0—6.0—2.0) and answers (2.0—6.0—1.0) |
| **Threat category** | Tampering with data (T) |
| **Risk** | Damage potential: 10<br>Reproducibility: 1<br>Exploitability: 1<br>Affected users: 8<br>Discoverability: 10<br>Overall rating: 6.0 |
| **Comments** | The attacks derived from this threat can be prevented by using a PANA/IKEv2 SA, sequence numbers, nonces, MAC and EAP-IKEv2 attributes (e.g. the *AUTH* payload). |

Table 11.28: **PANA/IKEv2 threat #6**

| **Threat description** | **Bidding down attacks against PANA/IKEv2** |
|---|---|
| **Threat target** | PaC (2.0) |
| **Threat category** | Spoofing identity (S) |
| **Risk** | Damage potential: 5<br>Reproducibility: 10<br>Exploitability: 3<br>Affected users: 1<br>Discoverability: 10<br>Overall rating: 5.8 |
| **Comments** | Some protection against this threat can be offered by repeating the list of supported EAP methods protected with the PANA/IKEv2 SA, in addition to the use of a distinct identifier for each authentication method needed. Full protection is provided if legitimate parties only accept the use of robust cryptographic techniques. |

Table 11.29: **PANA/IKEv2 threat #7**

| **Threat description** | **Blind resource consumption DoS attacks against PANA/IKEv2** |
|---|---|
| **Threat target** | PANA/IKEv2 authentication process (6.0) |
| **Threat category** | Denial of service (D) |
| **Risk** | Damage potential: 9<br>Reproducibility: 4<br>Exploitability: 4<br>Affected users: 8<br>Discoverability: 10<br>Overall rating: 7.0 |
| **Comments** | PANA/IKEv2 sequence numbers and cookies provide protection against this threat. |

Table 11.30: **PANA/IKEv2 threat #8**

| **Threat description** | **DoS attacks using PANA/IKEv2 termination messages** |
|---|---|
| **Threat target** | PANA requests (1.0—6.0—2.0) and answers (2.0—6.0—1.0) |
| **Threat category** | Denial of service (D) |
| **Risk** | Damage potential: 7<br>Reproducibility: 4<br>Exploitability: 5<br>Affected users: 1<br>Discoverability: 10<br>Overall rating: 5.4 |
| **Comments** | All messages exchanged during the termination phase need to be protected with a PANA/IKEv2-based MAC, which neutralises this threat. |

Table 11.31: **PANA/IKEv2 threat #9**

| **Threat description** | **DoS attacks using false PANA/IKEv2 success or failure indications** |
|---|---|
| **Threat target** | PANA requests (1.0—6.0—2.0) and answers (2.0—6.0—1.0) |
| **Threat category** | Denial of service (D) |
| **Risk** | Damage potential: 7<br>Reproducibility: 4<br>Exploitability: 4<br>Affected users: 3<br>Discoverability: 10<br>Overall rating: 5.6 |
| **Comments** | We use PANA/IKEv2 SA, sequence numbers and MAC to neutralise this threat. |

Table 11.32: **PANA/IKEv2 threat #10**

| **Threat description** | **IP address depletion attacks against PANA/IKEv2** |
|---|---|
| **Threat target** | PANA/IKEv2 authentication process (6.0) |
| **Threat category** | Denial of service (D) |
| **Risk** | Damage potential: 9<br>Reproducibility: 3<br>Exploitability: 3<br>Affected users: 9<br>Discoverability: 10<br>Overall rating: 6.8 |
| **Comments** | This threat can be prevented by deploying secure address resolution, such as the SEND mechanism (see section 6.2.1.2). |

#### 11.5.2.4 Sorting the Threats

Once we have calculated the overall risk rating of each threat, we can then sort the threats to each of the proposed authentication techniques in descending order (i.e. threats with a higher security risk first and lower-risk threats last). Tables 11.33 to 11.35 summarise and provide an average of the DREAD overall security risk ratings, which can be used to rank the threats applying to each of the entity authentication proposals.

Table 11.33: **Ranking the PANA/GSM threats by decreasing risk**

| Rank | Threat | Threat description | Risk |
|---|---|---|---|
| 1 | 7 | PANA/GSM signalling traffic tampering. | 7.6 |
| 2 | 3 | PANA/GSM session key disclosure. | 7.2 |
| 3 | 4 | MitM attacks against PANA/GSM. | 7.2 |
| 4 | 9 | Blind resource consumption DoS attacks against PANA/GSM. | 7.0 |
| 5 | 12 | IP address depletion attacks against PANA/GSM. | 6.8 |
| 6 | 2 | Permanent GSM user identifier disclosure. | 6.6 |
| 7 | 1 | PAA spoofing via GSM triplet exposure. | 6.2 |
| 8 | 5 | Service theft attacks against PANA/GSM. | 6.2 |
| 9 | 6 | SIM credential reuse and brute-force attacks. | 6.0 |
| 10 | 8 | Bidding down attacks against PANA/GSM. | 5.8 |
| 11 | 11 | DoS attacks via false PANA/GSM success/failure indications. | 5.6 |
| 12 | 10 | DoS attacks via PANA/GSM termination messages. | 5.4 |
| # | # | **Overall risk rating** ..... | **6.47** |

## 11.6 Mitigating the Threats

According to Howard and LeBlanc [81, p106], after determining the security risk for each of the captured threats in the previous section, the final step of the formal threat modelling process is to determine how to deal with them. We have four options when considering threats and how to mitigate them [81, p106-107]:

- doing nothing (which is rarely the correct solution);

Table 11.34: **Ranking the PANA/UMTS and PANA/Liberty threats by decreasing risk**

| Rank | Threat | Threat description | Risk |
|---|---|---|---|
| 1 | 2 | MitM attacks against PANA/UMTS or PANA/Liberty. | 7.2 |
| 2 | 6 | Blind resource consumption DoS attacks against PANA/UMTS or PANA/Liberty. | 7.0 |
| 3 | 9 | IP address depletion attacks against PANA/UMTS or PANA/Liberty. | 6.8 |
| 4 | 1 | Permanent UMTS user identifier disclosure. | 6.6 |
| 5 | 3 | Service theft attacks against PANA/UMTS or PANA/Liberty. | 6.2 |
| 6 | 4 | PANA/UMTS or PANA/Liberty signalling traffic tampering. | 6.0 |
| 7 | 5 | Bidding down attacks against PANA/UMTS or PANA/Liberty. | 5.8 |
| 8 | 8 | DoS attacks via false PANA/UMTS or PANA/Liberty success/failure indications. | 5.6 |
| 9 | 7 | DoS attacks via PANA/UMTS or PANA/Liberty termination messages. | 5.4 |
| 10 | 10 | Brute-force attacks against PANA/UMTS or PANA/Liberty keying material. | 4.8 |
| # | # | **Overall risk rating** ..... | **6.14** |

- warning the user of the problem, thereby allowing the user to decide whether to use the feature which is the focus of the threat;
- removing the problem, by removing the feature giving rise to the threat from the protocol; or
- fixing the problem, by using security techniques (see Chapter 2) — our chosen option.

In this section, a description of how to select the appropriate techniques to mitigate the threats is first provided (section 11.6.1). We then map these threat mitigation techniques to each of the entity authentication proposals, thereby deducing their mitigation status (section 11.6.2).

Table 11.35: **Ranking the PANA/IKEv2 threats by decreasing risk**

| Rank | Threat | Threat description | Risk |
|---|---|---|---|
| 1 | 1 | MitM attacks against PANA/IKEv2. | 7.2 |
| 2 | 7 | Blind resource consumption DoS attacks against PANA/IKEv2. | 7.0 |
| 3 | 10 | IP address depletion attacks against PANA/IKEv2. | 6.8 |
| 4 | 3 | Service theft attacks against PANA/IKEv2. | 6.2 |
| 5 | 5 | PANA/IKEv2 signalling traffic tampering. | 6.0 |
| 6 | 6 | Bidding down attacks against PANA/IKEv2. | 5.8 |
| 7 | 9 | DoS attacks via false PANA/IKEv2 success/failure indications. | 5.6 |
| 8 | 8 | DoS attacks using PANA/IKEv2 termination messages. | 5.4 |
| 9 | 4 | Brute-force attacks against PANA/IKEv2 keying material. | 4.4 |
| 10 | 2 | PANA/IKEv2 user identifier disclosure. | 2.8 |
| # | # | **Overall risk rating** ..... | **5.72** |

### 11.6.1 Mitigation Techniques

The first step in determining how to address the threats we have identified involves choosing the appropriate mitigation methods. As suggested in [81, p107], we again use the concept of STRIDE (see section 11.4.1). Table 11.36 lists some of the security techniques[21] that we could employ to mitigate the threats, classified according to the STRIDE model.

### 11.6.2 Mitigation Status

As described in section 11.4.2, we have placed *mitigation circles* below the least likely nodes in the threat trees presented in Figures 11.11 to 11.42. These circles already indicate how the threats are mitigated or, in other words, which *mitigation technique* is employed. As discussed in [81, p91], the mitigation circles have been added later, after the threat modelling process.

Additionally, Tables 11.1 to 11.32, which summarise the STRIDE category

[21] Many of these security techniques are discussed in Chapter 2, whilst others are described in [81, p108-118].

Table 11.36: **Partial list of threat mitigation techniques**

| **Threat type** | **Mitigation techniques** |
|---|---|
| Spoofing identity (S) | Appropriate authentication mechanisms.<br>Protect secret data.<br>Do not store secrets. |
| Tampering with data (T) | Appropriate authorisation mechanisms (e.g. access control lists, privileges, IP restrictions, and permissions).<br>Cryptographic hash functions.<br>Message authentication codes (MACs).<br>Digital signatures.<br>Tamper-resistant mechanisms (e.g. use of subscriber's smart cards, in the form of GSM SIMs or UMTS USIMs). |
| Repudiation (R) | Digital signatures.<br>Timestamps.<br>Audit trails. |
| Information disclosure (I) | Appropriate authorisation mechanisms.<br>Privacy-enhancing techniques (PETs — a special class of cryptographic protocols designed to enhance user privacy, e.g. by supporting anonymity).<br>Encryption mechanisms.<br>Protect secrets.<br>Do not store secrets. |
| Denial of service (D) | Appropriate authentication mechanisms.<br>Appropriate authorisation mechanisms.<br>Filtering (i.e. inspecting received data and making a decision to accept or reject the packet).<br>Throttling (i.e. limiting the number of requests to the protocol).<br>Quality of service (i.e. a set of components which allow the provision of preferential treatment for specific types of traffic). |
| Elevation of privilege (E) | Run with least privilege (i.e. always run with just enough privilege to get the job done, and no more). |

and the related DREAD risk metrics for each of the captured threats, also include a commentary field (called *Comments* — see section 11.5.2). This field contains complementary information that indicates the mitigation techniques adopted to neutralise the threats.

Building on the above observations, and after analysing Tables 11.33 to 11.35, which rank the threats by decreasing risk, we can deduce the *mitigation status*[22] of the threats. Tables 11.37 to 11.39 summarise the mitigation status

[22]As noted by Howard and LeBlanc [81, p92], the *mitigation status* consists of the answer to the question: 'has the threat been mitigated'? Valid entries are: 'Yes', 'No', 'Somewhat', and 'Needs Investigating'.

of the threats applying to each of the four proposed protocols.

Table 11.37: **Mitigation status of the PANA/GSM threats**

| # | Threat | STRIDE | Mitigation status |
|---|---|---|---|
| 7 | PANA/GSM signalling traffic tampering. | T | Yes |
| 3 | PANA/GSM session key disclosure. | I | Yes |
| 4 | MitM attacks against PANA/GSM. | S | Yes |
| 9 | Blind resource consumption DoS attacks against PANA/GSM. | D | Somewhat |
| 12 | IP address depletion attacks against PANA/GSM. | D | Yes |
| 2 | Permanent GSM user identifier disclosure. | I | Somewhat |
| 1 | PAA spoofing via GSM triplet exposure. | S | Somewhat |
| 5 | Service theft attacks against PANA/GSM. | E | Yes |
| 6 | SIM credential reuse and brute-force attacks. | I & S | Somewhat |
| 8 | Bidding down attacks against PANA/GSM. | S | Yes |
| 11 | DoS attacks via false PANA/GSM success/failure indications. | D | Yes |
| 10 | DoS attacks via PANA/GSM termination messages. | D | Yes |

## 11.7 Comparative Analysis

The aim of this section is to assess the four novel Internet authentication schemes proposed in Chapters 7, 8, 9, and 10 as candidates for a secure access procedure for heterogeneous network access supporting ubiquitous mobility. In particular, we make a comparative analysis of the services and properties possessed by the new authentication techniques, using as a benchmark the required Internet remote access services and properties established in Chapter 5 (mainly in section 5.3).

These latter services and properties are derived from two main requirement sets, namely *security* requirements (discussed in section 5.1) and *implementation* requirements (given in section 5.2). These requirement sets are used in conjunction with the results of the formal threat model (see sections 11.2 to 11.6) to provide a sound basis for the assessment of the candidate protocols.

Table 11.38: **Mitigation status of the PANA/UMTS and PANA/Liberty threats**

| # | Threat | STRIDE | Mitigation status |
|---|---|---|---|
| 2 | MitM attacks against PANA/UMTS or PANA/Liberty. | S | Yes |
| 6 | Blind resource consumption DoS attacks against PANA/UMTS or PANA/Liberty. | D | Somewhat |
| 9 | IP address depletion attacks against PANA/UMTS or PANA/Liberty. | D | Yes |
| 1 | Permanent UMTS user identifier disclosure. | I | Somewhat |
| 3 | Service theft attacks against PANA/UMTS or PANA/Liberty. | E | Yes |
| 4 | PANA/UMTS or PANA/Liberty signalling traffic tampering. | T | Yes |
| 5 | Bidding down attacks against PANA/UMTS or PANA/Liberty. | S | Yes |
| 8 | DoS attacks via false PANA/UMTS or PANA/Liberty success/failure indications. | D | Yes |
| 7 | DoS attacks via PANA/UMTS or PANA/Liberty termination messages. | D | Yes |
| 10 | Brute-force attacks against PANA/UMTS or PANA/Liberty keying material. | I | Yes |

Firstly, in order to evaluate the four protocols against the security requirements, we analyse and compare a number of aspects of entity authentication security for Internet remote access (section 11.7.1). Secondly, to assess the protocols against the implementation requirements, we analyse and compare features such as complexity, flexibility and performance (section 11.7.2). Thirdly, we assess the candidate protocols against the results of the formal threat model (section 11.7.3).

The results of the critical analyses made in sections 11.7.1 to 11.7.3 are then used to give an overall assessment of the four authentication techniques against the services and properties required of new authentication methods for Internet access (section 11.7.4).

Table 11.39: **Mitigation status of the PANA/IKEv2 threats**

| # | Threat | STRIDE | Mitigation status |
|---|---|---|---|
| 1 | MitM attacks against PANA/IKEv2. | S | Yes |
| 7 | Blind resource consumption DoS attacks against PANA/IKEv2. | D | Somewhat |
| 10 | IP address depletion attacks against PANA/IKEv2. | D | Yes |
| 3 | Service theft attacks against PANA/IKEv2. | E | Yes |
| 5 | PANA/IKEv2 signalling traffic tampering. | T | Yes |
| 6 | Bidding down attacks against PANA/IKEv2. | S | Yes |
| 9 | DoS attacks via false PANA/IKEv2 success/failure indications. | D | Yes |
| 8 | DoS attacks using PANA/IKEv2 termination messages. | D | Yes |
| 4 | Brute-force attacks against PANA/IKEv2 keying material. | I | Yes |
| 2 | PANA/IKEv2 user identifier disclosure. | I | Yes |

### 11.7.1 Security Assessment

We first analyse and compare the four authentication schemes against the security requirements described in section 5.1, examining a number of aspects of entity authentication security for Internet remote access. We now consider the security requirements one at a time.

#### 11.7.1.1 Client Authentication

As stated in section 5.1.1, we focus here purely on the authentication and key establishment processes, and not on subsequent use made of the authenticated channel and/or keys that may have been established. After examining the four authentication techniques proposed in this thesis, we observe that they all equally enable authentication of the client (i.e. the remote device) to the access network[23].

[23]In each of the four authentication protocols, a client identifier can be authenticated by verifying the credentials supplied by one of the users of the device, or by the device itself.

#### 11.7.1.2 Key Establishment

As discussed in section 5.1.2, a key establishment facility enables network remote access authentication schemes to be linked to an integrity service, to provide ongoing data origin authentication and integrity. To achieve this, the entity authentication protocol needs to be integrated with a key establishment mechanism, such that a by-product of successful authentication is a session key.

The four proposed authentication techniques all provide this facility, generating a shared secret session key called *MSK*. In every case, this session key is provided to the PaC as part of the EAP key exchange process; in each such process the PAA can obtain the session key from the EAP server via the AAA infrastructure.

#### 11.7.1.3 Use of EAP Methods

As discussed in section 5.1.3, since the EAP protocol is very flexible and can encapsulate arbitrary authentication methods, it is clearly a protocol that satisfies many of the requirements for a variety of authentication scenarios. As a result, all the Internet remote access authentication schemes proposed in this thesis make use of a tunnelled authentication mechanism carrying EAP.

#### 11.7.1.4 Mutual Entity Authentication

The authentication client and the network may be able to perform mutual authentication in some Internet remote access schemes. Indeed, only providing the capability for the network to authenticate the client may not always be sufficient[24].

[24] As discussed in section 5.1.4, and following [136], we claim that (mutual) entity authentication is not always an essential precursor for the establishment of secure communications. In some cases, the most important issue is to ensure that the properties of (implicit) key authentication and key freshness are provided for any established session keys.

In all four of the authentication techniques proposed in this thesis, the PaC and the PAA are able to perform mutual authentication.

#### 11.7.1.5 Key Freshness

As discussed in section 5.1.5, key freshness is a property useful in many applications in the Internet remote access environment. The absence of key freshness would enable an interceptor to force the verifier to keep re-using an 'old' session key, which might have been compromised. The key establishment processes of all four of the authentication protocols provide the key freshness requirement.

#### 11.7.1.6 Re-Authentication

As explained in section 5.1.6, authentication protocols provide assurance regarding the identity of an entity *only* at a given instant in time. Thus the authenticity of the entity can be ascertained just for the instance of the authentication exchange. If continuity of such an assurance is required, use of additional techniques is necessary. For example, authentication can be repeated periodically.

The four entity authentication schemes defined in this thesis are capable of supporting both periodic and on-demand re-authentication.

#### 11.7.1.7 Authorisation, Access Control, and Accounting

As discussed in section 5.1.7, after a PaC is authenticated by Internet remote access methods, it will be authorised for network access. While a backend authorisation infrastructure, e.g. RADIUS (see section 3.9.1) or Diameter (see section 3.9.2), might provide the necessary authorisation information to the access network, explicit support for authorisation functionality is outside the

scope of this thesis. Therefore, in assessing the four authentication schemes, we do not consider the possible need for the access network to provide service authorisation information to the authenticated PaC.

PaC remote access authentication should be followed by access control, to make sure only authenticated and authorised clients can send and receive IP packets via the access network. Although the four authentication schemes identify PaCs that are authorised to access the network, providing access control functionality in the network is outside the scope of this thesis.

Finally, as previously stated, issues associated with the transfer and management of accounting data are also outside the scope of this thesis.

#### 11.7.1.8 AAA Backend

The four Internet remote access protocols support interaction with a backend AAA infrastructure (i.e. Diameter EAP — see section 3.9.3), but such an interaction is not a requirement for their correct operation. If the access network does not rely on a specific AAA protocol, e.g. Diameter (see section 3.9.2) or RADIUS (see section 3.9.1), then the protocols use a proprietary backend system, or rely on locally stored information.

The details of the interaction between the access network and the backend authentication entities are outside the scope of this thesis.

#### 11.7.1.9 Secure Channel

None of the four authentication techniques assume the presence of a secure channel between the PaC and the PAA (see section 5.1.9). Indeed, as noted in section 11.4.2, the four schemes are able to provide a secure authentication service in networks which are not protected against packet eavesdropping and

spoofing. They also provide protection against replay attacks on both the PaC and the PAA.

#### 11.7.1.10 Denial-of-Service Attacks

The four entity authentication schemes are designed to be robust against Denial-of-Service attacks[25], in particular against those DoS category threats captured in section 11.4.2.

#### 11.7.1.11 Client Identity Confidentiality

As explained in section 5.1.11, some remote clients might prefer to hide their identity from visited access networks for privacy reasons. All four authentication schemes provide identity confidentiality for remote clients.

In particular, the PANA/IKEv2 scheme protects the privacy of the PaC and PAA identities against disclosure threats involving both *passive* and *active* attackers. However, in the other three authentication techniques, an active attacker that impersonates a given PAA in a first connection may learn the subscriber's permanent identifier.

However, in these latter authentication protocols, the PaC can refuse to send the cleartext permanent user identifier to the PAA if it believes that the visited access network should be able to recognise its pseudonym (as discussed in section 11.4.2).

### 11.7.2 Implementation Assessment

We now analyse and compare the four authentication schemes against the implementation requirements described in section 5.2, examining a number of im-

[25] As stated in section 5.1.10, such attacks could prevent network access by legitimate PaCs.

plementation features of such protocols.

#### 11.7.2.1 Client Identifiers

Since the four entity authentication schemes all use PANA as the target transportation environment (see Chapter 6), they can support a variety of client identifier types (e.g. username, Network Access Identifier, etc.), as well as a variety of remote device identifier types (e.g. IP address, link-layer address, port number of a switch, etc.), in addition to a binding between the client identifier and the associated device identifier upon successful authentication[26].

#### 11.7.2.2 IP Address Assignment

As discussed in section 5.2.2, assigning an IP address to the client of the authentication schemes is outside the scope of this thesis. We simply note here that the PaC configures an IP address before running each of the proposed entity authentication methods.

#### 11.7.2.3 EAP Lower Layer Requirements

As stated in section 5.2.3, EAP imposes many requirements on the underlying transport protocol that must be satisfied if EAP is to operate correctly. RFC 3748 [13] describes the generic transport requirements satisfied by the four schemes proposed in this thesis, since all of them make use of EAP.

[26]In order to prevent unauthorised access, all four authentication techniques support the cryptographic protection of the device identifier. The keying material required for this service needs to be indexed by the device identifier (see section 5.2.1).

#### 11.7.2.4 Flexibility

Since all four entity authentication schemes use PANA as the target transportation environment (see Chapter 6), they can support client devices with multiple interfaces, and access networks with multiple routers on multi-access links (as detailed in section 5.2.4).

In particular, the PANA/IKEv2 scheme provides flexibility through the public key based authentication option, while the PANA/Liberty technique supports flexibility by incorporating the Liberty Alliance framework and mechanisms of the GAA architecture.

#### 11.7.2.5 Performance

All four of the protocols handle the authentication process efficiently in order to gain network access with minimum latency. For example, since they all use PANA as the target transportation environment, they all have the ability to minimise the protocol signalling by creating local security associations. Also the schemes make use of two types of fast re-authentication, which contributes to potential gains in performance.

#### 11.7.2.6 Complexity

As discussed in section 5.2.6, in a number of situations it is highly desirable to minimise the number of round trips needed by the entity authentication procedure.

By using EAP to carry lightweight authentication methods, it is possible to create authentication solutions with low complexity at the application layer. This is particularly true for PANA/GSM, PANA/UMTS and PANA/Liberty, through the EAP encapsulation of lightweight authentication protocols used in

existing mobile telecommunications systems.

#### 11.7.2.7 IP Version Independence

Since the four authentication schemes use PANA as the target transportation environment (see Chapter 6), they can work with both IPv4 and IPv6.

### 11.7.3 Assessment using Threat Model Results

In this section, the main results of the formal threat modelling process (see sections 11.2 to 11.6) are combined to give an assessment of the candidate protocols.

Table 11.40 summarises for each protocol the overall security risk ratings (calculated in Tables 11.33 to 11.35) and the threat mitigation status (i.e. the number of threats which have been mitigated, and the number that have not been fully mitigated)[27].

Table 11.40: **Threat model main results**

| # | Authentication scheme | Overall risk rating | Threats mitigated (# Yes) | Threats not fully mitigated (# Somewhat) |
|---|---|---|---|---|
| 1 | PANA/IKEv2 | 5.72 | 9 | 1 |
| 2 | PANA/Liberty | 6.14 | 8 | 2 |
| 3 | PANA/UMTS | 6.14 | 8 | 2 |
| 4 | PANA/GSM | 6.47 | 8 | 4 |

[27]Indicated by the number of answers equal to or different of 'Yes', in the mitigation status entries in Tables 11.37 to 11.39.

### 11.7.4 Services and Properties Assessment

In this section, the results of the critical analyses made in sections 11.7.1 to 11.7.2 are first used to provide a combined security and implementation assessment of the four protocols (as defined in section 5.3). This combined assessment is then used in conjunction with the threat model results (given in the previous section) to provide an overall assessment of the proposed authentication schemes.

Table 11.41: **Security assessment**

| **Security requirements** | **Security services and properties of the candidate protocols** |
|---|---|
| Entity authentication service for remote network access. | Service provided by the four candidate protocols (section 11.7.1.1). |
| Key establishment services with key freshness property. | Services provided by the four candidate protocols (sections 11.7.1.2 and 11.7.1.5). |
| Use of a tunnelled authentication mechanism carrying EAP. | Property provided by the four candidate protocols (section 11.7.1.3). |
| Mutual authentication services between the remote client and the access network. | Services provided by the four candidate protocols (section 11.7.1.4). |
| Use of periodic and on-demand reauthentication techniques. | Service provided by the four candidate protocols (section 11.7.1.6). |
| Possible interaction between the network and AAA infrastructures. | Property provided by the four candidate protocols (sections 11.7.1.7 and 11.7.1.8). |
| Absence of vulnerabilities that can be exploited over insecure channels. | Property possible in the four candidate protocols (section 11.7.1.9). |
| Robustness against DoS attacks. | Property provided by the four candidate protocols (section 11.7.1.10). |
| Identity confidentiality service for remote clients. | Service possible in the four candidate protocols (particularly robust in PANA/IKEv2; see section 11.7.1.11). |

As shown in Tables 11.41 and 11.42, the four candidate protocols have the potential to meet all the identified requirements. In particular, the PANA/IKEv2 protocol provides a robust identity confidentiality service for remote clients (as given in section 11.7.1.11). On the other hand, PANA/GSM, PANA/UMTS, and PANA/Liberty possess low complexity in the authentication method (see section 11.7.2.6), while PANA/IKEv2 and PANA/Liberty provide a greater de-

gree of flexibility (as discussed in section 11.7.2.4).

Table 11.42: **Implementation assessment**

| **Implementation requirements** | **Implementation services and properties of the candidate protocols** |
|---|---|
| Support for multiple client and device identifiers. | Service provided by the four candidate protocols (section 11.7.2.1). |
| Satisfaction of the EAP transport requirements. | Property provided by the four candidate protocols (section 11.7.2.3). |
| Flexibility. | Property provided by the four candidate protocols (enhanced in PANA/IKEv2 and PANA/Liberty; see section 11.7.2.4). |
| Performance. | Property provided by the four candidate protocols (section 11.7.2.5). |
| Low complexity. | Property provided by the four candidate protocols (mainly by PANA/GSM, PANA/UMTS and PANA/Liberty; see section 11.7.2.6). |
| IP version independence. | Property provided by the four candidate protocols (section 11.7.2.7). |

Building on the above observations, and after analysing Table 11.40 which summarises the overall security risk ratings and threat mitigation status for each of the candidate protocols, it appears that the PANA/IKEv2 scheme is the most secure (with a risk rating of 5.72 and just one of the ten threats not fully mitigated), flexible and scalable method. We suggest further that PANA/Liberty can be classified in second position in our comparative analysis, if we consider its threat model results (it possesses a risk rating of 6.14, and two out of ten threats are not fully mitigated), as well as its low complexity and a degree of flexibility.

PANA/UMTS occupies third place in our assessment, because of both the threat model results (with a risk rating of 6.14 and two out of ten threats not fully mitigated) and the low complexity of the authentication method. PANA/GSM comes last, because of its threat model results (a risk rating of 6.47 and four out of twelve threats not fully mitigated).

Finally, it is important to note that the choice of PANA as the target trans-

portation environment for the four authentication protocols contributes significantly to the positive assessments of the four schemes in meeting all the identified security and implementation requirements (as discussed in sections 11.7.1 and 11.7.2).

## 11.8 Conclusions

As previously discussed, this thesis proposes, evaluates and compares new entity authentication protocols for Internet remote access. The main challenges addressed here include the investigation, development, and assessment of unified, secure and convenient authentication mechanisms that can be used in access networks of a wide range of types. The primary goal of this chapter has thus been to discover which of them is the most secure, lightweight, flexible and scalable Internet authentication method.

In this chapter, we have adopted a *formal threat model*, i.e. a security-based analysis that is used to determine the highest level security risks posed to an application, and how attacks can manifest themselves. The security analysis of the proposed authentication protocols has been performed using the threat modelling process described in Chapter 4 of Howard and LeBlanc [81, p69-124], the steps in which can be summarised as follows:

- formally decompose the protocols;
- determine the threats;
- rank the threats by decreasing risk; and
- employ mitigation techniques.

The main aim of performing this security-based analysis is to determine which threats to the new authentication techniques require mitigation and how

to mitigate them, reducing via a formal process of threat modelling the overall risk to the protocols to an acceptable level. We have also used this model to conduct a comparative analysis of the four authentication schemes proposed in Chapters 7, 8, 9, and 10.

Additionally, since these new techniques have been designed to meet the security and implementation services and properties required of new authentication methods for Internet access (as established in Chapter 5), we have analysed and compared the services and properties possessed by the four candidate protocols against each of those requirements. These security and implementation requirements were used in conjunction with the threat model results, to provide an overall assessment of the proposed authentication schemes.

Finally, our comparative analysis suggests that the PANA/IKEv2 technique is the best Internet authentication method of those proposed in this thesis. This is closely followed by the PANA/Liberty scheme. The PANA/UMTS and PANA/GSM protocols were ranked third and fourth.

# Chapter 12

# Conclusions

## Contents



The aim of this chapter is to summarise all the work that has been discussed here, focussing in particular on the original contributions of this thesis. In addition, suggestions for future work are also provided.

## 12.1 Summary and Conclusions

This thesis deals with Internet authentication procedures for remote access. The main focus has been on the investigation, development and assessment of unified, secure and convenient authentication mechanisms that can be used in Internet access networks of a wide range of types, all supporting ubiquitous mobility. A series of new solutions has been developed by adapting and reinforcing security techniques arising from a variety of different sources.

Firstly, background on security services and cryptographic techniques, in addition to a technical overview of entity authentication, was provided. A number of properties of authentication protocols, such as temporality, implicit key authentication, and the provision of key freshness, have been identified. A general authentication model was given. We have also distinguished between different perspectives related to Internet remote access.

We then described a number of possible approaches to constructing authentication protocols, and divided initial authentication for Internet remote access into two parts. The need for a higher layer authentication procedure for Internet access was then discussed. Possible tunnelled authentication mechanisms were considered, and a wide range of potential alternatives were reviewed. We summarised some of the existing authentication protocols relevant to this thesis, including legacy processes, public key based procedures, and mobile telecommunications methods.

Secondly, the problem domain was defined, a variety of different scenarios were described, and means to assess authentication protocols against Internet remote access requirements were developed. Two main sets of requirements, namely *security* and *implementation* requirements, were specified. To establish the security requirements we analysed potential risks associated with authentication protocols, examining a number of aspects of authentication security

for Internet access. To obtain the implementation requirements we analysed features such as complexity, flexibility and performance.

The results of this critical analysis were used to specify the services and properties needed to address the threats and to achieve the implementation features required from entity authentication schemes. We then discussed the selection of the PANA protocol as the target environment for transporting the new authentication techniques.

Thirdly, we have proposed four novel Internet authentication schemes, designed to meet the established requirements. We have focused on authentication protocols that can be carried both by the IETF PANA authentication carrier and the EAP mechanisms, and that make use of an AAA infrastructure. The core idea has been to adapt authentication protocols used in existing mobile telecommunications systems to provide security mechanisms for Internet remote access. A proposal has also been given for Internet access using a public key based authentication protocol.

We have thus presented four new, IP-compatible, flexible and scalable methods for authenticating a user to an access network, summarised below:

**PANA/GSM.** This lightweight method adapts the security techniques used in the GSM authentication mechanism to the PANA framework. PANA communicates, via Diameter EAP, with an AAA infrastructure interacting with an AuC in the GSM mobile network. PANA/GSM uses the EAP-SIM protocol, which encapsulates GSM parameters in EAP and provides enhancements such as stronger authentication and key agreement as well as mutual authentication. The use of 'triplets' in PANA/GSM minimises the necessary trust relationship between operators, thereby increasing the likelihood of successful use[1].

[1]From the user perspective, the PANA/GSM protocol works with a 'standard' GSM SIM card and requires only an appropriate Internet access device and a SIM card reader.

**PANA/UMTS.** This lightweight method adapts the security techniques used in the UMTS authentication and key agreement mechanism to the PANA environment. PANA communicates, via EAP, with an AAA infrastructure interacting with an AuC in the UMTS mobile network. PANA/UMTS uses EAP-AKA, which allows use of the AKA infrastructure in network scenarios in which mobile devices are equipped with a USIM. Use of UMTS authentication vectors minimises the necessary trust relationship between operators, thereby increasing the likelihood of successful use[2].

**PANA/Liberty.** This lightweight method reuses the cellular network authentication infrastructure deployed in subscriber smart cards, and offers an open SSO standard service. This protocol is based on the PANA/UMTS scheme, and incorporates the security techniques used in the UMTS mobile network and the GAA infrastructure into the PANA structure. This scheme is complemented by the Liberty SSO service, which can be used to extend a PANA/UMTS initial authentication to all Liberty-enabled SPs, and create a network identity infrastructure supporting all network access devices.

**PANA/IKEv2.** This method, which can employ either symmetric or asymmetric techniques[3], adapts the security procedures used in the IKEv2 public key based authentication mechanism to the PANA framework. PANA communicates, via EAP, with an AAA infrastructure. PANA/IKEv2 uses EAP-IKEv2, which allows use of the IKEv2 infrastructure defined for Internet key exchange in any scenario using EAP-based authentication.

Next, the four candidate protocols detailed above were evaluated and compared. The primary goal of this evaluation was to discover which of them is the

[2]From the user perspective, the PANA/UMTS protocol works with a 'standard' UMTS USIM (or even a 'standard' CDMA2000 (R)UIM) card and requires only an appropriate Internet access device and a USIM (or (R)UIM) card reader.

[3]Whereas the former requires the involvement of the home network during the initial authentication process between a user and a visited network, the latter allows for architectures that avoid the on-line involvement of the home network, since authentication may then be based on *public key certificates* (see section 10.1).

most secure, flexible and scalable Internet authentication method. We adopted a *formal threat model*, as described by Howard and LeBlanc [81, p69-124]. The main aim of performing this security-based analysis was to determine which threats to the new authentication techniques require mitigation and how to mitigate them, reducing via a formal process of threat modelling the overall risk to the protocols to an acceptable level. We also used this model to conduct a comparative analysis of the four authentication techniques.

Additionally, since these techniques were designed to meet the security and implementation services and properties required of new authentication methods for Internet access, we analysed and compared the services and properties possessed by the four candidate protocols against each of these requirements.

Finally, the above referenced requirements were used in conjunction with the threat model results to provide an overall assessment of the proposed authentication schemes. Our comparative analysis suggested that the PANA/IKEv2 technique is the best Internet authentication method of those proposed in this thesis. This is closely followed by the PANA/Liberty scheme. The PANA/UMTS and PANA/GSM protocols were ranked third and fourth.

## 12.2 Suggestions for Future Work

Many research issues remain in the area of Internet authentication for remote access, and major new problems are likely to emerge with the growth in ubiquitous Internet access, mobile computing and heterogeneity of the networking environment. Suggestions for future work in this area include the following.

- Authentication and key agreement are fundamental components of a secure procedure for remote access. We have already noted that the session key derivation mechanism in the current version of PANA/GSM depends

heavily on the EAP/SIM protocol. In addition, we observed that a malicious user could succeed in learning EAP/SIM keying material that is also used in the GSM network, as shown in the threat tree for SIM credential reuse and brute-force attacks (see Figure 11.16)[4]. Therefore, one interesting alternative might be to adopt one of the unified EAP session key derivation approaches currently being investigated, instead of adopting the existing scheme from EAP/SIM. One example of such an approach is provided by the Salowey-Eronen mechanism (given in section 3.6.6), which derives cryptographically separate keys for multiple applications independent of the EAP method in use.

- An analogous scheme to the PANA/GSM authentication technique would be to specify the General Packet Radio Service (GPRS) security and mobility management authentication protocol (see section 3.5.2) as an EAP method (e.g. Buckley et al. [31]). This would enable its use with PANA, in a scheme which we might call PANA/GPRS.

- The solution proposed in section 8.7 incorporates part of the GAA framework into PANA/UMTS, in which the EAP server operates as a *gateway* between the Internet AAA network and the UMTS AKA infrastructure, performing the retrieval of authentication vectors and the GUSS from the HSS. An analogous scheme could be specified to apply this solution directly to EAP-AKA, since (as discussed in section 8.7.2) the GAA *Zh* interface can be used to allow an EAP-AKA server to obtain authentication vectors from the HSS.

- As discussed in section 9.7.2, there are a variety of schemes that could potentially be used as the PANA inner authentication protocol instead of 3GPP AKA in the PANA/Liberty technique. In fact, these novel possibilities for PANA inner authentication may represent suggestions for further

[4]However, performing a brute-force search for a 64-bit key is a non-trivial task that could not be executed in real time; moreover, as previously stated, it is unlikely to be worth the effort of performing such a search just to steal network access.

research. A first alternative scheme that might be used as the PANA inner protocol is an EAP method encapsulating the 3GPP2 CDMA2000 1x identification and authentication message exchanges (i.e. EAP-CDMA, proposed in section 9.7.2); we might call this scheme PANA/CDMA.

- A second alternative method for PANA inner authentication is the transport of pre-shared key (PSK) based mechanisms by EAP (i.e. EAP-PSK, designed by Bersani and Tschöfenig — see section 3.6.7) and PANA, into a scheme which we might call PANA/PSK. As discussed in section 9.7.2, PANA/PSK would be a good candidate for use instead of PANA/UMTS as the initial authentication mechanism in the PANA/Liberty technique. PANA/PSK has the potential to meet the following design goals: *simplicity* — since EAP-PSK relies on a single cryptographic primitive (i.e. AES-128 — see section 3.6.7), *wide applicability* — since EAP-PSK is designed for authentication over insecure networks (such as IEEE 802.11 — see section 3.6.7), *security*, and *extensibility* (see section 1.1 of RFC 4764 [25]).

- A third possibility involves using PANA/IKEv2, instead of PANA/UMTS, as the PANA/Liberty initial authentication mechanism (see section 9.7.2). As previously described, PANA/IKEv2 uses the EAP-IKEv2 protocol (due to Tschöfenig, Kroeselberg, Ohba and Bersani — see section 10.4) as the PANA inner authentication mechanism. EAP-IKEv2 specifies a way of encapsulating the first phase of the IKEv2 protocol, which supports both symmetric and asymmetric authentication, within EAP. PANA/IKEv2 is thus a good candidate for use as the PANA/Liberty initial authentication mechanism. This is because the Liberty SSO service can be used to extend a PANA/IKEv2 initial authentication to all Liberty-enabled SPs, and create a network identity infrastructure supporting all network access devices. Therefore, we could combine the following benefits: the increase in flexibility provided by the public key based authentication option, the

gains in security given by the IKEv2 mechanism, and the gains in interoperability and scalability by incorporating the Liberty framework.

In parallel with the development of the new security schemes suggested above, there is a need to further evaluate and compare them with the existing proposals. This thesis is based on one particular draft of the PANA specification [65]. The latest version of this draft [66] was published as this thesis was being completed, and it would therefore be desirable to make any necessary changes in the schemes described in this thesis to reflect the changes to the PANA text.

# Bibliography


[1] 3rd Generation Partnership Project, Technical Specification Group Core Network and Terminals, Valbonne, France. *3GPP TS 24.109 V7.5.0 — Bootstrapping interface (Ub) and network application function interface (Ua), Protocol details (Release 7)*, December 2006.

[2] 3rd Generation Partnership Project, Technical Specification Group Core Network and Terminals, Valbonne, France. *3GPP TS 29.229 V7.6.0 — Cx and Dx interfaces based on the Diameter protocol, Protocol details (Release 7)*, September 2007.

[3] 3rd Generation Partnership Project, Technical Specification Group Core Network and Terminals, Valbonne, France. *3GPP TS 29.109 V7.7.0 — Generic Authentication Architecture (GAA), Zh and Zn Interfaces based on the Diameter protocol, Stage 3 (Release 7)*, September 2007.

[4] 3rd Generation Partnership Project, Technical Specification Group Core Network and Terminals, Valbonne, France. *3GPP TS 23.003 V7.5.0 — Numbering, addressing and identification (Release 7)*, September 2007.

[5] 3rd Generation Partnership Project, Technical Specification Group Services and System Aspects, Valbonne, France. *3GPP TS 33.222 V7.2.0 — Generic Authentication Architecture (GAA), Access to network application functions using Hypertext Transfer Protocol over Transport Layer Security (HTTPS) (Release 7)*, September 2006.

[6] 3rd Generation Partnership Project, Technical Specification Group Services and System Aspects, Valbonne, France. *3GPP TS 33.221 V7.0.0 — Generic Authentication Architecture (GAA), Support for subscriber certificates (Release 7)*, June 2007.

[7] 3rd Generation Partnership Project, Technical Specification Group Services and System Aspects, Valbonne, France. *3GPP TS 33.220 V7.9.0 — Generic Authentication Architecture (GAA), Generic bootstrapping architecture (Release 7)*, September 2007.

[8] 3rd Generation Partnership Project, Technical Specification Group Services and System Aspects, 3G Security, Valbonne, France. *3GPP TS 33.102 V7.1.0 — Security Architecture (Release 7)*, December 2006.

[9] 3rd Generation Partnership Project, Technical Specification Group Services and System Aspects, 3G Security, Valbonne, France. *3GPP TS 35.202 V7.0.0 — Specification of the 3GPP Confidentiality and Integrity Algorithms, Document 2: KASUMI Specification (Release 7)*, June 2007.

[10] 3rd Generation Partnership Project, Technical Specification Group Services and System Aspects, 3G Security, Valbonne, France. *3GPP TR 33.919 V7.2.0 — Generic Authentication Architecture (GAA), System Description (Release 7)*, March 2007.

[11] 3rd Generation Partnership Project, Technical Specification Group Services and System Aspects, Liberty Alliance and 3GPP Security Interworking, Valbonne, France. *3GPP TR 33.980 V7.6.0 — Interworking of Liberty Alliance Identity Federation Framework (ID-FF), Identity Web Services Framework (ID-WSF) and Generic Authentication Architecture (GAA) (Release 7)*, September 2007.

[12] R. Aarts and P. Madsen (editors). Liberty ID-WSF interaction service specification, version: 2.0-04. Liberty Specification draft-liberty-idwsf-interaction-svc-v2.0-04, Liberty Alliance Project, November 2005.

[13] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz. Extensible authentication protocol (EAP). Request For Comments 3748, Internet Engineering Task Force, June 2004.

[14] B. Aboba and P. Calhoun. (RADIUS) (remote authentication dial in user service) support for extensible authentication protocol (EAP). Request For Comments 3579, Internet Engineering Task Force, September 2003.

[15] B. Aboba, P. Calhoun, S. Glass, T. Hiller, P. McCann, H. Shiino, P. Walsh, G. Zorn, G. Dommety, C. Perkins, B. Patil, D. Mitton, S. Manning, M. Beadles, X. Chen, S. Sivalingham, A. Hameed, M. Munson, S. Jacobs, B. Lim, B. Hirschman, R. Hsu, H. Koo, M. Lipford, E. Campbell, Y. Xu, S. Baba, and E. Jaques. Criteria for evaluating AAA protocols for network access. Request For Comments 2989, Internet Engineering Task Force, November 2000.

[16] B. Aboba, H. Krawczyk, and Y. Sheffer. PIC, a pre-IKE credential provisioning protocol. Internet Draft (Work in Progress) draft-ietf-ipsra-pic-06, Internet Engineering Task Force, October 2002.

[17] B. Aboba and D. Simon. PPP EAP TLS authentication protocol. Request For Comments 2716, Internet Engineering Task Force, October 1999.

[18] B. Aboba, D. Simon, and P. Eronen. Extensible authentication protocol (EAP) key management framework. Internet Draft (Work in Progress) draft-ietf-eap-keying-19, Internet Engineering Task Force, October 2007.

[19] B. Aboba and J. Wood. Authentication, authorization and accounting (AAA) transport profile. Request For Comments 3539, Internet Engineering Task Force, June 2003.

[20] J. Arkko and H. Haverinen. Extensible authentication protocol method for 3rd generation authentication and key agreement (EAP-AKA). Request For Comments 4187, Internet Engineering Task Force, January 2006.

[21] N. Asokan, V. Niemi, and K. Nyberg. Man-in-the-middle in tunnelled authentication protocols. In B. Christianson, B. Crispo, J. Malcolm, and M. Roe, editors, *Security Protocols: 11th International Workshop on Security Protocols, Proceedings, Lecture Notes in Computer Science 3364*, pages 28–41, Cambridge, UK, April 2003. Springer-Verlag.

[22] M. Badra and P. Urien. EAP-Double-TLS authentication protocol. Internet Draft (Work in Progress) draft-badra-eap-double-tls-05, Internet Engineering Task Force, June 2006.

[23] J. Beatty, J. Sergent, and J. Hodges (editors). Liberty ID-WSF discovery service specification, version: 2.0-12. Liberty Specification draft-liberty-idwsf-disco-svc-v2.0-12, Liberty Alliance Project, November 2005.

[24] S. Beaulieu and R. Pereira. Extended authentication within IKE (XAUTH). Internet Draft (Work in Progress) draft-beaulieu-ike-xauth-02, Internet Engineering Task Force, October 2001.

[25] F. Bersani and H. Tschöfenig. The EAP-PSK protocol: a pre-shared key EAP method. Request For Comments 4764, Internet Engineering Task Force, January 2007.

[26] C. W. Blanchard. Wireless security. In R. Temple and J. Regnault, editors, *Internet and wireless security*, chapter 8, pages 147–162. Institution of Electrical Engineers Press, London, 2002.

[27] L. Blunk and J. Vollbrecht. PPP extensible authentication protocol (EAP). Request For Comments 2284, Internet Engineering Task Force, March 1998.

[28] L. Blunk, J. Vollbrecht, and B. Aboba. The one time password (OTP) and generic token card authentication protocols. Internet Draft (Work in Progress) draft-ietf-eap-otp-00, Internet Engineering Task Force, October 2002.

[29] D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. Nielsen, and D. Winer (editors). Simple object access protocol (SOAP) version 1.1. W3C Note NOTE-SOAP-20000508, World Wide Web Consortium, May 2000.

[30] C. Brookson. GSM (and PCN) security and encryption. http://www.brookson. com/gsm/gsmdoc.htm, 1994.

[31] A. Buckley, P. Satarasinghe, V. Alperovich, J. Puthenkulam, J. Walker, and V. Lortz. EAP SIM GMM authentication. Internet Draft (Work in Progress) draft-buckley-pppext-eap-sim-gmm-00, Internet Engineering Task Force, August 2002.

[32] P. Calhoun, S. Farrell, and W. Bulley. Diameter CMS security application. Internet Draft (Work in Progress) draft-ietf-aaa-diameter-cms-sec-04, Internet Engineering Task Force, March 2002.

[33] P. Calhoun, T. Johansson, C. Perkins, T. Hiller, and P. McCann. Diameter mobile IPv4 application. Request For Comments 4004, Internet Engineering Task Force, August 2005.

[34] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, and J. Arkko. Diameter base protocol. Request For Comments 3588, Internet Engineering Task Force, September 2003.

[35] P. Calhoun, G. Zorn, D. Spence, and D. Mitton. Diameter network access server application. Request For Comments 4005, Internet Engineering Task Force, August 2005.

[36] S. Cantor and J. Kemp (editors). Liberty ID-FF protocols and schema specification, version: 1.2-errata-v3.0. Liberty Specification draft-liberty-idff-protocols-schema-1.2-errata-v3.0, Liberty Alliance Project, May 2005.

[37] S. Cantor, J. Kemp, and D. Champagne (editors). Liberty ID-FF bindings and profiles specification, version: 1.2-errata-v2.0. Liberty Specifi-

cation draft-liberty-idff-bindings-profiles-1.2-errata-v2.0, Liberty Alliance Project, September 2004.

[38] W. Cheswick and S. Bellovin. *Firewalls and Internet Security*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1994.

[39] D. D. Clark and D. R. Wilson. A comparison of commercial and military computer security policies. In *Proceedings of the 1987 IEEE Symposium on Security and Privacy*, pages 184–194, Oakland, CA, April 1987. IEEE Computer Society Press.

[40] R. Clarke. Authentication: A sufficiently rich model to enable e-business. Review Draft of 26/12/2001, Department of Computer Science, Australian National University, Canberra 0200, Australia, available at http://www.anu.edu.au/people/Roger.Clarke/EC/AuthModel.html, December 2001.

[41] R. Clarke. Certainty of identity: A fundamental misconception, and a fundamental threat to security. *Privacy Law and Policy Reporter*, 8(3):63–65, July 2001.

[42] J. Daemen and V. Rijmen. *The Design of Rijndael: AES — The Advanced Encryption Standard*. Springer-Verlag, Berlin, 2002.

[43] A. W. Dent and C. J. Mitchell. *User's Guide to Cryptography and Standards*. Artech House, London, 2004.

[44] C. Dierks and C. Allen. The TLS protocol version 1.0. Request For Comments 2246, Internet Engineering Task Force, January 1999.

[45] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, June 1976.

[46] R. Droms. Dynamic host configuration protocol. Request For Comments 2131, Internet Engineering Task Force, March 1997.

[47] R. Droms, J. Bound, B. Volz, T. Lemon, C. Perkins, and M. Carney. Dynamic host configuration protocol for ipv6 (DHCPv6). Request For Comments 3315, Internet Engineering Task Force, July 2003.

[48] D. Eastlake 3rd, S. Crocker, and J. Schiller. Randomness recommendations for security. Request For Comments 1750, Internet Engineering Task Force, December 1994.

[49] C. Kaufman (editor). Internet key exchange (IKEv2) protocol. Request For Comments 4306, Internet Engineering Task Force, December 2005.

[50] G. Lambert (editor). Liberty ID-SIS contact book service implementation guidelines, version: 1.0-06. Liberty Specification draft-liberty-id-sis-contactbook-guidelines-v1.0-06, Liberty Alliance Project, July 2005.

[51] J. Arkko (editor), J. Kempf, B. Zill, and P. Nikander. SEcure Neighbor Discovery (SEND). Request For Comments 3971, Internet Engineering Task Force, March 2005.

[52] J. Kainulainen (editor). Liberty ID-SIS geolocation service specification, version: 1.0-12. Liberty Specification draft-liberty-id-sis-gl-v1.0-12, Liberty Alliance Project, April 2005.

[53] J. Uberti (editor). Liberty ID-SIS presence service implementation guidelines, version: 1.0-12. Liberty Specification draft-liberty-id-sis-presence-guidelines-v1.0-12, Liberty Alliance Project, July 2005.

[54] P. Saint-Andre (editor). Liberty ID-SIS presence service specification, version: 1.0-10. Liberty Specification draft-liberty-id-sis-presence-v1.0-10, Liberty Alliance Project, July 2005.

[55] S. Kellomaki (editor). Liberty ID-SIS contact book service specification, version: 1.0-11. Liberty Specification draft-liberty-id-sis-cb-v1.0-11, Liberty Alliance Project, July 2005.

[56] T. Wason (editor), S. Cantor, J. Hodges, J. Kemp, and P. Thompson. Liberty ID-FF architecture overview, version 1.2-errata-v1.0. Draft Specification draft-liberty-idff-arch-overview-1.2-errata-v1.0, Liberty Alliance Project, May 2005.

[57] G. Ellison, F. Hirsch, and P. Madsen (editors). ID-WSF 2.0 SecMech SAML Profile, version: v2.0-11. Liberty Specification draft-liberty-idwsf-security-mechanisms-saml-profile-v2.0-11, Liberty Alliance Project, November 2005.

[58] G. Ellison, F. Hirsch, and P. Madsen (editors). Liberty ID-WSF security mechanisms core, version: v2.0-12. Liberty Specification draft-liberty-idwsf-security-mechanisms-core-v2.0-12, Liberty Alliance Project, November 2005.

[59] P. Eronen, T. Hiller, and G. Zorn. Diameter extensible authentication protocol (EAP) application. Request For Comments 4072, Internet Engineering Task Force, August 2005.

[60] P. Eronen and H. Tschöfenig. Pre-shared key ciphersuites for transport layer security (TLS). Request For Comments 4279, Internet Engineering Task Force, December 2005.

[61] ETSI. *GSM Technical Specification GSM 04.08 (ETS 300 940): "Digital cellular telecommunication system (Phase 2+); Mobile radio interface layer 3 specification" (version 7.8.0)*. European Telecommunications Standards Institute, June 2000.

[62] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext transfer protocol — HTTP/1.1. Request For Comments 2616, Internet Engineering Task Force, June 1999.

[63] S. Fluhrer, I. Mantin, and A. Shamir. Weaknesses in the key scheduling algorithm of RC4. In S. Vaudenay and A.M. Youssef, editors, *Selected Areas*

*in Cryptography: 8th Annual International Workshop — SAC 2001, Proceedings, Lecture Notes in Computer Science 2259*, pages 1–24, Toronto, Ontario, Canada, August 2001. Springer-Verlag.

[64] W. Ford. *Computer communications security: Principles, standard protocols and techniques.* Prentice Hall, Upper Saddle River, New Jersey, 1994.

[65] D. Forsberg, Y. Ohba, B. Patil, H. Tschöfenig, and A. Yegin. Protocol for carrying authentication for network access (PANA). Internet Draft (Work in Progress) draft-ietf-pana-pana-10, Internet Engineering Task Force, July 2005.

[66] D. Forsberg, Y. Ohba, B. Patil, H. Tschöfenig, and A. Yegin. Protocol for carrying authentication for network access (PANA). Internet Draft (Work in Progress) draft-ietf-pana-pana-18, Internet Engineering Task Force, September 2007.

[67] D. Forsberg and J. Rajahalme. Secure network access authentication (SeNAA). Internet Draft (Work in Progress) draft-forsberg-pana-secure-network-access-auth-01, Internet Engineering Task Force, September 2002.

[68] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, and L. Stewart. HTTP authentication: Basic and digest access authentication. Request For Comments 2617, Internet Engineering Task Force, June 1999.

[69] A. Freier, P. Karlton, and P. Kocher. The SSL protocol version 3.0. Internet Draft (Work in Progress) draft-freier-ssl-version3-02, Internet Engineering Task Force, November 1996.

[70] P. Funk and S. Blake-Wilson. EAP tunneled TLS authentication protocol (EAP-TTLS). Internet Draft (Work in Progress) draft-ietf-pppext-eap-ttls-05, Internet Engineering Task Force, July 2004.

[71] J. T. Geier and J. Geier. *Wireless LANs*. Sams Publishing, Indianapolis, IN, USA, 2nd edition, 2001.

[72] D. Gollmann. What do we mean by entity authentication? In *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, pages 46–54, Oakland, CA, May 1996. IEEE Computer Society Press.

[73] C. Grahm, D. Castellanos, and J. Kainulainen (editors). Liberty ID-SIS geolocation service implementation guidelines, version: 1.0-15. Liberty Specification draft-liberty-id-sis-gl-guidelines-v1.0-15, Liberty Alliance Project, April 2005.

[74] M. Gudgin, M. Hadley, N. Mendelsohn, J. Moreau, H. F. Nielsen, A. Karmarkar, and Y. Lafon (editors). SOAP version 1.2 part 1: Messaging framework (second edition). W3C Recommendation REC-soap12-part1-20070427, World Wide Web Consortium, April 2007.

[75] N. Haller, C. Metz, P. Nesser, and M. Straw. A one-time password system. Request For Comments 2289, Internet Engineering Task Force, February 1998.

[76] D. Harkins and D. Carrel. The Internet Key Exchange (IKE). Request For Comments 2409, Internet Engineering Task Force, November 1998.

[77] H. Haverinen and J. Salowey. Extensible authentication protocol method for GSM subscriber identity modules (EAP-SIM). Request For Comments 4186, Internet Engineering Task Force, January 2006.

[78] J. Hodges and R. Aarts (editors). Liberty ID-WSF authentication service and single sign-on service specification, version: 1.1. Liberty Specification liberty-idwsf-authn-svc-v1.1, Liberty Alliance Project, May 2005.

[79] P. Hoffman. Algorithms for Internet Key Exchange version 1 (IKEv1). Request For Comments 4109, Internet Engineering Task Force, May 2005.

[80] R. Housley, W. Polk, W. Ford, and D. Solo. Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile. Request For Comments 3280, Internet Engineering Task Force, April 2002.

[81] M. Howard and D. LeBlanc. *Writing Secure Code*. Microsoft Press, USA, 2nd edition, 2003.

[82] R. Hulsebosch, C. Günther, G. Horn, S. Holtmanns, K. Howker, K. Paterson, J. Claessens, and M. Schuba. Pioneering advanced mobile privacy and security. In C. J. Mitchell, editor, *Security for mobility*, chapter 17, pages 383–432. Institution of Electrical Engineers Press, London, January 2004.

[83] Institute of Electrical and Electronics Engineers. *1363a–2004 — IEEE Standard Specifications for Public-Key Cryptography — Amendment 1: Additional Techniques*, 2004.

[84] Institute of Electrical and Electronics Engineers. *802.1X–2004 — IEEE Standard for Local and metropolitan area networks Port-Based Network Access Control*, 2004.

[85] Institute of Electrical and Electronics Engineers. *802.11–2007 — IEEE Standard for Information technology —Telecommunications and information exchange between systems — Local and metropolitan area networks — Specific requirements — Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, 2007.

[86] International Organization for Standardization (ISO), Geneva. *ISO 7498-2, Information processing systems — Open Systems Interconnection — Basic Reference Model — Part 2: Security Architecture*, 1st edition, 1989.

[87] International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), Geneva. *ISO/IEC 9798-1, Information technology — Security techniques — Entity authentication — Part 1: General*, 2nd edition, 1997.

[88] International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), Geneva. *ISO/IEC 18033-1, Information technology — Security techniques — Encryption algorithms — Part 1: General*, 1st edition, 2005.

[89] International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), Geneva. *ISO/IEC 18033-3, Information technology — Security techniques — Encryption algorithms — Part 3: Block ciphers*, 1st edition, 2005.

[90] International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), Geneva. *ISO/IEC 18033-4, Information technology — Security techniques — Encryption algorithms — Part 4: Stream ciphers*, 1st edition, 2005.

[91] International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), Geneva. *ISO/IEC 10118-1, Information technology — Security techniques — Hash-functions — Part 1: General*, 2nd edition, 2000.

[92] International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), Geneva. *ISO/IEC 10118-2, Information technology — Security techniques — Hash-functions — Part 2: Hash-functions using an n-bit block cipher*, 2nd edition, 2000.

[93] International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), Geneva. *ISO/IEC 10118-3, Information technology — Security techniques — Hash-functions — Part 3: Dedicated hash-functions*, 3rd edition, 2004.

[94] International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), Geneva. *ISO/IEC 10118-4, Information technology — Security techniques — Hash-functions — Part 4: Hash-functions using modular arithmetic*, 1st edition, 1998.

[95] International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), Geneva. *ISO/IEC 9797-1, Information technology — Security techniques — Message Authentication Codes (MACs) — Part 1: Mechanisms using a block cipher*, 1st edition, 1999.

[96] International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), Geneva. *ISO/IEC 9797-2, Information technology — Security techniques — Message Authentication Codes (MACs) — Part 2: Mechanisms using a dedicated hash-function*, 1st edition, 2002.

[97] International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), Geneva. *ISO/IEC FDIS 14888-2, Information technology — Security techniques — Digital signatures with appendix — Integer factorization based mechanisms*, 2nd edition, October 2007.

[98] International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), Geneva. *ISO/IEC 11770-1, Information technology — Security techniques — Key management — Part 1: Framework*, 1st edition, 1996.

[99] International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), Geneva. *ISO/IEC FCD 18033-2, Information technology — Security techniques — Encryption algorithms — Part 2: Asymmetric ciphers*, 1st edition, 2006.

[100] International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), Geneva. *ISO/IEC FCD 14888-1, Information technology — Security techniques — Digital signatures with appendix — Part 1: General*, 2nd edition, September 2007.

[101] International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), Geneva. *ISO/IEC 14888-3, Information*

*technology — Security techniques — Digital signatures with appendix — Part 3: Discrete logarithm based mechanisms*, 2nd edition, 2006.

[102] International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), Geneva. *ISO/IEC FCD 11770-3, Information technology — Security techniques — Key management — Part 3: Mechanisms using asymmetric techniques*, 2nd edition, October 2007.

[103] International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), Geneva. *ISO/IEC 18028-4, Information technology — Security techniques — IT network security — Part 4: Securing remote access*, 1st edition, 2005.

[104] International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), Geneva. *ISO/IEC 13239, Information technology — Telecommunications and information exchange between systems — High-level data link control (HDLC) procedures*, 3rd edition, 2002.

[105] International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), Geneva. *ISO/IEC 18031, Information technology — Security techniques — Random bit generation*, 1st edition, 2005.

[106] International Telecommunication Union - Radiocommunication Study Groups (ITU-R), Geneva. *Recomendation ITU-R M.1457: Detailed Specifications of the radio interfaces of International Mobile Telecommunications-2000 (IMT-2000)*, 1999.

[107] International Telecommunication Union - Telecommunication Standardization Sector (ITU-T), Geneva. *Recommendation X.800, Data Communication Networks: Open System Interconnection (OSI); Security, structure and Applications — Security architecture for Open Systems Interconnection for CCITT Applications*, 1991. Also published as ISO International Standard 7498-2.

[108] International Telecommunication Union - Telecommunication Standardization Sector (ITU-T), Geneva. *Recommendation X.509, Information technology — Open Systems Interconnection — The Directory: Public-key and attribute certificate frameworks*, March 2000. Also published as ISO International Standard 9594-8.

[109] P. Jayaraman, R. Lopez, Y. Ohba, M. Parthasarathy, and A. Yegin. PANA framework. Internet Draft (Work in Progress) draft-ietf-pana-framework-10, Internet Engineering Task Force, September 2007.

[110] D. Johnson, C. Perkins, and J. Arkko. Mobility support in IPv6. Request For Comments 3775, Internet Engineering Task Force, June 2004.

[111] S. Josefsson. The EAP securID(r) mechanism. Internet Draft (Work in Progress) draft-josefsson-eap-securid-01, Internet Engineering Task Force, February 2002.

[112] S. Kellomaki and R. Lockhart (editors). Liberty ID-SIS employee profile service implementation guidelines, version: 1.1. Liberty Specification lliberty-idsis-ep-guidelines-v1.1, Liberty Alliance Project, September 2005.

[113] S. Kellomaki and R. Lockhart (editors). Liberty ID-SIS employee profile service specification, version: 1.1. Liberty Specification liberty-idsis-ep-v1.1, Liberty Alliance Project, September 2005.

[114] S. Kellomaki and R. Lockhart (editors). Liberty ID-SIS personal profile service implementation guidelines, version: 1.1. Liberty Specification liberty-idsis-pp-guidelines-v1.1, Liberty Alliance Project, September 2005.

[115] S. Kellomaki and R. Lockhart (editors). Liberty ID-SIS personal profile service specification, version: 1.1. Liberty Specification liberty-idsis-pp-v1.1, Liberty Alliance Project, September 2005.

[116] S. Kent. IP authentication header. Request For Comments 4302, Internet Engineering Task Force, December 2005.

[117] S. Kent. IP encapsulating security payload (ESP). Request For Comments 4303, Internet Engineering Task Force, December 2005.

[118] S. Kent and K. Seo. Security architecture for the Internet Protocol. Request For Comments 4301, Internet Engineering Task Force, December 2005.

[119] V. Khu-smith and C. Mitchell. Enhancing e-commerce security using GSM authentication. In K. Bauknecht, A. Min Tjoa, and G. Quirchmayr, editors, *E-Commerce and Web Technologies: 4th International Conference — EC-Web 2003, Proceedings, Lecture Notes in Computer Science 2738*, pages 72–83, Prague, Czech Republic, September 2003. Springer-Verlag.

[120] H. Knospe and S. Schwiderski-Grosche. Future mobile networks: Ad-hoc access based on online payment with smartcards. In *13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2002), Proceedings*, pages 197–200, Lisbon, Portugal, September 2002. Institute of Electrical and Electronics Engineers.

[121] H. Knospe and S. Schwiderski-Grosche. Online payment for access to heterogeneous mobile networks. In N.P. Foteini and B. Arroyo-Fernandez, editors, *IST Mobile & Wireless Telecommunications Summit 2002, Proceedings*, pages 748–752, Thessaloniki, Greece, June 2002. IST.

[122] H. Knospe and S. Schwiderski-Grosche. Secure mobile commerce. In C. J. Mitchell, editor, *Security for Mobility*, chapter 14, pages 325–346. Institution of Electrical Engineers Press, January 2004.

[123] H. Krawczyk, M. Bellare, and R. Canetti. HMAC: Keyed-hashing for message authentication. Request For Comments 2104, Internet Engineering Task Force, February 1997.

[124] P. Laitinen, P. Ginzboorg, N. Asokan, S. Holtmanns, and V. Niemi. Extending cellular authentication as a service. In *First IEE International Conference on Commercialising Technology and Innovation, Proceedings*, pages 0_90–D2/4, London, September 2005. Institution of Electrical Engineers Press.

[125] A. Lior and A. Yegin. PANA AAA interworking. Internet Draft (Work in Progress) draft-ieft-pana-aaa-interworking-00, Internet Engineering Task Force, July 2005.

[126] B. Lloyd and W. Simpson. PPP authentication protocols. Request For Comments 1334, Internet Engineering Task Force, October 1992.

[127] J. Malcolm. Lightweight authentication in a mobile network (transcript of discussion). In B. Christianson, B. Crispo, J. Malcolm, and M. Roe, editors, *Security Protocols Workshop 2001, 9th International Workshop, Revised Papers. Lecture Notes in Computer Science 2467*, pages 217–220, Cambridge, UK, April 2001. University of Hertfordshire, Springer-Verlag.

[128] E. Maler, P. Mishra, and R. Philpott (editors). Assertions and protocol for the OASIS security assertion markup language (SAML) v1.1. Oasis committee specification, Organization for the Advancement of Structured Information Standards, May 2003.

[129] L. Mamakos, K. Lidl, J. Evarts, D. Carrel, D. Simone, and R. Wheeler. A method for transmitting PPP over ethernet (PPPoE). Request For Comments 2516, Internet Engineering Task Force, February 1999.

[130] M. Matsui. New block encryption algorithm MISTY. In E. Biham, editor, *Fast Software Encryption '97, Proceedings, Lecture Notes in Computer Science 1267*, pages 54–68, Haifa, Israel, January 1997. Springer-Verlag.

[131] D. Maughan, M. Schertler, M. Schneider, and J. Turner. Internet security association and key management protocol (ISAKMP). Request For Comments 2408, Internet Engineering Task Force, November 1998.

[132] A. Menezes, P. van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Boca Raton, 1996.

[133] C. Metz. OTP extended responses. Request For Comments 2243, Internet Engineering Task Force, November 1997.

[134] D. Meyer. Administratively scoped IP multicast. Request For Comments 2365, Internet Engineering Task Force, July 1998.

[135] C. Mitchell. The security of the GSM air interface protocol. Technical Report RHUL-MA-2001-3, Department of Mathematics, Royal Holloway, University of London, Egham, Surrey TW20 0EX, UK, available at http://www.rhul.ac.uk/mathematics/techreports, August 2001.

[136] C. Mitchell and P. Pagliusi. Is entity authentication necessary? In B. Christianson, J. A. Malcolm, B. Crispo, and M. Roe, editors, *Security Protocols: 10th International Workshop on Security Protocols, Proceedings, Lecture Notes in Computer Science 2845*, pages 20–33, Cambridge, UK, December 2003. Springer-Verlag.

[137] C. J. Mitchell. Cryptography for mobile security. In C. J. Mitchell, editor, *Security for Mobility*, IEE Telecommunications Series, chapter 1, pages 3–10. Institution of Electrical Engineers Press, London, 2004.

[138] A. Niemi, J. Arkko, and V. Torvinen. Hypertext transfer protocol (HTTP) digest authentication using authentication and key agreement (AKA). Request For Comments 3310, Internet Engineering Task Force, September 2002.

[139] NIST. *Federal Information Processing Standard, Secure Hash Standard (FIPS Publication 180-1)*. National Institute of Standards and Technology, U.S. Department of Commerce, Gaithersburg, MD, April 1995.

[140] NIST. *Federal Information Processing Standards: Data Encryption Standard (DES) (FIPS Publication 46-3)*. National Institute of Standards and

Technology, U.S. Department of Commerce, Gaithersburg, MD, October 1999.

[141] NIST. *Federal Information Processing Standards: Digital Signature Standard (FIPS Publication 186-2).* National Institute of Standards and Technology, Gaithersburg, MD, January 2000.

[142] NIST. *Federal Information Processing Standard, Advanced Encryption Standard (AES) (FIPS Publication 197).* National Institute of Standards and Technology, U.S. Department of Commerce, Gaithersburg, MD, November 2001.

[143] Y. Ohba, S. Baba, and S. Das. PANA over TLS. Internet Draft (Work in Progress) draft-ohba-pana-potls-01, Internet Engineering Task Force, October 2002.

[144] Y. Ohba, S. Das, B. Patil, H. Soliman, and A. Yegin. Problem statement and usage scenarios for PANA. Internet Draft (Work in Progress) draft-ietf-pana-usage-scenarios-06, Internet Engineering Task Force, April 2003.

[145] P. Pagliusi. A contemporary foreword on GSM security. In G. Davida, Y. Frankel, and O. Rees, editors, *Infrastructure Security: International Conference — InfraSec 2002, Proceedings, Lecture Notes in Computer Science 2437*, pages 129–144, Bristol, UK, October 2002. Springer-Verlag.

[146] P. Pagliusi and C. Mitchell. PANA/GSM authentication for Internet access. In P. Farkas, editor, *Joint 1st Workshop on Mobile Future & Symposium on Trends in Communications — SympoTIC'03, Proceedings*, pages 146–152, Bratislava, Slovakia, October 2003. Institute of Electrical and Electronics Engineers.

[147] P. Pagliusi and C. Mitchell. PANA/IKEv2: an Internet authentication protocol for heterogeneous access. In K. Chae and M. Yung, editors, *4th International Workshop on Information Security Applications — WISA*

*2003, Proceedings, Lecture Notes in Computer Science 2908*, pages 135–149, Jeju Island, Korea, August 2003. Springer-Verlag.

[148] P. S. Pagliusi and C. J. Mitchell. Heterogeneous Internet access via PANA/UMTS. In *the Proceedings of 3rd International Conference on Information Security, Hardware/Software Codesign And Computers Network — ISCOCO 2004*, Rio de Janeiro, Brazil, October 2004. To be published in the World Scientific and Engineering Academy and Society (WSEAS) Transactions.

[149] A. Palekar, D. Simon, J. Salowey, H. Zhou, G. Zorn, and S. Josefsson. Protected EAP protocol (PEAP) version 2. Internet Draft (Work in Progress) draft-josefsson-pppext-eap-tls-eap-10, Internet Engineering Task Force, October 2004.

[150] M. Parthasarathy. PANA enabling IPsec based access control. Internet Draft (Work in Progress) draft-ietf-pana-ipsec-07, Internet Engineering Task Force, July 2005.

[151] M. Parthasarathy. Protocol for carrying authentication and network access (PANA) threat analysis and security requirements. Request For Comments 4016, Internet Engineering Task Force, March 2005.

[152] B. Patel, B. Aboba, S. Kelly, and V. Gupta. Dynamic host configuration protocol (DHCPv4) — configuration of ipsec tunnel mode. Request For Comments 3456, Internet Engineering Task Force, January 2003.

[153] C. Perkins. IP mobility support for IPv4. Request For Comments 3344, Internet Engineering Task Force, August 2002.

[154] R. Perlman. Understanding IKEv2: Tutorial, and rationale for decisions. Internet Draft (Work in Progress) draft-ietf-ipsec-ikev2-tutorial-01, Internet Engineering Task Force, February 2003.

[155] F. Piper and S. Murphy. *Cryptography: A Very Short Introduction*. Oxford University Press, 2002.

[156] J. Postel. User datagram protocol. Request For Comments 0768, Internet Engineering Task Force, August 1980.

[157] J. Postel. Transmission control protocol. Request For Comments 0793 (STD 7), Internet Engineering Task Force, September 1981.

[158] J. Puthenkulam, V. Lortz, A. Palekar, and D. Simon. The compound authentication binding problem. Internet Draft (Work in Progress) draft-puthenkulam-eap-binding-04, Internet Engineering Task Force, October 2003.

[159] E. Rescorla. HTTP over TLS. Request For Comments 2818, Internet Engineering Task Force, May 2000.

[160] C. Rigney, W. Willats, and P. Calhoun. (RADIUS) extensions. Request For Comments 2869, Internet Engineering Task Force, June 2000.

[161] C. Rigney, S. Willens, A. Rubens, and W. Simpson. Remote authentication dial in user service (RADIUS). Request For Comments 2865, Internet Engineering Task Force, June 2000.

[162] R. Rivest. The MD5 message-digest algorithm. Request For Comments 1321, Internet Engineering Task Force, April 1992.

[163] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public key cryptosystem. In *Communications of the ACM*. 21:120–126, February 1978.

[164] R.L. Rivest. The rc4 encryption algorithm. Technical report, RSA Data Security, Inc., Redwood City, CA, March 1992.

[165] J. Salowey and P. Eronen. Guidelines for using the EAP extended master session key (EMSK). Internet Draft (Work in Progress) draft-salowey-eap-key-deriv-02, Internet Engineering Task Force, November 2003.

[166] B. Schneier. *Secrets & Lies: Digital Security in a Networked World*. John Wiley & Sons, New York, 2000.

[167] S. Schwiderski-Grosche and H. Knospe. Public key based network access. In C. J. Mitchell, editor, *Security for Mobility*, chapter 8, pages 171–189. IEE Press, January 2004.

[168] W. Simpson. The point-to-point protocol (PPP). Request For Comments 1661 (STD 51), Internet Engineering Task Force, July 1994.

[169] W. Simpson. PPP challenge handshake authentication protocol (CHAP). Request For Comments 1994, Internet Engineering Task Force, August 1996.

[170] W. Stallings. *Cryptography and Network Security: Principles and Practice*. Prentice Hall, Upper Saddle River, New Jersey, 2nd edition, 1999.

[171] D. Stanley, J. Walker, and B. Aboba. Extensible authentication protocol (EAP) method requirements for wireless LANs. Request For Comments 4017, Internet Engineering Task Force, March 2005.

[172] Third Generation Partnership Project 2. *3GPP2 SC.R5001-0 — 3GPP2 Vision*, 1st edition, June 2004.

[173] J. Tourzan and Y. Koga (editors). Liberty ID-WSF web services framework overview, version: 1.1. Liberty Specification liberty-idwsf-overview-v1.1, Liberty Alliance Project, May 2005.

[174] W. Townsley, A. Valencia, A. Rubens, G. Pall, G. Zorn, and B. Palter. Layer two tunneling protocol "L2TP". Request For Comments 2661, Internet Engineering Task Force, August 1999.

[175] H. Tschöfenig, D. Kroeselberg, Y. Ohba, and F. Bersani. EAP IKEv2 method. Internet Draft (Work in Progress) draft-tschofenig-eap-ikev2-08, Internet Engineering Task Force, January 2006.

[176] H. Tschöfenig, D. Kroeselberg, A. Pashalidis, Y. Ohba, and F. Bersani. EAP IKEv2 method. Internet Draft (Work in Progress) draft-tschofenig-eap-ikev2-15, Internet Engineering Task Force, September 2007.

[177] J. Walker. Unsafe at any key size; an analysis of the WEP encapsulation. IEEE Document 802.11-00/362, available at http://md.hudora.de/archiv/wireless/unsafew.pdf, October 2000.

[178] J. Walker and R. Housley. The EAP Archie protocol. Internet Draft (Work in Progress) draft-jwalker-eap-archie-01, Internet Engineering Task Force, June 2003.

[179] M. Walker and T. Wright. Security. In F. Hillebrand, editor, *GSM and UMTS: The Creation of Global Mobile Communication*, chapter 15, pages 385–406. John Wiley & Sons, New York, 2002.

[180] X. Wang and H. Yu. How to break MD5 and other hash functions. In R. Cramer, editor, *Advances in Cryptology: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques — EUROCRYPT 2005, Proceedings, Lecture Notes in Computer Science 3494*, pages 19–35, Aarhus, Denmark, May 2005. Springer-Verlag.

[181] WAP. *Specification WAP-261-WTLS-20010406-a — Wireless Application Protocol — Wireless Transport Layer Security, Version 06-Apr-2001, available from http://www.wapforum.org.* Wireless Application Protocol Forum, April 2001.

[182] M. Wasserman. Recommendations for IPv6 in third generation partnership project (3GPP) standards. Request For Comments 3314, Internet Engineering Task Force, September 2002.

[183] C. Wingert and M. Naidu. *CDMA 1xRTT Security OverView.* Qualcomm Incorporated, 1st edition, August 2002.

[184] A. Yegin, Y. Ohba, R. Penno, G. Tsirtsis, and C. Wang. Protocol for carrying authentication for network access (PANA) requirements. Request For Comments 4058, Internet Engineering Task Force, May 2005.