FIDIS

Future of Identity in the Information Society

Title:	"D3.8: Study on protocols with respect to identity and identification – *an insight on network protocols and privacy-aware communication*"
Author:	WP3
Editors:	Marit Hansen (ICPP, Germany)
	Ammar Alkassar (SIRRIX, Germany)
Reviewers:	Mark Gasson (University of Reading, UK)
	Jozef Vyskoč (VaF, Slovakia)
Identifier:	D3.8
Type:	[Deliverable]
Version:	0.8
Date:	Wednesday, 14 May 2008
Status:	[Final]
Class:	[Public]
File:	FIDIS_D3_8_Protocols_Final_V08.doc

Summary

This deliverable investigates identity-related properties of commonly used protocols and interesting proposed approaches for new protocols. This includes, categorising and showing dependencies between network protocols and the outline of privacy properties, based on personal data disclosed, linkability and identifiability. Further, it critically discusses whether privacy experts are – and should be – involved in the process of designing protocols. Protocols for communication in networks are analysed according to privacy-relevant data and techniques for privacy-aware communication and their associated protocols are explained. Finally in this document, new developments for Next Generation Internet protocols are described.

This deliverable assumes some prior knowledge, but references and further reading is there to help the reader.

The FIDIS NoE receives research funding from the Community's Sixth Framework Program

Copyright Notice:

Members of the FIDIS consortium

1. Goethe University Frankfurt	Germany
2. Joint Research Centre (JRC)	Spain
3. Vrije Universiteit Brussel	Belgium
4. Unabhängiges Landeszentrum für Datenschutz	Germany
5. Institut Europeen D'Administration Des Affaires (INSEAD)	France
6. University of Reading	United Kingdom
7. Katholieke Universiteit Leuven	Belgium
8. Tilburg University	Netherlands
9. Karlstads University	Sweden
10. Technische Universität Berlin	Germany
11. Technische Universität Dresden	Germany
12. Albert-Ludwig-University Freiburg	Germany
13. Masarykova universita v Brne	Czech Republic
14. VaF Bratislava	Slovakia
15. London School of Economics and Political Science	United Kingdom
16. Budapest University of Technology and Economics (ISTRI)	Hungary
17. IBM Research GmbH	Switzerland
18. Institut de recherche criminelle de la Gendarmerie Nationale	France
19. Netherlands Forensic Institute	Netherlands
20. Virtual Identity and Privacy Research Center	Switzerland
21. Europäisches Microsoft Innovations Center GmbH	Germany
22. Institute of Communication and Computer Systems (ICCS)	Greece
23. AXSionics AG	Switzerland
24. SIRRIX AG Security Technologies	Germany

Versions

Version	*Date*	*Description (Editor)*
0.1	18.06.2007	• Initial setup (Martin Meints, ICPP)
0.2	17.08.2007	• Integration of contributions from Stefan Köpsell, Sandra Steinbrecher, Stefan Berthold, Stefanie Poetzsch and Henning Waack, TUD, as well as Markulf Kohlweiss, Claudia Diaz, Stefan Schiffner and Karel Wouters, KU Leuven (Marit Hansen, ICPP)
0.3	11.09.2007	• Integration of contributions from Ammar Alkassar *et al.*, Sirrix (Marit Hansen, ICPP)
0.4	25.09.2007	• Filling gaps, modifying structure, adding sections, expanding abbreviations (Marit Hansen, ICPP)
0.5	26.-27.09.2007	• Rewriting policy protocol section, adding conclusions for every chapter (Marit Hansen, ICPP)
0.6	28.09.2007	• Writing Executive Summary (Marit Hansen, ICPP), producing review version
0.7	15.04.2007	• Editing and incorporating the reviewer's feedback into the document. (Ammar Alkassar, SRX)
0.8	13.05.2008	• Final editing ()

Foreword

FIDIS partners from various disciplines have contributed as authors to this document. The following list names the main contributors for the chapters of this document:

Chapter	*Contributor(s)*
Executive Summary	Marit Hansen (ICPP)
1 Introduction	Marit Hansen (ICPP)
2 Protocols for network communication	2.1.1+2.1.2: Stefan Köpsell, Sandra Steinbrecher, Stefan Berthold, Stefanie Poetzsch and Henning Waack (TUD) 2.1.3+2.1.4: Marit Hansen (ICPP) 2.2-2.5: Stefan Köpsell, Sandra Steinbrecher, Stefan Berthold, Stefanie Poetzsch and Henning Waack (TUD) 2.6: Marit Hansen (ICPP)
3 Protocols for privacy-aware communication	3.1: Stefan Köpsell, Sandra Steinbrecher, Stefan Berthold, Stefanie Poetzsch and Henning Waack (TUD) 3.2: Markulf Kohlweiss, Claudia Diaz, Stefan Schiffner and Karel Wouters, KU Leuven 3.3: Marit Hansen (ICPP) 3.4: Marit Hansen (ICPP)
4 Next Generation Internet protocols	4.1-4.4: Stefan Köpsell, Sandra Steinbrecher, Stefan Berthold, Stefanie Poetzsch and Henning Waack (TUD) 4.5: Marit Hansen (ICPP)
5 Summary and conclusions	Marit Hansen (ICPP) and Stefan Köpsell, Sandra Steinbrecher, Stefan Berthold, Stefanie Poetzsch and Henning Waack (TUD)
6 References	All

Table of Contents

Executive Summary

In computing, protocols are standards that control or facilitate the connection, communication, and data transfer between two endpoints. As communication is the basic foundation of our Information Society, protocols are highly relevant for all activities in information and communication technologies.

This deliverable investigates identity-related properties of commonly used protocols and interesting proposed approaches for new protocols. Firstly, general facts on network protocols are introduced: After an introduction in layered models to categorise and show dependencies between network protocols, possible privacy properties are outlined, based on personal data disclosed, linkability and identifiability as well as obvious or hidden identifiers. Further, it is critically discussed whether privacy experts are – and should be – involved in the process of designing protocols.

After these general remarks, protocols for communication in networks are analysed according to the privacy-relevant criteria given before. These protocols cover basic Internet, LAN and WLAN communications which are regularly used by each person participating in a network. This analysis shows basically that every protocol contains disclosure of identifiers which can be linked to other actions or directly to persons involved. Usually it is difficult, if not impossible to avoid the disclosure of privacy-relevant data in this context.

Techniques for privacy-aware communication and their associated protocols are explained in the next chapter. Three main areas are investigated: anonymisation services, user-centric identity management and privacy policy languages. In these areas the protocols and tools are not widely distributed and used, yet. However, the market is evolving fast in these areas, in particular driven by the demand for user-centric identity management which entails privacy policy protocols as well as – currently on a lower level – data minimising mechanisms.

Finally in this document, new developments for Next Generation Internet protocols are described. Although many of the proposed approaches are not yet implemented, it is evident that they aim to improve security features of protocols, having learnt from the shortcomings of today's Internet protocols. That said, on the specification level there is hardly any work done in the area of privacy properties of protocols.

Summarising, today's protocols pose a lot of privacy threats which normal users as well as many protocol designers are not aware of. Privacy experts should be more involved in the specification process of protocols to prevent further erosion of privacy by steady leakage of linkable data. Research and development as well as policy makers should direct their attention to cross-layer effects resulting from the interplay of the variety of protocols which today's citizens of the Information Society use day by day.

This deliverable assumes some prior knowledge, but references and further reading is there to help the reader.

1 Introduction

> *"In computing, a protocol is a convention or standard that controls or enables the connection, communication, and data transfer between two computing endpoints. In its simplest form, a protocol can be defined as the rules governing the syntax, semantics, and synchronization of communication. Protocols may be implemented by hardware, software, or a combination of the two. At the lowest level, a protocol defines the behavior of a hardware connection."*
>
> *(Wikipedia: Protocol (computing) 2007)*

Protocols determine how communication works. As communication is a basic activity for all kinds of transactions, interactions or other information exchange, it is the most basic foundation of Information Society. There are protocols on the social level as well as on the technical level where several sublevels can be identified. This deliverable deals with protocols on the technical level, i.e., how machines communicate with each other. The scope here is specifically limited to identity aspects of protocols - this means the analysis of privacy and identifiability issues, in particular threats to privacy and possibilities to circumvent or prevent them.

When discussing protocols, there is a need to distinguish between their specification and implementation. Although these should be one and the same, in practice implementations do not always properly adhere to what is laid down in the specifications – this may be done accidentally, but in some cases deviations from the specifications are intended, e.g., when implementing light-weight versions of the full specification or when contradictions are discovered in the documents which cannot be met. Here we restrict ourselves mainly to the analysis on the specification level.

This deliverable cannot serve as a comprehensive compendium or even tutorial of all potentially relevant protocols – indeed many specifications comprise more pages than appear here. Instead, a variety of protocols are tackled which represent typical applications or procedures in the networking world. These protocols are analysed with respect to their privacy relevance: Protocols for networking are evaluated according to their behaviour concerning personal data and linkability; protocols for privacy-aware communication are scrutinised with respect to their privacy achievements.

This deliverable is organised as follows:

- **Chapter 2** focuses on state-of-the-art protocols in various networking areas. It gives an overview of general facts concerning protocols and analyses privacy aspects in a structured way as introduced.
- This analysis is supplemented by the elaboration of privacy-aware protocols in **Chapter 3** which firstly depicts the functions of anonymising services, continues with protocols in the area of user-centric identity management, and finally describes different kinds of policy languages which also can be regarded as protocol-related.

- **Chapter 4** gives a brief outlook on a selection of Next Generation Internet protocols and their privacy features and threats. Similar to Chapter 2, the focus is put on basic networking protocols rather than specific privacy-aware protocol proposals. Further it poses the question of involvement of privacy experts in the design process of protocols and their specification.
- The overall summary of the deliverable is presented in **Chapter 5**.
- For interested readers, the **references** are recommended for further reading. The **list of abbreviations** in the appendix will help with the technical terminology typical to the protocol area.

This deliverable aims to give an overview of the identity-related aspects of protocols on different technical layers, regardless if they are commonly used today or interesting newly proposed approaches. A focus is put on the protocols and mechanisms of privacy-aware communication which can prevent some of the privacy threats of widely distributed protocols or application settings.

2 Protocols for network communication

The number of protocols which are currently used in networks such as the Internet is enormous[1]. Below we have selected a small sample of them. The selection was done in a way such that each layer of the Internet Protocol Model is represented by at least one protocol, but focussing on the most relevant Internet related protocols.

For each protocol we first briefly introduce its basic purpose and functionality. Thereafter we look at personal or identifying information, analyse the protocol's possible threats to privacy and data protection and discuss possibilities to circumvent them. The following criteria for elaborating privacy effects are taken into account:

- Identifiers and their uniqueness;
- Personal data being disclosed;
- Linkability: identifiability and profiling;
- Possibility of avoidance or circumvention of information disclosure.

These criteria are the baseline for the analysis of the selected protocols performed in this chapter. Note that in general we do this analysis by looking only at the protocol concerned – not at the protocols involved in lower layers. So when discussing protocols at a certain layer, we assume that all lower layers will not reveal or provide any identifying information. Of course when designing privacy-enhancing technologies and applications one has to consider carefully all layers involved (see also the FIDIS Deliverable D12.3 "A Holistic Privacy Framework for RFID" (Fischer-Hübner, Hedbom 2007)). Nevertheless in some cases we explicitly give advice and examples of how the identifying information or personal data provided at different protocol layers work together resulting in a much powerful threat to privacy compared to the simple sum of the threats arising form each layer.

2.1 Introductory facts on networking protocols

In this chapter, first the seven layers of the ISO/OSI reference model for protocols is introduced and complemented by the four-layer model used within TCP/IP. In Section 2.1.1 the criteria for the analysis performed in Chapter 2 are developed, dealing with identifiers, identifiability, linkability and disclosure of personal data. The reader's attention is directed to the possibility of hidden identifiers which may exist because of multiple reasons. The chapter is concluded in Section 2.1.3.

[1] See http://www.protocols.com/protocols.htm for a list of some of them.

2.1.1 Protocol layers

The ISO/OSI reference model, which is an abstract model to explain network architectures and protocol designs, defines seven protocol layers, which have different properties and functions.

The seven layers of the ISO/OSI model are:

- Layer 7: Application layer
- Layer 6: Presentation layer
- Layer 5: Session layer
- Layer 4: Transport layer
- Layer 3: Network layer
- Layer 2: Data link layer
- Layer 1: Physical layer

The basic idea is that all layers are independent from each other, e.g., a protocol at layer 2 takes input from layer 3, processes this input and hands it down to layer 1, but the functions of layer 2 are independent of those of layer 1 and layer 3. The only requirement is an input/output of a well-defined form. Therefore the ISO/OSI reference model clearly defines specified interfaces (cf. Tanenbaum 2003).

	Data unit	Layer	Function
Host layers	Data	Application	Network process to application
		Presentation	Data representation and encryption
		Session	Interhost communication
	Segments	Transport	End-to-end connections and reliability (TCP)
Media layers	Packets	Network	Path determination and logical addressing (IP)
	Frames	Data link	Physical addressing (MAC & LLC)
	Bits	Physical	Media, signal and binary transmission

Table 1: ISO/OSI reference model

The differentiation into the seven layers of the ISO/OSI reference model results in quite a complex protocol stack[2]. The ISO/OSI reference model is well suited for teaching network designs but was never adopted in practice.

[2] "A protocol stack (sometimes communications stack) is a particular software implementation of a computer networking protocol suite. The terms are often used interchangeably. Strictly speaking, the suite is the definition of the protocols, and the stack is the software implementation of them." [See Wikipedia, *Protocol stack*, http://en.wikipedia.org/wiki/Protocol_stack (as of Nov. 6, 2007, 21:13 GMT).]

Today's Internet architecture is based on the Internet reference model (also called TCP/IP model) which was developed before the ISO/OSI reference model and consists of only four layers. These layers are shown in Table 2.

TCP/IP layer	ISO/ OSI layer	Protocols									
Application	5-7	HTTP	FTP	SMTP	POP3	Telnet	HTTPS	DNS	SNMP	SSH	RTP
Transport	4	TCP						UDP		SCTP	
Internet	3	IP (IPv4, IPv6)						ICMP		IPSec	
Link / Physical/ Host-to-Network	1-2	Ethernet (CSMA/CD), WLAN, Token Ring, PPP, ISDN, Modem									

Table 2: Internet reference (TCP/IP) model

To achieve just four layers the functionality of the presentation and session layers were combined into the application layer and the physical and data link layers were combined with the host-to-network layer. Note that the general specification of the host-to-network layer is rather vague in the TCP/IP reference model - it is just stated that a host needs somehow to be able to connect to a network and send IP packets.

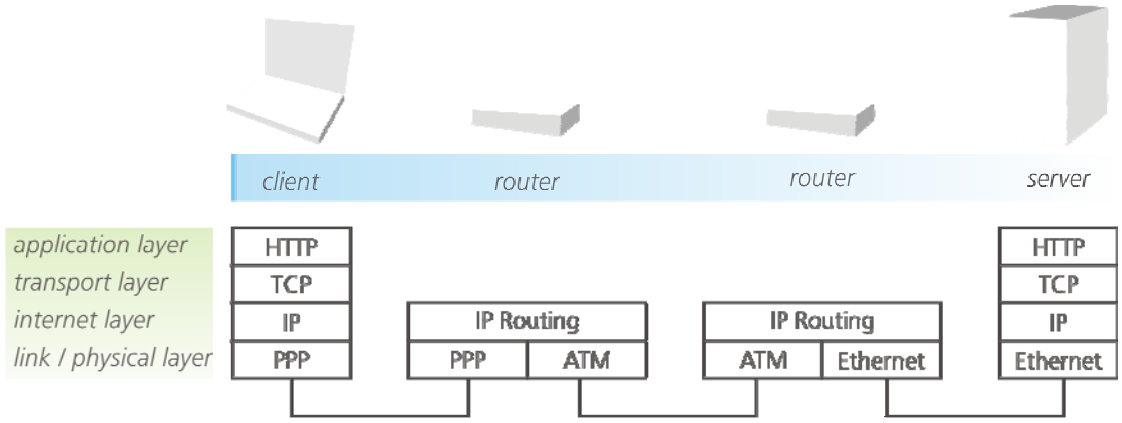

Figure 1: Example illustrating the involved layers of the TCP/IP model on various stages of and end-to-end communication

Figure 1 illustrates the orchestration of the four layers of the Internet protocol stack to establish communication between a client (in this case a web browser) and a server (in this case a web server). In this case we have an exemplary HTTP request from a client to a server via two routers.

2.1.2 Protocols – identifiers, identifiability, and personal data

First of all we need to clarify our understanding of the terms "identifying information" and "personal data". By "personal data" within a protocol we mean information transmitted in protocol runs which have a direct link to the user involved. An example would be if the

protocol requires the user to send his real name. Whereas the term "identifying information" denotes any information which could be used to link several protocol runs, i.e., it does not necessarily explicitly identify the user, but can be used to establish links between actions by the user or his machine using the communication network. Such information could for example reveal that two requests (which are otherwise independent) came from the same user, and so could be used to profile a user and his behaviour. Of course this profile in itself could be seen as personal data, and indeed if enough profiling information is available, the profile can reveal the identity of the user. More background information on profiling and its threats to privacy could be found in the FIDIS Deliverable D7.2 "Descriptive analysis and inventory of profiling practices" (Hildebrandt, Backhouse 2005).

In general, identifiers and identifying information within protocols which are usable to distinguish, identify or recognise machines, devices, applications and even users can be divided into two main classes: visible and hidden. Examples for the former are addresses used within protocols to address the sender or recipient of (protocol) messages. A well know and often quoted instance are the IP addresses used within the IP protocol to route packets from the sender to the recipient. Such obviously visible identifiers have the advantage (compared to the hidden ones) that it is much easier to become aware of the fact that they exist and to develop measures to circumvent them.

There are various reasons for the existence of hidden identifiers:

- **Protocol obscurities:** although protocols are specified and standardised (e.g., RFC[3] or ISO standards) the related documents and descriptions often purposefully do not cover every single detail of the protocol. For instance a specification usually just describes the meaning of protocol options and parameters without specifying explicitly an algorithm to calculate them. This is often left to the implementer of the protocol allowing him to adapt and optimise the protocol according to his needs. Also, specifications concentrate on the error-free runs of the protocol but cannot describe each possible fault situation and the related behaviour. This once again gives the implementer freedom of decision when implementing a certain protocol.
- **Misuse of protocol features:** nearly every protocol has parameters and options which can be freely chosen from a given domain. One can, for instance, use these parameters to distinguish between communication partners by using different parameter values when communicating with different partners.

  Example: Usually links within web sites (e.g., to subpages or embedded objects) should have the same address (URL) regardless of who is accessing the web site. But of course when delivering the web site the server can generate the link addresses "on the fly" so that each user gets different and unique addresses. If a user subsequently clicks on one of the unique links, the server will know that this has to be the user who

[3] Request for Comments (RFCs), which can be downloaded via http://www.rfc-editor.org/, are "technical and organisational documents about the Internet, including the technical specifications and policy documents produced by the Internet Engineering Task Force (IETF)" (http://www.rfc-editor.org/).

received the link beforehand. This way the server can track the whole browsing behaviour of a given user.

- **Manufacturing deviations:** Because of deviations introduced by the manufacturing process it is nearly impossible to build two devices with exactly identical (analogue) characteristics.
  Example: According to Lacknet *et al.* as well as Ellis and Serinken, wireless network adaptors can be distinguished according to their analogue signal characteristics (Lackner *et al.* 2006; Ellis, Serinken 2001).

The main problem with these hidden identifiers is that the possibilities of how to utilise the capabilities mentioned above for identification or distinction are endless. Therefore it is hard (if not impossible) to avoid all of them. The research in the past decade in the field of privacy-enhancing technologies has shown that whenever someone has believed to blur all identifying information, somebody else has found a new way to identify or link communicating parties.

2.1.3 Conclusion

This section has introduced commonly used layer models for categorising network protocols. These models show that usage of higher-layer protocols involves also usage of protocols on layers below, yielding possible dependencies in the analysis of privacy criteria. However, due to a reduction of complexity, such cross-layer effects are not tackled in this deliverable. However, the layers structure forms the basis for the analysis later on in this chapter.

Furthermore, this chapter has dealt with basic properties of potentially privacy-relevant data in protocols, i.e., identifiers, disclosure of directly personal data, and linkability enabling identifiability and/or profiling. It has also been stated that hidden identifiers may exist in protocol implementations.

2.2 Application layer protocols

The protocols at the application layer are familiar to many people, at least by name. HTTP is one of the most popular protocols. It is used when surfing the web, indicated by the prefix "http" in front of web page addresses. Also well known are the SMTP and POP3 protocols, which are used for e-mail access, and of course FTP, which is a file transfer protocol.

DNS is one of the backbone protocols of the modern Internet. It provides mapping between easy-to-remember textual host names like www.fidis.net and not-that-easy-to-remember IP addresses like 80.237.131.150. The importance of the real-time transport protocol RTP is growing steadily, especially now with Voice over IP and similar applications.

2.2.1 HTTP

2.2.1.1 Functional description

HTTP stands for "HyperText Transfer Protocol". It is a method used to transfer or convey information on the World Wide Web. Its original purpose was to provide a way to publish and retrieve HTML pages (Wikipedia: Hypertext Transfer Protocol 2007).

The HTTP protocol[4] is standardised by the Internet Engineering Task Force (IETF) the most popular version HTTP 1.1 is defined in RFC 2616. HTTP is a request/response protocol used between clients (also called user agents) and servers. A client is an entity sending a request to a server. The server answers with a response. The client requests information (or data) from the server by initialising a connection, normally a TCP connection[5]. The server processes the request and answers with a status message (e.g., "HTTP/1.1 200 OK") and the requested data. Resources which can be accessed by a client are identified by so called Uniform Resource Identifiers (URIs). A well-known subset of URIs are URLs (Uniform Resource Locators), which are used for HTML documents, for example.

There are eight request methods defined for HTTP, e.g., GET, POST, HEAD, DELETE. The communication between client and server is unencrypted. The protocol information is transported via human-understandable header fields and header values. Here, "human-understandable" means that a human can easily conclude which protocol information is exchanged just by looking at the raw protocol data.

2.2.1.2 Identifiers and their uniqueness

HTTP is a stateless protocol. This means that each request and response is independent of any former or later requests or responses. For example, a server will process two subsequent requests by a client for two images which are part of the same HTML page totally independent of each other. Likewise, the responses by the server are independent. Because of this stateless approach (e.g., no "session" information needs to be stored to link different requests) the very basic HTTP protocol itself contains only the URL as an identifying property. However, there are several ways to integrate additional identifying information:

1. Cookies
   If not explicitly forbidden by the user, any web browser allows web servers to set cookies. Cookies are small text files containing a limited amount of data. They are stored on the client machine of the user. If the user re-enters a web site which sets a cookie beforehand, that cookie will be automatically transferred to the web server. To track and identify users, the server can store a unique ID in a cookie. By means of that ID, a server can recognise a user and trace its activities. Because the cookie can store only a small amount of data, the whole trace and related profiling information would be stored in a database on the web site. Normally this tracing would only work within the domain of the web site, as cookies are only sent back to the server which set them. To enable cross-site tracing, advertisement companies, such as DoubleClick Inc. (now acquired by Google) sends a cookie together with their web advertisements (e.g., pictures), which are displayed on many internet sites. Since these advertisement objects are all severed from the same server (even though the site being viewed maybe different) the server can read the ID stored within the cookie. As the advertisement

[4] We use the two notations "HTTP" as well as "HTTP protocol" although the latter notation might look tautological. However, we prefer it over "HTT protocol" as that may be harder to read than the well known abbreviations (also just think of "I protocol" instead of "IP protocol").

[5] Details on the TCP protocol are given in Section 2.3.1.

company knows on which web site which advertisement is shown it can trace the surf behaviour (also known as "clickstream") of a user even across multiple web sites.

2. Session Ids

   Most servers support URL rewrite, which can be used to append a parameter to each link on web pages. Whenever the user clicks on such a link, the parameter is transmitted along with the requested resource to the server, thus allowing session management. The parameter often has the following form:

   http://www.someURL.org/text.html?sid=sdf3s3rf39asdlv974

   The last part of the URL, i.e., sid=sdf3s3rf39asdlv974, is the session ID identifying the current session.

3. Hidden form fields

   Forms on web pages can contain hidden fields which are by default not visible to the user. The hidden fields usually cannot be edited from within the browser and contain additional data for the server, like a session ID. This data is transmitted with the rest of the form whenever this form is submitted by the user.

This list only contains some examples of how identifying information can be added to the HTTP protocol to allow tracking and tracing of users. Many more possibilities exist, especially when used in conjunction with the web (e.g., HTML documents).

2.2.1.3 Personal data

HTTP does, to some extent, contain personal data. HTTP has header fields like the *Referer*, the *User-Agent, Accept* and *Accept-Language*, which might contain information which reveal personal data.

HTTP field	Description
Referer	The URL of the previous web page. Note that sometimes (as mentioned above) the URL might contain additional parameters like the user name used on the previous web page.
User-Agent	Information about the browser, operating system, hardware platform etc.
Accept	The data types the browser accepts, e.g., "text/xml", "image/png"
Accept-Language	The preferred language of the user.

Table 3: HTTP fields which contain personal data

Table 3 shows that HTTP headers contain personal data, though weak this may seem, fields like the *Referer* can contain quite sensitive data. Also the *User-Agent* can reveal interesting information: if the operating system of the user is "Linux" it might identify him as technically

interested; if the version of the browser he uses is the latest available version (or even a "test" version) it might identify him as belonging to the group of "early adaptors" etc. Of course also the preferred language of the user as transmitted by the *Accept-Language* header field will reveal personal information (like cultural background) of the user.

2.2.1.4 Linkability: identifiability and profiling

The server can identify the user within a session by using cookies, session IDs or hidden form fields. But to map the available session ID to real user data (like the user name, address, buying habits etc.), the user has to provide some personal data to the server, first. This could happen when a user registers an account and logs in later.

Although unlikely, (probably unreliable) identification could take place by using a combination of available HTTP header fields like *User-Agent*, *Accept* and *Accept-Language*.

2.2.1.5 Avoidance or circumvention of information disclosure

The session-spanning, reliable identification by cookies or session IDs can be avoided if the user does not give any data to the server. Otherwise, for example by, opening an account at an online store, giving away personal data like address and credit card information and logging in later with the provided data, then the server can identify the user. Other than this, identification by the HTTP protocol alone is not likely, as described above.

It has to be noted though that long-living cookies can weaken the privacy of a user. Remember the example of the advertisement companies like DoubleClick or Google, which have a widespread network and which can collect information from many sources. Cookies created by companies like these with a lot of accessible data can lead to the identification of single users. To avoid this, cookies should be deleted each time the browser is closed. In this way, the lifetime of cookies is reduced severely, which makes it harder to identify a user.

In order to enhance privacy, user agents like browsers should only provide what is strictly required in the header fields. For example, the *User-Agent* field is usually not needed for surfing the web. Further on, the *Referer* should be disabled whenever possible.

There exist many software implementations – commercial and non commercial ones – which help the user to enhance his privacy while surfing the Web. One example of such filtering software is called "Privoxy". "Privoxy is a web proxy with advanced filtering capabilities for protecting privacy, modifying web page data, managing cookies, controlling access, and removing ads, banners, pop-ups and other obnoxious Internet junk."[6]

2.2.2 FTP

2.2.2.1 Functional description

FTP is used to transfer data (files) between a client and a server. Clients connect to a FTP server in order to manipulate files, uploading or downloading them, renaming or deleting

[6] http://www.privoxy.org/.

them etc. The FTP protocol runs exclusively over TCP, UDP is not supported. This makes sense, since UDP does not guarantee faultless transmission of data. Commands from the client to the server are sent over one connection to a certain port, data is sent via another connection. The control connection is idle while data is transferred. The objectives of FTP, as outlined by its RFC 959, are described as follows (Wikipedia: File Transfer Protocol 2007):

1. To promote sharing of files (computer software and/or data).
2. To encourage indirect or implicit use of remote computers.
3. To shield a user from variations in file storage systems among different hosts and platforms.
4. To transfer data reliably, and efficiently.

Some well-known weaknesses of the FTP protocol are:

- The user name and password are sent in clear text from the client to the server.
- Multiple connections are used.
- A relatively high number of commands is needed to initiate file transfer, thus leading to a high latency.

Many FTP servers enable anonymous FTP, meaning that users can access (parts of) the server without a specified user name and password combination. The user name for an anonymous login is usually “anonymous”.

2.2.2.2 Identifiers and their uniqueness

The FTP protocol uses persistent connections, i.e., the client and server negotiate one or more ports over which they communicate. At least the connection for the control stream will remain open until the user logs out. Thus the server can trace the user’s actions, like which files have been downloaded. If the user logs in anonymously, the server has to track the user-interaction via its IP, which is given by the underlying network protocol TCP. The FTP protocol does not use any specific identifiers, except the log-in.

2.2.2.3 Personal data

If anonymous FTP access is used, the protocol does not contain any personal data itself, but underlying protocols do (like the IP address). When no anonymous log-in is used, the log-in data (i.e., user name and password) are personal data which is transferred to the server.

2.2.2.4 Linkability: identifiability and profiling

Users are identified via their user name and password, or, to be more precise, the user name and password are necessary to open a connection, which stays open as long as no time-out occurs. This open connection can be used to profile the user action, i.e., which directories have opened, which files up- or downloaded etc.

2.2.2.5 Avoidance or circumvention of information disclosure

The username/password can only be avoided by providing an anonymous log-in. If access to the data has to be controlled, some kind of identifier (authorisation token) has to be presented by the user.

In order to prevent eavesdropping on the connection, FTP can be enhanced by using an encrypted connection (like SSL). Thus the communication between client and server is run through a virtual tunnel, meaning that all requests and responses are encrypted. The FTPS protocol provides such measures. If FTPS cannot be used, IPSec and similar protocols at lower layers are available to protect the data.

2.2.3 SMTP

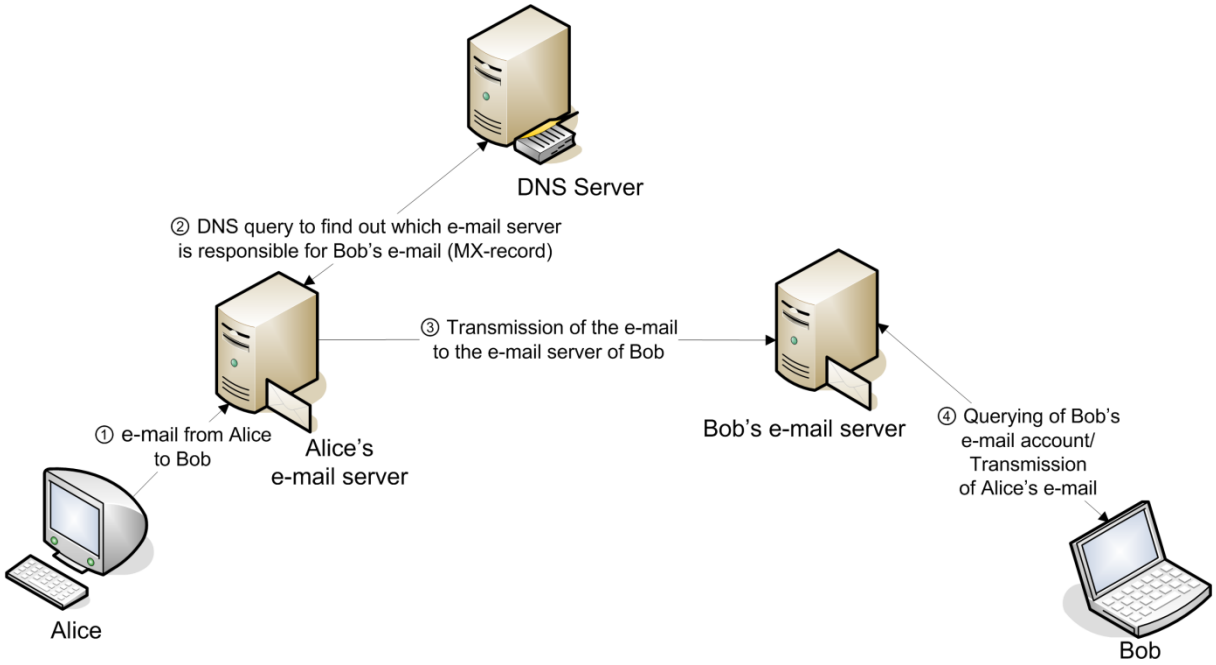

Figure 2: Example illustrating the protocols involved in an e-mail transmission

SMTP is a text-based mail protocol which offers push services by utilising a store and forward mechanism. SMTP is used by clients for sending plain text-messages to servers. Servers can forward text-messages to other servers via SMTP. To fetch e-mails, different protocols are used, like POP or IMAP. The general procedure and the different protocols involved are illustrated in Figure 2. In Step 1 and 3 the SMTP protocol is used, Step 2 is done by the DNS protocol and Step 4 by an e-mail fetching protocol like POP3 or IMAP.

Client	Server
	220 mail.example.com SMTP Foo Mailserver
HELO mail.example.org	
	250 Hello mail.example.org,
MAIL FROM:<hans.muster@example.org>	
	250 Sender OK
RCPT TO:<foo@example.com>	
	250 Recipient OK
DATA	
	354 End data with <CR><LF>.<CR><LF>
From: <hans.muster@example.org> To: <foo@example.com> Subject: Testmail Date: Thu, 26 Oct 2006 13:10:50 +0200 Testmail .	
	250 Message accepted for delivery
QUIT	
	221 See you later

Table 4: Sample SMTP conversation between e-mail client and server

An example of a run of the SMTP protocol between client and server is illustrated in Table 4 (Wikipedia: Simple Mail Transfer Protocol 2007). As can be seen in this example, no authentication takes place. Even so this can be seen as a good property from a privacy point of view it also poses a problem, since SMTP is a popular and widespread protocol, supported by many mail servers. Spammers can use this to send spam over open SMTP servers, which do not authenticate their clients. These SMTP servers are called “open relays”. To prevent this, SMTP extensions have been developed, like SMTPAuth or SMTP-After-POP. These extensions require some kind of authentication, in order to reduce the misuse of open SMTP servers by unauthorised users.

2.2.3.1 Identifiers and their uniqueness

The basic SMTP protocol *per se* contains no identifiers or identifying information. The sender has to state his domain, but the server cannot verify that the sender is really sending from this domain and has valid credentials. So an e-mail sent over SMTP could contain any sender address imaginable. The recipient of the e-mail has to be identified correctly; otherwise the e-mail cannot be delivered.

But in contrast to the first statement, SMTP might reveal a lot of identifying information if it is used in the usual way. Normally neither sender nor recipient will change their e-mail addresses for every e-mail they send/receive. Therefore one can link several SMTP protocol runs based on the sender or recipient e-mail addresses given. Moreover it is possible to send

more than one e-mail within a SMTP session. The SMTP sever can then conclude that all the e-mails transmitted within a single SMTP session are sent by the same user.

If SMTP-After-POP or SMTPAuth is used, the SMTP server can verify the sender by some credentials (like user name and password), thus identifying the sender and validating its given sender address. These authorisation credentials can also be used to link multiple SMTP session to the same sender (user).

2.2.3.2 Personal data

The content of an e-mail can be seen as personal data. If the content is not encrypted, it is sent in clear text and can be "read" by any SMTP server forwarding the e-mail.

Furthermore, the sender and receiver address are both personal data, more so in conjunction.

2.2.3.3 Linkability: identifiability and profiling

If the users have to authenticate themselves reliably against the server, servers can create certain profiles of the e-mails sent. Every forwarding SMTP server involved in sending an e-mail can log this data, e.g., the originator, the recipient, the date, etc. Furthermore, since the text is sent in plain text, every server involved can "read" the content of the e-mail.

Even if the content is encrypted, the header is sent in plain text, so every server can create profiles for e-mails sent. The connection from the client to the first SMTP server can be secured, e.g., by SSL or similar tunnelling protocols, but this will only secure the integrity and confidentiality to the first server. If the e-mail is forwarded to other servers, a server-to-server SSL connection is required in order to protect the e-mail. It has to be noted that the e-mail still exists in plain text on each server.

2.2.3.4 Avoidance or circumvention of information disclosure

Providing the real recipients address cannot be avoided without (complex) extensions like re-mailers, which are introduced in the next section. If the user has to authenticate himself against the first SMTP server, the sender's address can be verified. A user can prevent this by using SMTP servers which require no authentication, i.e., open relay servers.

The content of an e-mail can be secured using either symmetric or asymmetric cryptography, thus obtaining confidentiality and integrity of the data.

2.2.4 POP

POP is an acronym for "Post Office Protocol", the third and currently most used version is POP3 as defined in RFC 1939. POP uses the TCP protocol to retrieve e-mails from a remote server. POP3, in contrast to SMTP, uses a pull mechanism to get the e-mails from the server. Thus, POP3 supports users with dial-up connections who are not online all the time. E-mails retrieved via POP3 can either be transferred to the client's computer and then get deleted on

the server, or they stay at the server. POP3 supports MIME to send non-ASCII attachments with e-mails, like (binary) images.

A client authenticates himself with a user name and password to the POP3-server. This data is normally sent unprotected in plain text. POP3 extensions like APOP encrypt the password before sending it from the client to the server for authentication.

2.2.4.1 Identifiers and their uniqueness

POP3 requires the client to authenticate himself via a user name and password. This can be used as an identifier.

2.2.4.2 Personal data

If the e-mail is not encrypted, personal data in the e-mail body can be read by the server and each other server forwarding an e-mail. The header cannot be encrypted, thus the sender and recipient identity, which must be valid, are readable for each processing mail server. Thus, a communication can be reconstructed easily.

2.2.4.3 Linkability: identifiability and profiling

The POP3 server knows about all the communication of its clients. Clients can only protect the text content by encrypting it, but the recipient address must be readable for the mail server in order to deliver the e-mail. Every mail server in between can read-out the header, i.e., get to know who sends an e-mail to whom.

2.2.4.4 Avoidance or circumvention of information disclosure

The encryption of the e-mail body (text content) can provide confidentiality and integrity. The communication between two clients cannot be hidden; at least the recipient address must be valid in order for the mail servers to deliver the e-mail correctly.

In order to send an e-mail without any type of return address, i.e., sending an e-mail which the receiver cannot associate to the sender by means of the header, services exist which strip the header of e-mails and redirect them to the intended target. These services are called (anonymous) remailers.

Remailers cannot guarantee privacy for the sender though. Next-level remailers, called mixmaster remailers (or “type 2” remailers), are more secure. Mixmaster remailers use advanced techniques to avoid tracing of e-mails, but usually it requires special client software to use these services. Security advances compared with normal remailers are for example that each sent e-mail has the same size, is encrypted and messages are sent through a couple of remailers (chaining) before being delivered to the recipient. Further information can be found at (Wikipedia: Mixmaster-Remailer 2007).

Remailers provide anonymity for the sender of a message. In order to get recipient anonymity, another approach is needed. Nymserver take messages with a remailer as “first” receiver.

Additionally, an encrypted data block in the e-mail contains a symmetric key, the address of a second remailer and another encrypted data block. The first remailer, being able to decrypt the data block, can re-encrypt the message with the new symmetric key and send this and the received encrypted data block to the second remailer. This remailer decrypts the encrypted block, obtains a symmetric key, the address of a third remailer and another encrypted block. The system continues until one remailer at the end sends the e-mail to the recipient. The intended recipient decrypts the message with all given symmetric keys and gets the plain text content. This system requires a sophisticated infrastructure, where the key management (i.e., the spreading of the needed symmetric keys) especially poses a big problem.

2.2.5 DNS

The Domain Name System (DNS) is a hierarchical infrastructure for name resolution on the Internet. It allows the mapping of (numerical) IP addresses to user-friendly textual addresses, the well-known host/domain names.

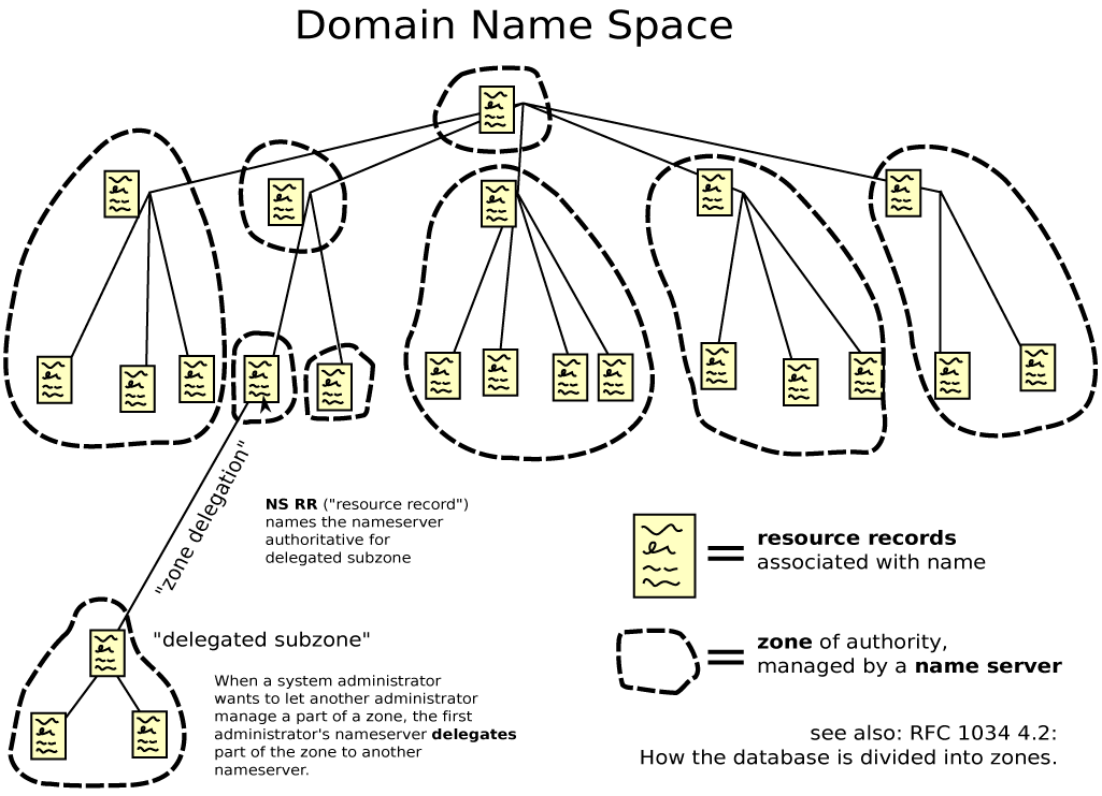

Figure 3: Architecture of the Domain Name System

The DNS system can be visualised as a tree (see Figure 3[7]). Each node and leaf holds at least one resource record, which itself holds information about the associated domain name. The

[7] Figure taken from Wikipedia: http://en.wikipedia.org/wiki/Image:Domain_name_space.svg.

DNS tree can be divided into sub-trees, called zones. Each zone consists of a collection of connected nodes and leafs. A zone is managed by a designated nameserver called "authoritative nameserver". If a nameserver is queried for a domain it is not responsible for (i.e., the queried nameserver cannot resolve the request), the nameserver can forward the query to another nameserver and its answer to the originator (recursive querying). Another approach is to return the address of a list of different nameservers to the user when a query cannot be resolved (non-recursive querying). Upon a positive answer from a nameserver, the querying nameserver can cache this answer for further use in order to reduce latency and load (Guha, Francis 2007). A sample DNS query is illustrated in Figure 4

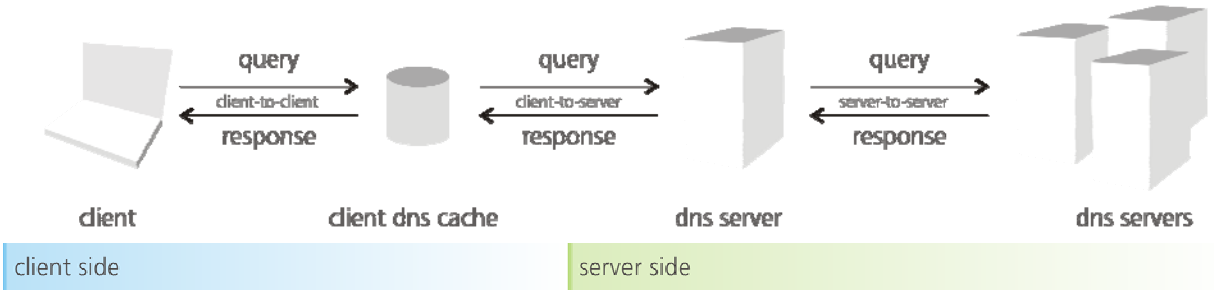

Figure 4: Sample DNS query

For each top-level domain there is at most one registry, for the German top-level domain ".de" the registry is the DENIC. The registries regulate the assignment of the possible mappings. A domain holder, also called a registrant, has to "lease" a domain from the registry for a certain fee.

With DNS, two types of affected entities must be distinguished:

1. Owner:
   An owner is a legal person having a server with an IP address and a domain name which the owner wants to register with the Domain Name System in order for users (see below) to access the server and/or its services.
2. User:
   A user is an entity (often a person) requiring a mapping, e.g., between domain name and IP address. This is needed for example when surfing the web and using domain names which have to be resolved to IP addresses.

The following sections will differentiate between the two affected entities "owner" and "user".

2.2.5.1 Identifiers and their uniqueness

Owners

As an owner of a server one has to register requested host name with the server's IP address with at least one DNS server so that the human readable addresses (like URLs) of the domain

can be resolved. The registration process requires the usage of real world information by the owner, like name, address, contact data, etc.

The mandatory registration data is published by the registry in a so called WHOIS database. This database contains information about each and every domain holder, and can be accessed with the help of the WHOIS protocol by everyone.

Relevant for the case where an owner wants to register a mobile device (e.g. a laptop) is the fact that a registered IP address to a given domain name may reveal additional information about the mobile device, especially his current location. If a so called dynamic DNS service like DynDNS[8] is used, which allows mobile devices with changing IP addresses to be reachable under a static domain name, the associated IP address may reveal information about the current location of that device. Services like DynDNS can normally be queried by everyone without any restriction (Guha, Francis 2007).

Users

A typical DNS query contains the information shown in Table 5.

Fieldname	Description
name	Either the (fully qualified) domain name or the IP address to resolve.
type	The type of the query, either *:a* for resolving a hostname to an IP address, or *:ptr* for the opposite.
class	A specified class for the DNS domain name.

Table 5: Fields of a DNS query

At first glance none of this data seems to contain any identifying values, at least none which can identify the entity sending the request. But at least the first DNS server knows who is sending this DNS query by underlying protocols like UDP providing information like the IP address. Thus a relation can be built between sender and requested host. In many cases the ISP of the requesting user maintains the first DNS server in order to reduce response time and to optimise by using a local cache.

2.2.5.2 Personal data

Owners

As described above, the registration of an IP address and/or domain name requires personal data about the registrant. Mandatory are fields like name, address, administrator contact information etc.

[8] DynDNS: http://www.dyndns.com/.

As described above, for mobile devices with a dynamic IP address but a static domain name, the IP address can reveal the location and thus information like movement patterns of the mobile device and consequently of the owner of the device (Guha, Francis 2007).

Users

In the usage of the DNS a user reveals which host-names or IP addresses he is interested in. Depending on them this could be seen as leakage of personal data (e.g., if www.aids.org or similar addresses are queried).

2.2.5.3 Linkability: identifiability and profiling

Owners

A lot of information can be gained by searching a WHOIS database. Not only is it possible to find out the owner of a certain domain name, but also all the domains a certain person/entity has registered. The location information for mobile users may lead to location profiling.

User

Primarily the first DNS server a user utilises for resolving domain names can collect a lot of data about the user. This DNS server is queried each time a mapping is needed (e.g., each time a new, unknown domain is used by the user), thus the DNS server can store the requests alongside the IP address of the user. If the DNS server is administrated by the ISP of the user, the ISP can easily link this data directly with all available personal data, like name, address etc.

2.2.5.4 Avoidance or circumvention of information disclosure

Owner

It does not seem to be necessary for the personal information about the owner of an IP address or domain name to be accessible publicly. At least a proxy should be available which masks this information up until a legitimate interest exists. A proxy could be a lawyer or a company which registers the domain on behalf of the user.

There are efforts to reform the registration process in order to provide more privacy to owners of domains and IP addresses. An ICANN key task force created a proposal in order to give more privacy options to domain name owners. The goal is to make requests for personal information of domains name owners much more expensive than they are now. This could be reached by giving the users the possibility to list e.g., a lawyer or a service provider as the contact person (Jesdanun 2007). Roessler gives additional information about DNS, WHOIS and privacy (Roessler 2002).

To avoid the leakage of location information for mobile devices, the DNS system has to be modified in order to prevent arbitrary people from requesting IP addresses to certain (private) domain names (Guha, Francis 2007).

User

Without further extensions it cannot be avoided that at least the first requested DNS server gets to know the sender of the request, because this DNS has to resolve the request, meaning it has to know the requested information (like IP address linked to a given domain name), and it has to send an answer to the correct user, meaning the DNS server needs to know the IP address of the user.

What can be done is the use of anonymisation services like "Tor" or similar approaches which (try to) hide the sender and maybe also the receiver of a message from eavesdroppers. Such services require a sophisticated infrastructure and often result in a high delay.

DNSSEC can be used to sign requests and responses from the client to its DNS server. What DNSSEC adds is primarily more secure name look-ups and reduced risk for manipulation of information and forged domains. But DNSSEC does not encrypt the DNS query itself in any way, so it does not achieve confidentiality. This means, there is no privacy in DNS queries:

> *"Most mobiles access a DNS server provided by the access network, which is typically configured with the DHCP protocol. The DNS server is able to record the names of the online servers contacted by the mobile. Even if the mobile connects to a VPN gateway and uses DNS services via a VPN tunnel, it may still rely on the local DNS server to resolve the VPN gateway name. This means that the DNS server in the access network learns the name of the organization to which the user belongs, and may enable it to identify the mobile with some accuracy. Furthermore, DNS requests are made recursively, which leaks the mobiles approximate location to the remote DNS servers. Thus, even if the actual data connections are forwarded via anonymizing proxies, the source of a DNS request may reveal the mobiles location to the peer endpoint." (Aura, Zugenmaier 2004)*

2.2.5.5 RTP

RTP is the acronym for Real-time Transport Protocol, which defines a standard for delivering real-time information (like audio or video) over the Internet. In today's Internet it is of growing importance since it is used as transport protocol for Voice over IP (VoIP).

RTP uses two communication channels, one for the control information (via the "RTP Control Protocol" RTCP which uses TCP) and one for the data, normally using UDP. The session establishment, i.e., primarily the call setup and tear-down, is managed by an extra protocol, e.g., SIP (Session Initiation Protocol), best known from Voice over IP usage.

According to RFC 1889, the first RTP standard, the services provided by RTP include (Wikipedia: Real-time Transport Protocol 2007):

- Payload-type identification – Indication of what kind of content is being carried;
- Sequence numbering – PDU (Protocol Data Unit) sequence number;
- Time stamping – allow synchronisation and jitter calculations;
- Delivery monitoring.

The following sections will distinguish between RTP and its control protocol RTCP.

2.2.5.6 Identifiers and their uniqueness

RTP

The RTP packets themselves do not contain any direct privacy relevant information, only media information. Of course RTP uses UDP, which itself contains the IP address of the sender of the packet.

RTCP

More privacy relevant information than in the RTP protocol is contained in the RTCP packets, i.e., the source description packets, which can include personal data. RTCP is not mandatory for RTP, but it helps the sender to synchronise and optimise a RTP stream. Although not mandatory, some media providers using RTP may require the use of RTCP.

The RTCP may contain data like the names and affiliations of participants in a communication. This data is user-defined, it depends on the application whether users can control the sending and the content of this information or not. Furthermore, RTCP sends a canonical name (CNAME) with each packet. This CNAME includes the IP address of the sender and the user name of the participant. The IP address is a unique identifier to a certain level, the user name is at least unique within a session. The uniqueness is dependant on the frequency of name changes a user applies.

2.2.5.7 Personal data

RTP

A RTP packet contains no personal data in itself, but of course it contains the media data, which itself may contain personal data, like voice data from a phone conversation.

RTCP

The RTCP may contain personal data like the names and affiliations of participants in a communication, but the provision of this data is optional. The usage of the CNAME field however is mandatory. This CNAME includes the IP address of the sender and his user name. Both can be seen as personal data, especially the IP address, which is often static. Even if the IP address is dynamic, it can leak interesting information like the location of the user. The IP address is also available from the TCP packets sent. But an important issue arises, if the user

works in an intranet with network address translation (NAT). NAT hides the user's internal IP address (i.e., the IP address in the intranet) with an external IP address. This enables many users to communicate with only one external IP address. But the IP address used in the CNAME field is the internal address, thus the receiver obtains both the internal and the external IP address.

2.2.5.8 Linkability: identifiability and profiling

RTP

Profiling can take place by eavesdroppers who analyse traffic which reveals who is talking to whom and when. This is possible because RTP neither hides the sender's nor the receiver's IP address, which is needed for delivering the communication data. Even if both IP addresses are dynamical, they are static within a (communication) session, so profiling is possible at least for one session.

RTCP

RTCP leaks the IP address of the sender of RTCP packet by the use of the TCP protocol. Additionally, the CNAME field leaks the IP address of the sender and his user name. This eases profiling, because there are two fields which can be observed.

2.2.5.9 Avoidance or circumvention of information disclosure

RTP

To protect the confidentiality of the sent media data, i.e., the conversation content, the RTP payload (data) has to be encrypted. The RTP standard provides support for both RTP and RTCP encryption. To encrypt RTP data packets, the payload may need some padding in order to have a length which is supported by the used encryption method (e.g., DES). This is illustrated in Figure 5 where in step 1 the padding is done and in step 2 the encryption takes place (cf. Perkins 2003). Typically used ciphers for encryption are DES (not considered as secure anymore), triple DES or AES.

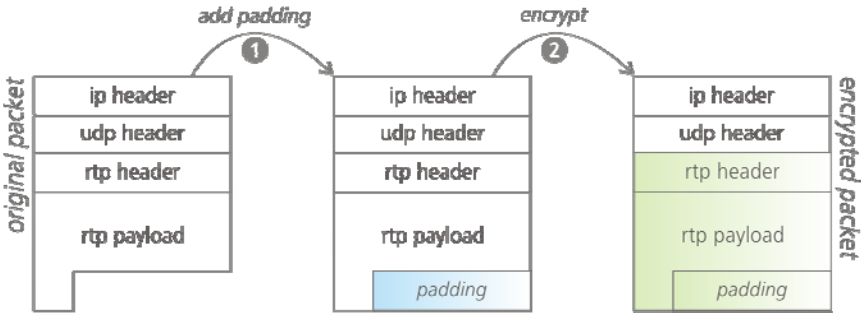

Figure 5: Encryption of the RTP payload

Alternatively SRTP (Secure RTP) can be used to provide confidentiality, and also authentication. Both SRTP and RTP with encryption rely on a secure key exchange via external protocols like MIKEY (Perkins 2003).

Another issue is the confidentiality of the circumstances of a communication, i.e., the information that a conversation is taking place at all. This problem cannot easily be solved as long as IP is used as the underlying transport protocol. One possibility is to use anonymisation services like Tor or AN.ON (cf. Section 3.1). However, these solutions add extra latency. As low latency is one of the key factors for quality of service concerning real-time communication, this prevents the adoption of anonymisation services in most cases.

RTCP

The RTCP may contain personal data like the names and affiliations of participants in a communication. This might not be a problem for a teleconference within a company, but it could be inappropriate for someone listening to a radio stream. As the transmission of the personal data is optional, applications should let users control the usage of this "feature".

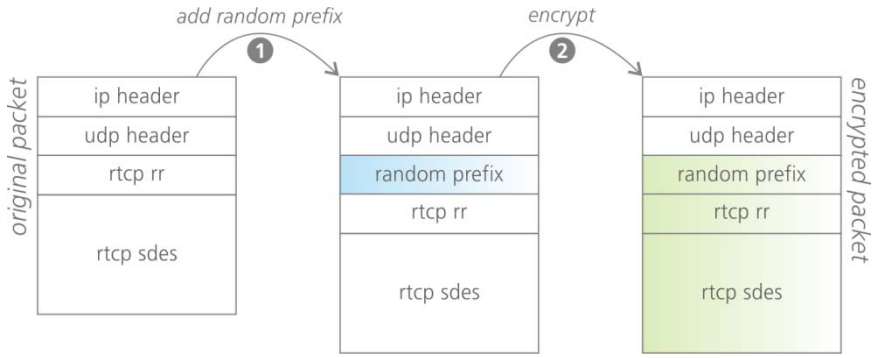

Figure 6: RTCP encryption

To provide confidentiality for the instructions and information contained in RTCP packets, encryption can be used, as depicted in Figure 6 (cf. Perkins 2003). In the first step a random prefix is added. This is necessary, since many fields of a RTCP are static and well-known, thus easing attacks. The second step involves the encryption of the random data, the RTCP receiver report (RTCP RR, information about the received data which is important for QoS adjustments for the sender) and the RTCP sender description (RCTP SDES).

The header fields IP header and UPD header cannot be encrypted, since this is information needed for the delivery of the RTCP packets. Thus traffic analysis is still possible - only the content of the RTCP packets can be protected by the encryption.

2.3 Transport layer protocols

The third layer in the protocol hierarchy contains protocols like TCP and UDP. These two protocols are the foundation for many other protocols from the higher layers. TCP is used for

reliable delivery of data streams, whereas UDP is mainly used for the fast, but unreliable transport of single data packets.

2.3.1 TCP

The Transmission Control Protocol (TCP) is one of the core protocols of the Internet protocol suite, supporting many important Internet protocols such as HTTP, FTP and POP3. TCP guarantees reliable (received packets are acknowledged by the receiver, else, they are sent again) and in-order (by the use of sequence numbers) packet delivery, thus making it the protocol of choice for applications, where loss of data is not acceptable.[9] Applications can send a stream of data via so called stream sockets to the TCP layer, which divides the stream of data into appropriately sized segments of data. These "data chunks" are extended by checksums, which allow the receiver of the data to check for corrupted packets. Note though, that the TCP checksums are in no way cryptographically secure. They just detect data corruption caused by the network, not by an active attack.

The TCP protocol allows the dynamic adaptation of the speed with which packets are sent. This is called congestion control. The TCP packets are passed to the next lower layer, the IP layer.

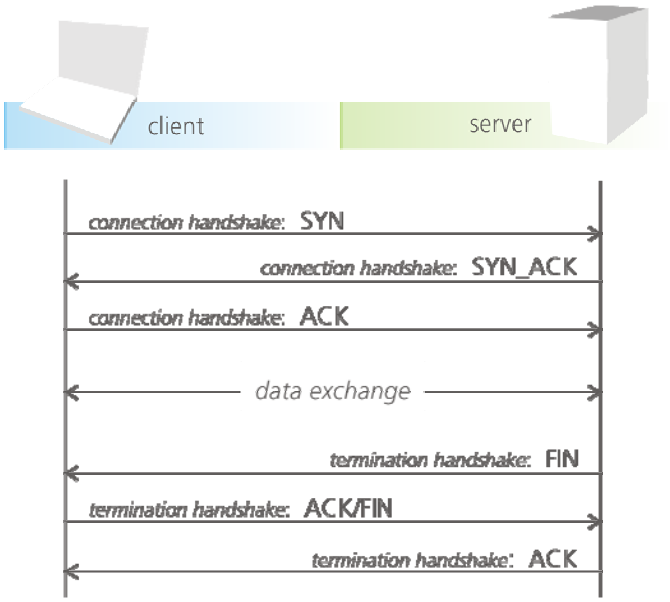

Figure 7: Three-way handshake in TCP

In order to send data via TCP, a connection between the sending and the receiving computer has to be established. A three-way handshake is used for this, as illustrated in Figure 7 which shows connection establishment, data exchange and two-sided connection termination. The termination of the session is realised by a handshake too (Wikipedia: Transmission Control Protocol 2007).

[9] UDP in contrast is a lousy protocol, but usually significantly faster than TCP. For details see section 2.2.2

Table 6 shows the header (bits 0 to 160/192) of a TCP packet.

Bits	0-3	4-7	8-15	16-31
0	Source Port			Destination Port
32	Sequence Number			
64	Acknowledgement Number			
96	Data Offset	Reserved	Flags	Window
128	Checksum			Urgent Pointer
160	Options (optional)			
160/192+	Data			

Table 6: TCP header

The source and destination port specify the port which is used for the data transfer. The sequence number is used for controlling the order of the sent packets. The acknowledgement number is a kind of receipt to acknowledge a received packet. The checksum is used to check for corrupted packets, the urgent pointer indicates packets with high priority. The options fields can contain non-mandatory information. Their usage is application dependant.

2.3.1.1 Identifiers and their uniqueness

The TCP protocol itself contains no data which can be used to identify a user (except such information is contained within the (unencrypted) data part of the packet). In addition to the lower level IP protocol, this is not true any more. The source and destination ports in cooperation with the IP address of the sender and receiver can identify both participating parties.

TCP is a good example for identifying information introduced by the “protocol obscurities” as described in Section 2.2 Although the meaning of the Sequence number field is defined – the initial value is not. It is up to the implementer how the initial sequence number is chosen (e.g., randomly). The same is true for instance for the Window field used for congestion control. As the congestion control has a key influence on the overall performance of TCP, many attempts to optimise them are done e.g., by operating system manufactures.

All these implementation depended information could be used for so called TCP/IP stack fingerprinting. The goal is to guess which TCP/IP stack and operating system is running on a remote machine. TCP/IP fingerprinting (as many other fingerprinting methods) can be done actively or passively; observing or modifying. A passive observing attempt would just listen on the networks links and tries to guess the TCP/IP stack from the eavesdropped data packets. An active observing attempt would also send data packets. The attacker would intelligently choose the data packets he sends to concluded as much information as possible from the remote host. But as it is still an observing attack the attacker would behave according to the protocol. Of course he could also violate the protocol rules. This typically gives him more possibilities for guessing the TCP/IP stack – but would make his attack also more obvious.

The software *Nmap* ("Network Mapper"[10]) is a well-known representative for tools which are able to do TCP/IP fingerprinting. According to the developers, Nmap currently has more than 1500 fingerprints in its database.

Of course the identifying information described above cannot only be used to guess the TCP/IP stack and operating system used but can also be used to link different TCP connections, especially if the guessed TCP/IP stack or operating system is a rather unusual one. Note that the information mentioned above can also be used to decide if TCP packets belong to the same TCP stream or not. Imagine for instance the situation that an attacker can monitor network traffic on different locations of the network whilst he cannot easily "follow" the TCP packets as an anonymisation service (operating at the IP layer) is used by the communication partners. Nevertheless chances are high that the attacker can still conclude who is communicating with whom just by looking at the TCP information.

2.3.1.2 Personal data

The ports can leak information about the application used without looking at the data content. This is possible since there are standard ports used in the Internet, like port 25 for FTP, or port 80 for HTTP requests. If an application uses the *Options* field, this can contain personal information too, but this is very unlikely, since most application will use the *Data* field for such information. The *Data* field therefore can contain the most sensitive information, i.e., the application data.

2.3.1.3 Linkability: identifiability and profiling

By monitoring and tracing TCP data (with tools like *TCPDump*[11]), it is possible to profile a communication or Internet traffic in general. A profile can contain header information and/or the data sent. If the data are not encrypted, an analysis of their content is quite easy.

Even if the connection is protected by low-level methods like IPSec, a traffic analysis can take place, as Bissias *et al.* have shown (Bissias *et al.* 2005). This can be used for profiling. To identify a user out of many, either the data fields or the lower-level IP address have to be examined.

2.3.1.4 Avoidance or circumvention of information disclosure

The data contents can be encrypted to prevent eavesdropping. A secure tunnel can be used in order to reduce the possibility of traffic analysis, but this must be designed and implemented carefully since traffic analysis can still be accomplished despite a secure tunnel.

The usage of TLS can protect the data sent with the TCP protocol. The TLS protocol operates above the TCP protocol, but beneath application protocols like HTTP. Thus, TLS can be easily integrated into available products, since it does not require any changes to the

[10] http://insecure.org/nmap/.

[11] http://www.tcpdump.org/.

application and transportation protocols used. TLS has been developed from the SSL protocol. The most common usage is the protection of sensitive data which is sent by the HTTP protocol over the Internet. TLS has the following security features:

- Peer entity authentication;
- Data confidentiality;
- Data integrity;
- Key generation and distribution;
- Security parameter negotiation.

TLS is a two layer protocol - the TLS Handshake Layer and the TLS Record Layer. Figure 8 shows an illustration of the handshake protocol. This handshake protocol is for establishing a secured connection between server and client. The handshake protocol has multiple purposes. First, it is needed for the exchange of certificates, to ensure the identity of the communication partner (note though that the client authentication is optional). Then, the handshake is used for the client and server to exchange their crypto-preferences, i.e., which cryptographic protocols to use for the encryption and the signing of messages. Lastly, the protocol is used for the exchange of the keys which are used for the cryptographic functions (Molva 1999).

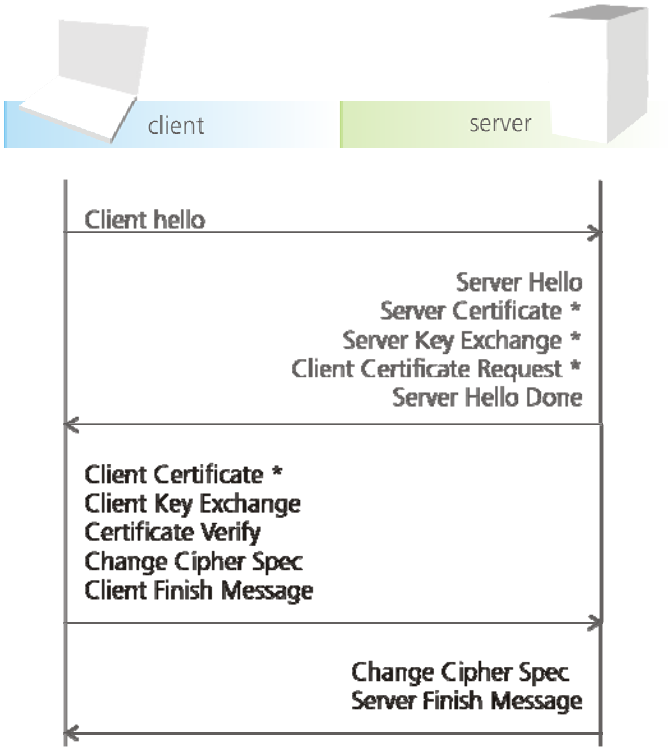

Figure 8: TLS handshake protocol ("*" marks optional fields)

After a successful handshake the TLS Record Layer is instantiated. This layer is responsible for the actual usage of the agreed security methods. This means, within the Record Layer, the data is encrypted and signed, thus protecting it against eavesdroppers and active attacks which try to change the communicated data. Furthermore, the record layer has built in replay protection and functionality for the key generation out of the exchanged master key.

As described above, TLS provides data integrity and confidentiality, as well as peer authentication, although this is optional. TLS does not protect against traffic analysis though. The protection level depends on the chosen cryptographic algorithms as well as on the keys used. It has to be noted that TLS only provides integrity and confidentiality on a hop-to-hop basis, e.g., directly from client to server, or from server to server. It does not provide end-to-end security, e.g., from client to client where data must pass through several servers. Figure 9 shows that the connection from *client* to *server1* is secured by TLS, as is the connection from *server1* to *server2*. But there is no directly secured connection from *client* to *server2*.

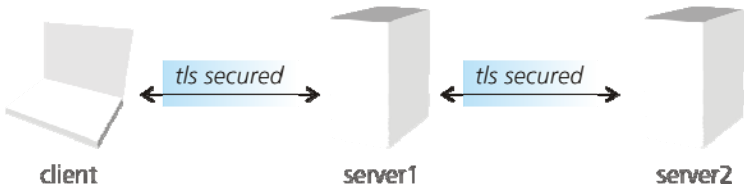

Figure 9: Illustration of the TLS hop-by-hop protection

2.3.2 UDP

UDP is the abbreviation of User Datagram Protocol and is defined in the RFC 768. UDP offers fast but unreliable data transmission - packets may get lost without automatic retransmission. No guarantee of order is given by UDP either. Because of these features, UDP is a lightweight protocol, which delivers each packet independently of other packets. UDP contains no state, cf. Figure 10 which shows request and response being stateless and independent of each other. Thus it is often used for services where small packets have to be sent in a fast way to many clients (or servers). Services which use UDP are: DNS, streaming media like IPTV, Voice over IP, online games, etc.

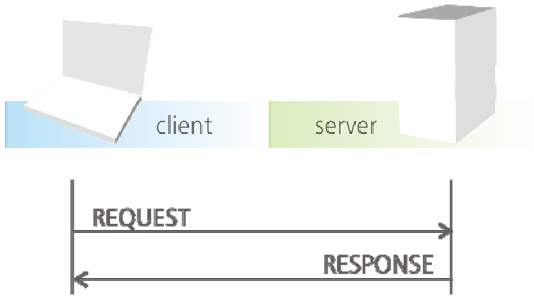

Figure 10: UDP data transfer

UDP, like TCP, uses ports to allow "parallel" sending and receiving of data in application-to-application scenarios. It has to be noted that UDP and TCP ports exist independently of each other, meaning that a port of a computer can be used at the same time for a TCP socket (stream socket) and a UDP socket (datagram socket). The UDP header has a very simple layout as shown in Table 7.

Bits	0-15	16-31
0	Source Port (*optional*)	Destination Port
32	Length	Checksum (*optional*)
32/48/64+	Data	

Table 7: UDP packet structure

As can be seen, a UDP packet is noticeable smaller than a TCP packet, and so such a packet can be processed faster than a TCP packet. But it is also evident that the lack of most of the TCP packet control fields leads to a loss of functionality.

2.3.2.1 Identifiers and their uniqueness

If UDP is used in the simplest possible case (sending just one packet) is does not reveal identifying information. But in practise a useful application often requires sending of multiple packets and implies the receiving of answers from the communication partner(s). Through the port numbers (in conjunction with the underlying IP protocol address information) it is possible to link all these UDP packets.

It might also be possible to decide if two UDP packets were sent from the same host, even if the underlying IP information is anonymised, by means of an IP level anonymisation service. This linkage can be done with the help of the source port number. TCP/IP stacks of different operating systems have different algorithms for choosing the source port number. If, for instance, it is known that the TCP/IP stack of the "victim" increases the UDP source port number one-by-one and the attacker can eavesdrop a sequence of UDP packets arriving at a certain recipient with rather random source port numbers (but not consistently increasing ones) then from an attacker's point of view the probability decreases that these UDP packets came from the "suspect".

Moreover, the data field may contain information which can be used for identifying participating parties.

2.3.2.2 Personal data

If the *Data* part of a UDP packet is not encrypted, personal data can potentially be read out. The only field which could be considered to be personal is the *Source Port*, because it identifies an open port to which answers should be sent. But this is more a security than privacy problem.

2.3.2.3 Linkability: identifiability and profiling

The combination of *Source Port* and *Destination Port* can identify packets which contain data belonging to the same application-to-application session. If the data parts of the packets are not encrypted, identifiable properties can be read out by an eavesdropper.

2.3.2.4 Avoidance or circumvention of information disclosure

Sending the content in clear-text can be avoided by using encryption, either application level encryption or network layer encryption like IPSec. Traffic analysis can take place even if network layer encryption is used.

The usage of the Source Port is optional. Therefore, the source port should be omitted (e.g., set to a random value) if it is not needed for a reply.

TLS, described above for TCP, cannot be used without changes for UDP. TLS requires a reliable transport protocol, like TCP or SCTP. To overcome this shortcoming, DTLS has been created. DTLS is an acronym for Datagram Transport Layer Security. DTLS is based on TLS meaning it will provide confidentiality and integrity as well as authentication. It has been developed in RFC 4347, and is now waiting for a widespread deployment.

2.3.3 SCTP

SCTP is a transport layer protocol, thus operating analogously to UDP and TCP. In fact, SCTP provides some similar services as TCP, particularly reliable and in-sequence submission with congestion control. SCTP has initially been developed because call setup for Voice over IP faced some severe problems when using TCP. The main problem is that the in-sequence delivery of TCP packets could disrupt the call-establishment of independent calls. The IETF working group SIGTRAN, then responsible to design a mechanism for reliably transporting call control signalling over the Internet, decided to create a different protocol to overcome these problems. The development of SCTP began.

The main difference of SCTP to TCP is that SCTP uses message-streams instead of byte-streams. SCTP allows multiple streams of messages within one association[12]. The message-streams are independent of each other, but each stream provides reliable and in-sequence data delivery. This is useful for applications in which multiple streams are related, but not dependent from each other. An example is a video conference in which the video and audio data is related, but not dependent. If the video is slowing down, the audio should run on smoothly if possible.

[12] In SCTP, an association is similar to a connection for TCP between two endpoints, e.g., two computers.

Further noteworthy features of SCTP are (Stewart, Amer 2004):

- Message Orientation: a message is delivered as a whole, like in UDP. This means, if the sender sends a 100 byte message, the receiver will get this 100 byte message in one read, no more, no less. Message boundaries are preserved.
- Unordered Delivery: next to the in-sequence delivery of messages, SCTP offers unordered delivery for applications, in which the order of messages is not important.
- Keep-alive function: a "heartbeat" is sent regularly in order to keep an association/connection alive when idling.
- Message time-to-live: a message can be tagged with a time-to-live (ttl) value, indicating how long a message is useful.

The packet structure of SCTP is shown in Table 8.

Bits	0-7	8-15	16-23	24-31
0	Source Port		Destination Port	
32	Verification Tag			
64	Checksum			
96	Chunk 1: Type	Chunk 1: Flags	Chunk 1: Length	
128+	Chunk 1: Data			
160+Chunk1: Length	Chunk 2: Type	Chunk 2: Flags	Chunk 2: Length	
192+Chunk1: Length	Chunk 2: Data			
...	...			

Table 8: SCTP packet structure

A SCTP packet can contain several so called "chunks". A chunk has a chunk header and a data body. The chunk header describes the type (e.g., *DATA*, *INIT*, *ERROR*), the flags for special type dependent properties and the length of the data field. The data field has header fields too which depend on the selected type. It is evident that SCTP is a more complex protocol than TCP or the simple UDP protocol.

2.3.3.1 Identifiers and their uniqueness

A SCTP packet can contain identifiers in its headers, but this is dependent on the type used. The *INIT* chunk for example contains all the IP addresses which can be used for the

communication. A *DATA* chunk does not contain any identifying properties, except those included inside the payload.

2.3.3.2 Personal data

A SCTP packet does not contain any personal information *per se*. Exceptions are the data carried as payload, which may contain personal data, and the IP addresses which must be provided in order to support multihoming. Multihoming is a SCTP service to increase the reliability of an association. It enables the two participating parties in an association to define multiple IP addresses, which can be switched between if required.

2.3.3.3 Linkability: identifiability and profiling

The combination of *Source Port* and *Destination Port* can identify packets which contain data belonging to the same application-to-application session. If the data part of the packet is not encrypted, identifiable properties can be read out by an eavesdropper. The additional IP addresses which can be specified in the *INIT* chunk can be used to link them to one entity.

2.3.3.4 Avoidance or circumvention of information disclosure

Sending the content in clear-text can be avoided by using encryption, either application level encryption or network layer encryption like IPSec. Traffic analysis can take place even if network layer encryption is used.

The provision of multiple IP addresses is optional, if the additional reliability provided by several IP addresses is not essential, it should be avoided.

TLS, described above for TCP, can be used with SCTP, too. TLS requires a reliable transport protocol, which SCTP is.

2.4 Internet layer protocols

The Internet layer, also called network layer, responds to service requests from the transport layer (e.g., from TCP) and issues service requests to the host-to-network layer. Simply put, the Internet layer is responsible for end-to-end packet delivery. In contrast, the host-to-network layer is responsible for node-to-node (also called hop-to-hop) delivery. The Internet layer is the first layer which implements from the data carrier independent services.

The Internet layer provides the functional and procedural means of transferring variable length data sequences from a source to a destination via one or more networks while maintaining the quality of service requested by the transport layer. The Internet layer performs network routing, flow control, network segmentation/desegmentation, and error control functions (Wikipedia: Network layer 2007).

The most important protocol on the Internet layer is the IP protocol. In order to protect the packets, IPSec can be used at this layer.

2.4.1 IP

The Internet Protocol (IP) is a stateless, data-oriented protocol, which is used to route data with the help of IP addresses and subnet masks to the intended destination. The data is encapsulated into packets (as shown in Figure 11: application layer (1), transport layer (2), Internet layer (3) and physical layer (4)) and sent through a so called packet-switched network. This means that there is no circuit setup of some kind before one host can send data to another host. The Public Switched Telephone Network (PSTN) works differently; here a circuit must be set up before a call can be established.

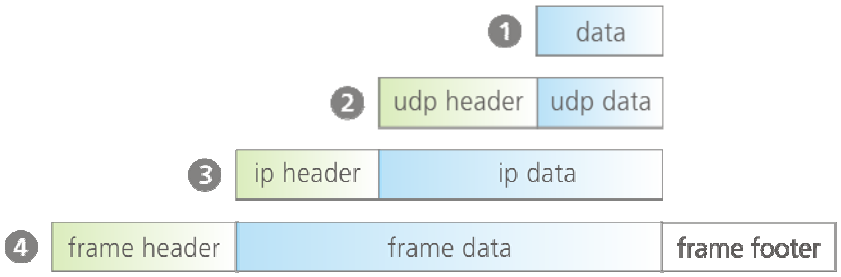

Figure 11: Data encapsulation over the four layers

Because of the abstraction of the Internet protocol, which is depicted in Figure 11, an IP packet can be sent independently of the underlying physical layer, thus the networks in between can be a mixture of different types, like Ethernet, WLAN, ATM, etc.

The Internet protocol is unreliable, thus making best-effort delivery without any guarantees. Possible errors are duplication, lost or corrupted packets, among others. The header is protected by a checksum, preventing corrupted headers resulting from network errors, but not active attacks by adversaries. It has to be noted that IP packets with a corrupted header are discarded on the spot (e.g., by a router) without notification to either the sender or receiver.

As stated above, packets are routed with the help of IP addresses. There are two versions used in parallel now, IPv4 and IPv6. The following sections will differentiate between these two protocols because there are some vital differences between both of them.

Both protocols have in common the need for unique addresses within any given subnet. A subnet is a set of devices which use one router in order to communicate with the rest of the network. A device in a subnet can be a computer or another router. To the rest of the network, all devices in a subnet are “represented” with the same external IP address. Within the subnet, the devices are differentiated by internal IP addresses, which depend on the subnet mask applied. Figure 12 illustrates a simple private network (the subnet) which is connected to the Internet by a router. Note that the router has an external IP address, to which devices from the Internet (like servers) connect. The computers in the subnet do not connect directly to the Internet, the router routes the requests and responses, thus “hiding” the infrastructure of the subnet from other hosts.

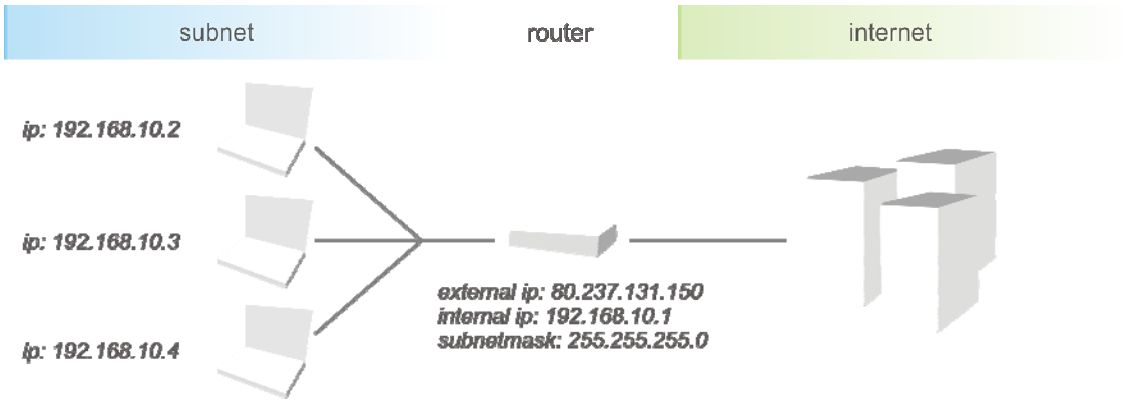

Figure 12: Subnet with three hosts connected to the Internet by a router

2.4.1.1 IPv4

An IPv4 address consists of 32 bits. The most common representation of an IPv4 address is in the form of four dot separated numbers in the range from 0 to 255 each. For example, the IP address of the domain http://www.fidis.net is 80.237.131.150.

The structure of an IPv4 packet is shown in Table 9.

Bits	0-3	4-7	8-15	16-18	19-31
0	Version	Header Length	Type of Service	Total Length	
32	Identification			Flags	Fragment Offset
64	Time to Live		Protocol	Header Checksum	
96	Source Address				
128	Destination Address				
160	Options (optional)				
160/192+	Data				

Table 9: IP packet structure

The *Type of Service* can be used for the prioritisation of the packet, which is useful for quality of service. The *Total Length* field contains the entire length (header and data) of the IP packet. The fields *Identification*, *Flags* and *Fragment Offset* are used for the reassembly of fragmented packets. The *Time to Live* indicates how long a packet is valid, whereas the *Protocol* specifies the protocol which is contained in the data section, e.g., TCP if the IP packet contains such a payload. The *Checksum* protects the header against network data corruption. The *Source* and *Destination Address* are both IPv4 addresses. The *Options* field contains mostly information about routing. The *Data* field contains the payload - the maximum is 65,515 bytes of content.

2.4.1.1.1 Identifiers and their uniqueness

The primary identifiers are the source and the destination addresses. Both are IPv4 addresses, i.e., 32 bit identifiers of the sending and receiving host. If the sending hosts uses a network address translation (NAT) service, the source address is the IP address of the NAT device, e.g., a router.

The source and destination addresses are unique within the network used. For the Internet, both addresses must be unique with respect to all addresses within the Internet. Normally Internet service providers assign unique IP addresses to each of their customers.

2.4.1.1.2 Personal data

An IP packet contains personal data in the form of the source and destination addresses, since both addresses in conjunction provide information about a communication relationship. Furthermore, with the help of the IP address in many cases the location of the user can be determined. At least the country and town can be read out of an IP address. So location privacy is not given.

Personal information can be contained in the data field of the packet. If the payload is not encrypted, i.e., the contents are sent in plain text, an eavesdropper can access all information contained in the packet.

2.4.1.1.3 Linkability: identifiability and profiling

An unsecured IP packet can be identified by its source address. The receiver of the packet can be identified by the destination address. Both properties can be used for profiling of an on going communication. The profile can contain a wealth of information, like the communicating partners, the time, the used protocol and the amount of data sent. All these single fields can give away information about the content of the communication and other interesting information.

Furthermore, if not encrypted, the content of the data field can contain identifiable information about the sending and receiving host.

2.4.1.1.4 Avoidance or circumvention of information disclosure

Sending the content in clear-text can be avoided by using encryption, either application level encryption or network layer encryption like IPSec. Traffic analysis can take place even if network layer encryption is used. Detailed information on IPSec is given in Section 2.4.2.

By the usage of IPSec one can try to hide the communication partner. This is possible if the user communicates with at least one proxy. The communication between sender and proxy must be secured by IPSec, thus providing a tunnel which hides the contents of the packet. The proxy relays the packets to the intended destination, after decrypting the IPSec packets. Still better, the proxy re-encrypts the packets and relays them to another proxy. Thus a proxy chain can be created. Likely designed as anonymous remailers, which were introduced in Section 2.2.4.4, this can help to give receiver, and also sender anonymity. Care has to be taken that an

eavesdropper cannot link the incoming with the outgoing traffic at a proxy, and vice versa. Remailer networks for normal Internet traffic are called mix networks. Two popular Mix technologies are AN.ON and Tor. Both route traffic through a series of servers. Each message is decrypted and re-encrypted at each server. Sophisticated techniques try to hide the relation between incoming and outgoing traffic. Because of these complex principles Mix networks are rather slow and cannot be used for real-time applications like VoIP right now. But they are sufficiently fast to protect the privacy of, e.g., users surfing the web.

2.4.1.2 IPv6

The IPv4 address space is quite limited. Although the 32 bit from IPv4 addresses result in a theoretical number of approx. $4.3 \cdot 10^9$ (with 10^9 being 1 billion) possible addresses, in reality it is much less because of reserved, private and multicast addresses. This results in a serious "IP address starvation", meaning that the number of publicly available addresses is very limited. Therefore a next generation protocol, the IPv6 protocol has been developed. IPv6 uses 128 bits for addresses, which results in over $3.4 \cdot 10^{38}$ available addresses.

The main advantages of IPv6 compared to IPv4 are:

- Larger address space as described above,
- simplified header format,
- optional extension header,
- authentication and encryption built in,
- auto configuration via the Stateless Address Autoconfiguration (SLAAC) and
- support for Quality of Service (QoS).

An IPv6 packet is shown in Table 10. Noticeable is that the header structure is simplified compared to the one of IPv4 packets. The source and destination address now use 128 bits. Additional fields contain information about the version, the *Traffic Class* (packet priority), *Flow Label* for QoS, *Payload Length* and *Hop Limit* to control the time to live of the packet. Noteworthy is the field *Next Header*. The header structure of an IPv6 packet could be simplified because of this field.

Bits	0-3	4-11	12-15	16-23	24-31
0	Version	Traffic Class		Flow Label	
32		Payload Length		Next Header	Hop Limit
96 128 160 192	Source Address				
224 256 288 320	Destination Address				
...	Extension Header				
...	Data				

Table 10: IPv6 packet structure

The Next Header element points to an additional header, thus enabling a header structure which adapts to the current needs of the sent packet. For example, the next header element could point to a TCP header if the payload of the IPv6 packet is a TCP packet. This feature enables routers to process packets faster and with less power, since routers only need to check and adjust header fields which are really needed. The additional headers are contained in the Extension Header field, which has a variable length. Figure 13 illustrates three different IP packets, the first without any extension header, the next with an extension header containing a TCP header and finally the third package has a TCP header as well as a routing header.

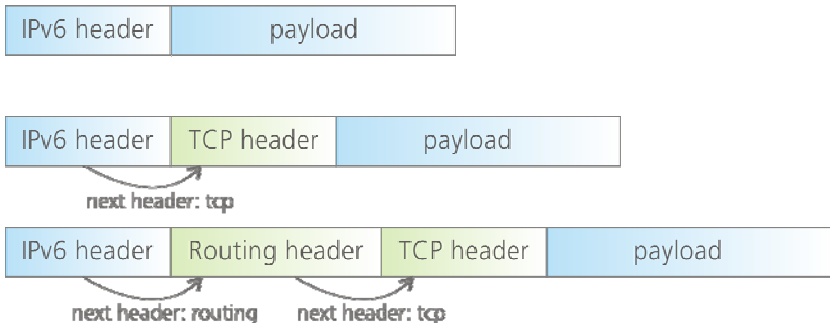

Figure 13: The extension header (the middle fields (green)) of IPv6

IPv6 has several ways for obtaining a valid IP address built in. Known from IPv4 are manual configuration and the usage of DHCP. DHCP describes a network service which assigns IP addresses to querying devices. DHCP was built to automate the assignment of network properties like IP addresses for devices as well as to provide protection regarding the multiple assignment of the same IP address to different devices. With IPv6, a device is able to compute its own IP address without the need of a DHCP service. Therefore the device uses its static link identifier, which is built among other things from the (globally unique) MAC address.

The transition from IPv4 to IPv6 will be a slow one. It is not possible, to switch from IPv4 to IPv6 at once. Thus the designers of IPv6 have taken care to ensure that devices using IPv6 can still be reached with an IPv4 address, and that IPv6 devices can utilise IPv4, called dual-stack devices.

2.4.1.2.1 Identifiers and their uniqueness

The primary identifiers are the source and the destination addresses, which identify the sending and receiving host. Since the IPv6 address space is so huge (128 bit), it is very likely that each and every device will get its own unique IPv6 address - solutions like NAT would not be used in the future. This property removes some of the "hiding capabilities" introduced by NAT routers in order to (unreliably) mask the identity of a host.`

The source and destination addresses have to be unique within the network used. For the Internet, both addresses must be unique with respect to all addresses within the Internet. It is predictable that Internet service providers will assign unique IPv6 addresses to each of their customers.

The payload may contain additional identifiers, which may be unique, too.

The extension headers may contain identifying properties too, it depends on which headers with which fields are used.

2.4.1.2.2 Personal data

An IP packet contains personal data in the form of the source and destination addresses, since both addresses in conjunction provide information about a communication relationship. Furthermore, with the help of the IP address, in many cases the location of the user can be determined. At least the country and town can be read out of an IP address, and so location privacy is not given.

Personal information can be contained in the data field of the packet. If the payload is not encrypted, i.e., the contents are sent in plain text, an eavesdropper can access all information contained in the packet.

2.4.1.2.3 Linkability: identifiability and profiling

An unsecured IP packet can be identified by its source address. The receiver of the packet can be identified by the destination address. Both properties can be used for profiling of an on going communication. The profile can contain a wealth of information, like the communicating partners, the time, the used protocols and the amount of data sent. All these single fields can give away information about the content of the communication and other useful information.

Furthermore, if not encrypted, the content of the data field can contain identifiable information about the sending and receiving host.

Location profiling is an interesting topic for mobile IP scenarios. The term mobile IP covers services which enable users to use IP services like the Internet while being mobile, e.g., while

moving around by car or foot. Scenarios of application are Voice over IP or WWW usage with the help of wireless access points. Since the user is moving from one access point to another, his IP address will change according to the current access point. The communication partner must be informed about the IP address change in order to send data to the correct destination. Since the IP address can be used to obtain location information, the knowledge about changing IP addresses can lead to location profiling.

The number of available IPv6 addresses will lead to the deployment of static IP addresses to many devices which currently either do not have an IP address or are assigned dynamic IP addresses. An example may be the oft proposed fridge which automatically orders new food whenever necessary. Although containing no critical information at first sight, a number of devices like this can lead to serious information leakage, usable for profiling, which can compromise the privacy of its owner.

2.4.1.2.4 Avoidance or circumvention of information disclosure

In order to protect the payload of an IP packet, this payload must be encrypted. This encryption can take place at higher layers (e.g., like the usage of PGP at the application layer for e-mail, or the usage of TLS at the transport layer for HTTP) or the encryption can take place at this layer by using IPSec. IPSec can furthermore protect the header. Further details about IPSec are given in Section 2.4.2.

A problem is that a client IPv6 address computed with SLAAC and the static MAC address of the network adapter results in a static and unique identifier. This could be prevented by changing this identifier repeatedly, thus gaining a dynamic instead of a static identifier. Concepts for this are discussed in RFC 3041 ("Privacy Extensions for Stateless Address Autoconfiguration in IPv6"), for example, where randomised IPv6 addresses are used, and in (Escudero Pascual 2002) where the concept of randomised addresses is analysed with respect to unobservability.

Changing the link identifier (IPv6 address) would make it more difficult to relate separate transactions to each other by using the given IP address. The rate of identifier changes must be balanced between the wish for (strong) privacy and the added load changing identifiers puts on the network device and the network itself.

In order to get sender and/or receiver privacy, concepts like Mix networks can be used. The concept does not change compared to its usage with IPv4, thus we refer the reader to the section above where Mix networks have been introduced.

In order to prevent the linking of IPv6 addresses across different web requests, Aura and Zugenmaier suggest using a new IP address for each TCP connection (Aura, Zugenmaier 2004). This is possible since one IPv6 enabled network device can control an arbitrary number of IP address: "*Using a new random or pseudo-random address for each connection makes it more difficult for an observer on the Internet to correlate two connections from the same host.*" (Aura, Zugenmaier 2004)

2.4.2 IPSec

Internet Protocol Security (IPSec) provides the following security functions in the IP layer:

- Data origin authentication;
- Data integrity;
- Replay detection;
- Data confidentiality;
- Limited traffic confidentiality;
- Access control.

IPSec also provides management services for the negotiation of sessions and session parameters. These parameters and security services are stored for each secured IP path in so called security associations (SA). Additionally, IPSec handles the key exchange between the communicating parties. IPSec itself is algorithm independent, although default algorithms are defined in the specification.

There are two basic security mechanisms built into IPSec. IP Authentication Header (AH) provides authenticated headers, i.e., headers for which the receiver can detect manipulations by an attacker. The second security mechanism is IP Encapsulated Payload (ESP), which secures the payload of IP packets, thus providing confidentiality. IP AH and IP ESP may be applied alone or in combination with each other.

There are two distinct modes which can be used for AH as well as ESP. The first mode is the transport mode. In this mode the data field and the transport protocol field (e.g., TCP or UDP headers) are encrypted. The original IP header is sent unencrypted. This mode can be used with IPSec unaware routers since the IP headers are not touched. The implementation and configuration of IPSec must take place at the client (and server) system. Thus the use of IPSec in transport mode is not transparent to the client and server system but to the routers in between. This method is usually used when people want to use a private network (like a corporate intranet) via a public network like the Internet.

In the second mode, called tunnel mode, IPSec encapsulates the complete original IP packet into a new IP packet, providing more communication privacy (see Figure 14). The original IP packet is encrypted, but the new IP header is not. Since tunnel mode IPSec can be implemented between cooperating IPSec enabled routers, the usage of this system can be transparent to the client system (Molva 1999).

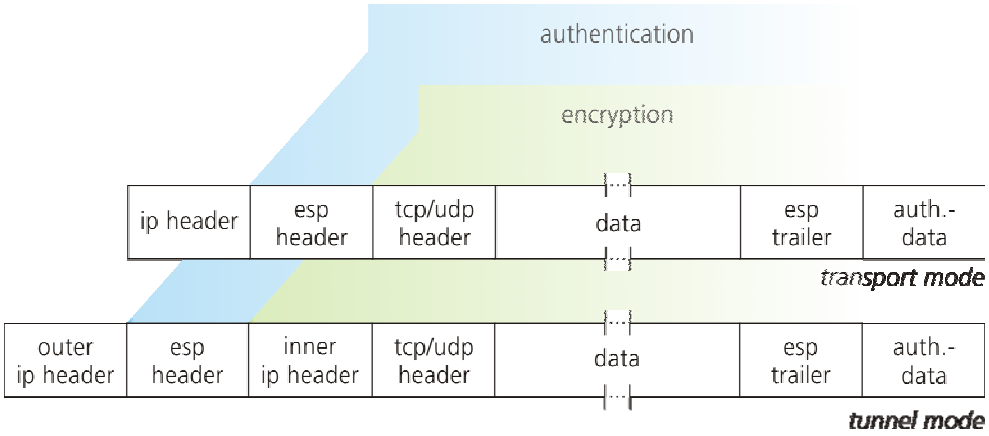

Figure 14: Packet structure of IPSec ESP packets in transport and tunnel mode

2.4.2.1 Identifiers and their uniqueness

If IPSec is used in the AH mode then no encryption of the encapsulated IP packet is done. Thus these types of IPSec packets contain all the identifying information contained in the IP packet transported in the payload of the IPSec packet. This can be avoided if IPSec ESP packets are used which provide mechanisms to encrypt the payload data.

Nevertheless, both IPSec AH headers contain a 32 bit Security Parameter Index (SPI) and a 32 bit sequence number. Both in conjunction could be used to reason whether two IPSec packets belong to the same communication or not.

2.4.2.2 Personal data

IPSec does not introduce any new personal data compared to the personal data included in the IP packets which IPSec tries to protect. Moreover, if IPSec is used for confidentiality, it can even protect personal data.

2.4.2.3 Linkability: identifiability and profiling

As already stated in Section 2.2.4.4, there is no additional information, but the data can be better protected than in the usual IP protocol.

2.5 Host-to-network layer protocols

The Host-to-network layer is the lowest layer of the TCP/IP reference model. It combines the link layer and the physical layer of the ISO/OSI model. At this layer, data is transferred between adjacent network nodes in a WAN or between nodes on the same LAN. The host-to-network layer provides the methods to transfer data between network entities. It also provides error detection and correction procedures, since the errors might come from the physical transfer. The host-to-network layer is responsible for physically transmitting the bit stream and reconstructing the “framed” data from a received bit stream for the higher layers.

The data transfer at this layer is normally not end-to-end transfer. It is in fact a data transfer from one node to another, where "the other node" might be the destination node or a node on the path to the destination.

In some networks, such as IEEE 802 local area networks, the host-to-network layer is split into the Medium Access Control sublayer (MAC) and the Logical Link Control sublayer (LLC). The LLC sublayer is the same for various physical media layers like Ethernet, Token Ring or WLAN. The main functions of the LLC's sublayer are the multiplexing and demultiplexing of data streams and providing flow control, detection and retransmission of dropped packets. The MAC sublayer is primarily responsible for framing of packets. A LLC header tells the link layer what to do with a packet once a frame is received. For example, a host receives a frame at the link layer and will look at the LLC header in order to find out where the packet is destined, e.g., for the IP protocol at the Internet layer.

The following section will introduce the protocols Ethernet, PPP, Token Ring and WLAN.

2.5.1 Ethernet

Ethernet is a large family of computer networking technologies for wired Local Area Networks (LANs). Ethernet relies on so called frames which are sent to all devices connected in a LAN. The different Ethernet standards (e.g., IEEE 802.3) comprise definitions for cable types and plugs, for the signalling at the physical layer and for the framing at the link layer. So Ethernet spans two layers in fact, the link and the physical layer. The addressing scheme used for Ethernet is the MAC address.

Ethernet has been in use since the 1990s. It is now the most popular LAN technology - other technologies like Token Ring are being displaced by Ethernet, while newer standards like WiFi, the wireless LAN standard, are becoming more and more popular

.

Ethernet has a number of disadvantages concerning LAN security:

- All stations in a LAN share a physical channel (except when the network is divided into sub-networks). This enables an attacker to eavesdrop on every frame, as everything a computer sends over the network can be received by all other stations connected to the LAN.
- The Ethernet protocol itself cannot authenticate the message's originator identity or verify a message's integrity. This enables an attacker to generate forged messages, to manipulate existing message streams or to replay previously intercepted frames.

Thus, because every device in this segment can read every frame, unsecured communication within a LAN segment is unwise.

Byte	8	6	6	2	up to 1500	4
Field	Preamble	Dest. Addr.	Source Addr.	Length	Payload	CRC

Table 11: Ethernet frame (IEEE 803.2 / 802.2)

The Preamble of the frame (shown in Table 11) contains information used for synchronisation. The destination and source address contain the MAC address of the receiving and sending device respectively. The payload of an Ethernet frame can be up to 1500 bytes. A 4 byte error correction code (Cyclical Redundancy Check (CRC)) is added at the end of the frame.

The sending adapter adds the preamble and the CRC. The receiving adapter removes these fields after analysing them. The receiving Ethernet adapter receives all frames, but only passes data to its host when the destination address is valid for the given host. A valid address is a unicast address with the adequate host address, or a valid broad- respectively multicast address. Furthermore, a network adapter can be programmed to pass all frames. This is called "promiscuous mode".

Since all stations in a LAN share a physical channel, so called collisions can occur. Since at any time only one station is allowed to send on an Ethernet network, collisions occur if two or more stations send data at the same time. For Ethernet networks the "Carrier Sense Multiple Access with Collision Detection (CSMA/CD)" media access control mechanism is used to detect collisions. After a collision, the involved stations wait for a random time and then send again.

2.5.1.1 Identifiers and their uniqueness

An Ethernet frame contains privacy critical information, i.e. the source and destination MAC addresses. These addresses are required since these addresses define the sending or receiving physical host. however, a MAC address is only valid within a LAN, if an Ethernet frame is sent to another LAN via a router, the MAC address changes.

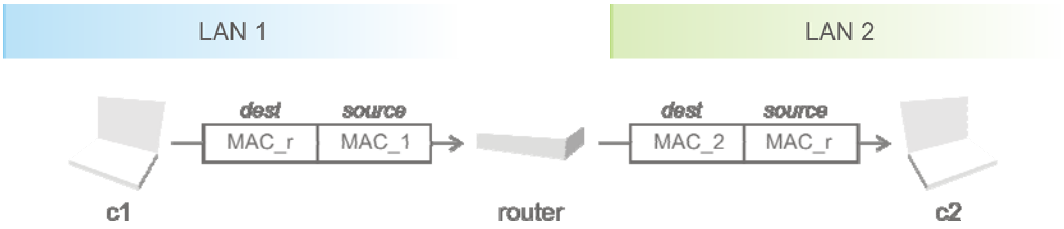

Figure 15: Illustration of an Ethernet frame with a change of MAC addresses

Figure 15 depicts the MAC address change of both destination and source addresses at a router. If *c1* in *LAN 1* sends a packet to *c2* in *LAN 2*, *c1* creates an Ethernet frame with the router's MAC address as the destination and its own address as the source address. The router replaces the destination address with that of *c2* and the source address with its own MAC address. The router either knows the MAC address of *c2,* or it uses the "Address Resolution Protocol" ARP to gain the MAC address of *c2.* Therefore the router has to analyse the payload of the Ethernet frame in order to get the IP address of *c2*. Because of this property of MAC address changing by routers, the MAC address can only be used within a separated LAN to identify a device.

The payload may contain privacy relevant information. If the payload contains an IP packet, this packet itself contains the IP address of the sending and receiving host. These addresses are unique to a certain degree. The payload of an IP packet may contain (unique) identifiers.

2.5.1.2 Personal data

An Ethernet frame does not contain any personal information except for any data contained in the payload.

2.5.1.3 Linkability: identifiability and profiling

Profiling may take place at the router. The router filters all traffic from one LAN to another. Therefore the router has to inspect all Ethernet frames and adjust them as needed for the routing. Therefore the router can extract fairly detailed picture of the data sent and the communication partners.

2.5.1.4 Avoidance or circumvention of information disclosure

The usage of the MAC address cannot be avoided.

The payload can be encrypted to a certain degree. Information like the destination IP address in the IP packet encapsulated in the Ethernet frame cannot be hidden. This data is needed to route the frame to its intended destination. However, using Mix networks can help by obscuring this information or the usage of IPSec can protect the content of an IP packet. More details on security and privacy on the Internet layer (where IP and IPSec are contained) are given in Section 3.3.

2.5.2 PPP

PPP is the acronym for Point-to-Point Protocol. PPP is used to establish a direct connection between two nodes, e.g., between two routers or a modem and a server. PPP is the most popular technique for transporting IP packets over a serial link between the user and his Internet service provider (ISP). A session between the user and the ISP is established by using the Link Control Protocol (LCP). PPP supports several ways for authentication:

- with a password via PAP,
- with a password and a challenge/response system (CHAP) or
- with a complex protocol called Extensible Authentication Protocol EAP, supporting certificates and other identifying properties.

PPP encapsulates high-level protocols like IP, thus making it usable for DSL modem dial-up via Ethernet, called PPP over Ethernet (PPPoE). Thus PPPoE enables a point-to-point connection between the user and the ISP in the normally multipoint architecture of Ethernet.

A PPP frame has the structure shown in Table 12. The *Flag* field indicates a frame's beginning or end, the *Address* is a broadcast address (11111111b) and the *Control* field contains information for the flow control. The protocol used (e.g., IP) is mapped to the *Protocol* field. The payload may have variable length. The last field for the error correction can be either 2 or 4 bytes long.

Byte	1	1	1	2	variable	2-4
Field	Flag	Address	Control	Protocol	Payload	CRC

Table 12: PPP frame as defined in RFC 1662

Figure 16 shows an illustration of a client to ISP communication over ADSL. The client uses a DSL modem for dial-up. The modem sends the data to the Digital Subscriber Line Access Multiplexer (DSLAM), a device operated by the ISP. The DSLAM connects multiple customer DSLs to the Internet backbone or, as depicted, to an access server which checks the user credentials provided

.

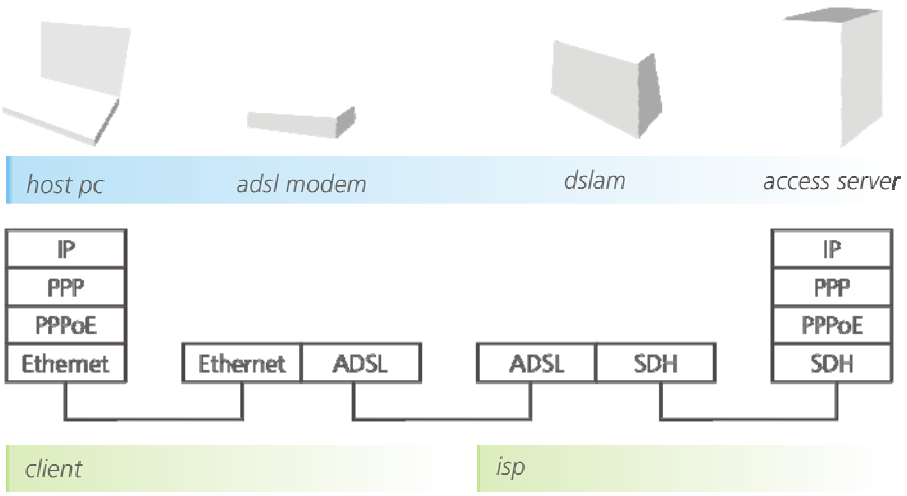

Figure 16: ADSL communication between client and ISP

2.5.2.1 Identifiers and their uniqueness

PPP itself contains no identifiers, although PPP does use authentication protocols like PAP, CHAP or EAP with which a user authenticates himself to the ISP. The authentication value is normally a unique identifier, like a user name / password combination or a certificate.

2.5.2.2 Personal data

PPP contains no personal data except the potential personal information contained in the payload.

2.5.2.3 Linkability: identifiability and profiling

Since the user is normally uniquely identified by the required credentials for authentication, the ISP with which the point to point connection exists can profile any network action the user takes. Therefore the ISP has to inspect the content of the PPP packets sent.

2.5.2.4 Avoidance or circumvention of information disclosure

In order to protect potentially privacy relevant data contained in the payload, encryption can be used. For example, the Microsoft Point-To-Point Encryption (MPPE) Protocol applies either 40, 56 or 128 bit keys for the protection of the data sent.

2.5.3 WLAN

WLAN stands for wireless LAN, and, as the name indicates, links computers without using wires. WLAN utilises radio waves for device communication over a limited area. WLAN is becoming more and more popular, mainly because it provides the users with mobile network access and gets rid of some cables. Furthermore, providers use WLAN as a means to provide customers with easy Internet access, e.g., at a coffee house or at public places like airports. Additionally WLAN is sometimes used to provide (relatively) fast Internet access to areas where no broadband connection via cable is possible.

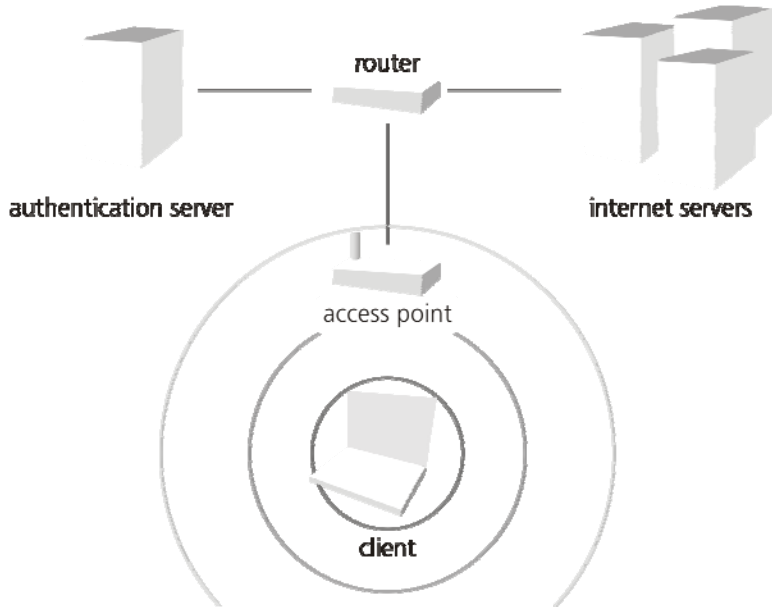

Figure 17: A schematic of the most common components of a wireless LAN

Figure 17 depicts the typical components of a wireless network: a client, an access point, a router, an authentication server and the Internet infrastructure. The client needs a WLAN card which uses the same standard as the access point. For example, both must support the IEEE 802.11g standard in order to communicate. The access point is responsible for routing data from the wireless network to the wired one where the data is sent to a router. The router could either route the data directly to the Internet, or, if capable, can first check client permission with the authentication server.

An access point is identified by a so called SSID, i.e., a Service Set Identifier. This SSID is added to each packet sent via the WLAN in order to relate the packets to the access point used. All devices which want to communicate with each other must use the same SSID.

Although there are a lot of advantages to WLAN (e.g., mobility, convenience of deployment etc.), there are some serious disadvantages, too:

1. Range
   WLAN has a limited range, which is essentially determined by physical facts like intercepting walls or buildings or electronic devices interfering with the radio waves thus weakening or destroying the original signal.
2. Reliability
   The reasons for range limitations also have an impact on the reliability of access. This means a WLAN signal in one room is not necessarily receivable in the next room, etc. This problem can be triggered by complex phenomena like multipath (a signal reaches its destination via more than one paths) or the Rician fading (cancellation of the radio signal by itself).
3. Speed
   WLAN is normally much slower than the usual 100 MBit wired LAN. Furthermore, latency is often higher, meaning that packets need longer to travel from the client device to the first device which is wired. This can pose a problem for real-time applications like VoIP.
4. Security
   Because of the usage of radio waves as a transport medium, physical access to the data sent is easy: an attacker has to be in reach of the radio waves only. On a normal, wired network, an attacker must overcome the physical limitation of tapping the actual wires, but this is not the case with WLAN and wireless transport of data. To prevent eavesdropping (and other security risks), security methods like WEP, WPA or WPA2 can be utilised.

The most popular WLAN protocols currently are from the IEEE 802.11 family. Wireless protocols from the 802.11 family all use the same basic protocol for their function. The difference between the single protocols is mainly in the frequency used, the throughput and data rate as well as the range. Furthermore, the modulation technique used varies from protocol to protocol. Modulation describes the way that the radio waves are modified in order

to transport the intended message (i.e., the data). The choice of modulation has an important impact on parameters like interference or multipath problems.

The 802.11 protocol covers the data link and the physical layer. 802.11 defines its own data link MAC layer, which is also responsible for some functions normally covered by upper layer protocols, e.g., fragmentation, packet resubmission and acknowledges. 802.11 defines three basic physical layers:

1. Frequency Hopping Spread Spectrum (FHSS);
2. Direct Sequence Spread Spectrum (DSSS);
3. Infrared (IR).

Figure 18 shows the three physical layers at the bottom of the image. Above the physical layers is the data link layer with its two sub layers, MAC and Logical Link Control. The MAC layer plus the three physical layers are defined in the 802.11x standards, whilst the Logical Link Control sub layer is defined in the 802.2 standard which is also used for wired LANs.

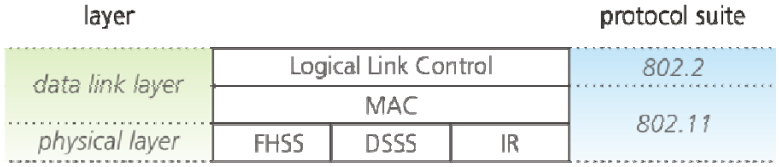

Figure 18: Layers as they are covered by the 802.11 resp. 802.2 protocol suites

The 802.11 standard defines the frame structure which is modelled in the MAC sub layer. There are three different frame types, which are:

- **Management Frames:** 802.11 management frames enable stations to establish and maintain communications.
- **Control Frames:** 802.11 control frames assist in the delivery of data frames between stations.
- **Data Frames:** 802.11 data frames can carry packets from upper layers like IP packets which themselves contain TCP packets etc.

The basic frame structure of all three frame types is the same and depicted in Table 13. The *Frame Control* field contains information about the 802.11 protocol version, frame type, and other indicators, such as whether WEP is enabled, power management, etc. Additionally, a 802.11 frame consists of the source and destination MAC addresses (*Addr 1* and *Addr 2*) as well as the address destination wireless station (access point) and transmitting wireless station (*Addr 3* respectively *Addr 4*).

802.11 data frames contain protocols and data from higher layers within the *Frame Payload*. Management and Control Frames use the Frame Payload field to carry their data.

Byte	2	2	6	6	6	2	6	0-2312	4
Field	Frame Control	Duration ID	Addr 1	Addr 2	Addr 3	Sequence Control	Addr 4	Frame Payload	CRC

Table 13: IEEE 802.11 MAC frame format

The frame displayed in Table 13 is encapsulated into the frame outlined by Table 14 if Frequency Hopping Spread Spectrum (FHSS) modulation is used. If another modulation technique is used (either infrared or Direct Sequence Spread Spectrum (DSSS)) the physical frame will differ slightly.

Byte	10	4	14-2326	4
Field	PLCP-Preamble	PLCP-Header	MAC Data	CRC

Table 14: 802.11 physical frame for FHSS modulation

The *PLCP-Preamble* (PLCP stands for "Physical Layer Convergence Protocol") contains synchronisation information and a delimiter which is used to define the frame timing. The *PLCP-Header* contains information about the length of the header, about the data rate to be used and finally a CRC code for the header. The *MAC Data* field is described in 2.5.3.1. The last field contains CRC data, this time for the whole frame.

Common threats to WLAN security are:

- Eavesdropping on the radio waves;
- Interception and modification of transmitted data;
- Spoofing, e.g., setting up a WLAN with a SSID already used by another WLAN hot spot in order to lure unaware users into using this spoofed access point;
- Denial of service (DOS), e.g., jamming some area in order to prevent clients from transmitting data;
- Free-loading (resource theft);
- Rogue WLANs, i.e., access points attached to a LAN without authorisation by the administrator.

Privacy threats are:

- Location privacy, e.g., by MAC addresses, untrusted network operators, precise positioning technology like triangulating the position by using several access points at once.
- Data privacy since WLAN data is sent by radio waves which can be received in a potentially wide area.

2.5.3.1 Identifiers and their uniqueness

Like wired Ethernet LAN, 802.11 WLAN also uses MAC addresses to uniquely identify network devices. A MAC address is made up of 48 bits, whereas the most significant 24 bits contain the unique identifier of the network device manufacturer. Each manufacturer assigns the other 24 bits of the MAC address with (more or less) unique values for each network card and stores the complete MAC address in the firmware of the device.

A WLAN frame does not only contain the MAC addresses of the source and destination device, but also the MAC address of the communicating access point.

WLAN is a good example for identifying information based on "manufacturing deviations" as described in Section 2.1.2. Research papers like (Toonstra, Kinsner 1996; Hall, Barbeau, Kranakis 2005) describe how the various characteristics of a radio signals and the specialties of the devices which emit them can be used to distinguish between devices from different manufactures but also distinguish between devices of the same model.

2.5.3.2 Personal data

An 802.11 frame does not contain any personal information except for any data contained in the payload. Note that most of the threats to privacy and personal data are not directly linked with the protocol but with the inherent broadcast feature and the usage of wireless networks. The former makes eavesdropping very easy to do and hard to detect. The latter is related to mobility which is greatly supported by means of wireless communication. But as the MAC address of a device is static and unique operators of large scale wireless networks (e.g., telecommunication companies operating access points at air ports and hotels around the world) can easily track the locations of the wireless devices (and thus most likely their users).

2.5.3.3 Linkability: identifiability and profiling

Profiling may take place at any access point which the user is communicating with, at any access point within the range of the sent radio signal and at any WLAN enabled device (like a laptop with a WLAN card) within the range of the radio signal sent. Profiling may include analysing of data sent, location and movement profiling.

Users can be identified at the MAC sub layer as well as the physical layer. The MAC sublayer provides the sender's MAC address, which is static and, to a certain degree, also unique. At

the physical layer attributes like signal-to-noise ratio, modulation peculiarities and other information obtainable from the sent signal could possibly lead to identification.

2.5.3.4 Avoidance or circumvention of information disclosure

In order to protect the data within the MAC data field encryption can be used. There are three standards which evolved within the history of WLAN.

The first standard was Wired Equivalent Privacy (WEP). This standard was introduced with the first 802.11 specification. The most important features are listed in Table 15. WEP is now considered insecure since it has several weaknesses like short keys, weak authentication mechanisms (only with pre-shared keys), virtually no key management, etc.

After the weaknesses of WEP became evident, WPA was developed. WPA has been created by the WiFi Alliance, thus it is not an IEEE standard. WPA supports strong encryption with quite long keys (AES with 128 bit keys) and key rotation. The authentication has been widely extended, WPA utilises EAP with RADIUS, certificates (PKI) and still shared key.

WPA was developed as an intermediate solution until 802.11i became available. WPA implements a subset of 802.11i, thus both protocols have many similarities. The main difference is that 802.11i introduces AES-CCMP for encryption and per session key management.

In order to protect the content of WLAN frames 802.11i or at least WPA should be used. WEP is unsecured and should not be applied anymore.

Protocol	Encryption	Authentication	Key Management
WEP	• RC4 with 40 bit key/28 bit hash • Static keys	• Pre Shared keys • Open System (SSID)	• Manual key rotation, i.e., no key management
WPA	• TKIP with 128 bit key (over RC4) • Constant key rotation	• 802.1x with EAP and RADIUS • Pre-shared key	• Per packet key rotation
802.11i	• TKIP with 128 bit key (over RC4) • AES-CCMP • Constant key rotation	• 802.1x with EAP and RADIUS • Pre-shared key	• Per packet key rotation (TKIP) • Per session key rotation (AES-CCMP)

Table 15: The most important security protocols for 802.11 WLANs

In order to prevent traffic analysis IPSec in tunnel mode can be used. IPSec has the advantage that it can be applied to secure the connection without need for updates to the WLAN soft- and hardware. Furthermore, IPSec is an established and well tested security solution which can provide a high level of security. however, the usage of IPSec is problematic because of several reasons:

1. The data sent by users is secured, but the WLAN infrastructure is not.
2. The mobility of users will be limited because of difficulties with roaming.

3. If IPSec is used to secure the connection between access point and WLAN device this could become a bottleneck since IPSec is quite resource intensive.

One threat becoming more and more severe because of the growing number of mobile Internet applications is related to location privacy. The term "location privacy" refers to the right of users to define who gets information about the current whereabouts of the users. The problem is that mobile users normally get different IP addresses at each access point. These IP addresses can be mapped to geographical data, meaning that the location of a user can be resolved by the IP address alone. Location based services (LBS) use this fact to offer users location dependent services and information. The current location of a person is personal data, thus users should be able to protect this data.

A lot of research has been done in the area of how location privacy can be provided or at least enhanced. There are solutions which try to "hide" a single user in a crowd such that the location data is unreliable. The technical term for this is k-anonymity, meaning that a privacy-relevant dataset is only released if there are at least k-1 other (distinct) datasets. All datasets must be indistinguishable from each other in terms of their identifying values (Gedik, Liu 2004). Another approach is to change the interface identifier (i.e., the MAC address) of the mobile device within certain intervals in order to provide IP addresses which cannot be linked by the MAC address (Gruteser, Grunwald 2005).

2.5.4 ISDN

ISDN is the acronym for Integrated Services Digital Network. ISDN is a circuit-switched telephone system. In contrast to the analogue public switched telephone network (PSTN), sometimes also referred to as "plain old telephone service" (POTS), ISDN has been designed for digital transmission of voice and data via copper wires. Therefore ISDN normally results in a better voice quality and a higher bandwidth per line.

ISDN consists of a set of protocols for establishing and ending calls as well as for advanced call features. Noteworthy features of ISDN are:

- Simultaneous usage of two (or more depending on the telecommunication provider) connections over one line.
- Fast call setup times (compared to analogue call setup).
- High voice quality and real-time service (not guaranteed over the Internet).
- Additional features are the delivery of caller ID to the receiver, the provision of Three-Way Calls and Call Forwarding.

ISDN specifies two types of channels, one for data called B channel (bearer channel) and one for signalling and control, called D channel (delta channel). There are three distinct ISDN implementations, which all vary in the number of B channels and D channels.

Mapped onto the 7-layer OSI model, ISDN can be seen at the network, data link and physical layer. At the data link layer the so called LAPD (Link Access Protocol – D channel) is defined. The format of a LAPD frame is given in Table 16. The LAPD protocol is used to establish a link (connection) between a user's endpoint (network termination NT) and the network itself.

Byte	1	2	1 or 2	0-260	2	1
Field	Flag	Address	Control	Information	FCS	Flag

Table 16: ISDN general LAPD frame format

The first *Flag* field is a frame delimiter and always set to a standard value. The *Address* field contains information about the terminal endpoint this frame is sent to. The *Control* field identifies the type of the frame, whereas the *FCS* field contains a frame checksum for error detection and correction. The last *Flag* is a delimiter.

At the physical layer one can distinguish between different interfaces. The S/T interface (S_0) is mainly used to connect terminals (e.g., ISDN telephones) with the network terminator. The U_{k0} interface is used on "the last mile" between the network terminator and the local telephone exchange. The format of the frames sent on the physical layer differ according to the transmission direction (i.e., from local exchange to the terminal or vice versa). Basically each frame contains the data from both lines (B1 and B2) as well as the data from the control channel (D). The U_{k0} interface frame format also defines so called maintenance data which allows the telephone company to communicate with the network terminator, e.g., to test for proper connection between the local exchange and the network terminator.

A call setup is established by sending network layer (layer 3) frames with appropriate values over the D channel. The ISDN layer 3 frame structure is given in Table 17.

Bits	1	2	3	4	5	6	7	8
0	Protocol Discriminator							
8	0	0	0	0	Length of Reference Value			
16	Flag	Call Reference Value						
24	0	Message Type						
32+	*Other information elements as required (up to 260 Byte)*							

Table 17: ISDN layer 3 frame format

The first field is the *Protocol Discriminator*, i.e., it defines which protocol is used to encode the remainder of the layer. Since the reference value can either be 8 or 16 bits long, the length is stored in the length field. The *Reference Value* identifies the call. The *Message Type* field contains information about the primary purpose of this frame, i.e., call establishment or clearing.

The last field is for *Information Elements*. These elements contain detailed information which is needed to process the request initialised by this frame. For example, an Information Element may contain the number of the party called.

2.5.4.1 Identifiers and their uniqueness

An established call contains the caller and the receiver ID, i.e., the telephone numbers of both parties. Information Elements which contain information which is privacy critical are:

- calling party number,
- calling party subaddress,
- called party number and
- called party subaddress.

All these fields contain information to uniquely identify the participating parties.

2.5.4.2 Personal data

Normally the data sent over the B channels, i.e., voice or other data, is the data which contains personal information, e.g., the contents of a communication. But the signalling data from the D channel also contains personal data like the telephone numbers of the communicating parties.

2.5.4.3 Linkability: identifiability and profiling

Profiling may take place at the telecommunication provider the end-user is connected to and all other digital local exchange parties involved in the delivery of a communication. Each frame sent can be associated to the communicating parties because of the established link carrying the data.

2.5.4.4 Avoidance or circumvention of information disclosure

Jerichow *et al.* propose the use of ISDN Mix networks in order to provide caller and receiver anonymity (Jerichow *et al.* 1998). The authors claim that their concept works with a minimal impact on the performance of the network. The proposed Mix networks also provide confidentiality next to anonymity.

In order to get confidentiality for ISDN connections extra hard- or software has to be installed since ISDN sends the data unencrypted over the network.

It is possible with ISDN to suppress the caller ID. Thus the receiver cannot recognise who is calling before answering the phone. It would be more privacy friendly if the caller ID was suppressed by default and only shown upon explicit caller request. Note that only the receiver cannot access a disabled caller ID, the service provider still has this information.

2.5.5 Bluetooth

Bluetooth is an industry specification for wireless Personal Area Networks (PANs). The purpose of Bluetooth is to transfer data between devices which are within short range, thus operating on a high radio frequency with low power consumption. Popular applications are Bluetooth headsets for mobile phones or wireless computer periphery like mice and keyboards.

The Bluetooth standard defines different profiles for different applications. For example, there is a profile for data transfer, one for voice data transfer, etc.

Bluetooth works in principle with a server/client or master/slave relationship. The server or master is a Bluetooth device which can communicate with up to seven slave or client devices. A group of connected devices is called a Piconet. Within a Piconet a total bandwidth of 1 MB/s (Megabyte per second) is shared between all devices. Up to 255 other devices can be inactive and connected to the master device, but only seven can be active.

Any Bluetooth device will transmit the following sets of information on demand (Wikipedia: Bluetooth 2007):

- Device name;
- Device class;
- List of services;
- Technical information, for example, device features, manufacturer, Bluetooth specification, clock offset.

Any device may perform an inquiry to find other devices. An inquiry is responded to with the above listed information. Most Bluetooth devices can be configured to only respond to inquiries after a user interaction if the sending device is unknown. If the device is known, the information is sent at once.

Every Bluetooth device has a unique 48-bit address. Normally these addresses are not shown, but the display name is sent. This can be problematic since identifying a device by its device name is error prone as the device name can be set by the user and need not be unique in any way.

The Bluetooth standard specifies three different levels of security:

1. **Non-Secure Mode:** No authentication of devices nor encryption of data sent.
2. **Service-Level Enforced Security:** The application layer is responsible for the selection of the security mechanisms
3. **Link-Level Enforced Security:** At the Link Layer, two security services are defined by the Bluetooth standard: secure authentication and encryption of data. The last is optional.

There are two possible ways for devices to create the keys needed for secure authentication and encryption. The first method involves the so called *Unit Key*. The Unit Key is a (normally unchangeable) unique symmetric key stored in the device. The second method utilises the so called *Link Key*. The Link Key can be seen as a temporary symmetric key which is used for one or more sessions. The Link Key is generated from several values, including a PIN (8 to 128 bit), the Unit Key, the MAC address of the devices and some random numbers.

The initialisation procedure of the keys reveals a major weakness of the Bluetooth security architecture. At least one the above mentioned parameters for the key generation must be shared between the devices. Since Bluetooth does not support certificates, the values have either to be submitted in clear-text between the involved devices, or the users have to manually input the values into their devices. Submitting such vital values in clear-text should not be an option since an attacker can easily obtain these values. Manual input by the users is problematic insofar that they will probably choose small values like a 4 digit PIN. The problem is that small values are weak and so ease brute-force attacks.

Once the required keys have been created, the devices can store those values and use them again later. Thus the initialisation process can be shortened by relying on existing values, making it harder for attackers to eavesdrop on the communication (Jakobsson, Wetzel 2001).

The format of the Bluetooth packets is shown in Table 18.

Bits	72	3	4	1	1	1	8	0-2745
Field	Access Code	Address	Type	Flow	Ack	Sequence	CRC	Payload

Table 18: Bluetooth packet format at the Link Layer

The Access Code is used to identify packets sent over a channel, i.e., all packets sent on the same channel have the same access code. By using the 3 bit Address field up to seven active devices can be addressed. The Type field specifies whether data or voice is transmitted. The last field before the Payload is an error correction code for the header.

2.5.5.1 Identifiers and their uniqueness

Bluetooth devices have a unique, permanent 48-bit Bluetooth Device Address (BD_ADDR).

2.5.5.2 Personal data

A Bluetooth packet does not contain any personal data except any contained in the payload, although the payload is encrypted by default. However the encryption relies mainly on the PIN. Some devices do not allow the user to enter the specified pin (e.g., headsets), thus the PIN is set to a default value (e.g., 0000) which enables a master device to start a communication with the slave device. This fact lowers the level of security severely (Hager, Midkiff 2003).

2.5.5.3 Linkability: identifiability and profiling

The Bluetooth Device Address is a unique identifier. A Bluetooth device is often contained within a personal belonging, like a mobile phone or a headset. Once a relation between device and person has been made, the person could be identified by means of the device. Especially if the person has more than one device, the conjunction of multiple identifiable devices may lead to profiling of the person even in cluttered environments with many other devices. Furthermore, profiling of movement and location may easily occur (Wong, Stajano 2005).

The identifier is revealed in several situations (Wong, Stajano 2005):

1. On the physical layer when an attacker observes the frequency-hopping scheme which is utilised by the parties. This is possible since the frequency hopping scheme is based on the BD_ADDR and the internal clock of the slave device.
2. When answering an inquiry the slave reveals its BD_ADDR.
3. When sending a page address a master device is revealing the BD_ADDR of the slave it is paging.

2.5.5.4 Avoidance or circumvention of information disclosure

In order to strengthen the confidentiality of sent data, upper layer encryption should be used which does not rely on the (often weak) PIN as the secret key and which probably supports strong authentication via certificates at the application layer. IPSec at layer 3 can be used to provide confidentiality, integrity, authenticity and to a certain degree prevent traffic analysis.

Using dynamic identifiers, i.e., changing the BD_ADDR frequently can significantly reduce the linkability of independent interactions with the same Bluetooth device, thus providing stronger location privacy. But if the dynamic identifiers are changed often and in a totally random way, Bluetooth cannot use the information about prior interactions for Piconet configuration. Thus frequent re-initialisation must take place between devices which formerly knew each other. This re-initialisation takes time, resources and is to some degree a security weakness with the current Bluetooth standard since in the initialisation process security and privacy relevant information is sent in clear-text over the air. A couple of other weaknesses with BD_ADDR pseudonyms are given in (Wong, Stajano 2005). But the authors also present a scheme which protects the participating parties' pseudonyms from linkability whilst still allowing re-initialisation based on previously negotiated values.

Since the BD_ADDR can be extracted from the frequency hopping pattern at the physical layer, the frequency hopping scheme must be independent from the BD_ADDR or the scheme must be derived from BD_ADDR pseudonyms.

2.5.6 Cable modem

A cable modem is a modem which provides access to the Internet over the cable television infrastructure. The working principle is that a cable modem utilises the unused bandwidth on

a cable television network in order to provide broadband access. Internet access by cable is often tied to a cable television subscription.

A cable modem is not a classical modem by its technical definition, but a network bridge. The available bandwidth of a cable channel is shared by several users. Thus the available bandwidth per user depends on the number of people using the connection. The cable operators have to ensure that not too many users use the same channel, else certain QoS parameters cannot be guaranteed. Data is send over coax cables in cable networks.

In Figure 19 the most important components of a cable network are given. The clients use cable modems to access the network. The modems connect to a CMTS, i.e., a Cable Modem Termination System. This CMTS is responsible for the connection to the IP-based Internet.

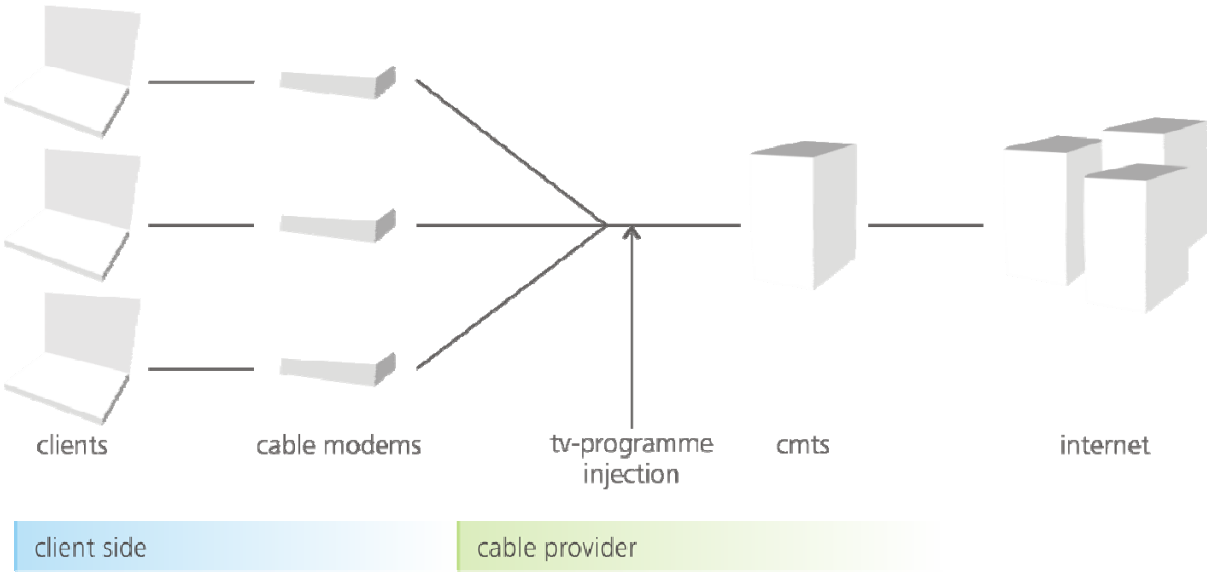

Figure 19: Components of a cable network with Internet access

Thus a CMTS can be imagined like a high-capacity router with an Ethernet interface on the one side (to the Internet) and a coax radio frequency (RF) interface on the cable network side.

The most established data transfer method for cable networks is the "Data over Cable Service Interface Specification", DOCSIS. The current version of this specification is version 3.0[13].

The DOCSIS protocol stack consists of four layers, i.e., the physical layer, the MPEG-2 transmission convergence layer, the MAC layer and finally the data link encryption layer. The physical layer describes the modulation on the cable network, which is different for up- and downstream. The MPEG-2 Transmission Convergence layer exists only for the downstream direction. Data (e.g., IP packets) from the Internet to the client are encapsulated in MPEG packets, thus they look like normal media packets but with an identifying header. Upstream data, i.e., data from the client to the Internet, is carried in Ethernet frames. The MAC layer primarily controls the upstream data by requesting slots from the CMTS in which data can be

[13] http://www.cablemodem.com/specifications/specifications30.html.

sent from the user to the provider. This is necessary to avoid collisions. The data link encryption layer employs *Baseline Privacy* or the newer *Baseline Privacy Plus* to provide some basic security. The security goals of Baseline Privacy are:

- Encryption of data flow between the cable modem of the user and the CMTS of the cable provider. The encryption is done using DES with a 40 bit or 56 bit key or AES with a 128 bit key. The DES encryption can now be considered to be insecure but for compatibility reasons it is still allowed.
- Providing protection against service theft for cable modem providers by using authentication methods.

The security measures are selected by the provider, the user cannot influence if and how data is protected. Also, there is no protection against tampering with the signals on the RF cable network (Fellows, Jones 2001).

2.5.6.1 Identifiers and their uniqueness

At initialisation, the user has to identify his terminal using a certificate. This ensures a distinct identification of the access point (i.e., cable modem) used.

After initialisation the user gets an IP address supplied by which this user is identified on the Internet. Often the cable modem provider acts as a kind of NAT-router, thus hiding the users behind its own IP address(es). But this is not strong protection. For more details see Section 2.4.1 on the Internet Protocol.

2.5.6.2 Personal data

No personal data except that present in the payload.

2.5.6.3 Linkability: identifiability and profiling

The user has to authenticate himself to the cable modem provider before the service can be utilised. Then a permanent link is established between the user and the provider. Thus the user can be identified, or to be more precisely, the cable modem terminal used can be identified. The cable modem provider can profile all user actions.

2.5.6.4 Avoidance or circumvention of information disclosure

The authentication via certificates against the provider cannot be circumvented because the provider needs this information in order to provide its services and for accounting.

Since Baseline Privacy is not very secure, users should apply upper layer security protocols in order to protect their privacy. By using the new Baseline Privacy Plus more secure algorithms

than DES can be used, but this has to be enabled by the provider. Again, Baseline Privacy protects only the data from the cable modem to the cable modem provider, no further. To protect data on the Internet, upper layer protocols have to be used by the user.

2.6 Conclusion

This chapter has analysed a variety of protocols on the application layer, the transport layer, the Internet layer and host-to-network layer with respect to privacy-relevant properties. In particular (unique) identifiers, personal data being disclosed and linkability characteristics providing identifiability and profiling have been elaborated regarding each of the protocols. Moreover, it has been discussed whether it is possible during the usage of these protocols to avoid or circumvent protocol-inherent information disclosure.

The analysis results in the observation that many protocols bear the risk of privacy threats which often are hard to counter. Typically identifying information usable for profiling is disclosed when making use of the protocols, yet this is not generally known by the mass of users. The scope of this chapter is limited to separate analysis per protocol, but it is clear that a cross-protocol or cross-layer analysis would yield even more privacy risks.

3 Protocols for privacy-aware communication

In the previous chapter, commonly used protocols for networking were analysed regarding their privacy properties. This chapter deals with privacy-aware communication and gives an overview of techniques and developments in three different areas:

1. Anonymisation services;
2. User-centric identity management;
3. Privacy policies and their enforcement.

The systems and properties described are not in all cases regarded as "protocols", but on the one hand the definition given in Chapter 1 (Wikipedia: Protocol (computing) 2007) covers the depicted techniques, and on the other hand they are important to understand current developments in the privacy area to cope with some of the privacy threats the use of protocols and ICT systems in general poses.

The three areas elaborated in this chapter are not orthogonal. Depending on the definition, "identity management" can be regarded as superordinate to the other two terms. However, in Section 3.2 a focus is put on credentials and standards for federated identity management rather than also comprising anonymity or policy features.

In principle the protocol layers introduced in Section 2.1.1 also apply for this chapter, but in current implementations the techniques for privacy-aware communication are realised in higher layers in the protocol stack (i.e., on the application layer) because it is assumed that today's existing infrastructure is being used and cannot easily be exchanged by privacy-enhancing developments also on the lower protocol layers.

3.1 Anonymisation services

The basics of anonymisation and related protocols are described within the FIDIS Deliverable D13.1 "Identity and impact of privacy enhancing technologies" (Cvrček, Matyáš 2007). Therefore we will concentrate in this deliverable on existing services offered to the public to anonymise the Internet usage of normal users, i.e., for example to allow them to browse the web anonymously.

In general one can distinguish between high latency and low latency anonymisation services. Thereby the terms "high latency" and "low latency" reflect the maximum tolerable latency from an application point of view to offer a reasonable level of quality of service to the user. A typical application for a high latency anonymisation service is e-mail communication whilst the most prominent area of application for low latency anonymisation services is browsing the web.

Mixminion (Danezis, Dingledine, Mathewson 2003) is an example of a high latency anonymisation service, whereas *AN.ON* (Berthold, Federrath, Köpsell 2001) and *Tor* (Dingledine, Mathewson, Syverson 2004) are examples for low latency anonymisation

services. AN.ON[14] has been developed in Germany by the Dresden, University of Technology (TUD) in cooperation with the University of Regensburg and the Independent Centre for Privacy Protection (ICPP) Schleswig-Holstein. Tor has its roots in the Free Haven Project[15] and the U.S. Naval Research Laboratory, while it now gets supported by the Electronic Frontier Foundation[16].

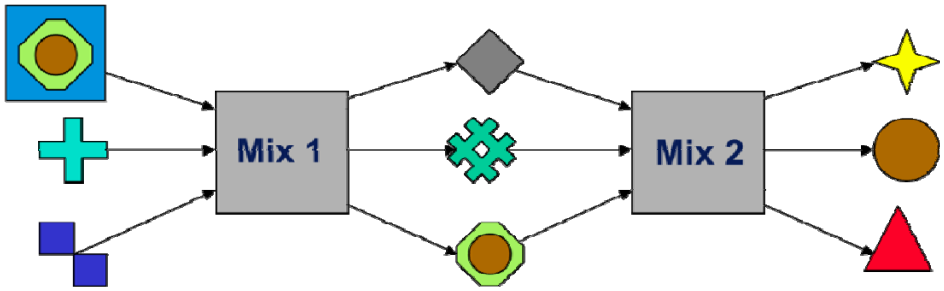

Figure 20: Basic principle of Mixes

All three services are based on the ideas of Mixes invented by David Chaum in the early 1980s (Chaum 1981). The basic idea – see Figure 20 – is to redirect the messages sent from a sender to the recipient through servers (which are called Mixes) on the Internet instead of sending them directly from the sender to the recipient. At each hop their coding changes. As the messages from multiple senders and recipients are "mixed" together an attacker who observes (all) the network links can only learn which set of senders communicates with which set of recipients but does not learn who is communicating with whom. This is an important property as anonymisation does not offer the users (and respectively his machine) a "magic hat" which makes him invisible within the network. Anonymisation just hides communication relations on the network.

Besides the general idea of redirecting the data traffic through servers, additional measures are necessary to reach this goal against even strong attackers who are able to listen on many (all) network links and to control parts of the anonymisation services (e.g., through operating or corrupting Mixes) and its users. In substance the following additional measures are necessary[17]:

- **Message padding:** each message sent through the anonymisation service is sliced and padded into fixed size packets (also called cells). This ensures that the attacker cannot link incoming and outgoing message just by looking at the message sizes.
- **Message recoding:** each message (or more precisely: each packet / cell) is recoded by each mix. This ensures that the attacker cannot link incoming and outgoing messages

[14] http://anon-online.de/.

[15] http://freehaven.net/.

[16] http://www.eff.org/.

[17] More details on Mixes could be found in the FIDIS Deliverable D13.1 Section 7.7 "Mixes".

just by analysing the bit patterns of them. This recoding is usually done by means of cryptography. The sender encrypts the message multiple times. For each layer of encryption he uses the encryption key of a different Mix of the path of Mixes from the sender to the recipient. If a Mix receives a message, it decrypts it using its secret decryption key. This way the Mix removes one layer of encryption. This will lead to a complete new bit pattern of the decrypted message. Thus an attacker would not be able to link incoming and outgoing messages after the decryption step done by the Mix.

- **Message buffering:** messages arriving at a Mix need to be delayed, otherwise an attacker could easily correlate incoming and outgoing messages just by looking at the timing of the receive/send events of a given message. How the buffering is done depends on the type of Mix. In general one can distinguish between Batch- and Poolmixes. A Batchmix collects *n* messages from *n* different senders before it forwards them all at once. Whereas a Poolmix has an internal pool (buffer) of stored messages. Every time a message arrives, it selects one or more messages from its message pool and forwards them. There exist several strategies of exactly how the messages are chosen (e.g., just randomly).
- **Message reordering:** the order of messages needs to be changed by the Mix, otherwise (e.g., if the Mix would work in a "first-in-first-out mode) an attacker could easily link outgoing to incoming messages. Different reordering strategies exist. One could either permute the messages randomly or sort them according to their bit pattern after decryption.

Note that not all anonymisation services implement all these measures. Especially the low latency systems do not implement a buffering strategy as the buffering might introduce an additional delay which would lead to an overall delay above the allowed maximum for offering an appropriated level of quality of service from an applications point of view.

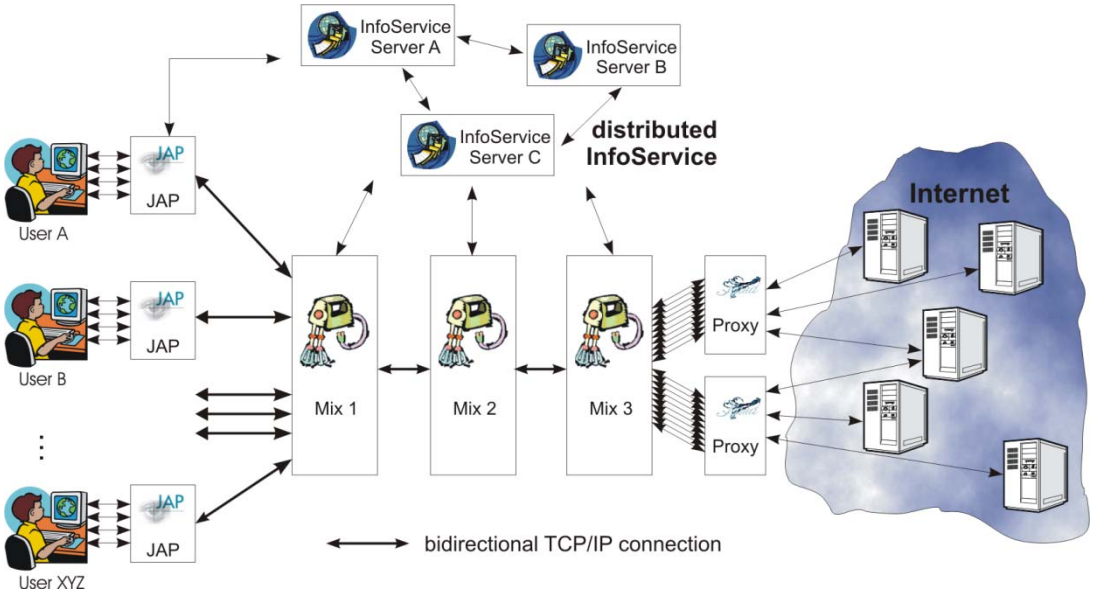

Figure 21: AN.ON architecture

One of the main differences between AN.ON and Tor is the topology of the overlay network spanned by the Mixes. AN.ON uses fixed chains of Mixes called cascades. The user can choose which cascade he wants to use but cannot select each single Mix. The Tor system uses free routes meaning that the user can freely select which Mixes he wants to have on his path from the sender to the recipient. Both concepts have their advantages and disadvantages as discussed in (Cvrček, Matyáš 2007) and in (Böhme *et al.* 2004). The general problem is that the user has to choose very carefully the Mixes (cascade) he wants to use. If all Mixes on his path are corrupt (e.g., working together with the attacker) all activities of the user can be deanonymised. To support the user in making an informed decision the anonymisation services offer a lookup service with information about the available Mixes (and cascades). Within the AN.ON system this service is called "InfoService", in Mixminion and Tor it is named "Directory Service". Besides static information like operator, location and software version of a Mix, dynamic information regarding the current number of users, the up-time and reliability as well as the traffic situation of the anonymisation network can also be queried. The architecture of AN.ON is shown in Figure 21.

In order to use one of the mentioned anonymisation services, the user has to install a client. This client software is called "Onion Proxy" in Tor and "JAP" in AN.ON (cf. Figure 22). Furthermore he has to reconfigure the applications (e.g., web browser, e-mail software, etc.) to use the installed client as a local proxy for accessing the network. This means that all network traffic is sent to the client instead of the recipient. The client is responsible for pre-processing the data according to the anonymisation protocol (e.g., for padding and encrypting the data etc.) and sends them to the first Mix in the chain of Mixes chosen by the user. The client also receives the data from the Mix and transforms it back to plaintext understandable by the application.

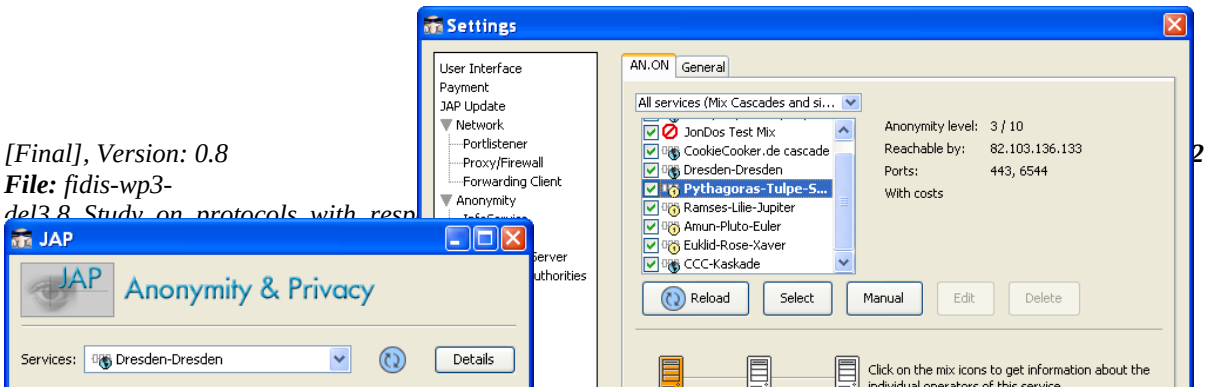

Figure 22: User interface of JAP

Tor, AN.ON and Mixminion are results of research projects and are under active development. They have not by far reached the desired level of protection (e.g., anonymisation even against strong attackers). The research in the field of privacy enhancing technologies has lead to many new attacks on anonymous communication which are at the moment not addressed by the mentioned anonymisation services. Nevertheless (depending on the attacker model and the individual protection goals of the users) we can greatly enhance the level of privacy protection of today's Internet.

AN.ON and Tor have a growing user base (roughly estimated at 50,000 users per system). One of the problems of the anonymisation services is that the Mixes are operated by volunteers. Although it allows the service to be offered to the public free of charge, it has the drawback that one has to find volunteer Mix operators and that the anonymisation services cannot give any guarantees with respect to the offered quality of service (e.g., in terms of bandwidth, latency, availability, etc.).

Therefore the AN.ON project recently started to "experiment" with commercialisation of their anonymisation service[18]. The first attempt (started in 1999) to offer a commercial anonymisation service which offers a high level of protection was done by the Canadian company Zeroknowledge Inc. But their "Freedom" service ran out of business (in 2002) as (according to Zeroknowledge Inc.) they could not attract enough users willing to pay for anonymisation. Whether or not the recent commercialisation attempt of the AN.ON project will succeed in filling a market niche, cannot be predicted at the moment. On the one side, more Internet users seem to be aware of the privacy problems related to the Internet, and also the operational costs are lower compared to 2000 (e.g., rent of servers and bandwidth). But on

[18] http://www.jondonym.de/.

the other side, the legal framework becomes much more complex, especially the European Directive on Data Retention and the uncertainties with respect to liability issues (e.g., in terms of misuse of the anonymisation service) makes the operation of Mixes risky. In general, one can say that five years ago the costs for servers and data traffic were the most hindering issues for (voluntarily) operating a Mix, whereas today the legal uncertainties are the major obstacles.

3.2 User-centric identity management

An important aspect of the right of privacy is the right to control the way personal data about oneself is collected and used. The substantial need for privacy protection is recognised by current legislation concerning the automatic processing of personal data. The Directive 95/46/EC of the European Parliament and of the Council "on the protection of individuals with regard to the processing of personal data and on the free movement of such data" (EU-DPD 1995) creates a legal framework that has to be followed by all member states of the European Union.

In traditional identity management systems identity information is hosted and managed by an identity provider using for instance directory services (Wikipedia: Directory service 2007). This has various advantages from the service provider's point of view, for instance being cost effective and easily scalable. However, by applying such an approach the user loses some control over his identity information. Recently, this has resulted in the emerging trend of *user-centric identity management* (Jøsang, Pope 2005). In user-centric identity management the user is put in the centre of interest by giving him control over his identity information. In particular, this means that the user himself decides which information should be forwarded and revealed to a particular service provider. This has the obvious advantage of better protecting the privacy of the involved user and gives the identity information back to who it belongs. The major drawback of a user-centric approach is that the user is in charge of storing and maintaining his credentials. Another drawback is due to the fact that this is a relatively recent phenomenon. As a consequence, the existing identity management standards offer only very basic support for user-centric identity management.

This survey into user-centric identity management and their support in federated identity management standards, such as SAML, Liberty Alliance, and Shibboleth is divided into four parts. In the first part we recall the functionality of anonymous credentials (also called private credentials or minimal disclosure tokens) as the most relevant cryptographic technique for user-centric identity management and discuss a change in terminology. In the next two sections we investigate the state of the art of XML security and federated identity management by looking at the most widely used (industry) standards in the field. In the last section we sketch a path for reconciling the existing federated identity management solutions with a more user-centric approach. In our opinion truly user-centric identity management can only be deployed in several phases, through stepwise deployment of new privacy enhancing technologies in the existing technological environment.

Part of the work on XML standards and credentials is based on a previous study on privacy enhancing techniques done for the Belgian IBBT E-HIP project (IBBT 2007).

3.2.1 Anonymous credentials or minimum disclosure tokens

Federated identity management is about the mechanics of identity and its power relations.[19] It is about exchanging data and establishing the validity of this data. A large part of it deals with compatibility between different systems and standards compliance.

The social dynamics that deal with determining who has the power or the customers confidence to hold identity information are outside of the scope of existing technical solutions for federated identity managements. So are the checks and balances that ensure that the dissemination of this information to other organisations follows the rules set by law or by the individual affected. The whole system is fundamentally based on trust in entities that are out of the immediate reach of the user and the technologist.

Unfortunately the attacker to privacy is not restricted in a similar fashion. Recorded data can be stored indefinitely, and powerful search and analysis tools can be used to construct a detailed picture about the user from many small puzzle pieces.

For protecting privacy, the goal set by cryptographers is to release as little unwanted information as necessary while still being able to execute secure transactions. In particular the reconstruction of the detailed picture from the small puzzle pieces should be prevented.

The best metaphor for this approach is that of a city (before the time of surveillance cameras and efficient face recognition software). Surrounded by millions of people, involved in a network of social activity, it was unlikely that any action outside of the immediate private sphere went unobserved. Someone was bound to see it, but contrary to small towns or small to medium sized companies there was little chance of information and rumours spreading.

The main feature that allows for this anonymity of the big city is something that does not exist in the digital world: ambiguity. If we see a complete stranger we may not be able to recognise him again later on. We know people in a certain context, e.g., as a regular customer to our shop, or as having certain roles: teacher, nurse, etc. It is when we start using the power of words that we start attaching names to people. At times we only know people by their first name. We meet them in a social context, and there is a long process of building up trust and mutually exchanging information. If we feel uncomfortable, we have the possibility to back off and escape into the crowd of other people, back into anonymity.

There is a strong tendency to anthropomorphise the Internet or as it is sometimes called the cyberspace, that is to use simple abstractions and to ignore the technical implications of what we are doing. These abstractions are useful, but they become dangerous when analysing the security properties of a system. When information is available in bits and bytes there is no ambiguity. Once an identifier or label has been attached to a piece of data, it is there, readable and searchable until it gets removed. We call a label that is used as replacement for a person's name a pseudonym. Not surprisingly persons that were subject to wide public attention such as writers or political figures already used pseudonyms in the pen and paper world to protect their privacy. We study the use of unlinkable user-generated pseudonyms as a replacement for the vagueness found in the real world.

[19] Knowledge is influence. Knowledge about identities is knowledge about people and is thus the essence of influence.

3.2.1.1 Pseudonymity and anonymity

Anonymity can be defined as: "... the state of being not identifiable within a set of subjects, the anonymity set." (Pfitzmann, Hansen 2007) Furthermore the authors define pseudonyms: "A subject is pseudonymous if a pseudonym is used as identifier instead of one of its real names." In the digital world pseudonyms are bit strings that are unique in a certain context. Depending on the scope of the context we can define different pseudonym types:

1. *person pseudonym:* Is a substitute of the holders name.
2. *role pseudonym:* The holder uses this pseudonym only in a given context while he acts in a certain role.
3. *relationship pseudonym:* The holder uses this pseudonym only for a single relation.
4. *role-relationship pseudonym:* The holder uses this only in a certain role and relation.
5. *transaction pseudonym:* The holder uses this only for one transaction.

These pseudonyms form a lattice according to the amount of information they reveal about the linkability to their holders. This lattice is shown in Figure 23 (cf. Pfitzmann, Hansen 2007).

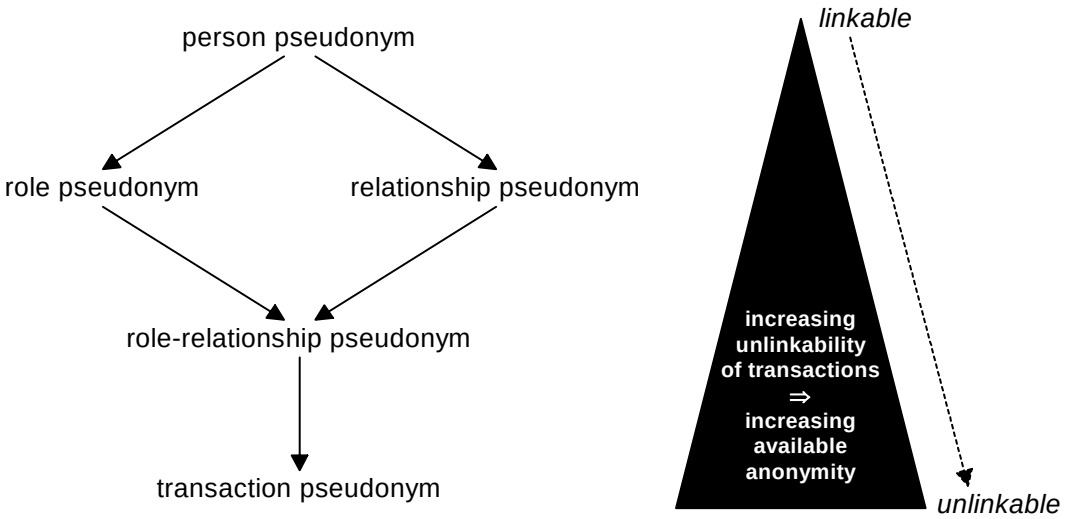

Figure 23: Pseudonym types

In wide area networks – such as the Internet – every transaction leaves a data trace. Without protection measurements these traces contain data like who accessed which data/service at which time. This information is revealed, even if the data content itself is encrypted. Already in a closed environment like a company the protection of the network against externals is a necessity but it is more difficult to protect sensitive transaction data against an internal attacker. For example in a hospital, the network administrator should not be able to access a

patient's personal data nor should he see who has accessed this data when. Similarly only the information that a user is requesting a certain service may put her privacy at risk.

If one wants to integrate databases in a federated manner the problems mentioned above are amplified. Access control mechanisms and basic encryption schemes can protect sensitive databases against external observers. If aggregated data is to be made public, data anonymisation techniques can be used to protect the privacy of individuals who contributed to the results.

With the help of pseudonyms we can do more. By revealing only the most necessary information, we can protect sensitive data even against too curious service/infrastructure providers. If pseudonyms are unlinkable the privacy attacker cannot construct the complete picture from the information he collected about the pseudonyms.

Pseudonyms help to protect a user's privacy; they also change the way conventional security properties like confidentiality, integrity and availability are guaranteed. These properties are often based on a long term relationship between the user and the organisation that wants to enforce the properties. If pseudonyms are used, the organisation needs to establish the access rights of the user for every pseudonym separately. It is not who you are, but what you are that determines the rights of the user. In this sense, the introduction of pseudonyms forces the security engineer to think more deeply about the connection between identity and security. It was already noted that this connection is one of the hard problems in network security engineering, even if only conventional security properties and mechanisms are considered:

Identifying a person by a unique identifier is not as simple as it may seem. Social security numbers and credit card numbers get stolen (Weinstein, Neumann 2003). Digital certificates have their weaknesses: Companies or persons can have the same or similar names. Certificate authorities generally disclaim any liability and any meaning to identity certificates. These problems are elaborated in more detail by Ellison and Schneier (Ellison, Schneier 2000). Consequently, the use of unique identifiers alone, while allowing tracking and profiling, does not provide adequate security.

In order to achieve accountability we need to connect identifiers or pseudonyms to authentic and certified information. Instead of identifying themselves, users show information and evidence of the correctness of this information under a pseudonym.

In the following section we show how to achieve this in a privacy-friendly way.

3.2.1.2 Using credentials

Giving users protected documents which allow them to prove statements about their relationships with public and private organisations is an alternative to keeping large centralised user records. Paper world examples of such documents are passports, driving licenses, and money. Further examples include credit cards, health insurance cards, cinema and public transport tickets, club membership cards, and game-arcade tokens.

Some of these documents can be easily forged while others are protected by special physical properties and the law. Documents providing information about the owner and his relationship with other entities which are protected against forgery are called *credentials*.

Credentials are issued by organisations that ascertain the authenticity of the information and can be provided to verifying entities on demand.

A common example is the ID, be it a passport, a driver's license, or some other identification card. An ID usually contains the essential personal information about its owner and is certified by the ID issuer. In certain situations it may be advantageous to reveal only parts of the information contained in a digital ID card, e.g., some lower limit for the person's age or the fact that the person is capable of driving a car.

Credentials can have several security properties and can be equipped with optional features in order to address the three main concerns of identity management: to protect the business of companies, to ensure the sovereignty of an open society, and to ensure the privacy of individuals:

- Firstly, and foremost, they have to be unforgeable to guarantee that all the information transferred is vouched for by their issuer. This also includes consistency in the sense that colluding users cannot combine credentials.
- Secondly, they should reveal only what the parties involved in the transaction agreed upon. This also includes relationship information. We discuss below how this property is connected to anonymity and the use of pseudonyms. For this we will split up this property into three sub-properties.

Since David Chaum (Chaum 1985) first defined the concept of *digital credentials* and *pseudonyms*, a lot of thought has been invested into giving individual users the best amount of privacy and protection against abuse while still providing companies and the society at large with the required security features to protect businesses and societal goals against misuse by individuals.

The anonymous credential system proposed by Chaum is sometimes also referred to as a pseudonym system (Lysyanskaya *et al.* 2000). This stems from the fact that the credentials of such a system use the psedonymity paradigm to control linkability. Credentials are obtained from and shown to organisations using different pseudonyms which cannot be linked. In certain extraordinary situations trusted organisations might be authorised to link two pseudonyms or even to reveal the identity of the user. This procedure is called anonymity revocation.

The introduction of pseudonyms (Chaum 1985) is a useful extension to anonymity. Pseudonyms allow users to choose a different name with each organisation. Generally these pseudonyms cannot be linked without the help of the user. Nevertheless certain statements about the relationship of a user with one organisation, under a pseudonym, can be shown to another organisation that knows the user only under a different pseudonym (Lysyanskaya *et al.* 2000). While pseudonyms allow organisations to create accounts for individual users, organisations cannot determine the real identities of their customers.

The extent of influence law enforcement will have on credential systems is still a matter of debate. Key escrow has been discussed in the nineties as a possibility for allowing law enforcement authorities to eavesdrop on encrypted connections (Abelson *et al.* 1997). Anonymity revocation serves a similar purpose by revoking the anonymity of communication

and credentials and may face similar resistance. While the cryptographic techniques for identity revocation already exist (Camenisch, Lysyanskaya 2001), implementing the required trust infrastructure and getting the necessary support from all parties involved is not an easy task. However, if the only alternative to anonymity revocation and trustee based tracing is to have no anonymity at all, the wisdom of fighting these privacy restricting technologies becomes doubtful. This however does not remove the need for privacy activism, as each restriction of privacy should be judged on its own ground to avoid a domino effect.

We formalise the properties required from a credential system using data structures and algorithms. We will also show conventional PKI algorithms that can be seen as a special case of anonymous credentials.

First we note that a credential *cred* is a secret entity that is known only to a user. Similarly pseudonyms consist of a public part $pseu_{pk}$ and a secret part $pseu_{sk}$. All of these datastructures are created through interactive protocols, to which both sides contribute randomness, and information computed in other algorithms. As is the case in most cryptographic algorithm, the first thing parties do is to generate keys: *UserKg* generates the users secret and public key (sk_U, pk_U) and *OrgKg* generates the keys (sk_O, pk_O) of organisations that want to issue or verify credentials.

Sign(*sk*, *m*)
Both the secret keys of organisation and users can be used as conventional signing keys. This allows users to sign messages under their identity. It is also possible to sign using the pseudonym secrets $pseu_{sk}$. The output of the algorithm is the signature σ.

Verify(*pk*, *m*, σ)
verifies a signature on a message with respect to a specific public key, or the public part of a pseudonym.

IssueCert(sk_O, *attrs*)
An organisation can use its key to sign attributes in order to create a conventional certificate cert. For an identity certificate the public key of the user pk_U and the user's name would be one of the attributes. The certificate consists of a signature σ and the attributes and can be verified by treating the attributes as the message using *Verify*(pk_O, *attr*, σ).

The issue protocol may be interactive to allow for a proof of possession, i.e., the user has to show that he actually possesses the secret key sk_U corresponding to the key pk_U. In order to allow for the full power of anonymous credentials we require interactive protocols in which users and organisations input different data and interact in multiple rounds of communication. We write an interactive protocols *Prot* as *ProtU*(*input*) ↔ *ProtO*(*input*).

RegisterNymU(*skU*, *unique?*) ↔ *RegisterNymO*(*unique?*) In order to create a pseudonym pair $pseu_{pk}$/$pseu_{sk}$ the user uses her master secret. The organisation does not learn anything about sk_U. In particular, it cannot link the transaction to the user's public key pk_U. The *unique?* flag allows restriction of pseudonym creation to one pseudonym per organisation. Each user can only register one unique peseudonym with each organisation. Unique pseudonyms and normal pseudonyms can coexist in parallel.

IssueCredU($pseu_{sk}$, pk_O, *issuespec*, *userattrs*) ↔ *IssueCredO*(sk_O, $pseu_{pk}$, *issuespec*, *orgattrs*). Using this protocol a user can obtain a credential that fulfills a certain specification from an organisation.

The credential will contain the user chosen attributes *userattrs*, as well as the organisation chosen attributes *orgattrs*. The first are only known to the user. In addition to the attributes, the specification may describe features of the credential, e.g., that it should be limited to a certain number of shows, or should allow for certain types of revocation or tracing. Some of the features may require additional hidden attributes to be added to the credential. These attributes may be chosen jointly at random with both the user and the issuing organisation contributing randomness. The user's output of the protocol is a private credential *cred* that consists of the attributes, a reference to the issue specification and a special type of signature known as a CL-signature. Note that big parts of *cred* are only known to the user, and that the credential contains the pseudonym and the user secret it was issued to, either by reference or by value.

ShowCredU($pseu_{skV}$, pk_{OI}, *showspec*, *cred*) ↔ *ShowCredO*($pseu_{pkV}$, pk_{OI}, *showspec*) This protocol allows a user with pseudonym $pseu_{skV}$ to show a credential fulfilling a show specification to a verifying organisation. $pseu_{skV}$ is the pseudonym at the verifying organisation and does not necessarily correspond to the pseudonym the credential was issued to. The show specification needs to be compatible with the issue specification. It contains a description of which attributes have to be revealed, but can also describe functions over multiple attributes or boolean expressions of different options that are to be shown. So, one can prove that one is of a certain minimum age and that the credential is not yet expired. The show specification also describes the credential features that are required. The result of the credential show is an *assertion* containing statements about the user and a proof that guarantees that the user is in possession of a credential such that these statements hold true.

The assertion can also contain statements about committed or encrypted data, e.g., that a certain binary object contained in the assertion is the encryption of the user's name, or a commitment to certain values of the user.

3.2.1.3 Discussion

Users are not forced to always use the same secret key, however they can only show credentials together, if all of the pseudonyms they were issued to were derived from the same user secret. In order to avoid users with multiple master secrets one can establish a root organisation that only issues a root credential to unique pseudonyms after verifying the real identity of the user. Organisations that want to avoid single users with multiple secret keys (so called Sybils), can require the show of a root credential before doing anything else. Now the only entity that can introduce Sybils is the root organisation itself.

The show credential protocol given above always shows credentials with respect to a pseudonym. This is not strictly required. The protocol can use a pseudonym created on the fly, or the *ShowCred* protocol could be simplified to require no pseudonym as input at all. While the model of most existing credential systems is slightly more restricted, it can easily be seen that it can be relaxed to allow for the full lattice of pseudonyms described above:

1. The user's public key can act as a *person pseudonym*.
2. A user could derive *role pseudonyms* from his secret key and use it with different organisations whenever he wants to act in a certain role.

3. Unique pseudonyms are organisation wide *relationship pseudonyms*.
4. Normal pseudonyms used only in a specific role are *role-relationship pseudonyms*.
5. By creating a new pseudonym for a new transaction the user establishes a *transaction pseudonym*.

A restriction of the current interface is that the assertion obtained by *ShowCred* only contains statements about one credential. This makes it impossible to express statements such as "the user has either a passport or a driver's license and we have an encryption of her name". Using general zero-knowledge techniques this is however possible. As a way to express this using the given API we propose to use meta-assertions. The user does a show of both a passport credential and a driver's license credential using a special show description that forces the user to either show the real credential and encode 1 in an additional commitment, or to show a fake credential that he made up by himself and encode 0 in the commitment. This results in two different assertions. The *CombineAssertions* protocol takes several assertions as input and allows the user to prove statements according to a meta show specification about it. In the example above the user would prove that the sum of the additional commitments is bigger than 1 or different from 0.

3.2.1.4 Security properties

As discussed, not only the content of transactions should be protected, but also the communication relations, i.e., who communicated with whom. Clearly anonymous credentials do not protect the identity of their owners, if this information is already given away through other channels, e.g., through the user's IP address, or through a picture recorded by a camera. Anonymity may not even be desired in all cases where anonymous credentials are used. For instance, the parties may know beforehand to whom they are talking, but may want to conceal this information from a third party. This is, for instance, often the case with anonymous e-cash, as it is with real money. The parties may also want to perform a transaction, but want to avoid the case that later on anyone can prove that the transaction took place. While credentials are not the only tool for achieving this, traditional certificates have none of these properties. Consequently, as anonymity is often not even in the picture, the term anonymous credential is often misleading. Different terminology such as private credential, private certificate, minimal disclosure proof, minimal disclosure token, and so on were used in the literature.

In particular we require such a primitive (whatever its appropriate name) to have the following privacy properties:

- Issue unlinkability: The issuer should not be able to link the issuing of a credential to its use.
- Show unlinkability: The uses of a credential should not be linkable among each other, nor should multiple credential shows under different pseudonyms allow linking of the pseudonyms to their owner. Often the two unlinkability properties are subsumed under one property.

- Minimal data disclosure: Credentials should not reveal more information about their attributes than specified in the show specification. Some credentials may only transfer a single bit, e.g., club membership. Most of the time however credentials vouch for different attributes of the user, of which only some algorithmically derived values are revealed in the assertion. Depending on the functionality of the system this derivation may be arbitrarily complex. Minimal data disclosure requires that no information besides that is revealed in a show.

3.2.1.5 Idemix

Idemix (Kohlweiss 2002) is a practical implementation of an anonymous digital credentials infrastructure which can help to achieve this goal.

The idemix software is currently divided into different layers. The mathematical abstraction layer can be seen as its own layer, but primarily there is the protocol layer, and the credential system protocol layer. These layers are also represented by their own packages: protocols and credsystem.

The credsystem package provides a high-level view on anonymous credentials with pseudonyms. Users can obtain private certificates, i.e., credentials, and can create proofs about them. The proofs are all with respect to one pseudonym, but can cover multiple certificates. A proof corresponds to an assertion that is expressed in XML but can also involve binary objects such as commitments and encryptions.

The protocols package should provide the cryptographic protocols for achieving this high level goal. At the moment it contains protocols for obtaining and showing individual credentials, and some zero-knowledge proof techniques for proving properties about commitments. Due to the lack of documentation it is currently preferable to use the protocols package alone, especially if one does not require a full-fledged credential system.

3.2.2 XML standards

With the rise of web services, XML has become the *de facto* standard to represent messages, exchanged between the components of those services. An XML syntax to transmit the result of security primitives and protocols based upon those primitives followed in a natural way, with the main advantage that they could offer end-to-end security instead of point-to-point security. In this section we describe those XML languages and the protocols that emerged from them.

3.2.2.1 XML Digital Signatures

XML Signatures (Eastlake, Reagle, Solo 2002) are more closely related to standards like PKCS#7 (RSA 1993), that define messaging syntax, than to raw cryptographic signature algorithms. The XML Signature standard defines how to relate source documents to their cryptographic signature by using XML. Additionally the XML Signature document can encompass cryptographic keys and certificates, e.g., X.509 certificates.

3.2.2.2 XML Encryption

XML Encryption (Eastlake, Reagle 2002) describes a standardised syntax for including encrypted data in XML documents. The encrypted data can be an XML element, the content of an XML element, or a whole document of a specified MIME type. XML Encryption is to confidentiality what XML Signatures are to authentication and non-repudiation, a standardised way of writing things down that in no way restricts the user of the standard in his choice of encryption algorithm or key material used.

3.2.2.3 WS-Security

The WS-Security standard (Nadalin *et al.* 2004) describes a set of SOAP extensions used to provide message integrity, message origin authentication and message confidentiality. WS-Security operates at the application level and thus is used to provide end-to-end message security and not just transport-level security (e.g., SSL/TLS). WS-Security gives the application the possibility to make the security decisions with a higher granularity, for instance allowing access to a certain web service method for an entity while denying access to other web service methods from the same entity. For instance a company might decide to offer certain functionality to the public while some sensitive functionality is only offered to partner companies.

WS-Security consists of a set of standards that specify how to protect the authenticity and confidentiality of SOAP messages. Message integrity and origin authentication of the SOAP message is realised by using XML digital signatures to ensure that messages originated from the appropriate sender and have not been modified whilst in transit. The confidentiality of the SOAP message is accomplished using XML Encryption to ensure that the sensitive parts of the message are not visible to eavesdroppers. The low level details for digital signature and encryption are handled by the XML Digital Signature and XML Encryption standards respectively. The WS-Security standard defines a higher-level syntax to place security information into SOAP messages. The security relevant information is placed in the SOAP message header.

3.2.2.4 WS-Trust, WS-MetadataExchange, WS-SecurityPolicy

While the XML standards, mentioned above, are basic components in web services and XML documents, WS-Trust (Nadalin *et al.* 2007a), WS-MetadataExchange (Ballinger *et al.* 2006) and WS-SecurityPolicy (Nadalin *et al.* 2007b) are mainly used in the Identity Metasystem as proposed by Microsoft together with others. For completeness, we introduce them here - the descriptions were taken from the standards' documentation.

- **WS-MetadataExchange:** Web services use metadata to describe what other endpoints need to know to interact with them. Specifically, WS-Policy describes the capabilities, requirements, and general characteristics of web services; WSDL (Web Services Description Language) describes abstract message operations, concrete

network protocols, and endpoint addresses used by web services, and XML Schema describes the structure and contents of XML-based messages received and sent by web services. To bootstrap communication with a web service, the WS-MetadataExchange specification defines three request/response message pairs to retrieve these three types of metadata: one retrieves the WS-Policy associated with the receiving endpoint or with a given target namespace, another retrieves either the WSDL associated with the receiving endpoint or with a given target namespace, and a third retrieves the XML Schema with a given target namespace. Together these messages allow efficient, incremental retrieval of a web service's metadata.

- **WS-Trust:** WS-Security defines the basic mechanisms for providing secure messaging. WS-Trust uses these base mechanisms and defines additional primitives and extensions for security token exchange to enable the issuance and dissemination of claims about a subject, as bound to X.509 certificates, Kerberos tickets, or even password hashes within different trust domains. In order to secure a communication between two parties, the two parties must exchange such security tokens (either directly or indirectly). However, each party needs to determine if they can "trust" the asserted claims of the other party. In WS-Trust, extensions to WS-Security are defined that provide methods for issuing, renewing, and validating security tokens and ways to establish and assess the presence of, and broker trust relationships.
- **WS-SecurityPolicy:** WS-Policy defines a framework for allowing web services to express their constraints and requirements. Such constraints and requirements are expressed as policy assertions. WS-SecurityPolicy defines a set of security policy assertions for use with the WS-Policy framework with respect to security features provided in WS-Security, WS-Trust and WS-SecureConversation. WS-SecurityPolicy takes the approach of defining a base set of assertions that describe how messages are to be secured. Flexibility with respect to token types, cryptographic algorithms and mechanisms used, including using transport level security is part of the design and allows for evolution over time. The intent is to provide enough information for compatibility and interoperability to be determined by web service participants along with all information necessary to actually enable a participant to engage in a secure exchange of messages.

3.2.3 Identity federation standards

There are currently two XML standards for identity federation which are supported by different organisations:

1. The Liberty Alliance is a consortium of around 150 companies supporting mainly an XML dialect called SAML (Security Assertion Markup Language).
2. Microsoft, IBM and VeriSign are supporting WS-Federation. WS-Federation is part of a more general specification of web services and partially overlaps with the approach of the Liberty Alliance due to the fact that various companies using the WS approach are also member of the Liberty Alliance.

It is still an open question which of the two approaches will finally get accepted by the community or if an integrated solution will be developed. Until then, organisations which are planning to establish an identity federation should agree on one of the two standards to decrease system complexity and avoid an additional source of error[20].

3.2.3.1 SAML

SAML was developed by the Security Services Technical Committee of OASIS. It is an XML-based framework for communicating user authentication, entitlement, and attribute information. This is done by making assertions regarding the identity, attributes, and entitlements of a subject (e.g., a human) to other entities, such as a partner company or another enterprise application. SAML can be extended by other standards, and this has happened through the Liberty Alliance Project and the Internet2 Shibboleth project.

Two main flavours are currently in use: SAML v1.1 was established in September 2003 and is the current basis for the Shibboleth project. From within the Shibboleth project, suggestions and improvements were taken up, together with comments from the Liberty Alliance's Identity Federation Framework, to form the improved SAML v2.0. At the time of writing, this version is supported by the Liberty Alliance project.

SAML can be used in three major ways:

- **Web single sign-on (SSO):** In web SSO, the user logs on to one web site and can access other web sites later on with the identity provided previously, without re-authenticating. This assumes an underlying trust relationship between the web sites' providers.

[20] There are software solutions which allow the communication between the standards, e.g., HP OpenView Select Access and IBM Tivoli Federated Identity Manager.

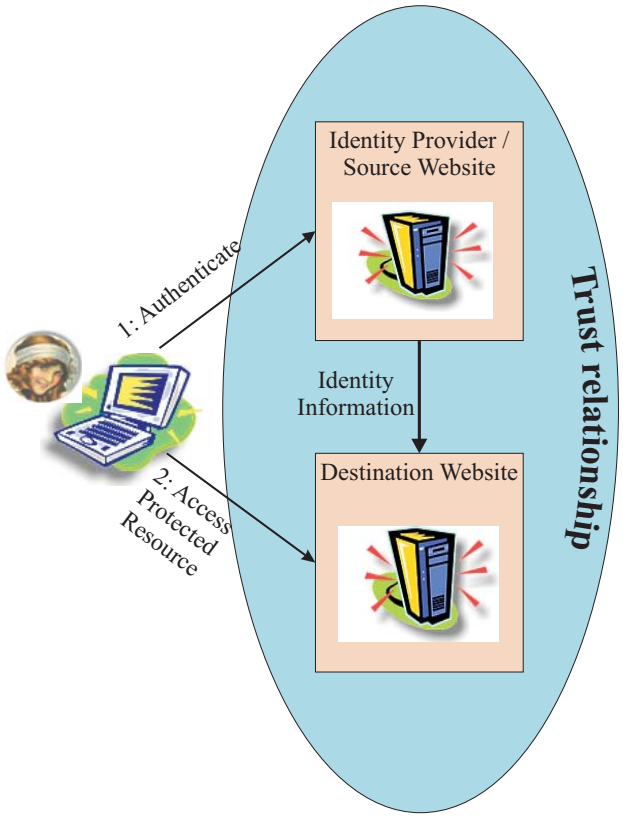

Figure 24: Web single sign-on

- **Attribute-based authorisation:** Similar to web SSO, but authentication to third party web sites happens by passing statements about the user, not necessarily his exact identity.
- **Securing web services:** When used in combination with SOAP, SAML can be used to transfer information about interacting users in web service transactions. This is specified in the OASIS WSS TC and used in Liberty Alliance's Identity Web Service Framework (ID-WSF).

The term federation is used when Web SSO is applied with different identifiers or user accounts. Federation refers to the establishment of a business agreement, cryptographic trust, and user identifiers or attributes across security and policy domains to enable more seamless cross-domain business interactions.

SAML is a platform-independent tool to achieve this.

SAML components

SAML is specified in terms of assertions, profiles, bindings and protocols:

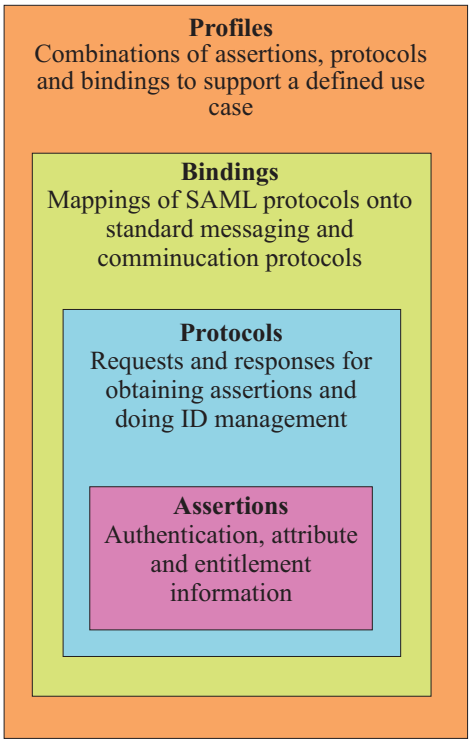

Figure 25: SAML components

- **Assertions** are the XML snippets in which SAML statements are contained. There exist three types:

  – Authentication assertions hold claims that a certain subject was authenticated using a given method at a given time.

  – Attribute assertions bind certain attributes to a given subject

  – Authorisation Decision assertions contain a request to grant or deny a certain subject access to a certain resource.

- **Protocols:** The SAML standard describes some request/response protocols to allow the exchange of SAML assertions. Service providers can request registration, mapping or termination of name identifiers, request assertions from SAML authorities and authentication assertions from identity providers, request simultaneous logout for a collection of related session, etc.

- **Profiles** of SAML specify constraints and/or extensions to make SAML work in a specific context. One of the first profiles for SAML was the Web SSO profile, which describes how SAML statements are communicated between the identity provider and the service provider to enable single sign-on for a browser user.
- **Bindings** map SAML protocols to standard communication protocols. Common bindings are the ones to SOAP and HTTP (using redirection).

More details on SAML v2.0 can be found in the core standard document (Cantor *et al.* 2005) and the Technical Overview of SAML (Ragouzis *et al.* 2006).

3.2.3.2 WS-Federation

WS-Federation was developed by IBM and Microsoft with the help of RSA-Security, BEA Systems, and Verisign. It is a specification but not a standard and forms part of the WS-* line of specifications. As such it is based on other WS-security specifications, in particular WS-Security and WS-Trust. It extends WS-Trust by introducing identity providers as special Security Token Services (STS). By involving identity providers WS-Federation allows different security realms to federate by allowing and brokering trust of identities, attributes and authentication between participating web services.

The use cases of WS-Federation are very similar to those of SAML. The more stringent focus and restriction to WS can be seen as the main dividing line.

- **Securing web services:** This is the main use case of WS-Federation.
- **Single sign-on:** WS-Federation allows single sign-on for distributed web service architectures.
- **Attributes and pseudonyms:** Attribute-based authorisation is similar to SSO, but authentication to third party web services is done based on information about a user's pseudonym or attributes. WS-Federation defines attribute and pseudonym services to provide this functionality. Attribute services should provide service providers with additional information about a subject (constrained by the subjects access control and privacy policies), while pseudonym services allow binding of persistent local pseudonyms to otherwise anonymous subjects.

While otherwise kept very generic, the specification of the WS-Federation defines two different profiles for clients:

Passive clients: A passive client does not know that he is used in a federation. This could for instance be a web browser which secures its connections with SSL (Secure Socket Layer).

Active clients: An active client knows about the federation system and thus can take advantage of the federation protocols to achieve an increased flexibility, security and productivity.

More details on WS-Federation can be found in the core specification (Nadalin, Kaler 2006) and a technical overview of WS-Federation (IBM, Microsoft 2003).

3.2.3.3 SAML, WS-Federation, and pseudonyms

Both SAML and WS-Federation support identity provider managed pseudonyms. In this way service providers may know users only under different pseudonyms. While this makes it harder for colluding service providers to collect information about a user, it gives more power to the identity provider. This was discussed in the increasingly active identity management blogosphere, e.g., in (Brands 2005). For instance an identity provider may "steal" some of a user's pseudonyms and link them to a different user.

3.2.4 Credentials and federated identity management

Until recently the necessary cryptographic mechanisms had not been fully implemented. The recent creation of an Idemix open source initiative as well as other attempts to make anonymous credential systems real for the first time created the possibility for developers to use credentials in real system. However, there are still many steps to take from the vision of privacy-enhancing identity management (as conceived by Chaum) to its realisation. The PRIME project[21] looked at all of these steps in more detail. The main contribution of this ambitious European research project is to gain an understanding of the dependencies between the different components in such a system. These dependencies are reflected by an identity management architecture, and by an integrated prototype (Andersson *et al.* 2005, Leenes, Schallaböck, Hansen 2007).

3.2.4.1 PRIME

In short, having anonymous certified data is not enough. These data also need to be used. An important missing component that needs to be added to anonymous credentials is thus attribute based access control. Such a system takes the certified attributes provided by a credential system and uses them to make access control decisions. In order to achieve accountability, such systems may also require an encrypted version of the requestor's identity that can only be decrypted by a special trusted authority, or a quorum of parties. This allows for conditional anonymity. The complex rules that govern who can access what after having given away which information are defined in policies. Policies can also stay attached to data after it has already been released.

[21] https://prime-project.eu/.

3.2.4.2 User interface component

As there is a natural degradation from full anonymity to full accountability which depends on the gradual release of more and more attributes, checks and bounds need to be provided to users to allow them to control the release of their personal data. This is called negotiation. Depending on the user's relationship with an organisation she will be willing to release more or less data. This raises many user interface question, as the amount of extra complexity users will tolerate to manage their identities is limited. In fact applications should not get more complicated, but should be redesigned to provide the same usability with added privacy.

3.2.4.3 Federated identity management

Another line of research looks at interoperability issues between conventional identity management infrastructures and anonymous credentials. The en vogue identity management paradigm that is currently hyped by the industry is federated identity management. It uses only conventional cryptographic techniques, has clear basic principles, and an already large product and standards portfolio. Still, the interoperability issues between different vendors and different domains define it as a moving target. This makes the integration of anonymous credentials and federated identity management a challenge from an engineering if not from a research perspective.

In federated identity management, we have an online request for identity information by the service provider (sometimes also called dependent party) the request is redirected through the user's client, most often this is a web browser, and the identity provider answers with an assertion signed using conventional signatures.

Most of the work for reconciling federated identity management with anonymous credentials hooks in at this conventional signature on the data provided about the user by the identity provider. Obviously the identity provider can recognise his own signature. This allows insiders to link transactions. More seriously, most of today's identity federation protocols require the user to explicitly tell the identity provider which services they want to access.

In order to achieve unlinkability the user obtains anonymous credentials from his identity providers. These credentials allow the user to do authorised transactions without communicating online with the identity provider. While we want to change the flow of communication, we do not want to change the standardised message formats defined by federation standards such as WS-Federation of the Liberty Alliance protocols (Liberty Alliance 2007). Thus, in order to achieve full unlinkability, the user himself has to create a fresh new signature every time he proves a statement about himself in an assertion to a service provider. On the other hand the signature must be restricted to certified statements that are consistent with the user's credentials. In short, what (Camenisch, Gross, et. al) and (Camenisch, Kohlweiss, et. al) say about this topic is that:

(A) the XMLDSIG standard which is used for this purpose needs to be extended

(B) the assertions proved by a credential need to be expressed using XML syntax, and

(C) the assertions made by the credential need to be translated to the assertions expressed by identity providers.

(A) can be achieved in two ways: Firstly, by using the signature proof of knowledge corresponding to a credential show directly as a signature in XMLDSIG. Instead of using XMLDSIG with DSS or RSA one would use non-interactive proofs of knowledge. Secondly, by using the credential to create and thus in some sense certify a fresh signing key, which will be used in the XMLDSIG, while the signature proof of knowledge of the credential acts as a certificate for the freshly created signing key. In WS-Security terminology such a generalised certificate is referred to as a security token (Dournaee 2002).

3.3 Privacy policy languages and protocols

Privacy policy languages are designed to support organisations and end-users in managing their privacy policies and preferences. In particular they may help regarding some or all of the following steps: writing, reviewing, testing, approving, issuing, combining, analysing, modifying, withdrawing, retrieving and enforcing privacy policies (Moses 2004, Kumaraguru *et al.* 2007). The development of privacy policy languages, the specification of their syntax and semantics, and the interaction with ICT systems, e.g., protocols for negotiating and matching policies, belong to a highly dynamic field. Since 1997 when W3C started the development of the Platform for Privacy Preferences (P3P), a variety of languages and protocols have been proposed which are specifically designed to manage privacy policies or – even if their main objective was less privacy-specific, can be applied for data protection purposes as well.

This section gives an overview of today's privacy policy languages which are categorised according to their main areas of use. Important properties are briefly discussed. Finally, upcoming work to standardise privacy policy languages and protocols is described.

3.3.1 Categorisation of privacy policy languages

Each proposal of a new privacy policy language or an enhancement of existing one has entailed discussions on advantages and disadvantages comparing the new approach with selected related work. However, in each scientific paper usually only two or three different policy languages to be used in the area of data protection are compared so that it is hard to get a broader overview.

The most comprehensive proposal for a categorisation was recently done by Kumaraguru, Cranor, Lobo and Calo who presented their research plan on the SOUPS (Symposium On Usable Privacy and Security) 2007 conference in July 2007 (Kumaraguru *et al.* 2007). The authors classified thirteen privacy policy languages according to their categorisation scheme which we extended in this deliverable (see Table 19). Another categorisation, based on the analysis of four privacy policy languages, was proposed by Madsen, Casassa Mont and Wilton who used their results to position Liberty's ID-WSF (Liberty Alliance 2007) as a privacy policy framework (Madsen, Casassa Mont, Wilton 2006). Also Anderson, who is active in XACML specification, compared several privacy policy languages to propose an own advantageous specification (Anderson 2005, Anderson 2005-2006, Anderson 2006). Further, in the area of semantic web, work is done on policy languages and their comparison (e.g., Tonti *et al.* 2003; Oldemilla 2007; Bonatti, Oldemilla 2007).

Combining the results from all that work, we enhanced the categorisation proposed by Kumaraguru *et al.* (Kumaraguru *et al.* 2007), see Table 19. The following sections will give more information on each listed privacy policy language and point to further references.

Area of use	Perspective	Examples
Sophisticated access control languages	Organisation	ODRL, SAML, WSPL, WS-PolicyConstraints, XACL, XACML,
	User	WSPL, WS-PolicyConstraints, XACML
Enterprise privacy policy languages	Organisation	CPExchange, DPAL, EPAL, E-P3P, PRML
Web privacy policy languages	Organisation	P3P
	User	APPEL, XPref
Context sensitive languages	Organisation	Geopriv, KAoS, PeerTrust, Ponder, Protune, Rei
	User	Geopriv, PeerTrust, Protune

Table 19: Categorisation of privacy policy languages (based on Kumaraguru *et al.* 2007)

According to Kumaraguru *et al.*, the following categories can be distinguished:

1. **Sophisticated access control languages:** These languages are based on access control languages, especially on Role Based Access Control (RBAC). Usually they implement security policies maintained by system administrators.
2. **Enterprise privacy policy languages:** These languages are used within an organisation to enforce what has been stated in their internal privacy policy. This category is sometimes regarded as part of the first category because it also deals with sophisticated access control. However, we list it as an additional category to demonstrate the difference between those languages which explicitly deal with data protection and privacy issues.
3. **Web privacy policy languages:** These languages represent standardised human-readable privacy policies from web sites formats which can be interpreted by machines.
4. **Context sensitive languages:** These languages take into account the context when interpreting the policy, e.g., in the semantic web. They can be used for providing personalised services.

In all cases where privacy policies are not only limited to the internal use within an organisation (as it is the case with Enterprise Privacy Policy Languages), the categories above can be separated into the user's perspective (defining preferences on how to deal with the data) and the organisation's perspective (defining promises on how to deal with the data).

3.3.2 Sophisticated access control languages

Access control languages do not solve all problems related to privacy or data protection. In particular principles like data minimisation or transparency are not in the main focus of access control languages. However, Sophisticated access control languages can implement many other properties of data protection and privacy, in particular pre-defined settings on "who is allowed to access personal data under which conditions?".

The most prominent examples in this category are SAML (Ragouzis *et al.* 2006) and XACML (Moses 2005a; Moses 2005b). The examples listed in Table 19 are briefly explained:

Open Digital Rights Language (ODRL) is a rights expression language for digital asset management and e-commerce. It aims at the Digital Rights Management (DRM) community and is proposed to be used for expressing right information over content (Iannella 2002). It provides mechanisms to support transparent use of digital resources in publishing, distributing and consuming of electronic publications, digital images, audio and movies, learning objects, computer software and other creations in digital form. ODRL has been accepted by the Open Mobile Alliance (OMA) as the standards rights expression language for all mobile content.[22]

Security Assertion Markup Language (SAML) is an XML standard for exchanging authentication and authorisation data between security domains (Cantor *et al.* 2005). SAML addresses in particular the web single sign-on problem. SAML is being developed by the OASIS Security Services Technical Committee. The latest version is 2.0.

Web Services Policy Language (WSPL) is a subset of XACML, i.e., the XACML profile for web services. It enables negotiating policies between entities on the level of constraints and assertions (Moses 2003). WSPL is a Working Draft in the OASIS XACML Technical Committee.

WS-PolicyConstraints has been developed by Anderson as a subset of WSPL to overcome the disadvantage that WSPL conflicts with WS-Policy and WS-PolicyAttachment (Anderson 2005-2006). WS-PolicyConstraints is a domain-independent language for expressing constraints for a web services policy.

XML Access Control Language (XACL) is an XML-based language to specify security policies to be enforced on specific accesses to XML documents (Hada, Kudo 2000). XACL's authorisation model is not restricted to granting or denying read, write, create or delete access, but these primitive actions can be combined with additional actions such as auditing, digital signature verification, encryption, and XSL transformations.

[22] http://odrl.net/docs/ODRL-brochure.pdf

eXtensible Access Control Markup Language (XACML) is a widely adopted standard language for access control, authorisation, and privacy policies (Moses 200). With XACML both privacy and security policies can be expressed in a machine-readable format. It is used for enforcing privacy policies. XACML is being developed by the OASIS eXtensible Access Control Markup Language (XACML) Technical Committee. The current version is 2.0, version 3.0 is in preparation.

3.3.3 Enterprise privacy policy languages

Enterprise privacy policy languages have been proposed since 2000. They represent the internal policies of an enterprise and support it to perform the actions as stated in its privacy policy. Often these languages are more fine-grained than web privacy policy languages.

The most prominent example in this category is EPAL. The examples listed in Table 19 are briefly explained:

The **Customer Profile Exchange (CPExchange)** specification integrates online and offline customer data in an XML-based data model for use within various enterprise applications both on and off the web (Bohrer, Holland 2000). It does not only standardise data formats for facilitating easier information exchange across the entire network, but also specifies metadata for associating privacy controls to the data. Privacy declarations are defined in a privacy header of a CPExchange document. The privacy information model is based on P3P, extending the Working Draft version from October 2000. The CPExchange specification is available in version 1.0.

Declarative Privacy Authorization Language (DPAL) is a formal policy language that evaluates and enforces all statements in the DPAL policy – (unlike in EPAL where the evaluation may terminate mid-policy on the first match (Barth, Mitchell 2004). Thereby it enables both local reasoning and combination of policies by concatenating separate policies. Every EPAL policy can be translated into a DPAL policy.

Privacy Rights Markup Language (PRML) is an XML-based language that allows for the definition of objects and a mechanism for linking these objects together to form privacy declarations (Zero-Knowledge Systems 2001). Objects can be roles, operations, data groups, subjects, purposes, constraints, actions and transformations. A PRML declaration specifies that a role can do an operation on a data group belonging to a subject for a purpose if (optionally) certain constraints are satisfied. A collection of PRML declarations form the privacy policy. On top of usual access control features, PRML can specify that an action should take place immediately after a defined event or before another operation can occur.

Zero-Knowledge Systems took IBM to court because it claimed that EPAL uses mechanisms from PRML and the jointly developed Enterprise Privacy Markup Language (EPML).

Platform for Enterprise Privacy Practices (E-P3P) defines the enterprise privacy enforcement system for enterprise-internal privacy policies (Ashley *et al.* 2002; Karjoth, Schunter, Waidner 2002). When personal data are processed, E-P3P can be used to ensure that data flows and usage practices of an enterprise comply with the privacy statement of the organisation. E-P3P is the predecessor of EPAL.

Enterprise Privacy Authorization Language (EPAL) is a formal language for expressing enterprise privacy policies to govern data handling practices in IT systems according to fine-grained positive and negative authorisation rights (Powers, Schunter 2003). An EPAL policy defines lists of hierarchies of data-categories, user-categories, and purposes, and sets of (privacy) actions, obligations, and conditions. With these elements, privacy authorisation rules are defined that allow or deny actions on data-categories by user-categories for certain purposes under certain conditions while mandating certain obligations. EPAL allows for general rules and exceptions.

EPAL was developed by IBM. Version 1.2 was submitted as a W3C standard in 2003, but stalled because of the lawsuit with Zero-Knowledge Systems concerning intellectual property regarding EPAL.

3.3.4 Web privacy policy languages

Web privacy policy languages represent human-readable privacy policies on the web server's side as well as the user's privacy preferences in machine-readable formats.

The most prominent example in this category is definitely P3P - on the user's side by now not many preference languages are used in practice. The examples listed in Table 19 are briefly explained:

The **Platform for Privacy Preferences Project (P3P)** enables web sites to express their privacy practices in a standard format that can be retrieved automatically and interpreted easily by user agents (Cranor 2002; Wenning, Schunter 2006). The user agents can show the content of the privacy policy in the desired natural language by interpreting the machine-readable form, i.e. in XML format. They can react automatically, e.g., by displaying warnings if the policy does not match the user's preferences. P3P 1.0 became a W3C recommendation in 2002. Today the P3P 1.1 Specification is published as a Working Group Note to give P3P 1.1 a provisionally final state. The P3P web site lists current implementations of P3P user agents, proxies, server-side support tools as well as policy generators, editors and checkers.[23] Among them are current browsers (Internet Explorer or Netscape) which at least partially support the interpretation of P3P policies.

Parallel to P3P 1.0, a language for expressing user's preferences was developed: **A P3P Preference Exchange Language (APPEL)**. The APPEL preference rules can be used by the user agent to decide (semi-)automatically regarding the acceptability of P3P policies (Langheinrich 2002; Cranor 2002).

XPath-based Preference Language (XPref) was proposed to overcome shortcomings of APPEL, in particular difficulties and ambiguities when writing APPEL preferences (Agrawal *et al.* 2003). Based on XML Path Language (XPath), XPref is as expressive as APPEL, but easier to handle.

[23] http://www.w3.org/P3P/implementations.html.

3.3.5 Context sensitive languages

Context sensitive languages are in particular developed for and on top of semantic web technologies. They represent policies that take into consideration context information, e.g., for offering personalised services.

As the proposed languages are still quite young, it is not clear which ones to highlight as the most prominent examples in this category. An overview on many of these languages is given by multiple publications and tutorials from Olmedilla (Olmedilla 2007; Bonatti, Oldemilla 2007). The examples listed in Table 19 are briefly explained:

Geographic Location/Privacy (Geopriv) is an authorisation policy language for controlling access to location information (Schulzrinne, Tschofenig 2007). The privacy-relevant properties of this policy language are the condition elements specific to location information in order to restrict access based on the current location of the target as well as location-specific transformation elements to reduce the granularity of the returned location information. Geopriv is proposed as Internet-Draft, i.e., as a working document of the Internet Engineering Task Force (IETF), and work in progress.

KAoS provides a framework for specification, management, conflict resolution and enforcement of policies (Uszok *et al.* 2003, Olmedilla 2007). This framework also offers distributed policy interaction and support for dynamic policy changes. System administrators can profit from KAoS Administration Tool when writing their policies. A KAoS policy contains authorisations, constraints and obligations. It is bases on the Web Ontology Language (OWL).

The **PeerTrust** language for expressing access control policies is based on guarded distributed logic programs (Gavriloaie *et al.* 2004). Version 1.0 is based on Java/Prolog and offers trust negotiation capabilities for servers and clients, with facilities to import and reason about access control policies, digital credentials, and metadata about local resources requiring protection. In contrast to a few other approaches, the authors of Peer Trust assume that the access control rules should be protected against unauthorised access too. PeerTrust is developed and maintained by members of the WG on Policy Specification, Composition and Conformance of REWERSE[24] - there are plans for a version 2.0.

Ponder is a language for specifying management and security policies for distributed systems (Damianou *et al.* 2001). Similar to KAoS, it deals with authorisation and obligation policies, but in addition it supports refrain and delegation policies. A key concept is the expression of organisational structures, e.g., relationship between different roles, in policies. Ponder can also be used for security management activities such as registration of users or logging and auditing events for dealing with access to critical resources or security violations. Although Ponder is aimed at distributed systems, it is not explicitly designed for semantic web technologies.

[24] http://rewerse.net/I2/.

Protune (Provisional trust negotiation) is the trust negotiation framework of REWERSE[25] (Bonatti, Olmedilla 2005). Its policy language is rule-based and supports sophisticated access control policies as well as credential release policies. Protune provides an automated trust negotiation mechanism among the involved peers. During negotiations, peers exchange credentials, declarations, and policy rules (including privacy preferences). Protune's policy rules can be filtered, i.e., "sanitised", before releasing it because they can contain sensitive data. There is a prototype implemented on top of PeerTrust. A more robust system is planned to be made publicly available on the Open Source software development web site Sourceforge[26].

Rei is a logic-based policy language which is named after the Japanese word for "universal" or "essence", to indicate the universal applicability of the policy language (Kagal 2002). It is based on domain-specific ontologies expressed in OWL and provides, on top of access control, features like constraints and obligations also logic-like variables which offer higher flexibility. Further it includes mechanisms for conflict resolution and supports remote policy management.

3.3.6 Discussion

The list of privacy policy languages given in the previous sections can neither be exhaustive nor can the specifications be explained in depth. However, the brief descriptions show the variety in this area and demonstrate which research groups from different disciplines within computer science work on that topic.

Further categorisation and comparison work has to be done to identify which policies and protocols best suit in different application contexts, legal environments or specific contexts.

A methodology for categorisation, partially related to the one shown previously in Table 19, was proposed by Madsen *et al.* (Madsen, Casassa Mont, Wilton 2006). They classify privacy policies according to the scheme illustrated in Table 20).

No.	Short name	Description	Example
P0	policy preference	"If I provide my credit card number I expect ..."	APPEL, XPref
P1	policy promise	"I will use your credit card number for ..."	P3P
P2	governs acceptance	"Don't accept credit card number unless …"	XACML
P3	governs internal use and/or release	"Only share credit card number when ..."	XACML, EPAL
P4	governs subsequent	"If you receive the credit card number, you must	XACML,

[25] REWERSE is a research Network of Excellence on "Reasoning on the Web" that is funded by the EU Commission and Switzerland within FP6. Web site: http://rewerse.net/.

[26] http://sourceforge.net/.

	use and/or release	delete after 3 days …"	ODRL

Table 20: Exemplary categorisation of privacy policies (based on Madsen *et al.* 2006)

Privacy relevant criteria for data handling policies are given by Ardagna *et al.* (Ardagna, De Capitani di Vimercati, Samarati 2006):

- "*Individual control.* Users should be able to specify who can see what information about them and when.
- *Consent.* Users should be able to give their explicit consent on how their personal data can be used.
- *Correction.* Users should be able to access their personal information to modify it when needed.
- *Security.* Adequate security mechanisms have to be applied, according to the sensitivity of the data collected."

Casassa Mont lists further requirements, concentrating on handling of privacy obligations (Casassa Mont, 2006).

Further criteria for semantic web policies are elaborated by Bonatti, Oldemilla and others (Bonatti *et al.* 2006; Bonatti, Oldemilla 2007).

The World Wide Web Consortium continues its work in the privacy policy area and has recently established an Interest Group on Policy Languages as part of the Privacy Activity (W3C 2007). This group is called PLING - Policy Languages Interest Group. Its objective is the discussion and coordination of policy languages and W3C's metadata framework. Quoted from the charter of this Interest Group (W3C 2007):

"The group will primarily focus on policy languages that are already specified and broadly address the privacy, access control, and obligation management areas; it is not expected to engage in the design of new policy or rule languages. The Interest Group will work towards identifying obstacles to a joint deployment of such languages, and suggest requirements and technological enablers that may help overcome such obstacles."

The described work shows that during the following years there will be further development in the area of privacy policies and their protocols. In addition there will be investments on interoperability of different policy languages and frameworks.

3.4 Conclusion

In this chapter three main areas of protocols for privacy-aware communication have been described:

1. Firstly anonymisation services have been tackled, in particular technologies for mixing IP packets and substituting IP addresses. Although theoretical models work quite well, today's practical implementations do not provide appropriate safeguards

against clever and powerful attackers. In addition, currently anonymisation services are not widely distributed; their usage often is inconvenient and increases undesired latency. The protocols and mechanisms of anonymisation services can be used on top of the protocols described in Chapter 2.

2. Secondly, user-centric identity management protocols have been introduced, especially concerning anonymous credentials and protocols for web services including single sign-on. This section shows several components which could be used in federated identity management to improve privacy and security. However, a high adoption in particular regarding anonymous credentials is crucial for a privacy assessment of ICT systems, otherwise the data disclosed from other protocol layers plus the additional information in the identity management area can be linked and yield detailed profiles on individuals.
3. Thirdly, the domain of privacy policies is a vast area which could only briefly be touched on. Protocols for negotiating policies and enforcing them will play a prominent role in the next years. As data minimisation is not an option in many practical cases, policies and policy enforcement have to step in. From today's perspective it is not clear which languages and protocols will prevail in which areas.

Summarising, this chapter shows work in progress and points to some research questions which will, among other things, affect specification and standardisation of protocols.

4 Next Generation Internet protocols

This section describes developments in the area of Internet protocols, so called "Next Generation Internet Protocols", NGIP. The Internet nowadays relies on protocols which for the most part have been developed some decades ago. The very important TCP protocol was standardised in the 1970s. The 1980s brought the IP, DNS and SMTP protocol specifications. POP3 dates back to the mid 1990s, similarly with the first HTTP description. Protocols like WLAN or IPv6 are of much younger age, but these protocols on the lower layers are still used by the older protocols on higher layers mentioned before.

The older protocols have especially been created within a totally different context than is existing today. When TCP was created, for example, Quality of Service was not an important topic, and SMTP was built with a small and rather closed community of users in mind. Authentication was not crucial back then. This "weak design" is one of the main reasons SPAM could evolve in the way it did. Another example is content delivery of the WWW. Proxies, load-balancing, content-transformation servers and other techniques aiming at delivering content from a virtual host to the clients require sophisticated approaches which are often proprietary and thus violate the original Internet design, architecture and philosophy (Cheriton, Gritter 2000).

It is important to design new protocols in order to straighten out protocol flaws created tens of years ago. Furthermore, since usage patterns have changed rapidly in the last years, new protocols must support new user demands so that novel applications can be developed and deployed, as well as allowing current applications like streaming media to be utilised without bringing the WWW we know to its knees. And last but not least, a new Internet architecture is important in order to provide more security necessary for governance, commerce and user-privacy.

Section 4.5 hypothesises that privacy experts usually do not participate in the design of protocols, although this may be necessary to conduct a proper privacy impact analysis beforehand.

The following subsections introduce main approaches in the area of Next Generation Internet protocols, i.e., Internet2, GÉANT2, TRIAD and FIND.

4.1 Internet2

The Internet2 is a consortium of universities and companies developing an infrastructure which is much more powerful than the current Internet. The Internet2 group created a very fast backbone, called "Abilene". It is a fibre optic network with a bandwidth of up to 100 Gbit per second. The Internet2 group started in 1997 by the "University Corporation for Advanced Internet Development (UCAID)". The Internet2 consortium is independent of the US military, which is noteworthy, since the original Internet was developed by the US military. Right now more then 200 US universities are connected to this high-speed infrastructure, and more than 60 companies are involved in its creation.

European countries and institutions are connected to the Abilene network, too. In Germany for instance, the DFN ("Deutsches Foschungsnetz") operates the X-WiN network which is connected to the Abilene infrastructure.

The Abilene network is merely a new physical infrastructure; it can handle any sort of Internet protocols. And in fact the Abilene network as it is deployed today uses the same old protocols as the rest of the Internet does. For example, addressing is done by using the IPv6 protocol. But there are also some new protocols developed for this high-speed network. For example a new bulk file transfer protocol is being developed by some researchers in order to speed up the transfer of often huge amounts of data needed by many research projects and applications (Shalunov *et al.* 2005).

The Internet2 consortium claims that it takes security seriously. For example, instead of the normal DNS, the Internet2 network uses DNSSEC for a more secure naming service. But the members acknowledge that there is still much to do in order to create a secure network. They try to implement security in all stack levels, and this is of course complex since many working groups are involved in such an effort. Authentication and authorisation are especially big topics, as well as aggravating Denial-of-Service attacks and other attacks against users in the network. Many security features are implemented on the so called "Internet2 Middleware", which is something like the "glue" between the network and the application. The middleware software "provides services such as identification, authentication, authorisation, directories, and security."[27] At a lower level, the networking layer, Abilene has the ability to filter traffic in case of an attack on the network itself or on its peers.

An interesting project within the Internet2 research is "Shibboleth". Shibboleth is a middleware project for federate identity-based authentication and authorisation system. It can be used for a single-sign-on system, meaning that a user has to identify himself once against a so called "Identity Provider", which then provides credentials after a successful login to so called "Service Providers". A Service Provider can decide by the credentials which services to open to the user. [28]

4.2 GÉANT2

GÉANT can be seen as the European answer to Internet2. GÉANT is the 7th generation of a pan-European research network infrastructure, meaning that GÉANT is, like Abilene, right now a network infrastructure for its participating partners. Participants are European universities, 30 of Europe's national research and education networks, 3500 research groups in 34 European countries. But GÉANT also reaches out of Europe, there are many connections to other, outer-European countries, like to the Abilene network in the US. The project, which currently has funding until August 2008, is getting most of the needed research money from the European Commission and the national research and education networks.

The objectives of GÉANT2 are to (Karapandzic 2007):

- Provide a gigabit-speed infrastructure to support European research and education – extend geographic reach of the network;
- Deploy the first international hybrid network: routed IP traffic combined with switched point-to-point circuits;

[27] http://middleware.internet2.edu/.

[28] http://shibboleth.internet2.edu/.

- Implement end-to-end QoS provision;
- Develop a wider range of network services:
  - Performance monitoring;
  - Security;
  - Bandwidth on demand;
  - Test-bed facility;
  - Mobility and roaming;
  - Provide user support and consultancy.

Right now the GÉANT initiative is putting together its request for further funding for the GÉANT3 project which will continue the work done by the former seven network generations.

4.3 TRIAD

The project TRIAD aims primarily at the problem of content delivery. The authors of TRIAD claim that many problems of the current Internet come from the fact that because of private networks (see the problems with NAT) and proprietary content, delivery techniques like load-balancing one of the most important properties of the Internet, i.e., point-to-point connections between hosts, is not given any more. In order to enable these point-to-point connections again in both directions, "TRIAD defines an explicit content layer that provides scalable content routing, caching, content transformation and load balancing, integration naming routing and transport connection setup." (Cheriton, Gritter 2000). The authors claim that TRIAD "provides attractive solutions to mobility, virtual private networking, policy-based routing and source spoofing". TRIAD is an overlay to the current Internet, it relies upon IPv4 (the authors claim that IPv6 is not needed with their solution since they support NAT), TCP and DNS. But TRIAD still requires some changes to current protocols like TCP, thus complicating a transition to this new overlay network in large since the Internet protocol stack has to be changed. However, the TRIAD project seems to have stopped its work, there have not been any publications since August 2004.

4.4 Future Internet Network Design

The "Future Internet Network Design" (FIND) is an initiative from the "National Science Foundation" NSF, United States. The FIND programme is a long-term research project investigating two main questions (FIND 2005):

1. What are the requirements for the global network of 15 years from now – what should that network look like and do?

2. How would we re-conceive tomorrow's global network today, if we could design it from scratch?

The main research targets are security and robustness, management capabilities, integration of new elements like sensors, embedded systems etc., identity management, protocols, etc.

4.4.1 Global Environment for Networking Innovations

Global Environment for Networking Innovations (GENI) is another project of the National Science Foundation NSF. GENI, when finished, can be used as a test-bed for problems like communications, networking, distributed systems, cyber-security and networking services. It can be usefully used to test new protocols and networking applications for the next generation Internet. "The emphasis is on enabling researchers to experiment with radical network designs in a way that is far more realistic than they can today." (GENI 2007).

Researchers can test their network implementations without being bound to the current Internet infrastructure. The use of GENI would not be restricted to the academic research world, but also would offer a test-bed for industrial experiments. However, it has to be shown if GENI can really provide researchers with a test-bed which is capable of implementing arbitrary networks.

4.4.2 Internet Research Task Force

The Internet Research Task Force (IRTF) must not be confused with the Internet Engineering Task Force (IETF): While the IETF's purpose is solving rather urgent problems of the current Internet, the IRTF's primary task is to "[...] promote research of importance to the evolution of the future Internet by creating focused, long-term and small Research Groups working on topics related to Internet protocols, applications, architecture and technology." The main sponsors of the IRTF are the IETF and "The Internet Society" ISOC.

4.5 Designing protocols with or without privacy experts

Protocols are the foundation of information and communication technologies (ICT). According to Lessig, protocols belong to the major regulators which have a profound impact on society and whose implications must be considered (Lessig 1999). This applies for all implementations of protocols, forming the architecture of ICT and providing today's possibilities for usage. In addition, the specifications of protocols already play a role as they are the blueprint not only for implementations thereof, but define interfaces to other specifications and implementations.

If protocols, i.e., their specifications and/or their implementations, are faulty, the applications on top usually cannot eliminate the mistakes, but often even intensify the consequences. As previously argued, linkability between different uses of protocols – possible by cross-layer analysis, possibly even by hidden identifiers – may already cause privacy threats which can yield undesired consequences for the individuals concerned (cf. Hansen, Meissner 2007). Considering the complexity of the area and the massive influence of protocols on the Information Society, a privacy and linkability analysis should be performed during the design phase of each protocol. Art. 20 of the Directive 95/46/EC deals with "prior checking" which

should be carried out when the processing operations are "likely to present specific risks to the rights and freedoms of data subjects" (EU-DPD 1995). In particular outside the European Union, e.g., in Canada, the United States, Australia and New Zealand, a similar procedure is also known as "Privacy Impact Assessment". Taking this seriously, privacy experts would have to be involved right from the beginning in each design process of communication protocol specifications.

The general participation of Data Protection Authorities (DPAs) and other trusted parties in the technology design process for better trust and trustworthiness was proposed by Köhntopp and Ruhmann who, as employees of a DPA, had first-hand experience determining factors regarding possible privacy impact assessment (Köhntopp, Ruhmann 1999). However, a limiting factor is the lack of resources of DPAs concerning budget for personnel, for travelling and participating in meetings where protocols are being specified as well as lacking skilled staff in all areas of research and development of protocols.

There are various reasons for not seeking participation of privacy experts when designing protocols:

- Specification work on protocols is usually application-driven where primarily functionality requirements are the focus. Privacy requirements are often regarded as a secondary objective which comes into play at a later stage.
- Naïve implementations – and also specifications – are often privacy-invasive, e.g., centralised storage of data, unique identifiers, and lots of debug information according to what engineers and computer scientists learn in their training. In many cases privacy-enhancing components add complexity in the design process.
- The work is mainly performed by technologists and implementers instead of legally skilled people who are aware of national or – with international protocols – even global legal regulations.
- The language of specifications is not always easy to understand for laymen, lawyers or also people skilled in technology assessment. This is a severe obstacle to their involvement.
- The complexity of today's ICT world with all its interdependencies is hard to cope with for everybody. Thus, a thorough analysis of protocols in an early stage may not cover all privacy-relevant issues arising from the interplay with other existing or upcoming protocols and applications.

Indeed during the last decades very few DPAs were involved when protocols were specified, and those involved usually participated only in the design of specific protocols and languages focusing on privacy and data protection (such as P3P or EPAL, as introduced in Section 3.3). However, all kinds of protocols have been discussed and criticised in the privacy community, e.g., because of shortcomings concerning important privacy concepts such as data minimisation, transparency or the user's self-determination, but usually after the finalisation of the specification process and without a serious attempt to modify the specification or even stop the implementation or use of a protocol.

4.6 Conclusion

It is too early to conclude much on Next Generation Internet protocols as such – it will take some time before more practical experiences can be collected. In any case, the development of the next generation of the Internet opens an opportunity to smooth away past mistakes in specifications and implementations. Still, the integration of all relevant stakeholders including privacy experts, as demanded in Section 4.5, will be missed again. Or, to phrase is differently, at most very few privacy experts - if any - will participate in this important area of specification and standardisation. The last section of this chapter has dealt with the design process of protocols and the immanent lack of participation of privacy experts, yielding in specifications which often do not fulfil privacy requirements such as data minimisation or transparency.

5 Summary and conclusions

The analysis of the protocols in this deliverable shows that virtually any commonly used protocol reveals identifying and linkable information usable for profiling. Some of them even disclose personal data. Avoiding or circumventing these threats for privacy and data protection cannot be done easily – especially considering the fact that the established architecture and protocol infrastructure of today's communication networks exists with all its shortcomings. Even the Next Generation Internet protocols will not be designed in a privacy-enhancing way; however, there is progress regarding ICT security features.

Anonymisation services or other data minimisation techniques on the lower protocol layers can be used to blur some of the traces one leaves while using the Internet. However, they neither offer a convincing level of protection nor have they achieved a level of stability and quality of service necessary for every day use by the masses. Nevertheless they are suitable tools at least for some use cases. An easy to implement measure (from a technological point of view) would be to use link encryption of every single data link. This would greatly enhance privacy against outsiders – e.g., eavesdroppers on the lines – who would neither learn the communications' content nor (most of) their circumstances.

On other layers, identity management functions can be supported by appropriate protocols, especially handling of anonymous credentials which combine accountability and privacy requirements. In addition protocols and languages for expressing, matching, negotiating and enforcing privacy policies are demanded where personal data are exchanged. After one decade of research and development in this area there is a varied selection of proposed privacy policy languages. It is not likely that this variety will be narrowed down to one or very few languages in the next years, so that in the coming years work will be done to find solutions for interoperable use of these languages.

This deliverable could not tackle cross-layer effects on privacy and data protection, e.g., basing on linkability of information from various protocol layers. Today this is an open research issue. The same is valid for cross-layer privacy-enhancing technologies which have to be further investigated. Most research and development projects and university work is limited to one or few protocol layers each, blinding out the impact of linking data from arbitrary layers. Also the composition of different privacy-enhancing technology tools is barely investigated now.

A major challenge is not only the understanding of today's protocol world, but also the design and specification of new protocols. In particular in those areas where right now standardisation work is being performed it would be advisable to integrate experts from the fields of identity and privacy in the processes. Naïve specifications and implementations of global standards will usually cement not so privacy-friendly information and communication technologies. Even if privacy-invasive requirements such as demanded data retention are an obstacle to pure privacy-enhancing design of protocols, data protection functionality could be massively improved. In addition, the impact of these protocols, their interdependencies and the whole specification process have to be made more transparent to decision makers and citizens because protocols are the backbone of our Information Society.

6 References

Abelson, H., Anderson, R., Bellovin, S.M., Benaloh, J., Blaze, M., Diffie, W., Gilmore, J., Neumann, P.G., Rivest, R.L., Schiller, J.I., Schneier, B., 'The risks of key recovery, key escrow, and trusted third-party encryption', *World Wide Web Journal*, (3), 1997, pp. 241-257.

Agrawal, R., Kiernan, J., Srikant, R., Xu, Y., 'An XPath-based Preference Language for P3P', *Proceedings of the 12th International Conference on World Wide Web (WWW '03)*, New York, NY, USA, 2003, pp. 629-639.

Anderson, A., *A Comparison of Two Privacy Policy Languages: EPAL and XACML*, September 2005, Sun Labs Technical Report TR-2005-147, http://research.sun.com/techrep/2005/smli_tr-2005-147.pdf (current September 2007).

Anderson, A., *XACML-based Web Services Policy Constraint Language (WS-PolicyConstraints)*, Sun Microsystems, Inc., specification, whitepapers, slides, proof-of-concept, 2005-2006, http://research.sun.com/projects/xacml (current September 2007).

Anderson, A., *Sun Position Paper: W3C Workshop on Languages for Privacy Policy Negotiation and Semantics-Driven Enforcement*, 2006, http://www.w3.org/2006/07/privacy-ws/papers/17-anderson-position/ (current September 2007).

Andersson, C., Camenisch, J., Crane, S., Fischer-Hübner, S., Leenes, R., Pearson, S., Pettersson, J.S., Sommer, D., 'Trust in PRIME', *Proceedings of the Fifth IEEE International Symposium on Signal Processing and Information Technology*, 2005, pp. 552-559.

Ardagna, C.A., De Capitani di Vimercati, S., Samarati, P., 'Enhancing user privacy through data handling policies', *Proceedings of IFIP WG 11.3 Working Conference on Data and Applications Security*, LNCS 4127, Springer, Berlin 2006, pp. 224-236, http://seclab.dti.unimi.it/Papers/samarati-ifip06.pdf (current September 2007).

Ashley, P., Hada, S., Karjoth, G., Schunter, M., 'E-P3P Privacy Policies and Privacy Authorization', *Proceedings of the 2002 ACM Workshop on Privacy in the Electronic Society (WPES '02)*, New York, NY, USA, 2002, pp. 103-109.

Aura, T., Zugenmaier, A., 'Privacy, Control and Internet Mobility', *Security Protocols Workshop 2004*, 2004, pp. 133-145.

Ballinger, K. *et al.*, *Web Services Metadata Exchange (WS-MetadataExchange) v1.1*, August 2006, http://www-128.ibm.com/developerworks/library/specification/ws-mex/ (current September 2007).

Barth, A., Mitchell, J.C., 'Conflict and Combination in Privacy Policy Languages', *Proceedings of the 2004 ACM Workshop on Privacy in the Electronic Society (WPES '04)*, 2004.

Berthold, O., Federrath, H., Köpsell, S., 'Web MIXes: A System for Anonymous and Unobservable Internet Access', *Workshop on Design Issues in Anonymity and Unobservability 2000*, LNCS 2009, Springer-Verlag, Berlin 2001, pp. 115-129.

Bissias, G.D., Liberatore, M., Jensen, D., Levine, B.N., 'Privacy Vulnerabilities in Encrypted HTTP Streams', *Proceedings of Privacy Enhancing Technologies Workshop (PET 2005)*, 2005, pp. 1-11.

Böhme, R., Danezis, G., Díaz, C., Köpsell, S., Pfitzmann, A., 'On the PET Workshop Panel Mix Cascades Versus Peer-to-Peer: Is One Concept Superior?', *Proceedings of Privacy Enhancing Technologies workshop (PET 2004)*, LNCS 3424, Springer, Berlin 2004.

Bonatti, P.A., Duma, C., Fuchs, N.E., Nejdl, W., Olmedilla, D., Peer, J., Shahmehri, N., 'Semantic Web Policies – A Discussion of Requirements and Research Issues', *Proceedings of 3rd European Semantic Web Conference (ESWC 2006)*, LNCS 4011, Springer, Berlin 2006, pp. 712-724, REWERSE Technical Report REWERSE-RP-2006-020, http://rewerse.net/publications/download/REWERSE-RP-2006-020.pdf (current September 2007).

Bonatti, P.A., Daniel Olmedilla, D., 'Driving and monitoring provisional trust negotiation with metapolicies', *Proceedings of 6th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2005)*, IEEE Computer Society, 2005, pp. 14-23.

Bonatti, P.A., Olmedilla, D., Rule Based Policy Representation & Reasoning for the Semantic Web, Presentation at "Reasoning Web 2007 – Summer School", Dresden, 5 September 2007, slides available at http://www.l3s.de/%7Eolmedilla/presentations/2007/20070905_REWERSE_SS.ppt (current September 2007).

Bohrer, K., Holland, B., *Customer Profile Exchange (CPExchange) Specification*, Version 1.0, Technical Report, October 2000.

Brands, S., *Identity corner: Analysis of "pseudonyms" in SAML 2.0 & Liberty Alliance*, February 2005, http://www.idcorner.org/?p=40 (current September 2007).

Camenisch, J., Lysyanskaya, A., 'An efficient system for non-transferable anonymous credentials with optional anonymity revocation', in Pfitzmann, B. (Ed.), *EUROCRYPT*, LNCS 2045, Springer, 2001, pp. 93-118.

Camenisch, J., Gross, T., and Sommer, D., 'Enhancing Privacy of Federated Identity Management Protocols -- Anonymous Credentials in WS-Security', WPES 2006.

Camenisch, J., Kohlweiss, M., Preneel, B., and Sommer, D., 'Assertion-based Signatures for XML Signatures', COSIC internal report, 21 pages, 2007.

Cantor, S., Kemp, J., Philpott, R., Maler, E. (Eds.), *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0*, OASIS Standard, 15 March 2005, http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf (current September 2007).

Casassa Mont, M., *On the Need to Explicitly Manage Privacy Obligation Policies as Part of Good Data Handling Practices*, Position Paper, 2006, http://www.w3.org/2006/07/privacy-ws/papers/03-casassa-mont-obligations/ (current September 2007).

Chaum, D., 'Security without identification: transaction systems to make big brother obsolete', *Communications of the ACM*, 28(10), 1985, pp. 1030-1044.

Chaum, D., 'Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms', *Communications of the ACM*, 24(2), 1981, pp. 84-88.

Cheriton, D.R., Gritter, M., *TRIAD: A new next generation Internet architecture*, 2000, http://www.dsg.stanford.edu/triad/triad.ps.gz (current September 2007).

Cranor, L.F., *Web Privacy with P3P*, Sebastopol, CA, USA, 2002.

Cvrček, D., Matyáš, V. (Eds.), *D13.1: Identity and impact of privacy enhancing technologies*, FIDIS Deliverable, Frankfurt a.M., Germany, 2007.

Damianou, N., Dulay, N., Lupu, E., Sloman, M., 'The Ponder Policy Specification Language', *Proceedings of the International Workshop on Policies for Distributed Systems and Networks*, LNCS 1995, Springer, London 2001, pp. 18-38.

Danezis, G., Dingledine, R., Mathewson, N., Mixminion: 'Design of a Type III Anonymous Remailer Protocol', *IEEE Symposium on Security and Privacy 2003*, IEEE Computer Society Press, 2003, pp. 2-15.

Dingledine, R., Mathewson, N., Syverson, P.F., 'Tor: The Second-Generation Onion Router', *Proceedings of the 13th USENIX Security Symposium*, USENIX Association, 2004, pp. 303-320.

Dournaee, B., *XML Security*, McGraw-Hill Osborne Media, February 2002.

Eastlake, D., Reagle, J. (Eds.), *XML Encryption Syntax and Processing*, W3C Recommendation 10 December 2002, http://www.w3.org/TR/xmlenc-core/ (current September 2007).

Eastlake, D., Reagle, J., Solo, D. (Eds.), *XML-Signature Syntax and Processing* – RFC 3275, W3C Recommendation 12 February 2002, http://www.w3.org/TR/xmldsig-core/ (current September 2007).

Ellis, K.J., Serinken, N., 'Characteristics of radio transmitter fingerprints', *Radio Science*, Vol. .36, No. 4, 2001, pp. 585-598.

Ellison, C., Schneier, B., 'Ten risks of PKI: What you're not being told about public-key infrastructure', *Computer Security Journal*, 16(1), 2000, pp. 1-7.

Escudero Pascual, A., 'Requirements for unobservability of privacy extension in IPv6', *Radio Vetenskap 2002*, Stockholm, Sweden, 2002, pp. 58.

EU-DPD: The European Parliament and the Council of the European Union – 'Directive 95/46/EC: On the protection of individuals with regard to the processing of personal data and on the free movement of such data', *Official Journal of the European Communities*, (L. 281):31-39, November 1995.

Fellows, D., Jones, D., 'DOCSIS Cable Modem Technology', *IEEE Communications*, Vol. 39, Issue 3, March 2001, pp. 202-209.

FIND – *Future Internet Network Design*, National Science Foundation, 2005, http://www.nets-find.net/ (current September 2007).

Fischer-Hübner, S., Hedbom, H. (Eds.), *D7.3: A holistic privacy framework for RFID*, FIDIS Deliverable, Frankfurt a.M., Germany, 2007.

Gavriloaie, R., Nejdl, W., Olmedilla, D., Seamons, K.E., Winslett, M., 'No registration needed: How to use declarative policies and negotiation to access sensitive resources on the semantic web', *Proceedings of 1st European Semantic Web Symposium (ESWS 2004)*, LNCS 3053, Springer 2004, pp. 342-356.

Gedik, B., Liu, L., *A Customizable k-Anonymity Model for Protecting Location Privacy*, Technical Report of Georgia Institute of Technology GIT-CERCS-04-15, April 2004.

GENI – *Global Environment for Networking Innovations*, National Science Foundation, 2007, http://www.geni.net/ (current September 2007).

Gruteser, M., Grunwald, D., 'Enhancing location privacy in wireless LAN through disposable interface identifiers: a quantitative analysis', *Mobile Networks and Applications*, Volume 10, Issue 3, June 2005, pp. 315-325.

Guha, S., Francis, P., 'Identity Trail: Covert Surveillance Using DNS', *Proceedings of the Privacy Enhancing Technologies Symposium (PET '07)*, 2007.

Hada, S., Kudo, M., *XML Access Control Language: Provisional Authorization for XML Documents*, Technical Report, October 2000, http://www.trl.ibm.com/projects/xml/xacl/xacl-spec.html (current September 2007).

Hager, C.T., Midkiff, S.F., 'An Analysis of Bluetooth Security Vulnerabilities', *Wireless Communications and Networking*, Vol. 3, 2003, pp. 1825-1831.

Hall, J., Barbeau, M., Kranakis, E., 'Radio Frequency Fingerprinting for Intrusion Detection in Wireless Networks', *IEEE Transactions on Dependable and Secure Computing*, Manuscript received July, 2005, http://www.scs.carleton.ca/~jhall2/Publications/IEEETDSC.pdf (current September 2007).

Hansen, M., Meissner, S. (Eds.), *Verkettung digitaler Identitäten*, report commissioned by the Federal Ministry of Education and Research, Germany, 2007, in German, Executive Summary available in English, https://www.datenschutzzentrum.de/projekte/verkettung/2007-uld-tud-verkettung-digitaler-identitaeten-bmbf.pdf (current November 2007).

Hildebrandt, M., Backhouse, J. (Eds.), *D7.2: Descriptive analysis and inventory of profiling practices*, FIDIS Deliverable, Frankfurt a.M., Germany, 2005.

Iannella, R. (Ed.), *Open Digital Rights Language (ODRL) Version 1.1*, W3C Note 19 September 2002, http://www.w3.org/TR/odrl/ (current September 2007).

IBBT, *E-Health Information Platforms*, 2007, https://ehip.ibbt.be/ (current September 2007).

IBM Corporation, Microsoft Corporation, *Federation of Identities in a Web Services World*, Version 1.0, Joint White Paper, July 2003, ftp://www6.software.ibm.com/software/developer/library/ws-fedworld.pdf (current September 2007).

Jakobsson, M., Wetzel, S., 'Security weaknesses in Bluetooth', *Proceedings of the 2001 Conference on Topics in Cryptology: The Cryptographer's Track at RSA*, LNCS 2020, 2001, pp. 176-191.

Jerichow, A., Müller, J., Pfitzmann, A., Pfitzmann, B., Waidner, M., 'Real-Time Mixes: A Bandwidth-Efficient Anonymity Protocol', *IEEE Journal on Selected Areas in Communications*, Special issue "Copyright and privacy protection", 4(1998), pp. 495-509.

Jesdanun, A., *Privacy for Internet names moves forward*, Associated Press, March 2007, http://www.theglobeandmail.com/servlet/story/RTGAM.20070320.wprivacy0320/BNStory/Technology/ (current September 2007).

Jøsang, A., Pope, S., 'User centric identity management', *AusCERT Conference 2005*, 2005.

Kagal, L., *Rei: A Policy Language for the Me-Centric Project*, Technical Report HPL-2002-270, HP Laboratories (2002), http://ebiquity.umbc.edu/_file_directory_/papers/57.pdf (current September 2007).

Karjoth, G., Schunter, M., Waidner, M., 'Platform for Enterprise Privacy Practices: Privacy-enabled Management of Customer Data', *Proceedings of the 2nd Workshop on Privacy Enhancing Technologies*, 2002, pp. 61-83.

Karapandzic, M., *GÉANT2 Newcomers' Introduction*, 3rd GÉANT2 Technical Workshop, January 2007, http://www.geant2.net/upload/pdf/GN2_structure_and_management_-_Newcomers.pdf (current September 2007).

Kohlweiss, M., *Idemix 2003 design*, unpublished, 2002.

Köhntopp, M., Ruhmann, I., 'Trust through participation of trusted parties in technology design', in: Müller, G., Rannenberg, K. (Eds.), *Multilateral Security in Communications – Technology, Infrastructure, Economy*, Addison-Wesley 1999, pp., 499-514.

Kumaraguru, P., Cranor, L., Lobo, J., Calo, S., 'A Survey of Privacy Policy Languages', *SOUPS 2007*, 18-20 July, 2007, Pittsburgh, PA, USA, http://cups.cs.cmu.edu/soups/2007/workshop/Privacy_Policy_Languages.pdf (current September 2007).

Lackner, G., Lamberger, M., Payer, U., Teufl, P., 'WiFi Chipset Fingerprinting', *Proceedings of DACH Mobility 2006*, 17-18 October 2006, Munich, Germany, 2006.

Langheinrich, M. (Ed.), *A P3P Preference Exchange Language 1.0 (APPEL1.0)*, W3C Working Draft 15 April 2002, http://www.w3.org/TR/P3P-preferences/ (current September 2007).

Leenes, R., Schallaböck, J., Hansen, M. (Eds.), *Privacy and Identity Management for Europe – PRIME White Paper V2*, 2007, https://www.prime-project.eu/prime_products/whitepaper/ (current September 2007).

Lessig, L., *Code and other laws of cyberspace*, New York, Basic Books, 1999.

Liberty Alliance, *Specifications*, August 2007, http://www.projectliberty.org/ (current September 2007).

Lysyanskaya, A., Rivest, R.L., Sahai, A., Wolf, S., 'Pseudonym systems', in Heys, H.M., Adams, C.M. (Eds.), *Selected Areas in Cryptography*, LNCS 1758, Springer, 2000, pp. 184-199.

Madsen, P., Casassa Mont, M., Wilton, R., *A Privacy Policy Framework – A position paper for the W3C Workshop of Privacy Policy Negotiation*, 2006, http://www.w3.org/2006/07/privacy-ws/papers/28-madsen-framework/ (current September 2007).

Molva, R., *Internet Security Architecture*, Computer Networks Vol. 31 No. 8, Amsterdam, Netherlands, 1999, pp. 787-804.

Moses, T. (Ed.), *XACML profile for Web-services (WSPL)*, OASIS XACML Technical Committee Working Draft 04, 29 September 2003, http://www.oasis-open.org/committees/download.php/3661/draft-xacml-wspl-04.pdf (current September 2007).

Moses, T. (Ed.), *eXtensible Access Control Markup Language (XACML) Version 2.0*, OASIS Standard, 1 Feb 2005, http://docs.oasis-open.org/xacml/2.0/XACML-2.0-OS-NORMATIVE.zip (current September 2007).

Moses, T., (Ed.), *Privacy policy profile of XACML v2.0*, OASIS Standard, 1 February 2005, http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-privacy_profile-spec-os.pdf (current September 2007).

Nadalin, A., Goodner, M., Gudgin, M., Barbir, A., Granqvist, H. (Eds.), *WS-Trust v1.3*, OASIS Standard, 19 March 2007, http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.pdf (current September 2007).

Nadalin, A., Goodner, M., Gudgin, M., Barbir, A., Granqvist, H. (Eds.), *WS-SecurityPolicy v1.2*, OASIS Standard, 1 July 2007, http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702/ws-securitypolicy-1.2-spec-os.pdf (current September 2007).

Nadalin, A., Kaler, C. (Eds.), *Web Services Federation Language (WS-Federation)*, Version 1.1. http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-fed/WS-Federation-V1-1B.pdf (current September, 2007), December 2006.

Nadalin, A., Kaler, C., Hallam-Baker, P., Monzillo, R. (Eds.), *Web Services Security: SOAP Message Security 1.0*, OASIS Standard, March 2004, http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf (current September 2007).

Olmedilla, D., 'Security and Privacy on the Semantic Web', in: Petkovic, M., Jonker, W. (Eds.), *Security, Privacy and Trust in Modern Data Management, Data-Centric Systems and Applications*, Springer 2007, pp. 399-415, REWERSE Report REWERSE-RP-2007-060, http://rewerse.net/publications/download/REWERSE-RP-2007-060.pdf (current September 2007).

Perkins, C., *RTP Audio and Video for the Internet*, Addison-Wesley, 2003.

Pfitzmann, A., Hansen, M., Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management – A Consolidated Proposal for Terminology, Version 0.29, July 2007, http://dud.inf.tu-dresden.de/Anon_Terminology.shtml (current September 2007).

Powers, C., Schunter, M. (Eds.), Enterprise Privacy Authorization Language (EPAL 1.2), W3C Member Submission 10 November 2003, http://www.w3.org/Submission/2003/SUBM-EPAL-20031110/ (current September 2007).

Ragouzis, N., Hughes, J., Philpott, R., Maler, E. (Eds.), *Security Assertion Markup Language (SAML) V2.0, Technical Overview*, OASIS Working Draft 10, 9 October 2006, http://www.oasis-open.org/committees/download.php/20645/sstc-saml-tech-overview-2%200-draft-10.pdf (current September 2007).

RSA Laboratories, PKCS #7: Cryptographic Message Syntax Standard v1.5, November 1993, ftp://ftp.rsasecurity.com/pub/pkcs/ascii/pkcs-7.asc (current September 2007).

Roessler, T., 'WHOIS: Datenschutz im DNS?', *Datenschutz und Datensicherheit* 26/11, 2002.

Schulzrinne, H., Tschofenig, H. (Eds.), Geolocation Policy: A Document Format for Expressing Privacy Preferences for Location Information, GEOPRIV Internet Draft, May 2007, http://www.ietf.org/internet-drafts/draft-ietf-geopriv-policy-12.txt (current September 2007).

Shalunov, S., *et al.*, *Design Space for a Bulk Transport Tool*, May 2005, http://e2epi.internet2.edu/transport/transport-design-space-10.pdf (current September 2007).

Stewart, R., Amer, P.D., 'Why is SCTP needed given TCP and UDP are widely available?', *ISO Member Briefing #17, Internet Society*, June 2004.

Tanenbaum, A.S., *Computer Networks*, fourth edition, Prentice Hall, Upper Saddle River, NJ, 892 pages, 2003 (first edition 1981).

Tonti, G., Bradshaw, J.M., Jeffers, R., Montanari, R., Suri, N., Uszok, A., 'Semantic web languages for policy representation and reasoning: A comparison of KAoS, Rei, and Ponder', *Proceedings of 2nd International Semantic Web Conference (ISWC)*, LNCS 2870, Springer, 2003, pp. 419-437.

Toonstra, J., Kinsner, W., 'A radio transmitter fingerprinting system ODO-1', *Canadian Conference on Electrical and Computer Engineering* Vol. 1, 1996, pp. 60-63.

Uszok, A., Bradshaw, J.M., Jeffers, R., Suri, N., Hayes, P.J., Breedy, M.R., Bunch, L., Johnson, M., Kulkarni, S., Lott, J., 'KAoS Policy and Domain Services: Toward a Description-Logic Approach to Policy Representation, Deconfliction, and Enforcement', *Proceedings of IEEE 4th International Workshop Policies for Distributed Systems and Networks*, IEEE CS Press, 2003, pp. 93-96.

Weinstein, L, Neumann, P.G., 'Privacy issues and privacy enhancing technologies', in Ito, J. (Ed.), *A Report of Research on Privacy for Electronic Government*. Neoteny Co., Ltd., March 2003, pp. 315-349, http://joi.ito.com/joiwiki/PrivacyReport (current September 2007).

Wenning, R., Schunter, M. (Eds.), *The Platform for Privacy Preferences 1.1 (P3P1.1) Specification*, W3C Working Group Note 13 November 2006, http://www.w3.org/TR/P3P11/ (current September 2007).

Wikipedia contributors, *Bluetooth*, 2007.

Wikipedia contributors, *Cable modem*, 2007.

Wikipedia contributors, *Directory service*, 2007.

Wikipedia contributors, *File Transfer Protocol*, 2007.

Wikipedia contributors, *Hypertext Transfer Protocol*, 2007.

Wikipedia contributors, *Mixmaster-Remailer*, 2007.

Wikipedia contributors, *Network layer*, 2007.

Wikipedia contributors, *OSI model*, 2007.

Wikipedia contributors, *Protocol (computing)*, 2007.

Wikipedia contributors, *Real-time Transport Protocol*, 2007.

Wikipedia contributors, *Simple Mail Transfer Protocol*, 2007.

Wikipedia contributors, *Transmission Control Protocol*, 2007.

Wong, F.-L, Stajano, F., 'Location Privacy in Bluetooth', *Proceedings of 2nd European Workshop on Security and Privacy in Ad hoc and Sensor Networks (ESAS 2005)*, LNCS 3813, Springer, 2005, pp. 176-188.

World Wide Web Consortium (W3C), *PLING - W3C Policy Languages Interest Group, Discussion and Coordination of Policy Languages and W3C's metadata framework* 2007, http://www.w3.org/Policy/pling/ (current November 2007).

Zero-Knowledge Systems, Privacy Rights Markup Language Specification, Version 0.9, June 2001, http://www.synomos.com/html/EPML/documents/prml-spec.pdf (current September 2007).

Appendices

List of Figures

List of Tables

List of Abbreviations

AC	Access Control
ACK	Acknowledgement
ADSL	Asymmetric Digital Subscriber Line
AES	Advanced Encryption Standard
AES-CCMP	AES-Counter Mode CBC-MAC Protocol
AH	Authentication Header
AN.ON	Anonymity Online
APOP	Authenticated Post Office Protocol
APPEL	A P3P Preference Exchange Language
ASCII	American Standard Code for Information Interchange
ATM	Asynchronous Transfer Mode
B channel	Bearer channel
BD_ADDR	Bluetooth Device ADDRess
CBC	Cipher Block Chaining
CCMP	Counter Mode with Cipher Block Chaining Message Authentication Code Protocol
CHAP	Challenge Handshake Authentication Protocol
CL-signature	Camenisch/Lysyanskaya signature
CMTS	Cable Modem Termination System
CNAME	Canonical Name
CPExchange	Customer Profile Exchange
CR	Carriage Return
CRC	Cyclical Redundancy Check
CSMA/CD	Carrier Sense Multiple Access/Collision Detection
D channel	Delta channel
DENIC	Deutsches Network Information Center (German NIC)
DES	Data Encryption Standard
DFN	Deutsches Forschungsnetz (Germany)
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DNSSEC	Domain Name System Security Extensions
DOCSIS	Data Over Cable Service Interface Specifications
DOS	Denial of Service
DPA	Data Protection Authority
DPAL	Declarative Privacy Authorization Language
DPD	Data Protection Directive
DRM	Digital Rights Management
DSL	Digital Subscriber Line
DSLAM	Digital Subscriber Line Access Multiplexer

DSS	Digital Signature Standard
DSSS	Direct Sequence Spread Spectrum
DTLS	Datagram Transport Layer Security
DynDNS	Dynamic Domain Name System
EAP	Extensible Authentication Protocol
EC	European Commission
E-HIP	E-Health Information Platforms
EPAL	Enterprise Privacy Authorization Language
EPML	Enterprise Privacy Markup Language
E-P3P	Platform for Enterprise Privacy Practices
ESP	EncapSulated Payload
EU	European Union
FIND	Future Internet Network Design
FHSS	Frequency Hopping Spread Spectrum
FP	Framework Programme
FTP	File Transfer Protocol
FTPS	File Transfer Protocol Secure
GENI	Global Environment for Networking Innovations
Geopriv	Geographic Location/Privacy
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IBBT	Interdisciplinair instituut voor BreedBand Technologie (Belgium)
IBM	International Business Machines Corporation
ICANN	Internet Corporation for Assigned Names and Numbers
ICMP	Internet Control Message Protocol
ICT	Information and Communication Technologies
ID	Identifier
ID-WSF	Identity Web Service Framework
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IMAP	Internet Message Access Protocol
IP	Internet Protocol
IPTV	Internet Protocol TeleVision
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
IPSec	Internet Protocol Security
IR	InfraRed
IRTF	Internet Research Task Force
ISDN	Integrated Services Digital Network
ISO	International Organization for Standardization

ISOC	Internet Society
ISP	Internet Service Provider
LAN	Local Area Network
LAPD	Link Access Protocol – D channel
LBS	Location Based Services
LCP	Link Control Protocol
LF	Line Feed
LLC	Logical Link Control
MAC	Media Access Control
MB	Megabyte
MBit	MegaBit
MIKEY	Multimedia Internet KEYing
MIME	Multipurpose Internet Mail Extensions
MPEG	Moving Picture Experts Group
MPEG-2	Moving Picture Experts Group Version 2
MPPE	Microsoft Point-to-Point Encryption
NAT	Network Address Translation
NGIP	Next Generation Internet Protocols
NIC	Network Information Center
No.	Number
NoE	Network of Excellence
NSF	Nation Science Foundation (United States)
OASIS	Organization for the Advancement of Structured Information Standards
ODRL	Open Digital Rights Language
OMA	Open Mobile Alliance
OSI	Open System Interconnection
OWL	Web Ontology Language
PAN	Personal Area Network
PAP	Password Authentication Protocol
PDU	Protocol Data Unit
PET	Privacy-Enhancing Technologies
PGP	Pretty Good Privacy
PIN	Personal Idenfitication Number
PKCS	Public Key Cryptography Standards
PKI	Public Key Infrastructure
PLCP	Physical Layer Convergence Protocol
POP	Post Office Protocol
POP3	Post Office Protocol Version 3
POTS	Plain Old Telephone Service
PPP	Point-to-Point Protocol
PPPoE	PPP over Ethernet, Point-to-Point Protocol over Ethernet

PRIME	Privacy and Identity Management for Europe (FP6 Integrated Project)
PRML	Privacy Rights Markup Language
Protune	PROvisional TrUst NEgotiation
PSTN	Public Switched Telephone Network
P3P	Platform for Privacy Preferences
QoS	Quality of Service
RADIUS	Remote Authentication Dial In User Service
RBAC	Role Based Access Control
RC4	Rivest Cipher 4
REWERSE	REasoning on the WEb with Rules and Semantics (FP6 NoE)
RF	Radio Frequency
RFC	Request for Comments
RFID	Radio Frequency Identification
RR	Receiver Report
RTCP	Real-Time Control Protocol
RTP	Real-Time Transport Protocol
RSA	Rivest, Shamir, Adleman
s	second
SA	Security Association
SAML	Security Assertion Markup Language
SCTP	Stream Control Transmission Protocol
SDES	Sender DEScription
SIGTRAN	Signal Transport
SIP	Session Initiation Protocol
SLAAC	StateLess Address AutoConfiguration
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SOAP	Simple Object Access Protocol
SOUPS	Symposium On Usable Privacy and Security
SPI	Security Parameter Index
SSID	Service Set Identifier
SSH	Secure SHell
SSL	Secure Socket Layer
SSO	Single Sign-On
STS	Security Token Services
SYN	SYNchronisation
TC	Technical Committee
UCAID	University Corporation for Advanced Internet Development
UDP	User Datagram Protocol
UK	United Kingdom
URI	Uniform Resource Identifier

URL	Uniform Resource Locator
U.S.	United States (of America)
TCP	Transmission Control Protocol
TKIP	Temporal Key Integrity Protocol
TLS	Transport Layer Security
TTL	Time to live
VoIP	Voice over IP / Voice over Internet Protocol
VPN	Virtual Private Network
WEP	Wired Equivalent Privacy
WiFi	Wireless Fidelity
WLAN	Wireless Local Area Network
WP	Work Package
WPA	Wi-Fi Protected Access
WPA2	Wi-Fi Protected Access Version 2
WS	Web Services
WSDL	Web Services Description Language
WSPL	Web Services Policy Language
WSS	Web Services Security
WWW	World Wide Web
W3C	World Wide Web Consortium
XACL	XML Access Control Language
XACML	eXtensible Access Control Markup Language
XML	eXtensible Markup Language
XMLDSIG	eXtensible Markup Language Digital SIGnature
XPath	XML Path Language
XPref	XPath-based Preference Language
X-WiN	X-WIssenschaftsNetz (Germany)