Framework for secure communication in medical systems

Tariq Mohammad Amin

Kongens Lyngby 2008

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk

IMM-PHD: ISSN 0909-3192

Summary

Telemedicine is the delivery of healthcare services on distance using communication technologies. It is not only a demand of the patients and doctors, but also of the time and development in this IT age. A specific application of telemedicine is the electronic patient journal which tends to replace paper journals with an electronic journal system and gather all information belonging to a patient at one place. Telemedicine in its totality has not yet been established in Denmark, though efforts are being done. Electronic patient journals should have been implemented in all hospitals at the end 2005, which unfortunately has still not been realized. A number of barriers must be overcome before the implementation can fully happen. The barriers are organizational, regulatory, technical and economical.

One of the main points of interest is data security. Both regulatory bodies and data protection authorities have put forward guidelines and requirements to be met for secure data handling and transfer. A secure framework is needed on which the parties involved can agree and base the development.

This thesis deals with different important aspects of data security. It puts forward a framework for secure communication that can be used in medical system. The framework covers three major aspects of communication, namely authentication, authorization and secure transfer. Unlike traditional frameworks, this framework provides dynamic authentication and multi-tiered access control mechanisms.

Resumé

Telemedicin er ydelse af health care over afstand ved hjælp af kommunikationsteknologier. Det er idag ikke blot patienter og lægers behov, men lige så meget tidens og udviklingens i dette IT-samfund. En konkret anvendelse af telemedicin af den elektroniske patientjorunal, som har til formål at udskifte de traditionelle papirbaserede patientjournaler med et elektronisk journalsystem, og samtidig samle alle informationer tilhørende den enkelte patient på ét centralt sted. Telemedicin er i sin helhed endnu ikke blevet etableret noget sted i Danmark, skønt en stor indsats gøres. EPJ skulle være indført på alle hospitaler i Danmark ved udgangen af året 2005, hvilket stadig (ved udgangen af året 2008) ikke er blevet en realitet. En række barrierer skal overvindes førend en implementation kan komme på tale. Barrierne er organisatoriske, regulatoriske, tekniske og økonomiske.

Et af de centraler emner er datasikkerhed. Både regulatoriske og datasikkerhedsinstitutioner har fremlagt retningslinier og krav som må opfyldes for sikker databehandling og -overførsel. Et sikkert framework er påkrævet, som de involverede parter kan blive enige om og basere udviklingen på.

Denne opgave behandler forskellige væsentlige aspekter af datasikkerhed, og fremsætter et framework for sikker kommunikation som kan bruges for medicinske systemer. Dette framework dækker tre af de væsentligste aspekter af kommunikation, nemlig autentifikation, autorisation og sikker overførsel. Modsat traditionelle framework, tilbyder dette framework dynamisk autentifikation og multi-lags adgangskontrolmekanismer.

Preface

This thesis was prepared at Informatics Mathematical Modelling, the Technical University of Denmark in fulfillment of the requirements for acquiring the M.Sc. degree in engineering. The thesis deals with different aspects of communication security setting forward a framework for secure medical communication.

It was written during the period Nov. 2007– Sep. 2008 under the supervision of Associate Professor Christian W. Probst, and submitted in October 2008 at the Technical University of Denmark.

Lyngby, October 2008

Tariq Mohammad Amin

Acknowledgements

I would like to express my sincere gratitude to my supervisor Associate Professor Christian W. Probst for his guidance and support throughout the project. Also thanks to Ph.D Terkel K. Tolstrup for his help to define the project.

I would like to thank the Danish medical company SimeHealth and especially Agnar Höskuldsson, ex-professor at DTU and director of SimeHealth, for their support and co-operation during the project.

I also want to express my thanks to all the parties I have been in contact with during the fieldwork. Here among the following with whom I have held meetings: Jesper Leck, Executive Director at IntraMed A/S, Ivan Brandslund, Laboratory Chief at Vejle Hospital, and his staff, Stephan Engberg, Director of the IT Security company Priway, and the staff at Chemometec A/S, who produce the medical device for SimeHealth. Moreover Ib Johansen og Jens Rahbek Nørgaard from the Danish Health Portal, Sundhed.dk, Ph.D. Jacob Andersen from the University of Aarhus, and Jeppe Spure Nielsen from Alexandra Instituttet A/S have spent a lot of time finding and sending useful information to me, and answering questions.

Special thanks to my boss, Leo Feldborg, administrative director of Feldborg Information, for giving me one year off to write my thesis.

Last but not least, I am deeply thankful to my family and near friends for missing my company for so long time and for their encouragement, which made it possible for me to complete this thesis. Thanks to all those who have willed to read the thesis, for proof-reading or out of curiosity and kindness.

Contents

Chapter 1

Introduction

The Internet has brought a lot of possibilities to the world. Through the Internet, everything from getting the latest news, watching television and making phone calls to buying products and transferring or withdrawing money from the bank is possible. Working from remote locations, such as working at home, is widely used. Now even remote teaching and education programs are available. Simply put, there are no limitations for the possibilities provided by the Internet. However, among all these services, the representation of health services is still very near the invisible.

The health ministries in many counties have become aware of this, and research and development in this field has begun. The field in which communication technologies are used in health care is called Telemedicine. In these days, there is a lot focus is on the development of telemedicine. Among the applications of telemedicine, especially the electronic patient journal (EPJ) has come into the limelight. In many countries EPJ systems are being developed, Denmark being one of them. Though, for a number of reasons implementations are still lacking behind, and the plans are as yet, very far from being realized.

Medical systems are filled with personal and sensitive information, which must be kept secure. The security must be even higher than for e-banking systems, as life is more valuable than money. At least many stakeholders believe so, although bank people would probably not agree. When medical systems come online, a huge number of security threats are to be faced. A medical system, or any system with sensitive information in general, must take all security precautions and do what is needed to prevent attacks.

General systems, such as EPJ, consist of different parts, which must be considered, for example the communication, the data storage, user logins and so on. This reveals the necessity of having a proper framework for secure communication. The main purpose of this thesis is to propose a framework for secure communication that can be used for medical systems, or principally any system in general. The main security aspects for such communication will be analyzed. Besides this, the thesis gives a short insight into the status of telemedicine and EPJ.

The research described in this thesis was performed at DTU in co-operation with the medical firm SimeHealth. The outline of the thesis is given below together with general reading guidelines.

1.1 Thesis outline

Chapter 2 introduces telemedicine and gives a brief history of it along with its propagation in Denmark. Moreover, the electronic patient journal (EPJ) is defined, and its benefits and drawbacks are analysed. Finally, The Danish Act on Processing of Personal Data, the governmental authority in Denmark that administrates the laws dealing with the personal data is introduced.

In **Chapter 3** the requirements of the framework proposed in the report are analysed and discussed.

The main principles of cryptography, which is a cornerstone in data security, are outlined in **Chapter 4**.

Chapter 5 analyses the communication part that plays a critical role in the framework. The TCP/IP Protocol Suite is described and security is considered in each of its layers. Also, the different types of firewalls and the common forms of VPN are described.

Authentication is the second main part of the framework. In **Chapter 6**, different ways of authentication are described. At the same time, certain attack forms are mentioned and best practise guidelines are given. Finally, two-factor authentication is discussed.

Chapter 7 outlines the traditional and classical forms of access control.

The framework is proposed in **Chapter 8**. Each of its main parts are described. Application of the framework is shown through some case studies. This is followed by an evaluation of the framework.

Finally, **Chapter 9** concludes the thesis discussing the usability of the framework and looks at other external factors having relevance for its usefulness in practice.

1.2 Reading guidelines

Each chapter has a short introduction and the different chapters can easily be read independently i.e. without having been through the other chapters.

Framework only
If the reader only wants to study the framework, he can start the reading from **Chapter 8**. Then, if the reader wants to dive more into some of the main parts, he can read the relevant chapter(s) between **Chapter 4-7**.

Security theory
If the reader is interested in the technical parts of information security, e.g. the TCP/IP protocol or authentication, he may start reading the **Chapters 4-7** that offer a theoretical analysis of the security mechanisms related to communication.

Medical history
If the reader likes to know more about telemedicine and the EPJ, **Chapter 2** should be studied.

Chapter 2

Telemedicine

With the rapid development and progress in the application of the Internet, telemedicine has come into focus. According to The Danish Society for Clinical Telemedicine telemedicine can be defined as Digitally supported health service aid over distance, [29].
The World Health Organization (WHO) provides a more detailed definition:

> The delivery of healthcare services, where distance is a critical factor, by healthcare professionals using information and communication technologies for the exchange of valid information and diagnosis, treatment and prevention of diseases and injuries, research and evaluation, and for the continuing education of healthcare providers, all in the interest of advancing health and communities. [36]

Likewise many other definitions of telemedicine exist with different wordings, all agreeing that telemedicine is about providing health care over a distance using telecommunication technology.

The above definition by the WHO gives a little hint on some of the problems healthcare services might have faced in the past and proposes a solution for them as well. Healthcare services have undergone several changes with time and telemedicine is a new innovation. This type of innovation provides better diagnostic and treatment tools for the doctors and patients, and thereby, in theory, it is beneficial for both parties.

At the moment telemedicine is very much being discussed and at many places telemedicinal systems have been established on trial basis. Telemedicine in itself is not a new technology but a new way of co-operating with the use and combination of existing technologies.

2.0.1 What is telemedicine?

Telemedicine only deals with matters for clinical purposes. Any communication regarding anything else such as education and research is technically not regarded as being a part of telemedicine. Instead the term telehealth which is much broader, is used for this all together.

The communication can be of any kind and between different sectors of the health services. Communication can be sending pictures for instance from one department of radiology to another department at another hospital or to a experienced radiologist at home, sending ECG data from an ambulance to a cardiologist, or general visualization and monitoring of the elder at their homes.

Basically the use of telemedicine is either real-time or store-and-forward, also called synchronous and asynchronous, respectively. The first is a live communication between the two parties (i.e., the doctor and patient or it may be two doctors) and they must be present at the same time. This can be everything from a phone call to robotic surgery. The most common technology used in real-time telemedicine is videoconferencing.

For example, if a doctor is having a patient with rashes on the skin and needs to send the patient to a dermatologist, the doctor starts a video conference with the dermatologist instead of sending the patient. Moreover, both the doctor and dermatologist are looking at the patient

at the same time and they can discuss the case live in the very moment, instead of writing and sending logs to each other.

This shows how convenient it is for all involved parties. The patient would normally skip one day of work to visit the doctor. It would cost another working day if the patient were to be sent to the dermatologist etc. Thus, the doctor and dermatologist get a better co-operation, and save precious time and efforts in getting to a diagnosis.

In the asynchronous type of telemedicine the medical data is first stored somewhere and then sent to the receiving end, who may or may not be online at that time, without requiring an instant answer. This is just like sending an e-mail, which also falls into this category.

Whether it is real time or store-and-forward, both serve to reduce or remove physical distances in regard to obtain medical aid. In isolated communities and regions without access to doctors the benefit of telemedicine is obvious but even in modernized societies telemedicine has a magnificent value. It is a highly useful and effective tool for the communication between doctors and specialists.

2.0.2 The need for telemedicine

The ministry of health in Denmark [75] reviews telemedicine in two ways. First of all, it is possible to measure the quality of the provided health services and the satisfaction patients feel. Secondly it is possible to measure the system economically and review the used resources [78].

The aim of every health ministry is to make sure that the patients get diagnosed and treated quickly. Besides this, they also want the patients to feel satisfaction with the whole system in its totality. This raises the need of implementing the telemedicine throughout the whole system. The quicker one can trace the sickness and the sooner the treatment can get started, the more healthy and satisfied patients there will be.

A patient with heart attack was previously transported in ambulance to the nearest hospital. Sometimes the patient had to be moved to a

more central hospital with the ability of offering a better treatment. In such situations this extra time could have been saved with the use of telemedicine and in some cases this could very well make all the difference between the life or death of a patient.

There is a need for telemedicine in the nurse care at home as well. It is easier for the nurse to take care of the patient by monitoring the patient, and the nurse is able to act quickly if needed. Many elderly people live alone and are visited maybe once during the night. Many incidents could take place, such as trips and falls. It has even come to pass that they keep lying there for several hours without any help at all. Many of them have internal bleedings, and other lesions and by monitoring them, the help can reach them sooner, potentially saving them from great discomfort or in some cases even death.

The need for telemedicine also exists between different departments at the hospital. For example, the expert can remotely see an x-ray of a patient at his monitor and discuss the case over phone/videoconference if needed. The same applies if a small hospital with no expertise in radiology wants some pictures analyzed instantly, the help would be nearby. Likewise, such usage of telemedicine could connect doctors from one town to another, or one country to another.

2.0.3 Background

In the last two or three decades the attitudes towards healthcare services in the society have changed dramatically. Earlier the patients did not have so many demands and expectations from the doctors. This is not the case anymore. Today the patients are also more involved in their disease and treatment. The patients in consultation are sometimes more knowledgeable than the ordinary doctor and may possess detailed knowledge, which was not the case earlier. The behaviour has changed with the introduction of the Internet to the common person. The undreamt of possibilities that the Internet has brought and its wide use in our daily life is beyond description. Although, the health sector is still behind. Thus the implementation of telemedicine is not just a demand by the patients but also a natural consequence of the time and development.

Another thing worth mentioning here is that the general opinion on quality of life has also changed in the last two or three decades. Everyone wants to be free from sorrows and traumas of any kind. People want to be healthy at any cost and as quickly as possible. They seek out a lot of precautionary medicine or vaccination against influenza and there is no patience to see in patients when it comes to finding out the diagnoses and the afterwards treatment, the patients do not exhibit any patience. It is often said that the world has become a global village, and the concept of having one family doctor who deals with all the health information of the family does not exist any more. It is normal that a patient is receiving treatment from different places at the same time. In most cases every doctor must have all the information about the different treatments. Today it is possible to shop through the Internet directly from one's home. The banks have also come online. When the technology is developing so fast, one can only agree with the patients that their demands seem to be more than rational. When almost everything is possible from one's home, it should also be possible to visit one's doctor through the Internet. This is the main reason as to why the implementation of telemedicine has been so much in focus in the recent years.

2.0.4 Telemedicine in Denmark

I 1996 there was a commission that evaluated a huge project in health service called Telemedicine. They concluded that Denmark compared with other countries was lacking behind in this field. Norway and Sweden, for instance, were much more advanced in this joint venture of communication technology and medicine. At that time, they recommended that expertise help from Denmark and abroad was gathered on four levels: technical, organizational, health and industrial. [78]

They recommended a national center for Telemedicine. It should guide, aid and give counsel to the government and coordinate other parties involved in such projects. It should be up-to-date with the worldwide development in telemedicine and make sure that all the lines of direction were kept. In Tromsø, Norway there were 5-10 people connected with such a center. Their job was to start, support and coordinate such projects. Similar approaches were recommended to Denmark.

At the time of writing, telemedicine in its totality has not yet been established in Denmark. One of the reasons behind this is the geography of Denmark. The Faroe Islands, Greenland and Denmark make up one country. In Denmark the population density is greater than the Faroe Islands and Greenland. Telemedicine is more needed in Greenland and the Faroe Islands because the distance to an expert is much greater and any delay in getting medical help can prove fatal. There is a telemedical service between the Faroe Island, Greenland and Rigshospitalet (in Copenhagen) in radiology with 24 hour service. This is due to the fact that there are not any permanent radiologists in Greenland. Therefore the hospitals there must have access to another hospital within or outside the country in order to treat the patients.

There is a project under EU called Mermaid. The Southern Jutlandic University Center and *Søfartsstyrelsen* are taking part in it. It includes the following elements:

- An electronic journal of patients. The paramedic on board will guide one through the clinical features and examination.
- A module of communication that ensures a correct exchange of information between the paramedic and the doctor.
- All diagnostical and therapeutical procedures are described.
- A database which includes the information of medical equipments and a detailed description of all the medicine which is brought onto the ship.
- Information about symptoms, observations and treatment for a series of illnesses and accidents.
- And a module of training, where the program can be tested.

Other similar projects are running at various places in Denmark, especially in the departments of radiology and cardiology. Especially the hospital in Vejle is working on telemedicinal projects. More information about actual projects can be found in [29] and [30].

2.0.5 Potentials and barriers

The use of telemedicine definitely brings a huge number of new possibilities that can improve the treatment and it gives the health sector a welcome face-lift. To get a realistic picture of the situation, an estimation of benefits and drawbacks is necessary.

The potentials and barriers will only be mentioned briefly in this chapter in order to get an overview of the problems telemedicine might face.

There are many benefits of telemedicine. First of all telemedicine can be viewed from the aspect of quality of the service the health sector can provide. Next, there is a social perspective, namely the advantages for and the satisfaction of the patients based on the service they get and its quality. The economical advantages are another way of benefit estimation.

The use of telemedicine will give the patient a number of benefits. Among them, quicker and better diagnoses. Fewer extra check-ups will be needed and there will not be unnecessary transport or extra waiting time. Moreover, in many cases the patient would not need to be admitted. Instead the patient can stay in his home with his relatives and in circumstances where he feels safe. This will be a huge psychological benefit for the patient.

In countries like Denmark, patients do not pay for the treatment, and in most cases even the transport from the patient's home to the place of treatment is paid for him. Thus the economical benefits will also be great. In hospitals the waiting lists are always long and therefore every saving of time will be highly beneficial. As most of the communication between the doctors and specialists can be done using telemedicine both the time and transport expenses are reduced. And in general the doctor's insight will increase. The shorter waiting time due to fewer physical visits, will make it possible for the government to fulfil their promise about "the guarantee for treatment" (Danish: behandlingsgarantien) which is a very hot topic these days. Hence, the hospitals will have more room for aiding "more" patients as the number of people visiting the hospital and also the number of admissions will decrease.

The communication between the hospitals and other places of treatment will increase and the planning can be better arranged. With better planning the resources can be optimized, resulting in huge economical savings. Moreover the doctors can ask and consult experts much more and in this way become better and more effective in their work.

The barriers can, in the same way, be set up in different groups, among which are organizational, regulatory, technical and economical. Also barriers based on attitude are weighty, sometimes even more than everything else. One may overcome all barriers but without being able to change the attitude of the people. Briefly mentioned below are the important factors that could possibly arise or need to be considered are briefly mentioned.

Organizational barriers
As a consequence of the introduction of telemedicine, the existing work processes must be systematically analyzed as the procedures for many tasks change. Changes must be made to obtain and fulfil the purposes and goals of telemedicine. The administration and management structure will be influenced by it too and new positions may be introduced. In relation to the organization telemedicine will increase the decentralizing as many tasks will be spread out and the examination process will include external assistance. However, at the same time telemedicine also gives more centralization as the whole treatment takes place in one place, involving several specialists and doctors not psychically present. For the patient it will give him a connected treatment process with only one case and one doctor responsible whom he can come back to, more or less like the old principle of having a single family doctor. Moreover the telemedical solutions will make new demands to the qualifications of the personnel in order for them to get insight in and work with the new systems. This will require education programs to be set up.

Regulatory barriers
The regulatory barriers contain both legal aspects and security related issues as well as problems related to the settling of accounts or payments.

First of all the legal aspect plays an important and central role. Especially the question of responsibility must be clear when the treatment is carried out with telemedicinal aid. In addition to this telemedicine puts forward

a number of claims for the information and consent, journal writing, professional secrecy, the patient's rights of accessing their documents etc.

As a number of technical systems are used in telemedical systems, a number of security issues will raise and there will be many doors to be locked in order to protect the data from unauthorized access. Along with the outer security threats, the inner threats will also need to be taken care of effectively. The security part can be split up into the three main parts: Confidentiality, integrity and availability. These are common security issues for almost every electronic system dealing with personal information, though, for every system the solution is unique. Moreover the security issues and the legal aspects overlap each other, as the security has to meet the requirement set forth by the legislation.

Telemedicine will change the traditional treatment process as a result of the involvement of several doctors and specialists and this affects the payment settlings as there is an "extra" consulting. Doctors and specialists will not agree on this extra support without getting anything for it. So, for one treatment the several involved individuals must be paid, and thus the question of how to calculate and settle this in proportion to their contribution will arise.

Technical barriers
Computer systems and other relevant hardware must be set up in both ends and this will cause great requirements to the patients as well as the places of treatment. Questions related to the purchase, installation and technical setup must be discussed. Lack of standards for the communication can damage the use and propagation of telemedicine severely.

Economical barriers
The initial costs related to the introduction must be analyzed. Limited economical resources will prevent the application of telemedicine. Also the costs related to the daily use and service must be considered and covered. It must be assured that the savings will exceed the expenses and acknowledged how long it will take before this happens. This is a hard task especially when similar has not been tried before, and so, there is no prior experience to rely upon.

Problems of attitude
The missing willingness to accept changes and adjust oneself to them must be taken into consideration. It is a common attitude that people prefer what they have always done, and what they feel safe with. It may be prejudices towards new systems in the patients' minds or the doctors' fear of not being able to handle them. To change societies' stand is not an easy task. Moreover new innovations always have some start problems and this may lead to the strengthening of people's conservatism if there are too many start-up issues.

The potentials and barriers mentioned in this chapter are primarily based on and deduced from the publication [74] released by the Danish Health Ministry and *Teknologirådet*'s paper [78] on the use of telemedicine.

2.1 Eletronic Patient Journal (EPJ)

The electronic patient journal (EPJ) is one specific application of telemedicine. The purpose of EPJ is to replace paper journals with an electronic journal system. All information belonging to a patient will be gathered and kept in one place. It will be accessible through the Internet to any authorized person from any place in the world and whenever needed. Although a person visits different treatment places, the doctors will only have access to one (and the same shared) journal for that person. Whatever they add, update or write will immediately be visible for others who are authorized to access and view the journal. This increases the possibility of co-operation between the doctors which is the very spirit of telemedicine. In 2003 *Indenrigs- og Sundhedsministeriet* in Denmark decided that all hospitals at the end 2005 should be using the electronic patient journals. [53]

EPJ should have been implemented in all regions in the 2005, but today in 2008 unfortunately, this has not been done yet. There is still no saying exactly when EPJ will be fully implemented everywhere. Moreover, most of the places where EPJ is implemented, are facing so many problems and difficulties with it that they only wish to get rid of the system and return to the former practice. The newspapers and weekly

magazines are filled with articles from doctors and IT-specialists harshly criticizing EPJ. In the previous chapter a number of barriers were mentioned for telemedicine. Many of the very same barriers exist for EPJ. In the following, some of the technical issues and concerns for EPJ will be discussed, and analysis of the EPJ system together with its goals and visions is analyzed.

2.1.1 Description

There is no unambiguous definition of what EPJ is. Many different definitions are given. *Sundhedsstyrelsen* defines EPJ as:

> An electronic patient journal is a clinical information system which directly supports a daily process orientated medical examination, treatment and care of the individual patient. [76]

Whatever the precise definition of EPJ is, there is general agreement that EPJ is a means to help during the treatment, diagnostics and a support for making decisions. EPJ is not different from paper based journals in the sense that the same kind of information is stored in both. The main difference is that in EPJ the data is stored digitally.

The EPJ can be seen as a complete health profile card for a person, as his total health information is gathered in one place. For the patient, it gives him a better and more effective treatment. At the same time the patient gains more insight in his own treatment as he can follow his information and even decide what level of access the different doctors should have.

Technically an EPJ system consists of a strong central computer with a number of client computers (work stations) attached to it. Together with the central computer there may be a separate database server where all the data is stored. Notice that the data itself is not different from the paper journals as the content of the journals is unaffected by whether it is stored electronically or in paper. Only the central computer has access

to the database server while all the clients have (limited) access to the central computer in order to receive and send the necessary data. The connection may be through the Internet, so that the client computers can theoretically be placed on an arbitrary place as long as an Internet connection is available. However, this is only an ideal case. In reality the setup is for local networks (for regions) only and thus a number of small EPJ-setups are quite conceivable in reality. The software installed on all the work stations is the same, or at least software that is compatible with each other and developed with the same requirements.

2.1.2 Functionality and improvements

Basically an EPJ is used for the same purpose as paper based journals. This includes registering patient journals, storing data and accessing the journal. However with the EPJ system the functionality increases. Among new properties are the following:

- Where a paper based journal can only be read by one person at a time, it is possible for multiple users (in different places) to read the same journal in the electronic version.
- Patient data is added to the electronic journal from several places and even simultaneously.
- Data can be sorted and the information may be differentiated, so that only selected and relevant information is shown.
- Calculations can be made on data and the result may well be shown graphically also.
- The electronic journal system may be able to send and receive data to and from other information systems.
- In the EPJ information can be found quickly by doing a search.
- All relevant information for one person is gathered in the electronic journal and only one patient journal exists no matter how many places he is treated.

- Access to the journals and all changes made can easily be logged.

The increased functionality of the EPJ does not come for free however. A number of security issues and problems arise. These are dealt with in the next section.

2.1.3 Paper journal versus the EPJ

Going from the paper journal to the EPJ a number of issues must be considered. Below some main issues are discussed.

Placement of the journal
Paper journals are physical objects placed somewhere, usually at the ward offices. In order to read them one must enter the room where the journals are and find the journal. At small clinics, the journals are usually placed at a fixed place, but at hospitals and bigger clinics, this may seldom be the case. Especially when a person is admitted, his journal is transferred to that ward or it may be at the office of the doctor responsible for his treatment. Moreover if a person is transferred to another hospital, his journal will also be transferred to that hospital. The point is, it may not always be absolutely clear where the journal is at a given time. Therefore to find the location of the journal itself, is also a task in itself for a would-be intruder. First of all he must locate the position of the journal. Next, he has to personally travel to that place and find a way to get into the room. However even this is not without difficulties. The wards are seldom completely empty of people and the staff knows each other very well. Even if he manages to enter the room, he may need to browse through a huge number of journals to find the desired one. To carry out all these steps requires deep insight into of the system, for instance, to know how to locate the journal, how to break in and pretend to be from the police or such.

As soon as the paper journal is converted to an electronic version all these hindrances that naturally existed are gone. The journal is no longer a physical object but a piece of information placed on a database server. The database would require a high level of protection, continuous moni-

toring, be placed behind secure walls making it difficult for unauthorized persons to break into it. Only a handful of people would have access to it directly. As it is only the raw data that is stored in the database even the doctors do not have access to it. Instead the access to and from the database would happen through some graphical user interface. But to break into the room where the database is would in itself never be the target for an intruder, as there is no reason for this. Instead the intruder would try to break in through the graphical user interface by hacking it. The larger the network is, the more vulnerable it is. For instance, when the electronic patient journal is extended so that it is reachable from the Internet, then every person connected to the Internet can potentially be a hacker. The threat increases to an unpredictable degree and therefore this will require appropriate precautions.

Accessing and transferring the journal

If one person is reading a paper based journal it is not possible for another to get the same journal and find information in it until the first person has finished reading it. Besides, given that the journal is only located at one place it must be transferred from one place to another if it is needed somewhere else, because copies are not usually made. This transport may happen with the hospital's own transport system or using the public mail system. Usually copies are not made of the patient journals. First of all because the journal may not always be so simple because of the dissimilarity of the journal content (such as have x-rays, graphs and other kinds of pictures along with the text). Another reason is that there would be no way to prevent the journals against unwanted spread and this propagation of journals would quickly come out of control. It could end with multiple journal copies with different contents leading to a problem of defining which the original main journal is. Hence the security around the journal transfer depends on the external transferring medium. With the electronic journal system there is no need for transferring data in order to read the journal as long as the same EPJ system is being used. Whenever the journal is asked for, data will be sent from the database and shown in the graphical user interface as a virtual copy of the journal. It also means that the restriction of how many persons can have the journal simultaneously is lifted. Although it has been stated above that the journals do not have to be transferred from place to place when reading it as it is only a matter of logging into the system, in reality it

is just the opposite. At every valid authorized request journal data is sent from the database server to the client terminal where it is shown and this technically counts as a transfer. This puts forward a strong condition for the protection of the data transmitted over the network from the database to the terminal. The threat is actually greater than before, as the data earlier could only fall into wrongs hands by troubling the used post system and this would only be possible in the local area where the delivery person travels. In contrast to this, when data is sent over the network, all those who have access to that network can interrupt the transmission. Again, when the system is extended to the Internet, in theory everyone could potentially disturb the transfer.

Browsing the journal and event logging
It is not possible to track what a person does with the paper journal, e.g. it cannot be traced what he did read and what he did not. Moreover papers could be taken out of the journal and destroyed, something could be put in that was not there before or other kind of changes could be made. Assume that a doctor gives some wrong medicine causing the death of a patient or serious illness. Now the doctor could make up some text and insert in into the journal to prove that the patient needed the given medicine and that it was rightly given in the hope of averting legal proceedings. The chances of catching this act would be small, especially if there is reason to suspect the doctor. Likewise there could be many other reasons for a doctor, an intruder or some third person to make changes in the journal, not to mention the authority that the secretary has. If a person wants to know which people have access to his journal and for what reason, then this is simply not possible, in any provable and trustworthy way.

Besides the fact that tracking is not possible, another problem is that whoever is authorized to read a journal will have full access to it, even information without any relevance to him. Likewise if an unauthorized person gets access to the journal everything in the journal is unveiled. When it is not possible to find out who is reading a journal, how can it at all be possible to prevent the journal from being seen by unauthorized persons?

A huge advantage in the electronic version is that all kinds of activities

concerning a journal can easily be logged and traced. It is therefore no longer feasible that unjustified or hidden alteration and modifications take place in a journal without being logged. The electronic system will be able to show every entrance to the journal and even the purpose for viewing the journal can be made a requirement to state. Though, such possibilities exist, it requires strong systems and a lot of backup and file handling as the log files can easily become very large.

Using adequate access control mechanism all the mentioned problems can be solved. In this way the patient can follow who has what rights at any given time. In addition, an access control mechanism makes it possible to allot specified roles to the authorized persons so that only relevant information is shown. However this will require some kind of grouping of the information into categories which for instance could be according to the content. Access control mechanisms play an important role in electronic system storing personal information and a number of issues are connected to their use.

Searching a journal
To find a journal or to look for something in a journal requires one to flick through every page in the journal. If the person wants to know the number of patients with certain diseases he must go through all the journals individually. It may be that the intruder manages to wangle his way into the room where the journals are, but in spite of this, does not succeed in finding the desired journal.

In the electronic journal system, data is placed in a database and is searchable which means that both to find a person, to look after something in his journal and to search for something through all the journals at once will be possible. When this is viewed from the doctor's side it will have time saving benefits and better statistics can be made. But at the same time the security concern again becomes significant; if an unauthorized person breaks into the system he may easily scan through all journals searching for different words. Dirty sellers and researchers may pay hackers to bring them information about the holders of certain diseases, which is called screening. Therefore limitations must necessarily be put on the system making it difficult to carry out screenings. And of course even an authorized person should only be able to search through

the data he is allowed to access.

Writing to the journal
Another problem with the paper journal as discussed above is that there is no differentiation in the shown information, e.g. everything is exposed to all viewers of the journal. With user roles this problem is solved. At the sametime, it also solves another issue, namely about writing in the journal. User roles can be made so that a person is not allowed to read anything but only write to certain parts of the journal. Assume that a person is playing football and gets injured. He visits the doctor who adds the log of his visit to the person's journal. In such cases the doctor can actually fulfil his task without having access to any part of the person's journal. The main rule that has to be followed is that the least access rights necessary for the performance of a task must be given. This increases the security as unnecessary doors (i.e. read and write rights etc) will not be opened when they are not needed and the database will only send the necessary data. The same principle is being followed in other secure systems. For instance in certain online banking systems (such as Nordea, Danske Bank etc.) one can select whether he wants to login to view his account, to transfer money or if he wants full login.

For the journal writing in general there are laws for what to fill in but with the paper journal, the doctor can decide to fill them out or leave them empty. Certainly this will always apply but in the EPJ at least a basic validation of required fields can be made. This can also catch some unintentional errors, and mistakes due to misinterpretation of a doctor's hand writing will not occur.

Besides the mentioned points there are many other issues that can be discussed like the doctor's professional secrecy, the patient's right of access to the document and similar. Such problems are not directly connected to the IT system and security but are rather related to the law and belong to the regulatory and organizational aspect. For that reason they have been left out from the above discussion of issues that are primarily of the security aspects.

2.1.4 The need for EPJ

Whenever a person visits a doctor, be it is his personal doctor, at the hospital, his dentist, his club doctor etc. The doctor appends a log regarding the person's visit in a local journal, which the doctor has for every person that once or regularly comes to see him. Thus it is normal that a person has a number of journals, one for each place he visits. Earlier these journals were all paper-based journals but today almost all institutions have some local electronic journal application installed where the data is inserted. In countries like Denmark where all people normally have a personal doctor, it is preferred that the personal doctor gets informed about his or her patient's visit to other treatment places but this is done only by the agreement of the patient. Hence it is the patient's choice whether information or details about his visit to any doctor should be sent to his personal doctor or not. In many cases it is a great help for the doctor to have knowledge of his patient's visits to other doctors. However, the decentralized journal system, i.e. where all institutions have a local journal, sets a number of limitations and does not allow much co-operation. The different journals are stored locally and created with different software without any common format. There are no standards and the transfer of journal logs from one treatment place to another is done either with post or an e-mail. The receiver can then read the received journal data and insert it into his own system. Many times a person himself is asked to bring his data, for instance from one dentist to another, the patient may be responsible for bringing his previous dentist logs. Moreover if a person wants to read his journals he must visit the different places and ask for his journal. This may not be possible for him if he gets admitted in a hospitable in another country during his holidays.

The solution for this is to build a centralized electronic journal system with remote access and access roles for all authorized treatment places. This could be placed on a nationwide network or extended even more.

2.1.5 Technical requirements of EPJ

In 2001, the Danish Government published a document about the implementation of EPJ, [73]. Recently, in 2008, a new publication has been released in which the requirements are mentioned, [27].

The main points in [73] are the following:

- Proper user authentication.
- Providing different user rights and roles.
- Differentiated access control of both services and patient information.
- Being able to support emergency situations.
- Protection against data revealing to unauthorized parties.
- Security with data transfer in form of integrity and confidentiality.

In [3] [27] the same points are followed up with more details, especially the access control part is discussed.

There are three requirements for accessing data that have to be followed at all times.

1. Information can only be provided when it is necessary, e.g., when a patient is being treated.
2. Only the necessary and needed information must be read, if this is technically possible. This means that the part of a patient journal which is not relevant for the doctor should be kept hidden.
3. If the patient does not want his information or parts of it to be read, the system should be able to put such limitations.

For doctors and other parties that the patient has approved access for, the above three rules apply. But for all others who can get access, point 1 and 2 are limited further, by adding that information should only be provided to the doctor from the treatment place, and that only information about the actual treatment is revealed.

However, besides the three main rules, there is a rule which allows the responsible party (e.g. the doctor) to get access to data in case of emergency, or if the patient is not able to take care of his affairs himself (e.g., mentally retarded persons).

Regarding the security, data must be encrypted at different levels. For sensitive data strong encryption is required. Moreover, stronger authentication for remote working places should be considered.

These are the main requirements for EPJ. Any implementation of EPJ further needs to be approved by The Danish Data Protection Agency.

2.2 The Danish Act on Processing of Personal Data

The Danish Data Protection Agency (Danish: Datatilsynet) [24] is the governmental authority in Denmark that administrates the laws dealing with the personal data, referred to as the Danish Act on Processing of Personal Data (Danish: Persondataloven). It receives complaints, responds to queries and makes statements about bills and makes drafts for publications. Moreover it makes inspections of organisations and keeps an eye on any act that is included in and linked to the act of personal data.

The act applies to the processing of all kinds of personal data, such as the handling of data collection, transmission and storing. Personal data is defined as information that directly or indirectly can refer to or disclose the identity of a physical person. It is not dependent on any medium meaning that it concerns both electronically stored digital information and paper based information as well as images, sounds and video and even biological material. Only anonymous data that cannot, in any way, be traced back to its origin are not under this act, which means that even encrypted data comes under the act.

Any organization, institution, firm or private person that works with personal data and is established in Denmark must observe the act regardless of which EU country the work is carried out in. If the institution is established in Sweden for instance and the experiments are carried out in Denmark, the Swedish act of personal data will be applied. [23]

No matter if the firm is private or public, as long as it is established in Denmark, the law must be observed. In addition to this, the law is not passive, meaning that firms cannot start working and follow the act by themselves. Every firm must be approved by the Agency, which again means that prior to commencement of processing personal data, the Agency must be informed and authorize the work. Then a number of requirements must be met and certain things have to be implemented. This is done in order to ensure the security so that it reaches the level that is required by the agency. Moreover it has to be documented and

proved by the firm to the Agency that the requirements are met. Besides this, the Agency makes random inspections to verify the observance of the legal terms.

In this way it is governmentally ensured that any legal Danish firm dealing with personal data live up to the minimum security level.

The act in itself is quite long. According to [23] the most important conditions are the following:

- Electronically stored information must be password protected or encrypted.
- Transmission over the open network requires strong encryption and through the intranet protection against unauthorized access is necessary.
- Non-electronic stored information must be kept in securely and locked and biological material must be protected against loss, depreciation and destruction.
- The physical conditions, rooms, where the data is kept must be guarded against unauthorized persons.
- The information stored is not to be handed over to a third party without the confirmation of the Agency.
- If the project in which the information is used ends and the data is no longer needed, it has to be deleted or made anonymous, both in such a way that it cannot be recovered.

Chapter 3

Framework requirements

In the past chapters, telemedicine has been described with focus on it's development and implementation in the society. Therein one specific application of telemedicine has been introduced, namely the electronic patient journal (EPJ), often referred to as the electronic health record (EHR) in English and American literature. With the increasing propagation of electronic health records, and in general, the growing application of the Internet in medical systems, the need of protecting the privacy of the patients and the vulnerability of personal and sensitive information security comes in focus. Security is critical for any electronic healthcare system as a breach can have decisive consequences. Leakage of private information may threaten the patient's privacy. Improper use of medical services may threaten his health or even his life if malicious modification of diagnosis data is made. The damages of an insecure healthcare system are inestimable, for both the patients and hospitals, not to mention insurance companies.

The main purpose of this thesis is to propose a framework for secure communication for medical systems. In the previous chapters the advantages and benefits of both telemedicine and the electronic patient journal was

briefly touched upon, along with their needs and usage. The intended usage of this framework is online electronic patient journals and applications of telemedicine. However, the framework should be presented at a higher level so that it can be applied to any system dealing with sensitive information. Moreover the framework should neither be dependent on any platform nor go into specific software or algorithm details in order to get greater compatibility. In this chapter the framework requirements are analysed.

The key factors that must be considered for a secure data communication are confidentiality, identification and authentication, integrity and authorization.

Data in transit flow through long distances passing many intermediate servers and networks. While some data are neutral or insensitive, some data transfers are highly confidential. Therefore different levels of confidentiality must be considered. Sometimes only confidentiality of the data being transferred matters, but in many cases it is necessary to prove the origin and integrity also. For authorization, proper access control mechanisms are needed.

Users and authentication
In a health care system there will be a huge number of users, having one or more of several roles, such as patients, doctors, nurses, secretaries and other administrative personnel. From time to time new roles are needed, and modifications of existing roles may occur as well. The users are located widely in many different places, for instance, at the hospital, in the clinics, at home, at public places etc. (cf. Figure 3.1)

Proper identification and authentication must be done, both for the purpose of security and for trust. If a patient just wants to download a receipt, this may not require the strongest authentication and identification. Instead, if a patient wants live online consultation with his doctor, it would be required that the patient can have trust in whom he is communicating with on the other end. Moreover depending on the roles, the authentication level may vary; hacking a patient's user account which only has one access right (read) and only to his own data, although it can have severe consequences, cannot be compared to the case, if the super-

privileged administrator's account is compromised. Therefore some roles may require stronger authentication than others.

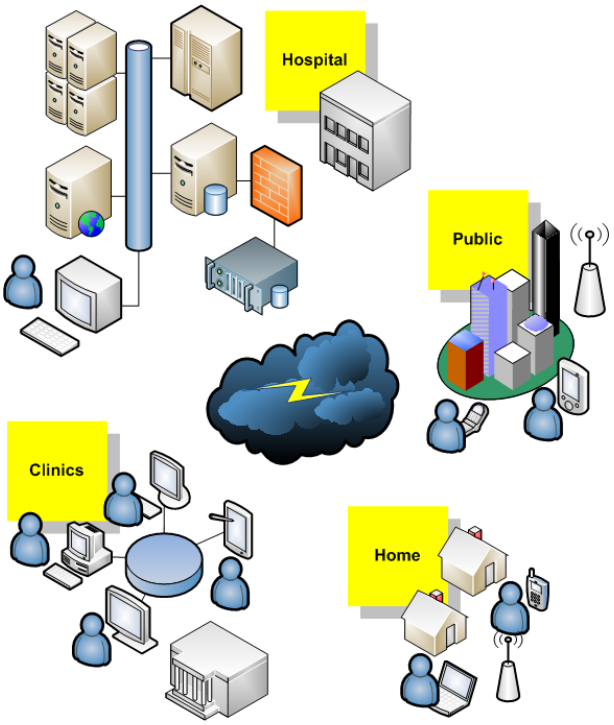

Figure 3.1: Users of a healthcare system connected through the Internet

Data and authorization

In the scenario of EPJ, data is usually medical records having information of varying sensitivity. A medical record can be divided into parts. For example, one part records the personal information only, while another one contains the medical information. The record will be classified further according to its use, such as the history health record (e.g. for reference purpose), general health record (e.g. for normal treatment purpose) and critical health record (e.g. for life saving purpose). Beside the different views of a record, there are also different actions, among which are reading, writing, updating, searching, etc. Consequently a proper access control mechanism must be set up for the handling of rights management and protection of information.

Primarily access control includes two aspects; the denial of access to users not having access rights, and to provide necessary access to the authorized users according to their exact permissions. The goal is to develop a need-to-know access control mechanism providing exactly the data needed, neither more nor less. In principle a user should only have access sufficient to accomplish his task, though it is not easily achieved. A patient should not have access to see other patients' records, doctors should only be allowed to view their own patients, and sometimes only the part of their patients' records required for the treatment. In emergency situations, the doctor may need to have instant access to a patient's data even without the patient's consent. Likewise, there may be situations where a doctor needs to see more information than permitted under normal circumstances. It can also be that there is some information which the doctor initially does not have, for example, before an operation, which should be available afterwards.

To protect the patient's privacy, the system must assure that the medical patient information remains confidential and that it is not disclosed except according to the law and/or the patient's explicit consent. The latter requirement is particularly important in countries where medical insurances are privatized and the patient's medical data has monetary value. At the same time, the system must be immediately available to provide high-quality health care. Fulfilling both requirements is not an easy task. The access control mechanism must be simple and fast, in order to provide immediate availability. Super-fine granularity, only granting access under precise conditions and circumstances, in a complex medical system with a large number of health records, many users with several roles, makes the system slow. Moreover, it is impossible to design an access control mechanism that models every circumstance. In practise, there will always be some more information shown than absolutely necessary. Hence, the choice of access control mechanism is a trade off between simplicity and speed on one side and getting perfect granularity on the other side.

Access control mechanisms are not just limited to the medical data, but also addresses the different health services provided by the system, i.e. to allow and deny services.

Data transfer and communication
Most of the telemedicine services include communication between two or more parties. A patient communicates with his doctor, whether it is audio or video conference or two doctors exchanging medicinal information. Hence, a secure channel for the communication is required. The security level of the connection varies according to the needs of protection. While some data may be transferred unprotected on an insecure channel, other data would need a high level of confidentiality and integrity. The Internet protocol suite is used for all Internet communication today. The framework should consider security for its different protocols.

Before proposing the framework, an analysis or state-of-art review of the different aspects and major fields (cryptography, communication, authentication and access controls) related to the framework are described and explained in Chapters 4-7. The framework is presented in Chapter 8.

Chapter 4

Cryptography

When discussing communication security techniques, the field of cryptography cannot be left out as it forms the basis for the majority of the security schemes. Regardless of the topic, be it protocol security, user authentication or access controls, all make use of cryptographic techniques in order to provide security. For this reason, an overview of the main elements in this field is given in this chapter. The reason for placing the chapter first is that all the following chapters somehow mention cryptography. The best understanding can only be obtained for the reader if the techniques and terms being used are described and known already. Alternatively the different elements of cryptography could be described in the very place they appear in the different chapters. This way, the reader might find it confusing and unnecessary repetition would occur.

As the purpose here is only to bring an outline of the cryptographic working fields, the mathematical aspects will be omitted. For a comprehensive treatment of the *state of the art*, the reader is referred to the many textbooks in the field.

4.1 Types of encryption algorithms

Different encryption algorithms are used to obtain the goals of cryptography. Encryption algorithms are mainly of two kinds: Symmetric and asymmetric encryption.

Before going on to describe the two types, some of the important terms used with encryption are shortly introduced. The message that is to be encrypted is called plaintext and after being encrypted it is called ciphertext. There is always an encryption method bringing a plaintext to ciphertext and a decryption method to bring the ciphertext to plaintext. Several different encryption methods may be used on the same text, i.e. multiple encryptions, in order to strengthen it. They have each a key that is given to the encryption methods as argument when doing the encrypting. The same key which may be secret or public may be used for the decryption.

4.1.1 Symmetric encryption

Symmetric encryption is also called private-key encryption or traditional encryption. The sender and receiver share the same private key which must be unknown to all others. The encryption and decryption methods are usually public but as the key is secret they are not of any use for an attacker. Moreover the decryption function is normally the inverse function of the encryption function. It is built in such a mathematical way that it can be seen as a one-way function. In this way an attacker cannot use a ciphertext and the encryption function to solve the private key.

A very simple example of such an algorithm could be two persons sending PIN-codes to each other. Whenever they send, the sender adds a number to the PIN-code on which they have agreed in privacy. The receiver subtracts the same number when he receives the original PIN-code. If an intruder fetches the message during the transmission he does not know which number is added to the original message.

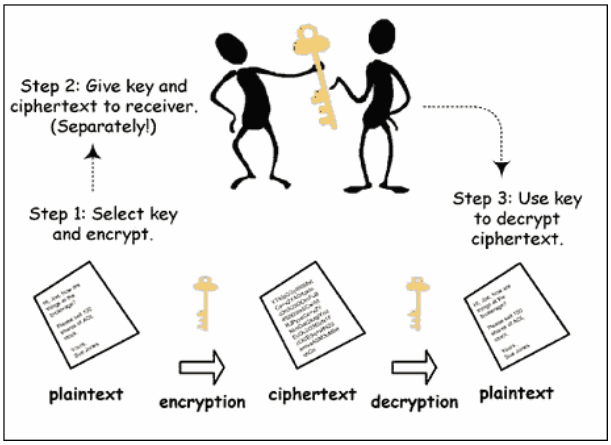

Figure 4.1: Symmetric encryption

The example above illustrates a very simplified method of making symmetric encryption. In practise, very advanced methods having several different and long keys are used in order to prevent attacks. A number of issues are necessary to consider regarding this encryption form.
Whenever symmetric encryption is used the communication parties must have agreed on the secret keys beforehand. This shows that a secure channel is needed and the question is how to actually do this. The purpose of the encryption is exactly to establish secure communication, but it is a prerequisite that keys have been exchanged over a secure channel already.
The next problem is that the key length determines how hard it is to break the encryption. As mentioned the algorithms are normally public and therefore an attacker might try all the keys up till or of the key length. If the key length is short, it is easily done.
Thirdly, as both the sender and receiver share the same key, there is no way to determine who encrypted the message. If a disagreement occurs between them, both can refuse to have sent the message and instead accuse the other for having sent it. The more parties knowing the secret key, the greater the problem is. Moreover the security decreases with the number of people having the secret key as all of them can potentially come to reveal it. This shows the difficulty of using this form when having multiple parties communicating.

There are many symmetric encryption algorithms in use. Some of the famous ones are DES [57], 3DES (RFC3217), IDEA (RFC 3058), RC4 [44] and RC5 [64].

4.1.2 Asymmetric encryption

Asymmetric encryption is often called public-key encryption. In this form the sender has a key-pair of one private key and one public key. The public key is used for encryption and the private key for decryption. For example, a person named Bob publishes his public key somewhere e.g., on the Internet.
Whenever someone wants to send an encrypted message to Bob, he uses Bob's public key to encrypt the message. Bob, having the private key, is the only person who can decrypt the message. Not even the encrypting party can decrypt it. Again such functions are chosen so that it is impossible (i.e. extremely difficult) to calculate the private key through the public key.

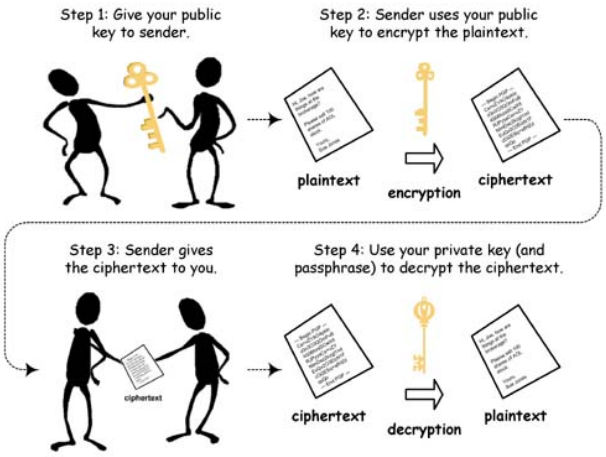

Figure 4.2: Asymmetric encryption

The problem of how to securely agree on a key between sender and receiver does not exist here. Moreover the key-pair is uniquely attributed

to one person, as the private key is not shared with anyone. Thus the problem of multiple parties communicating is also solved.

The public-key encryption has a lot of benefits but compared to symmetric encryption is very slow. According to [31] it can be more than 1000 times slower. In real applications, this makes a huge difference, especially when a lot of encryptions are done.

The most well known public key encryption algorithm is the RSA [43].

4.1.3 Hybrid public/private key encryption

In practice, almost all encryption systems are not merely symmetric or asymmetric. Instead a hybrid of both is used. The public key encryption is slow but can be used without being dependent on a secure channel to exchange the key first. Therefore public key encryption is used in such hybrid systems for key exchange, and the actual encryption is then done using private key encryption.

4.1.4 Message digest functions

With time new function classes have become popular and are used with encryption. One of the most important is the message digest functions, also called hash functions.

> A message digest function generates a unique (or nearly so) pattern of bits for a given input. The digest value is computed in such a way that finding an input that will exactly generate a given digest is computationally unfeasible. Message digests are often regarded as fingerprints for files. [31]

Typically a message digest is between 128 and 256 bits in length [31]. In a powerful message digest function a number of mathematical properties

must exist. Given a piece of data and its message digest, it should not be computationally possible to find another piece of data with the same message digest. Or more generally, it should be computationally impossible to find a corresponding data stream. In the message digest every bit should be influenced by every bit in the function's input. If any bit in the functions' input changes, every bit in the digest has a 50% chance of changing.

It should be noticed that message digests are not used for encryption and decryption in itself, but rather for creating digital signatures and message authentication codes (MAC). As mentioned, a digest serves as a fingerprint, and if the abovementioned properties are followed, it is a powerful tool for detecting even small changes in a data stream. Before sending a data stream, one calculates its digest and places it in a public place. When the receiver receives the data stream, he calculates the digest himself and checks if it matches the one calculated by the sender. If it does not, bits may have been lost in transit or intentional altering has happened.

Among the most used hash functions today are the HMAC (RFC 2104), MD5 (RFC 1321) and SHA-1 (RFC 3174).

4.1.5 Digital Signatures

In the real world, when two parties communicate identification takes place either by recognizing the person if he is known on beforehand, or more practically passports and driver's licences establish identification as they are issued by trusted authorities. *Digital signatures* deal with establishing identity electronically. Digital signatures are based on the idea of public key encryption. Every person has a public and private key. If a person wants to send something he can sign it with his private key. The receivers can then check through the public key of the person, whether it has been signed with his private key or not.

4.1.6 Digital Certificates

The mentioned techniques are very useful. However, to be used for large systems, having a central authority that can be relied on is required. It is not practical if everyone has to manage all their contacts' public keys and at the same time be sure that they are not fake or if they have been compromised and therefore should be updated. Instead, a *central authority* (CA) must handle these things. A central authority issues *digital certificates* and is trusted by all the users. A digital certificate can be thought of as a passport and the CA as the passport issuer. It contains a number of information including a person's name and public key. As long as the certificate is valid, the communicating parties do not need to spend any further efforts to assure that they are communicating with the intended person. VerySign [83] is a very well known CA.

By integrating digital certificates, public-key cryptography and certificate authorities, Public Key Infrastructures (PKI) make a network security architecture. A PKI binds public keys to the users and manages issuance of certificates, along with revoking, renewing etc. PKIs primarily provide the services: Authentication, integrity, confidentiality along with signature services. [1]

Chapter 5

Communication

One of the most used and common communication channels today is the Internet. More and more sensitive data is being transferred through the Internet, and so most banks have an online banking system today. Likewise healthcare is a sensitive area, as both the patient and the medical data are regarded as highly sensitive. Although the Internet has become so influential, it is basically insecure. Therefore, whenever an online system is modelled having sensitive data the transfer security issue cannot be let unhandled.

There are many ways of securing data. Both regulatory bodies and data protection authorities have put forward guidelines and requirements to be met for secure data transfer. This chapter deals with data transfer over the Internet.

When data is transferred a number of questions arise. [11] puts up and discusses the following questions:

1. Has the transferred data been read by any unauthorized person?

2. Who sent the data? How can the sender prove his identity?
3. Is the received data the same as that sent by the sender?

From the cryptographic point of view there are three main aspects related to secure data transfer. These are confidentiality, identification, authentication, and integrity. Short definitions of these are given in Appendix B These aspects, which are also the main working fields in cryptography, bring the answers to the above questions. Confidentiality answers the first question, identification and authentication the second, and integrity the third.

There are several algorithms dealing individually with each of the aspects. An example of this is the Secure Socket Layer (SSL) protocol for securing web traffic. It is supported by almost all web browsers and servers today and it is amongst or the most used protocol for secure transfer. SSL supports 9 different encryption algorithms dealing with confidentiality, 14 key exchange algorithms handling identification and authentication and 2 algorithms that deal with integrity. [11] [9]

The data protection can be a complex task when it is looked at from a technical point of view. Many models have been set up to handle this. Models are necessary as they reduce the complexity. An example of a security model is the one by Bruce Schneier called 'The Security Onion' [11]. In this model Internet security is conceived as an onion. The onion has a kernel that is surrounded by different layers. Figure 5.1 illustrates the onion.

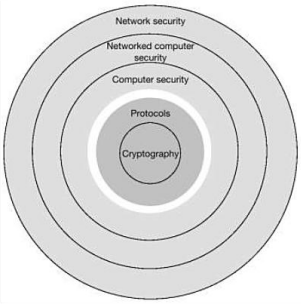

Figure 5.1: The security onion, [11]

The kernel is based on cryptography and security protocols.
Cryptography was described briefly in the previous chapter and it is a branch of mathematics and computer science. Cryptographic algorithms exist for all the mentioned aspects, such as confidentiality, integrity, and authentication.
A security protocol is a protocol that performs security related functions and achieves certain security goals by combining cryptographic algorithms and infrastructure requirements. Some of the aspects are entity authentication or identification, non-repudiation, and secure data transport by transferring encrypted data.

The combination of cryptography and security protocols has the potential to completely secure the transferred data. There are even some cryptographic techniques which can give complete end-to-end security. For this reason these two fields make up the kernel in Schneier's model. The Security Onion does not in fact limit the security issues related to data protection at transfer time but is a general model for data protection.
The interaction with surrounding layers is necessary as some of the security goals cannot be achieved only by cryptography or protocols.
Around the kernel is the computer security layer. This deals with controlling access to the data and the use of it, along with the software.
Next is the networked computer security layer. When transferring data, computers are connected (networked) to each other. This layer deals with the control of the data flow between the computer and the network. It must be determined which data flow is allowed and trusted and which is not. The last layer is for the network security as the networked computer form a network. It deals with all kinds of attacks against the network, granting, blocking or modifying communication, and allowing access to data resources.

[11] sums this up in the following table:

Layer	Security aspects and issues
Cryptography	Algorithms for confidentiality, identification/authentication, integrity
Protocols	Security protocols (S/MIME, SSL/TLS), online/offline certificate distribution methods
Computer security	Rights management, uninterrupted power supply, physical access protection
Network computer security	Filters, malware detection, access control on resources
Network security	Firewall, virtual private networks, tunnelling, e-mail security, intrusion detection, other

The security onion helps defining the security by splitting up the security into layers. This is useful as there is no absolute way to define security. The security onion brings a way to evaluate the level of security layer by layer. Security is always only as strong as the weakest link in the chain. Therefore even though the network security is very high, if there is no computer security, the data cannot be deemed protected and secure.

5.1 The TCP/IP protocol stack

The TCP/IP Internet Protocol Suite [19] is the set of communication protocols that can be used to communicate across any set of interconnected networks. The Internet runs on a protocol stack that is implemented by this. TCP/IP consists of the two main standards Transmission Control Protocol (TCP) and Internet Protocol (IP).

The TCP/IP protocol is organized as series of five layers of levels, each one built upon the one below it [77]. The purpose of each layer is to offer certain services to the higher layers and without giving any detailed information about the actual implementation of the services offered. In this way, using layering, the management of a complex protocol becomes easier for the (security) designer. Figure 5.2 shows the TCP/IP layered model.

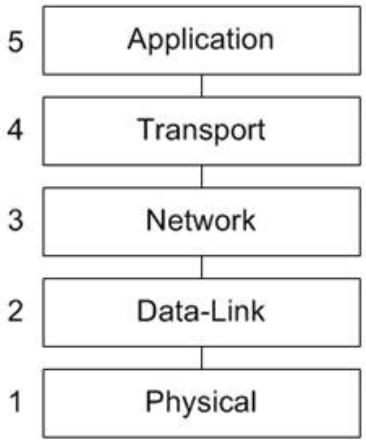

Figure 5.2: The TCP/IP model

From lowest to highest, the layers built upon the physical layer are the data link layer, the network layer, the transport layer and finally the application layer.

When a certain layer (layer x) on one machine wants to communicate with another machine, the communication is carried out on with the same layer (layer x) on the other machine. The rules and conventions that are followed during the communication are called the *layer x protocol*. A protocol is a contract or agreement made by the two communicating parties on how to carry out the communication. In practise, the data is not transferred directly from layer x on the first machine to layer x on the second. Instead each layer transfers the information to the layer just below it. This continues until the lowest layer, which is the physical medium (e.g. Internet). From this, the information is forwarded to the receiver's machine and transferred up layer by layer until it reaches the same layer from where it was sent on the sender's machine. The communication between two application programs is referred to as end-to-end.

All networks organized into layers work in the same way as described above. Before looking further into the layers, it is worth considering where the security belongs to in the protocol stack. There is probably

no single answer to this, as the security issues are not restricted to any specific place. Like it was the case with the security onion, every layer contributes to maintain the security.

Physical security could be as simple as door locks, alarms or security guards. Although security in the physical layer does not touch the technical part directly, it is certainly still of high importance. For what use is the most secure system if it is left without any protection in public space? However, security in the physical layer will not be discussed in this part.

In the data link layer it is possible to encrypt data as it is transmitted between two points in a network. This method, which is called *link encryption*, is easy to add to any network and often very useful. The plaintext, which is the data, is encrypted at the time of leaving the host, and then decrypted at the next link. The next link could be a host or relay point. When it continues to the next, it is re-encrypted and then decrypted again as before until it reaches the end destination.

Link encryption takes place in the lowest protocol layer (the data link layer). In situations where the security of the transmission line is not assured, this method is useful as it protects the message in transit. However, the solution breaks down if the packets have to go across multiple routes. This is because the packets are decrypted at each host and thus vulnerable to attacks from inside the host. Especially in the case where the host is not known, the vulnerability increases.

Within organizations and the military where the security of each link can be assured, the method has widely been used, though, it is not so practical over the Internet, because intermediate links are neither accessible nor secure. It does not allow some sessions to be protected, for example those that involve online purchase by credit card.

In the network layer, security can be obtained through the use of firewalls. Firewalls are appliances located at the connection point of the internal data system and external network. The addresses of all communication that are directed into the enterprise are read by the firewall. It determines whether the communication should be allowed to pass into the internal

network or not.

In the transport layer, it is possible to encrypt entire connections, *end-to-end*. It can help preventing eavesdropping, message forgery and tampering when applications communicate across the network. Moreover authentication and privacy can be provided over the Internet with the help of cryptography.

In terms of security, the application layer is very weak as it has many protocols providing vulnerabilities and access points to the hackers. Moreover as the information the hackers want resides inside the application, the application layer is very attractive for attacks. Security in this layer can be provided by security add-ons to the protocols which themselves are basically insecure or with the help of lower layer technologies. Common for the majority of the solutions in the application is that they make use of cryptographic techniques.

In the following each layer is discussed with focus on the security protocols. Notice that the description will not be technical and neither will it be going into technical details of each layer. Instead it will be on a general level where only protocols and techniques related to the security will be discussed. A thorough description of the TCP/IP architecture with all its protocols is beyond the scope of this thesis.

5.1.1 Data Link Layer

The data link layer handles the transfer of data across the physical link. It responds to service requests from the network layer and issues service requests to the physical layer. [77]

Many data link protocols have been developed. In the beginning only a single transmission between two machines was possible. Later on with the growth of the Internet error control and other functions were added. There are two main data link protocols that have widely been used in the Internet, SLIP (RFC 1055) and PPP (RFC 1661).

The SLIP (Serial Line Internet Protocol) is very simple and does not

allow error control nor address control. For this and a number of other reasons, it quickly became obsolete. It does not provide any kind of authentication, and it has never been approved as an Internet Standard.

Besides SLIP, there is the PPP (Point-to-Point Protocol) which is a much more developed protocol and better suited for the transmission over the Internet. PPP is an official Internet Standard and was developed by the IETF solving the problems that existed with SLIP. It is described in RFC 1661 and further elaborated on in several other RFCs, such as RFC 1662-1663. PPP works with several other network layer protocols, here among the Internet Protocol (IP). PPP itself is not of main interest here and therefore it is not described further. The reason mentioning it is the Point-to-Point Tunnelling Protocol (PPTP, RFC 2637) that depends of PPP and also operates in the data link layer.

5.1.1.1 Point-to-Point Tunnelling Protocol (PPTP)

The Point-to-Point Tunnelling Protocol (PPTP) is a protocol that was created by Microsoft and a group of network product vendors. PPTP works in the data link layer and depends on another data link protocol, namely PPP. It allows PPP to be tunnelled across an IP network and the data payload is encrypted providing authentication and confidentiality. In this way it becomes possible to create a *virtual private network* (VPN).

PPTP can be used for connecting remote clients to a Remote Access Server (RAS) on a corporate intranet or connecting a local area network (LAN)-connected computer to another LAN. An example of a remote client connection could be a person who is abroad from his corporate network and wants to connect to it. He will first have to make a PPP connection to a local ISP's dial-up server and then connect to the corporate intranet via PPTP using a VPN connection. In this way the PPP frames will be carried to the corporate PPTP-server through TCP/IP Internet.

The actual data (PPP frames) are carried out by PPP using an extended version of Generic Routing Encapsulation (GRE). PPTP encapsulates the GRE packet which can then be routed over the Internet as any other

IP packet.

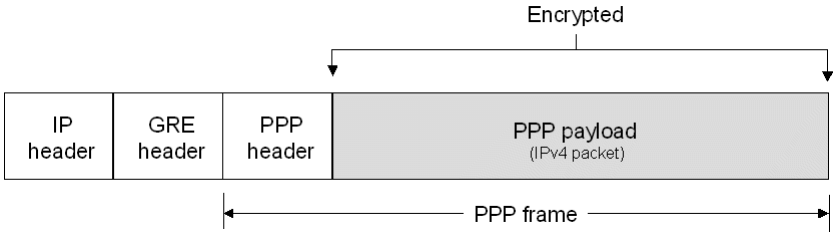

Figure 5.3: PPTP packet structure, from technet.microsoft.com

Specific cryptographic algorithms are not specified by PPTP. Instead it has a framework for negotiation of such algorithms, relying upon the Compression Control Protocol (CCP), the Challenge Handshake Authentication Protocol (CHAP) and the Encryption Control Protocol (ECP) and others.

It is possible to encrypt the PPP payload to obtain confidentiality. It is done using Microsoft Point-to-Point Encryption (MPPE) which is based on the symmetric RC4 encryption algorithm. Authentication is obtained using one or several protocols specified in the PPP. It is done with clear password, hashed password (Windows NT hash function) or Microsoft's version of the CHAP protocol.

All Microsoft Windows operating systems support PPTP. However, it has proven not to be secure and vulnerable against both authentication and confidentiality. Weaknesses in the RC4 encryption algorithm have been found. Actually, the most common authentication protocols have vulnerabilities and also cryptanalysis have shown many vulnerabilities. However, over time Microsoft has launched patches several times. [41]

There are also other reasons as to why PPTP is not a good choice. Eavesdroppers can snap the information as the communication over the control channel is done in clear text. As the information is sent in text, denial-of-Service are also possible to happen. The PPP frames are encrypted to get confidentiality but there is no way to prevent the sent information from being altered.

Generally the data link layer itself is not so practical for implementing data security. Dedicated links are needed between the host and router, and the security (only) holds from point to point. In unknown networks, which often occur over the Internet, this would not be trustworthy at all.

5.1.2 Network Layer

Above the data link layer is the network layer. The role of the network layer is to transmit packets from the source all the way to the destination, where the data link layer only concentrates about delivering from point to point, i.e. from one end of the wire to the other. For this reason the network layer is the lowest layer in which end-to-end transmission is possible. The Internet Protocol (IP) is the far most popular and widely used protocol in the network layer. [77] writes that IP is "the glue that holds the Internet together".

5.1.2.1 IP Security (IPSec)

IP Security (IPSec) is a security protocol which operates at the network layer and gives secure, authenticated, and reliable communication over IP networks. It was developed by the Internet Engineering Task Force (IETF) due to the need of having secure Internet. Specifications for the protocol are found in RFC 2401-2411 and 2451.

IPSec was developed especially for the IP protocol, and works with both IPv4 and IPv6. In the latter, it is a mandatory element. In order to obtain security, IPSec adds two new protocols to the IP, the IP Authentication Header (AH) and the IP Encapsulation Security Payload (ESP). Both protocols add a new header to the IP datagram. They may be used individually or in a combination with each other.

AH provides integrity and authenticates the sender, but does not give confidentiality. It uses message authentication code (MAC), and applies the hash function to the data part of the IP datagram and some of the fields (the non-mutable as they will not be altered in transit) from its

header. The result is stored in a new header which is added to the IP datagram. To be compliant, an AH implementation must support the HMAC hash algorithm with MD5 or SHA-1 algorithms.

ESP provides confidentiality along with integrity and authentication which is optional but highly recommended. The datagram is encrypted before sending it, but unlike AH, the header is not encrypted. A number of encryption algorithms are available such as DES, 3DES, IDEA. As AH is not well compatible with NAT (Network Address Translation), which is a widespread technology most implementations of IPSec use ESP.

IPSec can operate in one of the two modes, the tunnel mode and the traffic mode. Primarily the transport mode protects the upper layer protocols. It is used for end-to-end communication between two hosts. In transport mode only the payload is protected by encryption and optionally authentication. In the case of ESP, the IP header is not touched at all, while AH authenticates the non-mutual parts of the IP header also. The header (of ESP or AH) is inserted rightly after the original IP header, and for ESP its trailer and authentication fields are inserted at the end of the IP datagram. In tunnel mode, the entire packet is protected, i.e. both the payload and the header. A tunnel is created between two networks or hosts, or a network and a host. It creates a new IP datagram which has the old IP datagram inside. The old datagram (both payload and header) is encrypted and/or authenticated using ESP or AH and the ESP/AH fields are placed just as before but before the original IP header, not after. A new IP header is added in the beginning.

IPSec does not depend on specific algorithms. The two parties communicating have to agree on the encryption algorithm to be used. This is called Security Association (SA) and it is an important and fundamental concept in the IPSec architecture. The architecture does not require any specific algorithm but there are few algorithms that it must be able to provide in order to be considered compliant. These are the common encryption algorithms used such as the Data Encryption Standard (DES) with a 56-bit key and Triple DES (3DES) with a 168-bit key. Among the common hash functions are the Message Digest 5 (MD5) and the Secure Hash Algorithm 1 (SHA1).

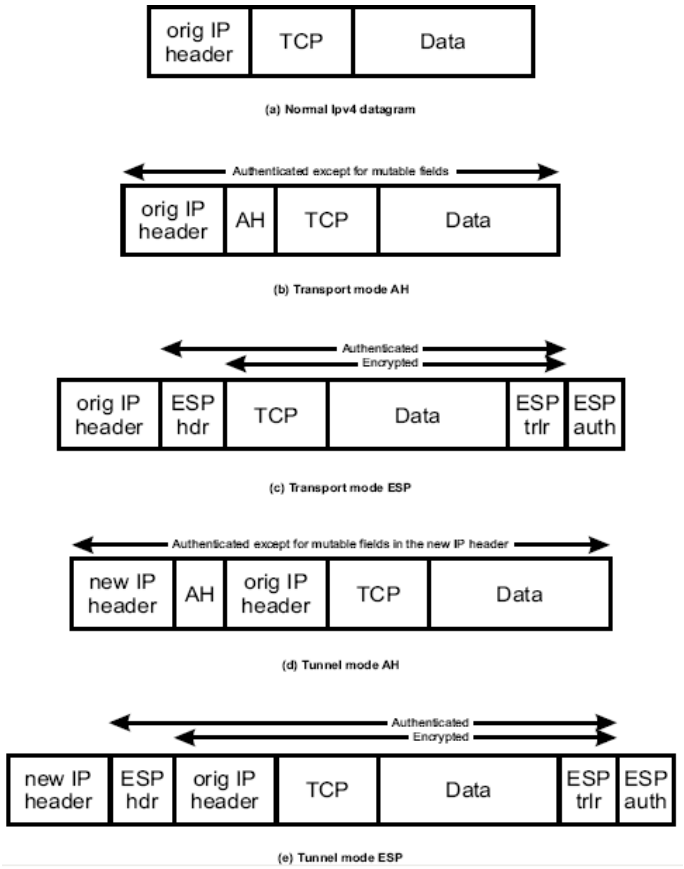

Figure 5.4: AH and ESP datagram formats in transport and tunnelmode, [51]

The SA is a one-way relationship between the communicating parties containing all the necessary information for the secured communication. This includes, for example, which encryption algorithm they have agreed upon, the keys being used, their lifetimes etc. The mode of IPSec and the host and destination addresses are also included in it. Regarding the keys, they should be changed frequently and the same keys should not be used during a whole session. Moreover as the keys are also an essential part of the security there must be a mechanism through which these can be exchanged securely. In small environments the keys can be created and installed manually by the system administrator and distributed on

both entities. But in a large environment some kind of key management protocol is needed that can be agreed upon just as the encryption algorithm. Among the key management protocols is the Internet Key Exchange (IKE, RFC 2409). Another key exchange protocol is Oakley (RFC 2412) which is based on the Diffie-Hellmann algorithm [45]. It is generic and does not set specific protocols. This is done in ISAKMP (Internet Security Association and Key Management Protocol, RFC 2408), which is a framework for Internet key management, providing specific protocol support.

To sum up, transport security is provided by IPSec for all users of IP. It does not require changes in the interface and the upper layer protocol does not need alterations or even know that they are being protected. This kind of transparent security is good as it can be applied to existing applications without making changes. But at the same time, it cannot provide any specific security service as it does not give user-to-user or application-to-application security, but host-to-host only. It should be noticed that the IPSec security services are not bound to any encryption algorithm or key management protocol. This means, if any algorithm or protocol was found to be flawed it could easily be replaced; IPSec would not need to be replaced.

5.1.2.2 Firewalls

Security on the network layer can also be obtained by the use of firewalls. [32] defines a firewall as "a network security device controlling traffic flow between two parts of a network". Firewalls are of different types and they can be used on this layer and the two upper layers also, the transport layer and the application layer. For this reason, instead of discussing firewalls in each subchapter they will be discussed in one place in a subsection right after the description of security in the different layers.

5.1.3 Transport Layer

The transport layer is the heart of the protocol hierarchy. If this layer was not there, the concept of layered protocols would not make much sense. The main task of the protocols in this layer is to provide reliable, and cost-effective data transport from source to destination without dependence of which networks are used.

The two main transport protocols of the Internet are the connection-orientated TCP protocol (Transmission Control Protocol) and the connectionless UDP (User Datagram Protocol). TCP was designed especially for the purpose of providing reliable end-to-end byte streams between two nodes of an unreliable network. Originally TCP was defined in RFC 793 but with time numerous errors and flaws were found. The requirements were then changed and extensions were made. This can be found in RFC 1122 and RFC 1323. Besides these, TCP is discussed in several other RFCs.

Every machine that supports TCP has a TCP transport entity that manages TCP streams and interfaces to the IP layer. The TCP entity takes data streams and sends it as IP datagrams after splitting it up into pieces. When the IP datagrams arrives to the destination they are passed through the TCP entity that reproduces the original data stream. The IP layer does not guarantee proper delivery of the datagrams and they may come in a wrong order. It is therefore the task of the TCP entity to keep track of this and assure proper delivery, detect packet loss, that the order is correct etc. The UDP in contrast to TCP, does not provide error recovery or end-to-end reliable communication. It is very simple and basically just IP with an extra header added.

TCP is used widely by a number of the most popular application protocols on the Internet and the resulting applications, such as WWW, E-mail, FTP and SSH. UDP is used mostly for broadcasting, multicasting and in other real-time multimedia communication where packet loss can be accepted to some degree. TCP is the transport protocol managing individual conversations between web clients and servers.

Although TCP is a very useful protocol, it has some vulnerabilities due

to its implementation. It lacks strong cryptographic authentication, data integrity and confidentiality. For this reason the Secure Socket Layer (SSL) protocol was introduced. As SSL plays a main role in communication security, SSL is described in details in the following part, though without going into the many parameters and properties. This way an understanding of the procedure of SSL is obtained without being too specific.

5.1.3.1 Secure Socket Layer (SSL)

SSL is a transport layer security protocol developed in the 90s by Netscape Communication Corporation. The first version was never released and version 2.0 was launched in 1994 but with several flaws. In 1996 version 3.0 of SSL was designed which was a totally different protocol compared to version 2.0. With this version, the popularity of SSL began for real. Based on SSL 3.0, IETF developed another security protocol, Transport Layer Security (TLS). TLS is a standard, while SSL is not.

TLS and SSL are both cryptographic protocols. Although there are small differences between these protocols, conceptually they provide the same security service, which is a secure channel between the client and the server. The data that is transmitted between both ends is kept secret and tampering is detected (data integrity). In the following SSL is described but as TLS only differs slightly from SSL, it can be taken as a description of both.

Basically an SSL connection is divided into the two parts. First comes the handshake followed by the data transfer. The handshake has three parts.

1. The client and the server agree on the cryptographic algorithms that will be used one the data, and the ones to authenticate each other.
2. They establish the set of the cryptographic keys that will be used for the protection of data.

3. The client is authenticated by the server, and optionally it can authenticate the server also.

When the handshake has passed successfully, the data transfer starts. The data is split up into fragments that are transmitted in series of protected records. Data integrity is provided by computing a message authentication code over every data fragment. Thereafter both the fragment and the message authentication code are encrypted. The mentioned procedure of an SSL connection can be illustrated by the following example.

SSL WWW-example
SSL is mostly used to secure the HyperText Transfer Protocol (HTTP) in order to secure the communication through the WWW. It is called Secure HTTP (HTTPS) or HTTP over SSL and it is supported by the majority of the web-browsers.

When creating a secure connection between a client (e.g. web-browser) and a server through secure HTTP, the client connects to the server. Usually traffic on the HTTP is routed through port 80, but in the case of SSL, the client connects to server through TCP port 443, and the URL scheme https://. Usually most servers automatically redirects from http to https, so that the user does not have to bother with this. When the connection is established, it is shown in the status bar of the browser (e.g. Internet Explorer, Firefox etc.) in form of a little yellow lock. For example, when http://www.campusnet.dtu.dk is entered in a web-browser, the browser redirects the user to a https:// address for the login at the DTU CampusNet. In the status bar the yellow lock is shown indicating that a secure connection over SSL has been established.

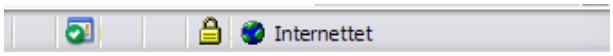

Figure 5.5: The yellow lock indicating the SSL connection

In the status bar the yellow lock is shown and moving the mouse pointer onto it reveals the encryption key bit length.

SSL layer architecture

SSL is a two-layered protocol that works on the top of a reliable transport protocol (e.g. TCP). It does not depend on TCP and it can run under other protocols as well. Though, the security protocols (SSL and TLS) do not work with UDP. A DTLS based on TLS has newly been developed been supporting UDP also. [49]

In the lowest layer of SSL there is a SSL Record Protocol (SRP). In the layer above this there are three higher layer protocols, Handshake Protocol (HP), Change Cipher Spec Protocol (CCSP) and Alert Protocol (AP). SRP provides the basic security services for the higher layer SSL protocols which provide support for SSL session establishment and management.

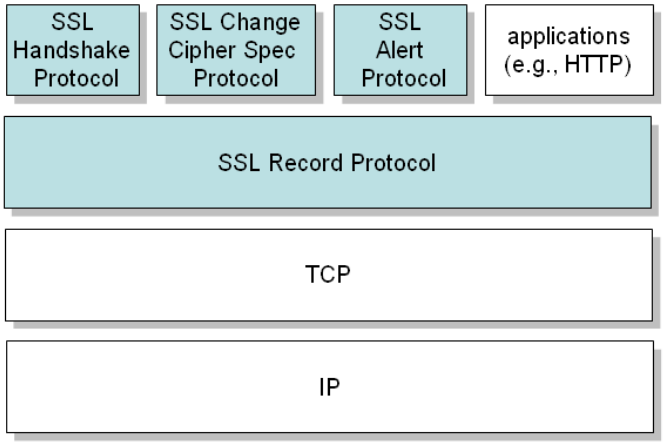

Figure 5.6: SSL architecture, [13]

SSL has two important states, a connection state and a session state. In the specification (RFC 2246) they are defined as:

Connection: A connection is a transport that provides a suitable type of service. For SSL, such connections are peer to peer relationships. The connections are transient. Every connection is associated with one session.

Session: A SSL session is an association between a client and a server.

Sessions are created by the Handshake protocol. Sessions define a set of cryptographic security parameters, which can be shared among multiple connections. Sessions are used to avoid the expensive negotiation of new security parameters for each connection.

The SSL Record Protocol

This protocol mainly provides two services for the SSL connections, confidentiality and message integrity. By encrypting the data, it provides confidentiality and message integrity is obtained using message authentication. Besides these two other keywords are fragmentation and compression.

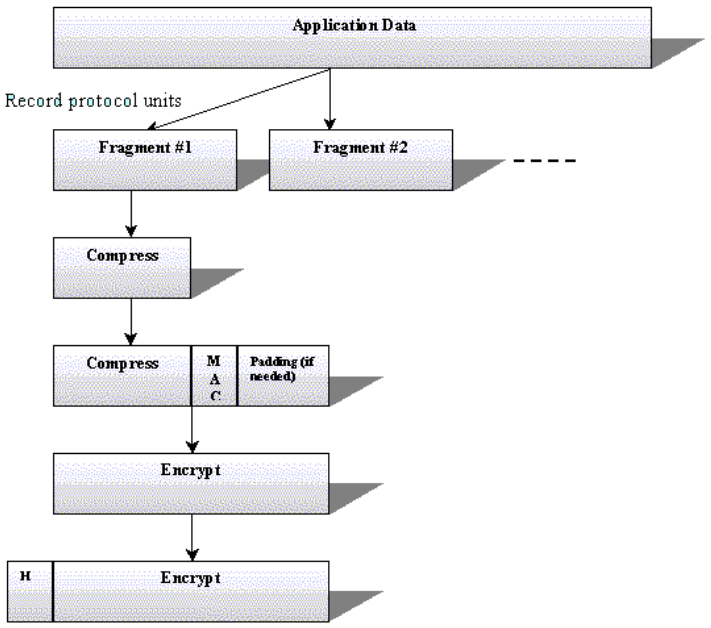

Figure 5.7: Record data process flowchart, [63]

When a message is to be transmitted, this protocol takes the message and fragments it into blocks of a certain size (2^{14} bytes or less). Optionally compression can be done afterwards, but as default it is null, i.e. no compression algorithm is used. The compression must not lose data or have a length exceeding $2^{14} + 1024$ bytes. After this, a MAC is calculated for the data using a secret key shared between both parts. The MAC is

not computed over the data only but a combination of the message, the secret key and optionally some padding. The receiver knows the secret key and computes the MAC in the same way as done by the sender. If the MAC he has received and the one he calculated are equal, the message has not been modified in transit. As an attacker (probably) does not know the secret key he will not be able to compute a MAC if he changes data. SSL supports both MD5 (128-bit hash) and SHA-1 (160-bit hash) hash algorithms.

The fragments together with their MACs are then encrypted using symmetric encryption. A number of encryptions are available having key sizes from 40 to 168 bits. Among these are DES, 3DES, IDEA and RC4. Finally a header is added including content type, versions and compressed length.

The SSL Change Cipher Spec Protocol
This protocol is used to indicate the end of the SSL handshake. It consists of a single byte (8 bit) with the content '1'. Receiving this message, the parties move on from the pending state to their current state. They will stop using the key-exchange algorithm and go on to the encryption and message authentication code algorithms defined in the handshake phase.

The SSL Alert Protocol
The task of this protocol is to alert error messages that may arise during the whole session. The error alerts are of two levels, fatal alerts and warnings. If a fatal error occurs the connection is cancelled immediately. The session ID is invalidated meaning that no new connection can be established within this session.

The alert message consists of two bytes. The first byte indicates the level, i.e. whether it is a fatal alert or a warning. The second byte holds the specific alert. Fatal alerts could be errors in the negotiation of security parameters, or inconsistency within fields of the Handshake protocol, while a warning could be that the certificate has expired.

The SSL Handshake Protocol
This protocol is the most complex part of SSL. It is responsible for creating a secure session between the client and the server. It can be divided

into several stages:

1. the authentication of client and server to each other (client authentication is optional)
2. the negotiation of common cryptographic algorithms that both support
3. the exchange of cryptography parameters using public key encryption
4. the establishment of an encrypted SSL connection

The handshake is done before transmitting application data. It consists of exchanging a series of messages between the client and the server which is shown in Figure 5.8.

As indicated, the handshake happens over four phases. For each of the four phases a description is shortly given in the figure above. A detailed description of the parameters can be found of the SSL specifications.

5.1.3.2 Advantages of transport layer security

Layering security protocols on the TCP, security to higher level protocols is provided. An advantage of these security protocols is that they can easily be used underneath various communication protocols including but not restricted to HTTP. Hence the applications do not have to carry out the encryption and/or authentication. As SSL or TLS is added just below the application, only minor changes must be done in the application, which is manageable. As these protocols only provide a secure channel, non-repudiation is not provided. When the transmitted data leaves the secure channel at the receiving side, the cryptographic protection is removed. There is no digital signature on client data. Therefore electronic information systems should implement a non-repudiation mechanism on top of the secure channel.

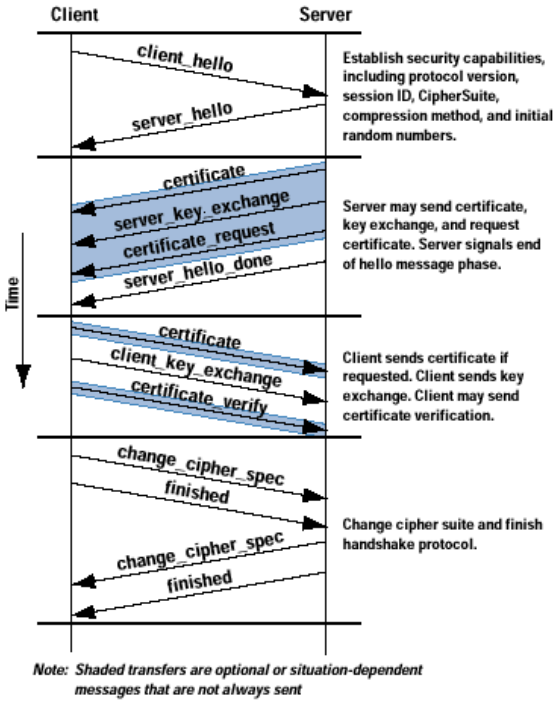

Figure 5.8: Handshake Protocol, [72]

The server is always authenticated and the client can be required to authenticate too. Not all electronic information systems do rely on the client authentication feature of the secure channel, but instead many implement a client authentication mechanism on top of this channel.

5.1.4 Application Layer

The application layer is the top layer of TCP/IP and it has a lot of protocols. The most well-known protocols are:

Protocol	Description
DNS: Domain Name Service	mapping of IP addresses to Internet domain names
FTP: File Transfer Protocol	download and upload of files
HTTP: Hyper Text Transfer Protocol	the WWW protocol for text, pictures, sound etc.
POP3: Post Office Protocol	picking up e-mail
RIP: Routing Information Protocol	exchange of routing information by network devices
SMTP: Simple Mail Transfer Protocol	sending electronic mails
Telnet: Network Terminal Protocol	text communication for remote login

Some of the protocols, such as FTP and Telnet are directly used by both users and applications, while other protocols like HTTP and SMTP are behind applications. Most of the protocols in the application layer are implemented as 7-bit ASCII (text) protocols. The text communication aspect has some benefits especially when testing and troubleshooting TCP/IP applications. But at the same time there are also drawbacks. It is very easy to eavesdrop on the conversation or to write programs that generate correct text messages in order to impersonate others.

Like the lower layer protocols, the application layer has its own security challenges. As there are many protocols in the application layer providing numerous vulnerabilities and access points for attackers, it is a weak link in terms of security. Moreover the application layer is of special interest for the attackers and a natural place to address their attacks as the information in which they are interested is in the application.

Two of the main categories of risk at the application layer are web security and email security.
For web security a balance between security and accessibility is necessary. Besides this, information privacy is important along with prevention of virus, worms etc.
For emails, either some kind of web-mail is used, or applications using the email protocols SMTP, POP and IMAP are used.

Pretty Good Privacy (PGP)
Pretty Good Privacy (PGP, [20]) was written by Phil Zimmermann and released in 1991. It is very popular and used worldwide for protecting emails and files. PGP is based on cryptographic techniques and supports secrecy, digital signatures, key management, and data compression. It does not invent new cryptographic algorithms but relies upon existing such as RSA, IDEA and MD5.

PGP is a hybrid encryption system. RSA is used for key management and IDEA for the bulk of encryption of data. It provides confidentiality through IDEA, integrity through the use MD5, authentication using public key certificates and non-repudiation using cryptographically signed messages.

PGP is available both as a standalone application and an integrated email program, or plug-in for email system. It provides end-to-end security for the sender and receiver.

One of the problems with PGP is the key management. A key never expires and the key-holder must distribute key revocation certificates if the key is compromised. If the contacts do not know this they can come to use the compromised key for a long time. Moreover, as a key never expires, the secret key for every public key distributed, must be kept for "all time".

The validation of the public keys in PGP happens by a web a trust. The users certify each others keys, believing that the keys are original. Moreover they can put their trust in individuals vouching for the authenticity and by signing their keys. For small communities such a trust model works, but for larger ones, it becomes problematic. It can be difficult to exchange keys among each other. They can be uploaded to servers on the Internet, which will just store the keys and send copies to those who want. However, most people choose to put their public key on a personal web page.

S/MIME
The Internet standard for sending files with binary attachment is MIME (Multipurpose Internet Mail Extensions, RFC 2405). S/MIME (RFC

2633) is an extension to MIME designed to add security services to the emails. The security services it offers are message confidentiality and integrity, authentication and non-repudiation. It provides confidentiality through user-specified encryption algorithms, integrity through user-specified cryptographic hash functions, authentication through public key certificates, and non-repudiation through cryptographically signed messages.

S-HTTP
Secure HTTP (S-HTTP, RFC 2660) was designed to provide security for web-based applications running over the HTTP protocol. It provides mechanisms for confidentiality, integrity and authentication. It is not tied to any particular cryptographic system, format or key infrastructure. The type of encryption being used is negotiated with the client. Moreover, it does not require the client to have public key certificates as it use symmetric keys. The keys are exchanged in advance through another secure channel or out of band connection.

S-HTTP, which is not to be confused with HTTPS, was developed before SSL and is not used so much and according to [31] the popular browsers, Netscape and Microsoft, failed to implement it in their browsers.

SSH
Secure Shell (SSH, RFC 4251) is a protocol for allowing secure remote login and data transfer over an insecure channel. It was designed to provide the missing security in protocols as Telnet, rlogin, rsh and other insecure remote shells, which send data (e.g. passwords) as plaintext. It offers confidentiality through encryption, authentication using public key cryptography. Mostly it is used for remote logins, but can also be used for tunnelling arbitrary TCP applications.

The use of SSH through SSH software is very widespread and especially universities use it a lot to offer remote login for their students. It is available for free for all operating systems.

Besides these, lower layer technologies can support the application layer in providing security. It has been seen how security can be provided to some of these application layer protocols using the lower layer technolo-

gies such as IPsec or securing HTTP through SSL.

5.1.4.1 Advantages of application layer security

The biggest advantage of placing security at the application layer is that applications can be extended without involving the operating system. Moreover the application can understand the data and can provide appropriate security. Application layer security is implemented in the end-hosts. Complete coverage of security requirements, such as confidentiality, integrity and non-repudiation, can be offered at this layer.

At the same time, it may be a drawback that security mechanisms have to be designed independently of each application. As mentioned earlier the security challenges cannot just be solved from a single place or layer, rather it is a result of interaction between security services in several layers. Practically all applications do use application security but at the same time they may use security services from lower layers such as SSL in the transport layer and firewalls placed at the network layer.

5.2 Firewalls

The main purpose of a firewall is to control traffic, whether to allow or deny it. Using firewalls is often the main protection against Internet attacks.

Basically a firewall is a system preventing unauthorized access to a specific network or parts of it. Firewalls are mostly used for preventing unauthorized Internet users from accessing a company's private networks that are connected to the Internet. A firewall is often used as a first line of defense in protecting private information from unauthorized access. Firewalls can also be used to control the type and amount of the traffic passing from one network segment to another. The firewall itself must be immune, so that it does not become the weakest link in the chain.

As mentioned earlier firewalls can operate on the three upper-layers in the TCP/IP protocol stack. There are 4 types of firewalls, [6] [17]. In the following each of the types will be touched briefly. The upsides and the downsides of each will be discussed as well.

5.2.1 Packet filtering firewalls

The first firewalls from 1988 were developed by engineers from Digital Equipment Corporation (DEC), [6]. Packet filters act at the network layer by inspecting fields in the packet headers. A set of specified rules are used to validate the packets and if a packet comply with these rules, it gets permission to become transferred. If it fails complying with these rules it gets discarded. This kind of firewalls is easy to setup and maintain, and the data flow through the firewall is pretty high even on elder platforms/hardware. All Internet communication always occurs in packets.
A packet consists of a series of information [17] [26]:

- The data, acknowledgment, request or command from the originating system
- The source IP address and port
- The destination IP address and port
- Information about the protocol (set of rules) by which the packet is to be handled
- Error checking information
- Usually, some sort of information about the type and status of the data being sent
- Often, a few other things too - which don't matter here.

Packet filtering only examines the header information of each of the packets. It does not examine the contents or the context, and as this type

of firewalls only work on the third layer (network layer) malicious code occur on other layers without being detected. A downside of the packet filtering firewall is that spoofing of source addresses can cause that non-wanted data, that should have been discarded, gets permission.

Another issue is that the attacker can specify the route the packets must follow before reaching the firewall, and as the last address can be a valid address, the non-wanted data find its way through the firewall.

5.2.2 Circuit-level firewalls

Circuit-level firewall [17] is a firewall approach operating on the transport layer that validates connections before allowing data to be exchanged. This means that the firewall does not just allow or prevent packets but also determines if the connection between both ends is valid according to a set of rules or not, and then opens a session and permits traffic only from the allowed source. Whether a connection is valid may for examples be based upon, [17]:

- destination IP address and/or port
- source IP address and/or port
- time of day
- protocol
- user
- password

Every single session of data exchange is validated and monitored and all traffic is prohibited unless a session is open. This way the circuit level firewall takes control a step further than a packet filter firewall.

Among the advantages of a circuit relay is that it can make up for the shortcomings of the ultra-simple and exploitable UDP protocol, wherein

the source address is never validated as a function of the protocol. IP spoofing can be rendered much more difficult.

A disadvantage is that circuit level filtering operates at the transport layer and may require substantial modification of the programming which normally provides transport functions (e.g. Winsock). Also the performance is lower than the packet filter firewalls as the amount of computations is bigger in the circuit level firewall. As it was the problem with packet filters, malicious code can occur on the higher layers (i.e. the application layer) without being detected. [32] writes that this type of firewall is rarely used in practise anymore.

5.2.3 Application-level firewalls

In this approach, the firewall goes still further in its regulation of traffic. The application level firewall [17] operates at the top layer of the TCP/IP protcol stack, i.e. the application layer.

An application-level firewall acts as a proxy for applications, performing all data exchanges with the remote system in their behalf. This can render a computer behind the firewall all but invisible to the remote system. This proxy is the only entity the outside world sees while it is transparent for the internal users except for filtering.

It can allow or disallow traffic according to very specific rules, for instance permitting some commands to a server but not others, limiting file access to certain types, varying rules according to authenticated users and so forth. This type of firewall may also perform very detailed logging of traffic and monitoring of events on the host system, and can often be instructed to sound alarms or notify an operator under defined conditions.

Application-level gateways are generally regarded as the most secure type of firewall. They certainly have the most sophisticated capabilities. A disadvantage is that setup may be very complex, requiring detailed attention to the individual applications that use the gateway. An application gateway is normally implemented on a separate computer on the network whose primary function is to provide proxy service.

[32] compares packet filters with telephone call barrying by number while application-level proxies monitor telephone calls by listening to the conversation.

5.2.4 Stateful packet inspection firewalls

Stateful inspection [59] is more secure than other firewall technologies such as packet filtering as it opens small "holes" through which traffic can pass. For example, instead of permitting any host or program to send any kind of TCP traffic on port 80, a stateful inspection firewall ensures that packets belong to an existing session. Furthermore, it can authenticate the user when the session is established, determine whether the packets really carry HTTP, and enforce granular constraints at the application layer (e.g. filtering URLs to deny access to black-listed sites).

Stateful inspection is an advanced firewall architecture that was invented by Check Point Software technologies in the early 1990s. It is also known as dynamic packet filtering, and has replaced static packet filtering as the industry standard firewall solution for networks. This kind of firewall is also called multi-layer firewall as it simultaneously operates at the network, transport and application layer.

Stateful inspection provides enhanced security by keeping track of communications packets over a period of time. Both incoming and outgoing packets are examined. Outgoing packets that request specific types of incoming packets are tracked; only those incoming packets constituting a proper response are allowed through the firewall. In contrast to static packet filtering, in which only the headers of packets are checked, stateful inspection analyzes packets uo to the application layer. In a firewall that uses stateful inspection, the network administrator can set the parameters to meet specific needs. In a typical network connected to the Internet, ports are normally closed unless an incoming packet requests connection to a specific port and then only that port is opened to the packet. This prevents port scanning, a well-known technique used by hackers to gain entry to networks and individual computers connected to the Internet.

5.2.5 Personal firewalls

The mentioned firewalls are used to protect the system from outside threats. Although this gives a good protection, it can be necessary to protect the personal computer also. It may be that the outer firewall fails or the attack may come from machines internally connected behind the same outer firewall. For this purpose is the personal firewall software that is intended to be installed on personal computers. If the firewall is configured well it can act as a defense line very close to the PC, just like a lock on a door. For trained users it may be an advantage as they in most cases cannot configure the hardware firewall. A huge number of software firewalls exist and can be configured to give a very high protection. Though, the same may be a drawback for the average user, as the configuration may require too much knowledge.

Most of the personal firewalls react on both internal and external attempts of opening a door. If the user runs a program that attempts to a open a door, the firewall will ask the user to approve or deny this. In the same way, it asks the user whenever some external visitor wants to get access. The user can approve programs once for all, in order not to manually take a decision every time. Sometimes this is dangerous as the user just allows access to programs without considering them in order to get rid of the alert boxes.

Some of the famous firewalls providers are Norton, McAfee and ZoneAlarm, but as mentioned there are thousands of personal firewalls, including freeware. Moreover, the operating systems (e.g. Windows) also come with built-in (simple) firewall solutions that can be used.

There are many differences between a software and a hardware firewall. Though, it should not be regarded as a either this or that. Instead they both can and should be applied simultaneously to get the best protection.

5.3 VPN

Virtual private networks (VPN) are private links made across shared or public network infrastructure (e.g. Internet) using encryption and tunnelling techniques. The idea is to create a dedicated private link on a shared or public network that enables data to be sent between to computers in a way that resembles the properties of a point-to-point private link. This is done by encapsulation of the data with a header providing routing information with allows it to go across the public network to its end destination. The data is encrypted for confidentiality, so packets that may be intercepted cannot be read. A VPN connection is the link in which the data is encapsulated and encrypted.

There are many advantages of VPN connections. They allow users to make remote connections to internal company networks. This makes it possible for users to work from home or other external places. For the user the connection seems like a point-to-point connection between his computer (client) and the remotely connected computer (server). Organizations having servers placed geographically different locations can communicate using VPN connections as if they were connected locally.

VPNs are normally split into two distinct categories, [82]:

- Site to site VPNs - between two or more offices or datacentres
- Client to site VPNs - between a desktop client and a central office or datacentre

Client-to-site VPN is also called remote access VPN. To set up a remote access VPN, a VPN server and a VPN client is needed. The VPN client is mostly software, but can also be implemented in the hardware.

VPN solutions can be implemented using several different technologies. Among the protocols VPN can be based on are PPTP, L2TP, IPSec and SSL. These have already being discussed in the previous sections. Mostly VPN is used as either IPSec VPN operating at the network layer or SSL

VPN operation at the application layer. Both have their advantages and disadvantages.

5.3.1 IPSec VPN versus SSL VPN

Traditional IPSec VPN (IV) solutions requires all users to have client VPN software installed on their computer. The client software can have many advantages for the company providing it, such as verifying anti-virus software, proper firewall settings etc. Though, today when the users need remote access from all kind of devices, such as PDA, mobile phones it may give a compatibility problem. Moreover often users use clients in kiosks, net cafes, libraries where they may not be allowed to install software. The solution for this is the SSL VPN (SV) which is much more flexible.

SV solutions are the security solutions for web-based traffic. IV encapsulates all IP packets without differentiating between the higher-level protocols (such as HTTP, FTP etc.). It supports all IP applications. In contrast to this, SV only support web-applications using the HTTP protocol. It has both advantages and disadvantages.

A big advantage of SV over IV is that no client VPN software is needed on the client's computer. SV uses the web browser and it is pre-built into most web browsers. Hence, SV is known as clientless solution.

When deploying IV, users must learn how to use the software. This may also require a lot of user training and support. As there are not any client software to install with SV, the support and troubleshooting problems are much more limited then for IV. The issues of installing the software on client computer, sending updates and patches, licensing etc., do not arise for SV. Moreover as most users are familiar with web browsing, user training is a lesser issues. Though, for advanced ActiveX controls and applets, user training may also be relevant for SV.

When discussing whether it should be IV or SV, the company therefore must consider the user community (e.g. where they need access from) and the client types (e.g. PDA, mobile phones).

SV connects remote users to specific applications and network resources inside trusted networks, whereas IV connects remote devices to trusted networks. For IV this poses a risk because all users can potentially access sensitive resources. SV eliminates that vulnerability by the restriction to a specific application or resource.

As SV uses the web browser, the remote application must be browser based. Sometimes the application may run as a Java applet or made accessible through Active-X plug-ins. Though, this is not always possible. The company using SV must be forced to develop ActiveX or Java based software etc., and this is done, often on pr. application basis. Moreover, this would often require the client (web browser) to open up for its browser's security settings to support the download of ActiveX controls and Java applets. Both of these have documented vulnerabilities.

A problem with SV is that the user may leave behind cookies and data that were cached during a browsing session. As the client is not trusted, the user is left vulnerable to keyboard utilities, Trojan horses, worms. Among the advantages of IV is that the client software can be programmed to carry out certain tasks, for example scan the computer for virus, verify the security settings on the client. This is not possible, in the same way, for SV. It may provide some of the features by forcing the browser to verify anti-virus software etc., through for example Java applets, but the possibilities of real-time applications (client software) are much bigger.

SV operates at the application layer, whereas IV works at the network layer. IV can enforce policies at the network layer, where SV enforces policies at application level. For IV, this for example allows control by IP address while it for SV may make it possible to control access more granularly for the specific application.

Mostly SV uses the digital certificates for server authentication, and other methods can used for client authentication also.

One of the main reasons for preferring SV over IV is the simplicity and clientless nature of SV. But the need for having ActiveX, Java applets and other Win32 controls and the need for configuring to maintain application

computability, may contradict the clientlessness and simplicity.

5.3.2 IPSec or SSL

As seen, both SSL VPN and IPSec VPN do have benefits and drawbacks. It cannot be said that one of them is the ultimate choice. In some cases SV is the best choice, while IV is chosen at other times, depending on a number of factors at discussed. Some implement both possibilities.

[79] concludes the following in the question of best choice:

> IPSec and SSL VPNs are effective and efficient means of providing network access to remote users and from site-tosite. Each excels in different scenarios, and many complex organizations could see reason to use both. One is not, therefore, "better" than the other; they are each ideal for what they do best. Because both methods secure data in transit, a key consideration is how to provide effective authentication ensuring that the user is who they say that they are.

[61] also points out that neither is absolutely "better".

> Whether an IPSec or SSL VPN is the right choice ultimately depends on the extent of your company's secure remote access needs:
>
> - IPSec VPN technology is designed for site-to-site VPNs or for remote access from a small finite number of tightly-controlled corporate assets. If these are the primary needs of your company, IPSec performs these functions quite well.
> - SSL VPN technology, on the other hand, works much better for secure remote access. SSL VPN technology is an ideal replacement for-or adjunct to-IPSec, because

> it increases productivity by allowing access to more resources from more end points; lowers costs by easing administration with clientless (and easy-as-clientless) access and centralized control; and adds security with granular access and end point control. Best practices for transitioning to an SSL VPN include establishing a corporate security policy, conducting a lab environment pilot and implementing a phased migration.

Many seems to prefer SSL VPN. [60] concludes:

> Most major analysts have voiced cautionary opinions about SSL VPNs in the past. However, if the precautions listed above are observed, SSL VPNs are now ready for prime time. In actual implementation projects, between 90% and 95% of an organization's remote users only require web access and e-mail. The percentage of users whose needs are met is even higher if some "standard" applications (as defined by the enterprise) are supported through plug-ins. There will be a continued need for IPSEC-based remote access VPNs in order to tunnel communication to unsupported applications, to serve clients on Macintosh and Linux, and for site-to-site VPNs. However, SSL VPNs offer near-equivalent functionality for most enterprise remote access users. By offering SSL as the default option to all remote users, and providing IPSEC VPN clients only to the few users who need non-standard application support, the enterprise can reduce the complexity of the overall remote access infrastructure, while enabling access from more places, including airport Internet kiosks and web-enabled wireless Personal Digital Assistants (PDAs). Major IPSEC VPN vendors are starting to offer integrated SSL VPN functionality in their existing products at little or no initial cost to the customer. In the future, all mainstream VPN systems will offer both IPSEC and SSL functionality, making the issue of "choice" moot.

[37] also prefers VPN based on SSL over IPSec. According to [56] there

is no discussion. SSL VPN is the ultimate choice in the future:

> In the end, you can see the numbers speak for themselves. An SSL VPN solution quickly pays for itself and goes on to reap significant security and productivity benefits. If you all the way with an SSL VPN-extending it not only to a broad group of mobile and home users but to your business partners as well, you get maximum ROI benefits. SSL VPNs are expected to become the primary access method for the majority of business employees who work remotely. Clearly, organizations are catching on to the cost-savings, productivity, and security gains to be had through switching to an SSL VPN.

Some considers IPSec as the best choice. [39] writes:

> Organizations and companies may choose whichever VPN secure protocol they desire, but it is evident that IPSec has more to offer an organization seeking secure data communications with employees, telecommuters, and regional businesses than SSL.

But even [39] does also mean that both have advantages and one cannot be ultimately deemed the best one.

> They both have their strengths and weaknesses, which are considered when a decision between the two is necessary. It is possible to use both IPSec and SSL, but the cost factor of using them together usually put organizations in favor of one over the other. IPSec is more useful to users which seek to use all applications and resources remotely as if they were physically connected to the organizations LAN. SSL is useful to users who need mobile access to applications like E-mail and file sharing, users who are not going to be physically at one PC location, and will be accessing the organizations network via PDA's, laptops, and cell phones. Both of these secure

> protocols can be very useful in their own way and provide organizations with the security they need. Many factors are to be considered when deciding which secure protocol to use and with the information provided, organizations can make that decision easier and with more confidence.

Common for all the conclusions is that both IPSec and SSL VPN have their strengths and the choice depends on the usage. As ActiveX components, Java applets etc. are developed so well-featured, SSL VPN is sufficient in most cases because of the clientlessess and the mobility. Moreover there is no reason to give access to the whole system, in the cases where it is only necessary to access one application. IPSec may be a better solution for those who want to use all applications and resources remotely as if they were physically connected to the remote organization. As [60] mentioned major vendors are starting to offer integrated IPSec and SSL VPN solution and that all mainstream VPN systems will offer both IPSEC and SSL functionality in the future. Hence, instead of choosing one or another, both should be implemented side by side.

Chapter 6

User authentication

No matter how secure a system may be, at the end of the day the security depends significantly on the users of the system as they have to be authenticated by the system. There are users who are well-aware of the threats and know which precautions must be taken when accessing the system. However, this group only constitutes a small percentage of the total users. For instance, the users of the electronic patient system may be all the citizens of a country, among which, many may not even know how to operate a computer correctly.

The security staff designing the system can lock all the doors for entrance to the system but at least one must be left open so that the users can enter the system. This is often the sorest spot of any system and a place where the security designers face a dilemma. On the one hand, they must do everything to strengthen the login security, but on the other hand they must bear in mind that the login procedure should not be too complicated as this will make it too user-unfriendly. In this chapter the problems related to user authentication are outlined. Different methods will be discussed with particular attention to the strengths and weaknesses of each.

6.1 Ways of authentication

When the user wants access to a system, he is most often asked to identify himself. In any system, the user will have some kind of unique identification key. For electronic systems this is normally called a username or a user ID. The process of verifying the claimed identity of a user is called *authentication*, [55]. That is, the user proves that he really is the one claimed to be. There are a number of ways in which the authentication can be performed. Figure 6.1 shows some of the common ways of authentication and how they are related to the human body.

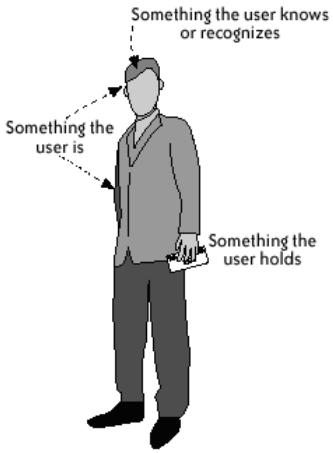

Figure 6.1: The figure shows how the common authentication forms are related to the human body, [22]

[9] lists the following four ways of authentication:

- What the user knows
- What the user has
- What the user is
- Where the user is

Besides the above mentioned, [32] adds a fifth approach which is about *"what the user does"*.

The four or five ways do not contrast or mutually exclude each other, which means that they can appear side by side, and most often, a combination of at least two of them is implemented. A process where several ways are combined is called an *N-factor* or *multi-factor authentication* where N indicates the number of (different) factors involved.

6.1.1 What the user knows

The simplest example of an authentication based on the user's knowledge is the traditional password login, which is also the most widespread. The user enters his password and the system will validate if it matches the correct password belonging to that user ID.
The idea of authentication based on knowledge has been used for a long time. Well-known examples are when making telephone queries to for instance the library in order to renew or order some books, the doctor to book a time, the bank to ask a question about one's account etc. The person calling may be asked to tell his birth date, home address, and other personal information.

Perhaps when renewing books from a library this kind of security would suffice but today when users operate systems dealing with more and more private and sensitive information, this mode of authentication is far from being sufficient. Of course a user defined password only known to the user, is safer than supplying public information such as one's home address which anyone can find out without many efforts. However user-defined passwords also have severe problems. Every possessor of the correct password to some user ID can impersonate the real user and log into the system. This is, for instances the case in the most popular e-mail systems like Hotmail or Yahoo where password is the only way of authentication. Most often there may not even be a way of determining if it was the real user who logged in or if it was some intruder holding the password.

Today tools are available online on a large scale for breaking password

strings, hence cracking passwords is no longer considered an impossible venture. However, this does not mean that static username and password check is not used any longer. At the same, the static password has become the integral part of almost every secure system - combined with other factors. Consequently, it will be necessary to examine the essential issues linked to the use of passwords.

6.1.1.1 Cracking passwords

There are several ways of cracking a password. The simplest way is to start from one end and try all combinations. This attack type is called *brute force* or *exhaustive search*. The probability of guessing the right code is 1, since the right password *will* be hit, sooner or later, as all possible combinations are being tried. In practise, passwords have a minimum length and they are composed of non alpha-letters, digits, and both upper and lower case as well. In this way, the time it will take to break the password increases. Another technique for preventing brute force attacks could be to make some arrangement so that passwords cannot be entered too many times in a row, and put some delay or even block the user if it happens. This kind of solutions will be discussed in a later section in this chapter.

In a brute force attack, every single combination of the letters is tried. This can easily turn out to be a cumbersome task. There is another attack method known as *the dictionary attack*. In a dictionary attack not all combinations are tried out, instead a list of meaningful words (e.g. from a dictionary) are picked out and tried repeatedly, by trial and error. This attack method cannot guarantee that the right password is found but as the majority of the users often choose meaningful words as their passwords, there is a good probability of finding the password in this way.

Besides these, many other techniques of cracking passwords exist. Often users do not change the system default passwords assigned at the user creation. Lists of default passwords for almost every system can be found on the Internet today. A hacker might try all users of a certain system with the default password. Unfortunately, there will always be a little

percentage of systems for which he will succeed in this way.
Password guessing can never be disabled. As long as the user can log on an intruder will also have the opportunity to guess the password. As seen in the case of brute force attack the probability of breaking in is 100%. Thus it is of crucial for the security how the password validation in the authentication system is realized. If there is a system where users can have a password of, for example, 4 digits and there is no logging of how many times a user tries to log in, successfully or invalidly, using brute force attack, the password can be cracked in a matter of few minutes. Merely, if the same system restricts the login session to at most three invalid login tries, the possibility of brute force is ruled out. Instead another problem will arise, namely that the hacker can exploit this to block all the users' accounts. This is called the *denial of service (DoS) attack*. This also shows, that solving one problem, often give rise to many other.

Below some central problems related to the use of password will be discussed, which will throw more light on the threats and possible solutions. Best-practise guidelines for choosing password are given.

6.1.1.2 Choosing password

Password-based authentication is the most used authentication today. It is not possible to make a password that cannot be guessed. However choosing the password in a smart way, the possibility can be decreased to such an extent that it is practically impossible to crack it.

The simplest techniques a password hacker would follow, exhaustive search (brute force) or intelligent search (dictionary attack), have already mentioned. As seen the brute force attack tries all combinations. Therefore, a long password decreases the chance of completing a brute force attack successfully. To avoid dictionary attacks, passwords should not be allowed to contain meaningful words, at least the common ones. Making the password case sensitive, (i.e. using both upper and lower case symbols) will impede both types of the attacks a lot. It will slow down the dictionary attack because every word must be tried in all combinations of upper and case lower.

As brute force attack systems are given a letter range for which they try a password, passwords should not only contain alphanumerical letters but also other chars types. This together with the case sensitivity, widens the range to a degree where the this attack can be deemed impossible.

To help the user, the system administrator could provide a password strength calculator. Many systems have random password generation features for creating strong passwords. Some random generators or individual persons let their passwords follow certain patterns. For example, to have a password of two chars followed by two digits and then two chars again etc. Whenever they are to change the password, they just replace one of the digits and letters. This should also be avoided as the hacker might find one or two of the previous passwords, and figure out the pattern. Then he can easily set up a brute force attack for this pattern.

The above suggests that a password should be long, non-meaningful, contain mixed case letters and have both alphanumerical and non-alpha chars. However, a number of limitations arise due to the human memory as humans are required to memorize the passwords. The stronger the password is, the more difficult is it for the user to remember it. Many users do not like random password generation as they want to define the password themselves, so that it is easier for them to remember. In systems where passwords are generated randomly, users tend to quickly change the random password into something they can remember. Hence, it is not reasonable for the administrator to lean against supplying random passwords.

As long as the password simple and memorisable, the user will keep it in mind, but if it becomes too hard to remember, he will be compelled to write it down Thus, a new way of password disclosure opens. Systems the user does not access frequently, he would probably write down the password even if it is easy due to long-term memory problems.

It shows that password selection is a trade-off between what is easy to remember but also easy to crack, and what is difficult to crack but also difficult to remember.

The above guidelines are not totally sufficient for securing the authentica-

tion thus other safety precautions must be taken from the administrator side.

As all attacks are based on repeatedly trying to guess the password, arrangements must be made to prevent passwords from being entered too many times within a given time interval. A brute force attack can run through thousands of passwords within a minute. If this can be prevented, it will slow down the performing time of such attacks incredible. Of course a real user cannot or will not try more than two-three passwords in a minute. If it is assumed that a system can only validate three passwords in a minute, it sums up to only 4320 a day. This is too slow for any brute force system especially in a system where the password is often changed.

Thus the system could set up to allow x login tries only. [32] mentions that the account should be locked completely after x wrong tries. This cannot be recommend, as it will give rise to the DoS-attacks, since the attackers will then enjoy locking valid users' accounts just to cause annoyance to the users. Of course, the password would not be compromised, but it will give other problems. Therefore, the better way to insert some idle time between the tries as in the example above. It will slow down the attack enormously for the hacker without bothering the user too much. Some systems use a combination of these methods locking the account for maybe twenty minutes or an hour after x failed tries.

Although a password is secure, with time it may be revealed or some attacker may carry out an attack over a long time span and finally hit the right password. Assume the brute force attacker is not deterred away by the waiting time inserted, and launches an attack knowing that it may complete within a year or two. Thus a password may eventually come in wrong hands especially if the same password is used for too long time. Even if the brute force attack is not there, having the same password over long is not without risk. Therefore, the users should be asked to change their password periodically.

This is a bothersome task as the user has to remember new passwords all the time. In some cases it may even be absurd, for instance if an authentication system with password aging asks the user to change the

password after three months, but the user visits the system only twice a year. Moreover when password aging is applied, other problems arise, such as how to ask the user to change the password. Users can be forced to make this change when they login the first time after the password has expired, but suppose that a user does not login for a year. The security designers may rest assured thinking that the password aging is enabled and therefore not paying attention to this attack type although the user's password is not changed at all. The system could be made to disable the account if the password is not changed, until the user changes it. But in this, how will the user then login himself to change the password? Hence, security cannot rely solely on password aging.

In systems where the password expires, the user should not be allowed to change "back" to a previously selected password. This is hard to prevent without logging all the previous passwords. Most system therefore put an upper limit, x, of the number of the previous passwords that are remembered. The users can, however, easily change the password x+1 times to be able to have the same password again. A medical doctor informed, that their system had password aging of three months, and the previous 11 passwords could not be chosen. His solution was to remember 12 names, so that there would always be one he could chose. Thus, he did not need to remember new passwords all the times.

This shows how easily such limitations can be passed by. Moreover it shows that many users tries to find ways to make the process easier for them. Non-IT folks, such as doctors, do not care so much about the security; they only want to use the system - which already may be a too great task for many, the elder ones.

Giving guidelines to the users is good, but it is not sufficient or something the administrator should lean against. As one of the fundamental principals of robust programming states, one must assume that many users do not read or understand the guidelines.

Besides strengthening the password, event logging is an effective way to discover attacks. Every login attempt must be logged, whether successful or failed. Many systems only log if the username is found, but it should also be logged if the username is wrong, as it can be used to catch brute

force attacks against user names. Moreover, it should never be revealed whether it was the username or password that failed, or any other specific information. For a hacker, every single piece of information given by the system is useful. Many hackers use such kind of failed-password error messages to find working usernames.

The password logging makes it easier for an administrator to see if an attack (e.g. brute force) is being made. The user may also use it to detect uninvited guests in his account, or spoofing attacks which are discussed below.

6.1.1.3 Spoofing attacks

When a user enters his login information, can he be sure that the receiver of the information is the intended, or that the route in which his login data travels is the supposed one? A situation where a hacker or a program exists somewhere between the user and the server in order to snap the login data is called *spoofing*. In spoofing attacks the main purpose for the hacker is to let the user enter his information in a wrong form set up by the hacker which resembles the real one. Stories about fake boards placed in front of bank terminals are well known to most bank users.
A spoofing attack is carried out in the following way. The hacker (it may be a legitimate user) sets up a fake login screen, (e.g. run it on the terminal). At places like the databars at DTU, students always see a login screen where they enter their password. They do not start up the machine themselves. A student may login to his account and start a program that matches this login screen precisely. What happens next is that another student comes and enters his credentials without knowing that the login screen is fake. Most fake login screen captures the password and returns a "wrong password" error to the user, and at the same time redirects the user to the real login page. The user just thinks that he made a typing error in the password and retypes the login information carefully. In this way the password is tricked out of the user without causing any suspicion about the act.

A more professional way used by spoofers is that they do not return an error message, but instead try to submit the login information data to

the real site after copying the credentials. Thus the user will not even have to retype his login information. This kind of spoofing attack is also called *login spoofing*.

Spoofing attacks occur in many other forms also. In *email spoofing* fake emails are sent to users in order to let them believe that the mails are from a trusted authority. It may be fake product offers or other kinds of links that they want the user to click on. Emails formed like "Update your private information" or "Update your email password now!" are received almost daily nowadays. The easiest way to detect email attacks are by checking the mail headers. As it is very easy to write anything in the from-field the email will most often appear to be from another source then the actual originator.

Another kind of spoofing attack is the *webpage spoofing attack* in which a user is tried to be deceived by a fake webpage resembling the real one. This attack is also known as a *phishing attack*. The attacker will change the URL so that it goes to the fake site but the change will be carried out in a discrete manner which is not so easily detected. For instance, changing a letter only in a word or putting something in front or at the back of the real URL. The only way to guard against this attack is to carefully examine the URL when entering one's login data. Unfortunately many legitimate sites temporarily redirect the user to very odd URLs for the login, such as when logging in at hotmail, and some redirect to the DNS address. This makes it hard for the common user to distinguish between what is real and what is fake.

Many other types of spoofing attacks exist, such as *IP spoofing* which is more complicated and done at network level.

Millions of people use Microsoft Messenger software every day. Very often one is offered to log in to some 'magic site' that will disclose if someone is blocking one. Many people fall into this trap, without considering for even a second that they are actually going to give out their email username and password to some hackers!

Unfortunately there is not much to do against spoofing from the programmer's side. How can a programmer prevent that a user does not

enter his password in a wrong form on a wrong a website? Spoofing attacks exploit users' lack of knowledge or the fact that they may not be concentrated enough to discover the fake arrangement.

Some general guidelines have already been mentioned above along with the attacks. Beyond these, the user should do his outmost to inspect sites before entering his data. An idea is to make a habit of always writing the password wrongfully first time in order to trick a possible login spoof. What the programmers can do is to show a list of failed login tries whenever the user logs in, so that he can inspect the list and make sure that it coincides with his recollection. For example, if a person receives a login error but it is not shown on the login list, he surely should change his password. The user should also be instructed to be careful and not to login from external sites, but in the spirit of robust programming, it must be assumed when designing security that the user will not care about any instruction or even read it (in fact, people sometimes forget!).

Moreover, even if guidelines are followed fully, it is not always enough, for example, in the case of *key loggers*.

6.1.1.4 Key loggers

Key loggers are one specific application used for password spoofing. If a hacker already has access to the computer (he could be a legitimate user himself), he can install key logger software on it. A key logger is a program running in the background, from where it records every keystroke and sends it to the hacker or saves it in a log the hacker can read. Such applications do not require much knowledge and anyone can download freeware key logger software from the Internet and install it on a computer. Moreover key logger does not only limit to software, there are hardware key loggers and kernel/driver key loggers. [66]

6.1.1.5 Caching and password saving

Writing down passwords is a problem as mentioned above. But, even worse are features like auto-save password. Many people save their password on the computer thinking that they are the only users of the computer. If a hacker breaks into the system, he can easily find and decrypt the stored passwords. Secure systems should therefore not offer auto-login features. Unfortunately browsers and even firewall software sometimes "force" the user to let it remember the login credentials, or annoy him until he does.

Besides this there is a problem with caching. When passwords are entered they are not sent directly to the server, but stored in buffers, caches etc for some time. It may be that the password is kept long enough to let the hacker grab it. In buffers the password is often kept totally unencrypted, just as the user entered it.

6.1.1.6 Compromising the password file

Any system with users has a password file where all the passwords are stored, usually in a database. Such a file or database is an obvious goal for a hacker. For this reason, it is necessary that the content of the database is encrypted. Actually a password should always be stored one-way encrypted so that the passwords are secured from being read, even by the legal authorities. The administrator himself could misuse user passowrd. He could read the user-chosen password to this system, and then try the same password against the same persons' email etc, as many people share the same password for all their systems. However, there is no reason in a secure system to save passwords unencrypted or decryptable. The password validation can easily happen encrypted and if a user forgets his password, a new should be generated.
Even if passwords are stored encrypted the password file still needs to be kept private so that unauthorized parties cannot access it. Both brute force and dictionary attacks (after encrypting the words) can be made in offline mode if the file is compromised. Password salting is often applied to protect against offline attacks on a password file, if it should be com-

promised. Password salting is a way in which additional information is appended to the password before encryption.

There are different ways of protecting a database. The main techniques involve either cryptographic protection or access controls, or a combination of these. This is not touched further here, as it is discussed in a later chapter.

6.1.1.7 One-time passwords

So far, ways a password can be compromised have been presented, and techniques to prevent this. Password aging turns out to be a good way, as frequent password changes makes it difficult for a hacker to find out the password. What if the password age is decreased so much that a password only holds for one successful login? This technique has recently become very popular and is called one-time password.

[9] defines a one-time password as "a password that is invalidated as soon as it used". One-time passwords could theoretically be seen as the solution for a number of security concerns that exist for ordinary passwords. However, when it comes to the implementation, a number of practical issues arises and needs be taken into consideration.

A system could generate a whole dictionary of passwords which is given to the user. Whenever he uses a password, he crosses out the used password and takes the next on the list the next time and so on. Another way could be that the system simply asked the user to change password at every successful login. Such ideas would practically not hold and would be unnecessary troublesome for the user, as he would either have to walk around with password directories or have to remember the new password entered every time.

The purpose of a one-time system is that the password is changed every time. But how can a user, for example, be prevented to re-chose the same password (immediately after), or shop between two or three passwords? The failure and risk in of having a log of the last passwords was seen above. It may be that if the user does not know the upper limit and

that it is rather big, that many users most probably would think that it is impossible to reuse a password. In one-time password systems where the new password is randomly or automatically generated the problem of password reuse do not apply (or seldom).

The most considerable problems of a one-time password system are how to generate a new password and how to synchronize the user, the system, and the generation of the new password. Basically one-time passwords come in of two types. (RFC 2289)

The first type is based on mathematical algorithms. In this approach a one-way function is used to generate the next password each time from the previous password. If a password is compromised the hacker needs to know the inversion function and the number of times it must be inverted (though, he could try after each inversion). As a one-way function cannot be inverted, this is an impossible task. In reality no one-way function exists but there are functions, such as cryptographic hash functions, that are so difficult and infeasible to invert that they can be regarded as one-way functions.

The other way of generating one-time passwords is the time-synchronization technique which is mostly related to use of a physical hardware token. Every user has a token which is time-synchronized with the server. Whenever the user wants a password, the token calculates the password based on the current time. The token will have to be specific for every user and of course a small time interval (of a least one-two minutes) would be required in which a user can enter the password and submit the login request to the server.

Mobile phones could be used to transmit the one-time password. Today almost everyone already has a mobile phone. Hence it would not require another piece of hardware for the user. In this case it would have to be programmed into the mobiles in a way that prevent sending the code in as an unencrypted SMS message.

One-time passwords alone cannot prevent brute force or dictionary attacks. One-time passwords only help in the case where the password is reused while brute force attacks do not reuse the password after reading

it during a transmission etc. but guesses it when it is needed. As seen in the examples there must be a second factor (for instance a token) to implement a one-time password system. Hence one-time passwords are mostly used as a part of a two-factor authentication.

6.1.1.8 Social engineering

Social engineering is a term that covers another way of robbing peoples' password and getting sensitive information out of people. In difference to other methods, this is a non-technical way, not based on any script. The attack does not directly target the system. Instead the hackers use social fraud of either physical or physic character. It has some resemblance to spoofing attacks as the hacker is pretending to be someone he is not. In social engineering of physic character, the attacker acquires some knowledge of the company, who's system is to be hacked, or a private person. Next, he personally contacts the target person mostly by phone and pretends to be from some consultant firm or offers other kind of support, in order to get the password out of the user with the excuse of, for example, a system update.
Another way is that the hacker does not directly ask for the password but has setup up a fake site (webpage spoofing) which he guides the person through. For instance he could inform that the login system has been hacked so it is necessary that all users go to this temporary login site and changes their password. This is like user-guided phishing.

In social engineering of physical character, the hacker searches in trash bins, obsolete software manuals, scraped hardware, backup CDs etc. in order to find sensitive information. Often in and around printers there may be papers that may reveal some information. Employees may get access to the administrator's office shortly during lunch breaks, and look for passwords or other useful information written down anywhere.

6.1.1.9 Conclusion

Even in the best case where all precautions are taken, passwords are kept completely safe, and the system is totally protected against any attack, still no one can guarantee that passwords will not be guessed or compromised. A password can always be guessed. Moreover there is no way at all to distinguish between a legitimate user and an intruder when using passwords. All this discussion leads to the fact that the authentication based on what the user knows only cannot be deemed sufficient for a system where sensitive information such as health data is dealt with. In fact the above mentioned points would be necessary but not sufficient. A combination of one or more of the other ways of authentication therefore has to be set up. Watching the security setup of e-banking systems, it is clear that no bank is satisfied with this mode only, instead another factor (e.g. a private key) is used. This is being described in the next part.

As authentication based on what the user knows, and especially passwords, are the most widespread, this chapter went into details with the problems in and around the use of password in general.

6.1.2 What the user has

Another way of authentication is based on something the user has. An example of this is a physical key. The lock can only be opened by a person having the key. The student card at DTU, for example, implements this kind of authentication.

A problem with this authentication is that the object itself actually is not uniquely attached to the person holding it. The object can be lost or stolen, and freely used by other persons as well. Every user of it will have exactly the same rights. Moreover physical objects, like keys, can also be copied, and there is no way to find out how many persons actually do have a copy of such an object. In practise, this kind of authentication is mostly used in combination with other modes. For instance most cards do have a PIN code also, which establishes a combination of what a

person knows and what he has.

In fact, in many secure online systems, this combination, i.e. what a user has and what he knows, is used. Examples are banking sites where digital signatures are used to login. The user must enter his secret code and complement it with the signature key file which is an example of something the user has.

Although the combination increases the security level, the risk of being impersonated is still there. There are several ways of stealing a private key or a credit card, and different ways of obtaining a password have already been discussed in the previous chapter. Though, it will require some more work for the hacker, yet it is not impossible.

Today, the combination of what one knows and has is among the most used authentication forms. What makes this kind so attractive in real life is that there often is some kind of arrangement through which a person quickly can report if his credit card, mobile etc. is lost or has been stolen. Normally it would take the hacker some time to break the password attached to the object. Meanwhile the object is instantly blocked and made unusable as soon as the owner notices the loss. Moreover if the password or secret code itself is chosen securely following the guidelines mentioned, the addition of this second object of course can only increase the security even more, to a practicably unbreakable extent. One must keep in mind that many "secure" sites, (e.g. mostly all e-mail providers) only use the password authentication still.

6.1.2.1 Conclusion

For both factors, something a person knows and something he has, it is common that they are not unambiguously attached to any person. Hence it does not matter who the person using them is as long as the two factors are given correctly. A unique identification therefore can never be obtained merely from a combination of these two. Sometimes credit cards or other identification cards are provided with a picture of the card holder also. This is a third factor, namely what the user is, which leads to the next discussion.

6.1.3 What the user is

Authentication can be done based on what the user is. This is the way people daily recognize each other at work, at school etc. Personal photos in driver's licenses, passports and on identity cards are all examples belonging to this category. It should be something "attached" to the user that cannot easily be changed but remains with him and identifies him (more or less) unambiguously. Although the use of photos is the most wide spread, misuse is easy since the one validating often does not know the fake user or the one on the photo too well. Moreover, a person's look changes by time.

However, with time biometric authentication has advanced, and it is still being developed. [9] defines biometric as "the automated measurement of biological or behavioral features that identify a person".

Hence biometrics is the use of technology for the recognition of individual human features such as fingerprints, voices, eyes, faces. As each human's fingerprint is considered unique, the use of it as a means of identification has significant security advantages. Moreover a fingerprint cannot be forgotten and is unlikely to be lost or misplaced. Using biometric authentication even twins can be distinguished from each other.

Historically fingerprints have been used a lot, for instance by the police for identification in criminal cases. This is the reason why fingerprints often are associated with something done to "criminals". However, many laptops today have built-in fingerprint censors. This shows that fingerprints are no longer used only by the police but slowly turn into a natural thing to have and operate with.

6.1.3.1 Biometric systems

In biometric systems first an enrolment is made. For instance, for fingerprints the user is asked to give samples of his fingerprints a number of times. These are called *reference templates* and they are stored in some secure database. Whenever the user wants to log in, his fingerprint is

compared against the stored reference templates.

Biometric authentication is primarily used for one of two purposes, namely *verification* and *identification*, [32] [42].

1. When used for identification the fingerprint is tested against all the templates in the database. As the fingerprint is unique for every person, the fingerprint that matches best within some allowable boundary will be regarded as the user's.
2. In verification the biometric system is used to verify that a user with a given identification really is the given user. This is done by matching the fingerprint template with the template stored for that specific user.

6.1.3.2 Problems

The technology today, offers ideal tools by which biometric traits can be utilized for security purposes. It has both strengths and weaknesses.

Already at the enrolment stage, problems can arise as the system may well fail to enrol a user. The *rate of failure to enrol* (FER) defines the frequency by which a system fails to enrol a user. This may be due to the fingers being so worn down that the biometric system cannot measure any good quality template. If fingerprint authentication is implemented in country wide systems such as electronic patient systems, what to do with the percentage that will be rejected?

Secondly, if a password is compromised it can be changed but a fingerprint cannot be changed. Thus, if a fingerprint is compromised there is no way to change it.

Next, fingerprints and biometrics traits in general may be unique but at the end they are not that secret. A person leaves his fingerprints wherever he goes, his voice can easily be recorded etc. Moreover it has been shown [48] [68] that rubber fingers can be constructed without too much difficulty that can break most fingerprint recognition systems.

The use of fingerprints and other types of biometric authentication also involves some social or ethical problems. At places where fingerprints are used for securing valuable things, reports come now and then in the newspapers about thieves chopping off fingers in order to get the thing. Two years ago the story [15] about car thieves in Malaysia gave rise to many ethical and social discussions of the use of biometrics. Four men attacked a Mercedes owner in Malaysia in order to get the keys of his car. As the car was security protected by a fingerprint sensor, they eventually had to cut his finger and they left him naked and fingerless on the road.

The debate is about whether the human body should be put into risk in order to secure something material. If keys or smart cards are stolen, there are solutions to quickly report this. They may beat him or even kill him, but in most cases the thieves would be satisfied when getting the keys, smart cards whatsoever. In the case of, for instance, fingerprints, it is most likely that the thieves will go after the fingers, the hands or even the arm in order to pass through the fingerprint authentication, thus causing the owner irreversible damage.

As the use of fingerprints increases, the incidents of finger-cutting thieves follow suit leaving physiological affects in people's minds. The classical guideline saying "do not venture your life, obey the thieves and give them what they ask for" cannot even be applied any longer, as no one can or will hand over his finger. Such physiological issues may grow fear in people's minds against the user of fingerprints.

However, at the time of writing a lot of development is still being done in the field, and it is not unlikely that the biometric systems in the future will all have built-in aliveness detection. A number of techniques for detecting aliveness are being tested, here among temperature, conductivity and pulse, [35]. In this way the problem with using fake fingers and chopping off fingers can be put out of the way.

6.1.4 Where the user is

A fourth form of authentication mentioned by [9] is the user's location. In this form, it is taken into consideration where the user is trying to log in

from. This may be his geographical placement but also which terminal, he is using, for example, based on his IP. This kind is used at offices where the users are only allowed to log in when they are at the office or through a special secured computer. Sometimes this is combined with remote login in the way that a user cannot login to system A from his home, but he may establish a remote connection to the system B from where he can enter system A. Such arrangements are done for many purposes for letting system B be the only system allowed to access system A. System B may have the right security arrangements and certificates installed, proper access logging etc.

Authentication based on geographical location may be more specific and applicable to mobile devices having GPS readers. In systems like the electronic patient journals where you want the user to be able to login from almost any terminal this factor is not always suitable to be included, though it could be useful some places, for example, to access special services only allowed from certain places.

6.1.5 What the user does

[32] mentions a fifth authentication which is based on what the user does. According to [32] this category constitutes the user's performance of tasks that is specific to the individual. As examples of this, he mentions

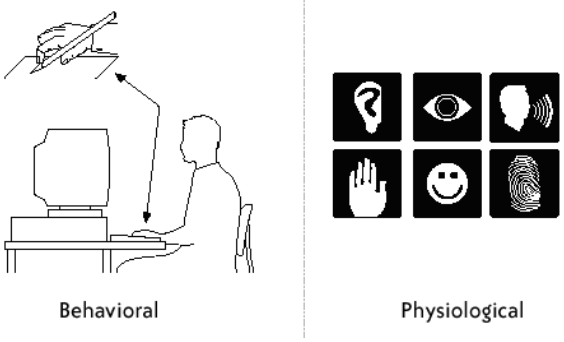

Figure 6.2: An overview of the main categories in biometric authentication

handwritten signature and typing speed on a keyboard. Actually these belong to the biometric authentication as it is divided into two main categories, as shown in Figure 6.2.

First there are the physiological features such as the fingerprint, eyes etc. Secondly there are behavioral features such as ones voice, signature, typing speed etc.

As these two categories together form the biometric systems this fifth type of authentication is normally treated under the type of what the user is. This may be the reason that [9] did not mention it separately either. In many books even the fourth form above is not mentioned either as it is regarded somewhat specific to mobile devices and not a general authentication factor. There is no doubt that authentication based on knowledge, possession and what the user is are the most interesting factors.

6.1.6 Comparison

Above different types of authentication have been discussed. The simplest and most widespread is the password authentication. Some of the problems related to the use of this kind of authentication have been covered. Today, there is a lot of focus on biometric authentication. One question raised is whether or not the biometric authentication brings a solution for the many problems related to authentication, and, in case, to what degree. The most popular type of biometric authentication is the fingerprint authentication. Below an overall comparison between the password authentication and fingerprint authentication is made.

Problems with input
A password can be stolen at the time of being entered with the help of key loggers. Nothing prevents fingerprints sensors from being logged in the same way. In fact, in order to get a fingerprint, the hacker does not even have to strike at the time in which the password is being entered. Instead, the fingerprint can be read afterwards, from places the person has touched with his finger, just as the police do in criminal cases.

Problems with transmission
The password is sent to a server through the Internet or intranet. No matter which transmission channel is used, it must be secure against sniffing and the password must be transmitted encrypted. The same problem applies for the fingerprint as it is also just a stream of bits being transferred from one point to another. It is already mentioned that the hacker can attack the weakest of the transmission channels through which a fingerprint is sent, as the fingerprint is the same. This does not necessarily apply for passwords, since they are most probably different.

Problems with storage
The passwords are stored in some database. The database needs to be secure and protected so that unauthorized users cannot access it. Moreover the passwords have to be kept encrypted, preferably with a one-way encryption. Again the same applies for fingerprints since they are also just a bit stream as the password.

Social engineering
Using social engineering, a person can be led to reveal his password through the telephone, email etc. This cannot be done with fingerprints, but instead the hacker only needs to pay one visit to a place where the user has been and placed his fingers.

Human problems
When passwords are lost, they can be changed as soon as the theft is reported. The user does not get any personal damage. But if the attacker starts chopping off fingers irreversible damage is caused to the person. Besides this physiological fear fingerprints are, for many, considered a too private thing and many relate it to something related to the criminals. The reason may be that the biometric information is uniquely attached to people. This is merely a question of a change in attitude, though not always an easy task. It is somewhat funny that fingerprints are regarded so sacred today, when they actually do not have much practical use in the society today and PIN codes are even more valuable and vulnerable. One advantage of fingerprints over passwords is, that the user does not need to remember his fingerprint, while he has to remember the password.

Although it may be against the spirit of security, individual passwords

may sometimes be shared for instance among couples. The wife may be asked to log in and submit health data etc. Sometimes a person may need to call a very close relative and ask him to log in and check something important for example if the person himself is stuck in the traffic or at some airport without access to Internet and instantly needs some information stored at his computer at home. The person may give out his password in this case having in mind that he can easily change his password next time he gets access to a computer and of course having trust in the second party. This is not possible at all with fingerprints.

The password is exact, only the exact bit stream representing the password will be accepted. Whenever the correct password is entered it will be accepted. Using fingerprints, one seldom hits the precisely same fingerprint template but rather inside some boundaries. A person may need to give his fingerprint a number of times before it is accepted. In periods where the finger is injured and plastered there may be no way to access the systems. The finger may need to be cleansed thoroughly as dirt even in small amounts can confuse the sensor. If the fingerprint machine rejects after several tries, is there anything to do?

Problems with availability
Wherever the user can access a computer, he can enter his password in order to log in. But biometric authentication requires hardware devices supporting the desired biometric functionality, such as fingerprint sensor.

At the time being, biometric authentication is not taken into use by many (relatively) and a lot of research is still being done at the moment in this field. Definitely a number of considerations, here among the above mentioned, need to get clarified before the shift should happen.

In the above discussion only fingerprints were treated. It is far more unimaginable that the other kinds can do it any better. For this reason, they have not been discussed but many of the discussion points can be generalized for all the biometric forms.

6.2 Two-factor authentication

Different forms of authentication have been presented in this chapter. Among these the central and most important ones are the three based on knowledge, possession and what the user is.

Today a two-factor authentication is regarded as secure. The Danish Data Protection Agency [24] also requires that at least two-factor authentication is being used. The same applies for most of the Danish banks. A single factor cannot be deemed secure as the discussion clearly has shown.

Two-factor authentication means that two different authentications forms must be combined. Therefore having two passwords for instance would not be regarded as a two-factor authentication. This is because, the authentication form is the same and the chance of cracking either of the passwords is equal. Another case is online transactions where a person uses his MasterCard. He is asked to enter the card number, expiry date and a security code that is stated on the backside of the card. It may look like a two-factor authentication but it is actually not, nor is it based on possession, which one could think as you have a card. This authentication is technically only based on knowledge and the card just functions as a paper on which the code is written. It is even more insecure than an ordinary password which is remembered by heart. Here the password is written on the card, so the hacker only has to get hold of the card for two minutes in order to read the info. In fact when paying online it is strictly not an authentication and the owner can contact his bank and deny having done the transaction. Though, the hacker may wait until the card owner, for instance, takes holidays and will not have access to his account for some weeks.

A third example is the online banking. It was mentioned earlier (and it is usually said) that it has a two-factor authentication, though it is strictly not correct to say so. The fact that one keeps his private key on a USB and has the code in mind, could resemble a two-factor authentication. The reason why it is not is that the private key is a bit file (a long bit stream) and the password phrase a shorter bit string. Both are strings that could be remembered if personal memory would allow, or written

down somewhere. Thus both are based on knowledge and therefore not a perfect two-factor authentication, although it resembles.

In continuation of this and support of the statement, it is actually being discussed that the banking sites should turn over to a new solution based on two-factor authentication. In the new solution [86] which is hoped to be ready in 2009 the private key will be dynamic and one-time, generated every time with for instance a token. In this way, although the hacker may snap the password from the user, and even the key file, it will not be of much use as the key file changes every time. This is a combination of the one-time password systems mentioned above and a simple password. It also supports the statement above that the current validation with static key files is strictly not a two-factor authentication. If the current authentication was already with two-factors, this discussion would not be of interest.

As long as the transmitting medium is the same, it would never truly be two-factor as a hacker who for example is acting as the man-in-the-middle could in principle be careless about which factors the user enters, he just needs the information when it is being sent. Instead, if there was a solution based on two-factors sending over two different mediums this would be a true two-factor systems. An example could be that a person first has to send an SMS to some server, and thereafter he would be able to login through the Internet with a code that was sent back on to his mobile along with some predefined password. In this way it is not enough for a hacker to monitor the outgoing traffic on a users' pc. This way of authentication is called *out-of-band authentication* and it has been used for some banking systems for some years now, for instance SMS approvals at time of making money transfers.

6.2.1 Breaking two-factor authentication

Although two-factor authentication at the time is regarded secure, there are examples of two-factor based systems being hacked, [80] [5] [21]. Below two examples are presented.

The online banking system of the Dutch bank AMR Amro was hacked

through phishing. Hackers sent the customers emails falsely claiming to be from ABN Amro. If recipients opened an attachment, software was installed on their machines without their knowledge. When customers visited their banking site, the software redirected them to a hacker-controlled mock site that requested their security details. As soon as the hackers received these details they were able to log into a customer's account at the real ABN Amro site, before the expiry of the fob-generated number. They could then transfer the customer's money. [5]

Secure Science Corporation spotted a phishing website targetting Citibank's Citibusiness service. This attempted to steal both the user name and password as well as the temporary password issued by the security token. The site furthermore acted as a middleman that relayed the information to the Citibank server for authentication. It prompted users if the information they entered was incorrect. [21]

It must be mentioned that many experts ([12] [67] [81]) actually do not regard two-factor authentication secure enough and has written about the insufficiency of two-factor authentication. One of the leading critics is Bruce Schneier [67]. According to Schneier two-factor authentication only solves problems that existed 10 years ago but does not face the problems of modern day. He mentions the problems with password, which has already been discussed in this report, and says that one-time tokens has been there for almost two decades as a solution to this. Earlier the attack types were of passive character, like eavesdropping and offline password guessing but now it has become more active in the forms phishing and Trojan horses.

The two main security issues against two-factor authentication he mentions are man-in-the-middle attacks and Trojans horses. A Trojan horse attack is an attack where software is installed on the victim's computer so that the hacker functions as a shoulder watcher, or as Schneier calls it, a piggybacker. According to Bruce Schneier the two-factor authentication does not solve anything. After mentioning the two attack types he writes:

> "See how two-factor authentication doesn't solve anything?
> In the first case, the attacker can pass the ever-changing part

> of the password to the bank along with the never-changing part. And in the second case, the attacker is relying on the user to log in. The real threat is fraud due to impersonation, and the tactics of impersonation will change in response to the defences. Two-factor authentication will force criminals to modify their tactics, that's all." [67]

Schneier's points are understandable and they have also been discussed just before. Though, it is an exaggeration to state that the two-factor authentication does not solve anything. Below the two issues he mentions are examined.

One of the factors mentioned by Schneier and the critics is the man-in-the-middle attack where an attacker gets between two transacting parties and either monitors or changes the messages without either participant's knowledge. This is difficult to avoid and it is also the basis in the phishing attacks, where some fake receiver of the credentials is set up. The main difference is that the man-in-the-middle attack usually functions as an intermediary sending the data fourth and back. The phisher takes the information and acts on behalf of the user while the user thinks some error has occurred. The phisher may show the error message that the system is down and will come up again in an hour. This way the user will not care more about it for at least one hour in which the hacker can work in peace.

Both of the attack types are not specifically attached to the two-factor authentication but of a general character. Before considering the attacks, one must first realize the purposes of cracking an authentication. There are two main purposes. The first is to get the login information (i.e. username and password) while the second is to break into the system without having the information.

The first can be hindered for the man-in-the-middle attack considerably by (one-way) encryption as the attacker will no be able to decrypt the credentials. The second goal is a bit harder to prevent, but it is possible using out-of-band authentication where two different mediums are used for the two-factors. Doing this the middle-man has to sit between both mediums which definitely makes it harder for him. Though, it may be

said that an out-of-band authentication does not affect the middleman's work. The middleman may not even care about the second medium, as he will be happy to let the user do this. For instance, if the second medium is sending an SMS, the hacker will gladly let the user send the SMS and then wait for the user to send the credentials online. A possible solution to this problem is to make the two media dependent on each other. The first mediums may send some information that will make the second specific to the user. For instance in the case of sending an SMS, the user could send his actual host address or IP address through the SMS, telling the server what the originator of the second factor will be. This will make it harder both for the middleman and especially the phisher to make any use of the gathered information. In this example, the hacker then also has to impersonate the user's IP. Remember that security can never be absolute. Whenever one solution is found, the hackers will find new ways to break through. However, the different solutions cannot be said to be useless as every step makes attacks more and more difficult.

What goes preventing the first attack goal (i.e. getting login data) for the phishing attacks, it is nearly impossible to perfectly guard against this as the possibility of trapping the user to give his information will be there - just by placing a gun at his head while asking him to login.

The second point that the critics stresses is the Trojan horse attack. Again, this kind of attack is not specific for authentication, but a general attack form. The Trojan horses actually does not care about the authentication process at all, it just functions as a monitor, watching everything that the user is doing. Thus the first attack purpose (e.g. getting the credentials) is not achieved in this attack unless the hacker installs fake login prompts. In this case it will be regarded as phishing, which has already been looked at. The Trojan horse attack may be effective but when functioning as a monitor, the attacker cannot, by himself, take actions. He may be able to watch a bank customer checking his account, but he cannot make the user transfer his money. Though, there are certain types of Trojan attacks that can overtake the control also.

6.2.2 Conclusion

To sum up, the conclusion is that two-factor authentication is currently regarded as secure and many banks and online systems are shifting to the use of it. The most popular two-factor authentication is based on knowledge and possession, usually a password and a one-time token, but biometric authentication is also making progress rapidly.

Despite the fact that two-factor authentication is considered to be strong enough, the critics' voices and concerns have also been presented and discussed in this chapter. The main points of the critics do not directly relate to the authentication and they can be prevented in other ways, for instance using firewalls. The authentication is just one part of the system, and of course all parts have to be secure in order to label the system secure in its totality. It is of no use having doors of concrete if the windows are left open to a house. Therefore, for the sake of authentication only, properly made two-factor authentication can be deemed secure and useful, at this time. At least according to the Danish experts and *"IT- og Telestyrelsen"* [54].

Chapter 7

Access Control

In modern information systems such as medical systems, a lot of sensitive information is stored. It is of high importance that only legitimate persons can access the information. Moreover not every authorized person is supposed to have the same permissions, also called *access rights*. Employees at a hospital, for example have, different roles. The secretary, the nurse and the doctor may all want access to the same journal, but each of them with a different query. Information access differentiation and protection is necessary in all systems with users and sensitive information. There are many ways of accomplishing this and at differing levels of granularity. Some solutions work on the network level, like firewalls, while some touch the information directly, such as encryption. The purpose is to control access. Almost every network uses some kind of access control to grant and deny access according to the users' privileges. Most operating systems also have native access control arrangements. A system may use different tools to protect different resources, such as for the protection of web applications, database servers, mail servers, file servers, applications etc. As the resources that need to be protected can be so different, implementing an access control is not always an easy task. Every resource has its own and specific requirements and it is essential

to have the accurate access control tools. Determining an access control mechanism can in itself be an issue as there are so many solutions existing today.

When discussing how fitting an access control mechanism is, the term *granularity* is used. The finer the granularity is, the lesser probable it is that users get unnecessary and inappropriate access rights. Likewise, the coarser the granularity is, the more inappropriate the rights are given. It follows that a perfect system will have the finest granularity. It has the least privileges which means, defining the access so that the users see precisely what they need, not more or less. Though, a disadvantage is that the finer granularity result in slower and more labour-intensive systems.

No matter which solution is selected, the access rights are configured by a limited number of parameters. Some solutions require a lot of manual configuration while others are based on configuration policies. To attain the level of least privilege totally is not possible. As getting the finest granularity requires too much work, it is a trade off between finer granularity and the labour intensity required and the speed of the system.

Most access control solutions are built as combinations of basic access control types. Hence, it is necessary to understand these. In this chapter the basic and traditional access control mechanisms are described. A good understanding of these access controls makes it possible to make the decision of access control in any scenario according to it's needs and requirements.

7.1 Access control matrix

A system can be divided into objects and subjects. The objects are those that need to be secured while the subjects are the active entities. Usually the objects are files, folder and other kind of resources, and subjects are users and processes associated with them. However, there is no clear distinction between subjects and objects. In some process an entity may be the subject while it is the object in another. In this way the process

determines what the subject and object are.

The interaction between subject and object can be looked at from either the subject's view or the object's. That is, it can be defined what a subject is allowed to do, or it can be defined what can be done with an object.

The access control matrix model characterizes the rights of the subjects against the objects. It tells which subject has exactly which rights in respect to any object in the system. This is done in a matrix, from where the name stems.

Below an example of an access control matrix is shown:

	report.pdf	attachment.doc
Bob	{ read, write }	-
Adam	{ read }	{ read, write }
James	{ read, write, own }	{ append }

The table shows the protection state in some system for two files and three users. The subjects are represented in the rows, while the columns represent the objects. The set of rights in this example are { read, write, own, append }.

The file report.pdf may be read and written to by Bob while Adam can only read it. James can also read and write to the file and at the same time, he is the owner. James has rights to append to the file attachment.doc. Adam can read and write to it, while Bob has no access rights for this file.

Theoretically an access control matrix is an ideal mechanism for controlling object access rights. Though, it is not suitable for direct implementation in large systems with a lot of subjects and objects. Instead several optimizations of it allow systems to use more suitable and simpler versions of the access control matrix. Among these models are Capabilities and the Access Control List.

7.2 Capabilities

As mentioned before there are two ways of specifying the interaction between the subject and the object. When the access rights are kept with the subject every subject gets a capability. The subjects' capabilities correspond to their rows in the access control matrix. The example matrix given above can be rewritten in terms of capabilities as follows:

Bob's capabilities: report.pdf: read, write;
Adam's capabilities: report.pdf: read; attachment.doc: read, write;
James's capabilities: report.pdf: read, write, own; attachment.doc: append;

When a subject wants access to an object, the subject's capability list is examined in order to determine if the subject has the necessary access rights for that object.

[32] writes that the capabilities have not become much widespread although it is an old concept. One of the reasons is that it is difficult to get an overview of which subjects has what access rights to a given object. It would require all subjects of the systems to be scanned. In practical, the question about subject's access rights on objects is asked most frequently. For this purpose Access Control List (ACL) are more suitable.

Access Control List
An Access Control List (ACL) keeps the access rights to an object with the object itself and thus it corresponds to a *column* in the access control matrix. Writing the ACL for the above examples gives:

ACL for report.pdf: Bob: read, write; Adam: read; James: read, write, own; ACL for attachment.doc: Adam: read, write; James: append;

The security models in most commercial systems base the permissions on an ACL of some type.

The UNIX operating system also uses a variant of ACL. In this variant, it is easy to get a per file overview over which rights users have to this

file, but an overview of which rights different users have cannot be derived without trouble. For this purpose, Capabilities exist. Thus before selecting either ACL or Capabilities, it must be decided how the system is supposed to be used.

7.3 Fundamental Access Controls

Basically there are two types of access control models: The Discretionary and the Mandatory Access Control model, [32] [9].

7.3.1 Discretionary Access Control

In the Discretionary Access Control (DAC), the individual owner specifies and controls which subjects have what access to his objects. The model has its name because access is granted at the discretion of the owner.

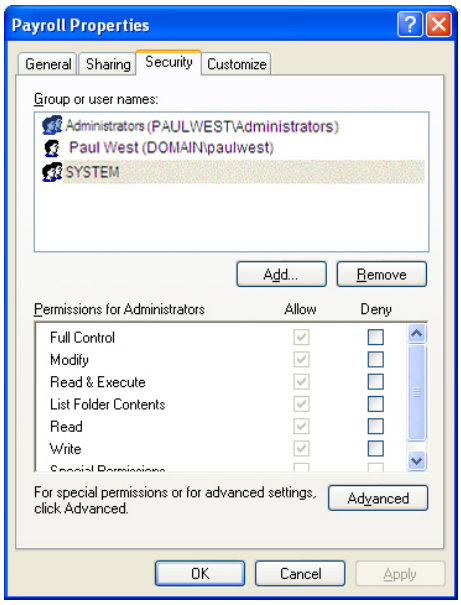

Figure 7.1: DAC used in Microsoft Windows

The access rights defined by the owners are most often implemented through ACLs enforced by the operating system. A good example of the DAC model is when browsing the file or directory properties in Windows (or another operating system) as Figure 7.1 shows.

The native access control feature in Windows allows one to define the access rights for a specific user, computer, or group of users. It should be noted that ACLs in Windows only apply to files stored on an NTFS formatted drive. The type and level of access is specified for a user or a group for any file or folder. One user may be granted Read and Write access, while another user group may only have Read access. Permissions are not restricted to files or folders, but can also be set up for other network objects, such as printers. This specification of access rights is an example of an instance of ACL enforcing the DAC model.

Theoretically, DAC can have a very fine granularity, but not without being very labour-intensive. Moreover, as DACs are mostly enforced through ACLs, it means that a user cannot get a full overview of the permissions for all his files and folders without scanning the whole system. The problems that exist with ACLs, of course, also appear here.

Access is granted and denied based on a subject's identity, which may be a user identity or group membership. This can lead to some complications as any program running on behalf of a user inherits all the users' access rights. Thus, it is vulnerable to Trojan horse attacks, for example.

7.3.2 Mandatory Access Control

The second model is called Mandatory Access Control model (MAC), in which the individual user (even if he is the owner) cannot edit the access rights to objects. Instead access control is regulated by the system mechanism controls. Owners do not have full freedom to determine the access rights of their files. They may do so, but the operating system makes the final decision and can override the access right defined by the user if they are in conflict with the rules. Therefore, MAC is also called rule-based access control, as it is the rules that describe the final conditions under which access is allowed.

The MAC model uses security labels to enforce the security. All users have a clearance associated with them and the objects have security labels containing the classification of the data. When making access decisions, the clearance of the subject is compared to the security label of the object. Users can only access data that is classified to be equal or lesser then their own user status.

7.4 Traditional Access Controls

Besides the two fundamental access controls, there are a number of traditional access controls, [47]. These are outlined in the following.

7.4.1 User Based Access Control (UBAC)

In User Based Access Control the administrator sets the access right for each user individually. This type of access control is also known as Identity-Based Access Control (IBAC). UBAC has the potential of giving very finely grained permissions, but at the same time, it is very labour-intensive. The administrator would need to know exactly which files all the different users need access to and update the permissions on a daily basis. In this kind of access control far more access rights are often given than actually needed as the administrator cannot manage to make the precise updates. It is worth noticing that the permissions are not set automatically, but manually. Moreover, speed limitation is also a factor to consider when considering this type of access control.

7.4.2 Role Based Access Control (RBAC)

In Role Based Access Control, roles are created with certain access rights. The different roles are assigned to users, and a user may have more than one role assigned to it. There may also be roles without any users assigned to them. A role can also be seen as a set of access rights that a users could have been given individually. Instead, it is made a role, which can

easily be applied to all users having the specific access rights defined in that role. This is called *implicit assignment of rights* which means that the rights are assigned to a group (role) from where the user inherits the rights. When rights are assigned specific to the individual user the assigning is called *explicit*.

The administration of access rights is significantly easier compared to MAC, as only roles have to be administered and not every single user. It uses a centrally administered set of controls to determine the interaction between subjects and objects. RBAC, therefore, is useful in companies where employees have different roles.

RBAC is also called non-discretionary access control (NDAC) because the assignment of a user to a role is unavoidably imposed.

The following is an example of RBAC. In a class with 25 students, there are two assistant teachers. The teacher wants to share some files. The assistant teachers should have read and write access, while the students should only be able to read files. The teacher should have all the rights being the owner of the file. Instead of assigning these rights to every student, three roles are made: student, teaching assistant and teacher. The role student is given the read right only and the role is then assigned to all the students. The two assistants are given the teaching assistant role and teacher the teacher role. Now if the teacher wants to give the students upload rights, he can get off with editing the student role. In the MAC setup he would be compelled to change the rights for each student individually.

In the UNIX operating system file, permissions are given using RBAC. The users are divided into three groups (i.e. three roles), the owner, group owner and others.

7.4.3 Policy Based Access Control (PBAC)

In Policy Based Access Control policies are created to control the users' access rights. PBAC is also known as Rule Set Based Access Control (RSBAC) as the policies are formed by a set of rules indicating which

actions are permitted between subjects and objects. An example of a policy may be a limitation in the number of documents that can be downloaded by a user in a given time span. Before the subject can access the object, it must be checked that the subject meets the predefined rules.

PBAC, unlike DAC, is not identity based. As PBAC is just a set of rules, it is not addressing the users individually thus it cannot give a fine granularity. However, this can be achieved by combining it with other access control types. PBAC mostly occurs in combination with other access controls to strengthen them with a set of general rules.

7.4.4 Content Dependent Access Control (CDAC)

In Content Dependent Access Control, access is permitted and denied according to the content of the resources. The access rights are based on the sensitivity of the information rather then the identity of the subject. Mainly it is used with databases that have sensitive data stored. For electronic patient journal systems it could be a choice, as not all legitimate readers of a patient's journal should be allowed to read everything. If a patient gets his knee injured during a soccer match, there would be no reason for the visiting doctor to know that the person may have had social problems with his wife. Likewise, other information in a journal may change the doctor's view on the person. CDAC is very useful as it can give access control even inside one resource, whereas other access controls may allow multiple people to read the journal without any content differentiation. The problem is on the other hand that it will require every single resource to be totally examined and labelled according to its content. In old systems it would thus be difficult to implement. If the system is built from scratch with an empty database, it will be possible to require every entry to have content dependent labels. As every resource must be scanned thoroughly, the speed of systems based on CDAC is usually not so high compared with other access control systems.

7.4.5 Context Based Access Control (CBAC)

Context Based Access Control is mostly used with firewalls to protect traffic. Often it is confused with Context Dependent Access Control (CDAC) as the names have some similarity. However, there is a clear difference between them. Moreover they can also appear together in a combination. While the access rights in CDAC are given solely based on the content of a resource, CBAC also includes another factor, related to the context in which access is sought. The factor may be time, location, temperature etc. Notice that CBAC itself does not have to be content dependent at all. It depends on the state instead of identity of the subject or content. Before giving access, first the state or situation is examined. A medical example could be that a doctor is only allowed to read certain parts of a patient journal when the operation has started, and the nurse may only see the journal after the operation has ended successfully. Another example could be an online examination system where the solutions cannot be shown before the examination time has passed.

CBAC makes use of strict security policies with which it decides the access rights. Therefore it is not necessary to specify user rights individually.

7.4.6 View Based Access Control (VBAC)

View Based Access Control splits a resource into sub-resources and users are assigned to the different sub-resources. VBAC is primarily applied in database systems, but it is also used for files and applications.

In VBAC, the resource is given an outer interface (a view) so that users only can interact through these. A patient journal is an example where VBAC may be used. Nurses need to access one sub-section of the journal, while doctors need others or all of it. VBAC is most useful for information stored in databases. Most (all) databases today have built-in VBAC functionality in forms of views. Assume, a table with patient journals. Important factors do have their own column in the table, such as blood

type. Views can then be created showing the patients with a specific blood type. Another view can be made that only shows the patient's name and the blood type. In this way the original table containing all the patient information is not directly accessed.

As with other access control mechanisms, granularity can be made fine but it is labour intensive. Moreover a lot of views must be made to fit user permissions. Besides this there may be problems with inserting and updating data, as the view is virtual and one is not actually seeing the resource itself.

Chapter 8

Framework

The framework requirements were outlined in Chapter 3. In this chapter the framework is proposed. It consists of three major schemes:

- Authentication scheme for the identification of users
- Access control scheme for the protection of information, i.e. authorization
- Secure transfer scheme for the protection of the confidentiality and integrity of information in transit over the network between server and client

Authentication involves the process of identifying the user. It is necessary in order to allow entrance to the system, and forms the basis of every further authorization mechanism.

Authorization involves the process of identifying the users' rights. Authentication identifies the user, while authorization defines which resources an identified user should have access to. Not only the access

is defined, but also which type of access, i.e which actions (read, write, etc.) the user is allowed to carry out to the resources, he is permitted to access.

Secure communication involves the process of transferring messages and data securely among the users and services of the health care system.

In the following the three schemes are presented.

8.1 Authentication scheme

The first major part of the framework is the identification and authentication scheme. Before allowing users access to the system, the identity must be established. Usually authentication schemes are static having either one or two authentication factors. As the security requirements of medical services vary a lot in health care systems, this framework presents a dynamic authentication scheme providing several authentication mechanisms.

Figure 8.1 shows the architecture of the authentication scheme. The different steps shown are the following:

1. The client registers at the system
2. The system notifies the access request agent
3. The access request agent queries the service provider agent for available services
4. The service provider agent responds with a service list
5. The access request agent sends the service list to the client
6. If the client needs service, it responds with an access request for a service
7. The access request agent sends the access request to the authentication agent

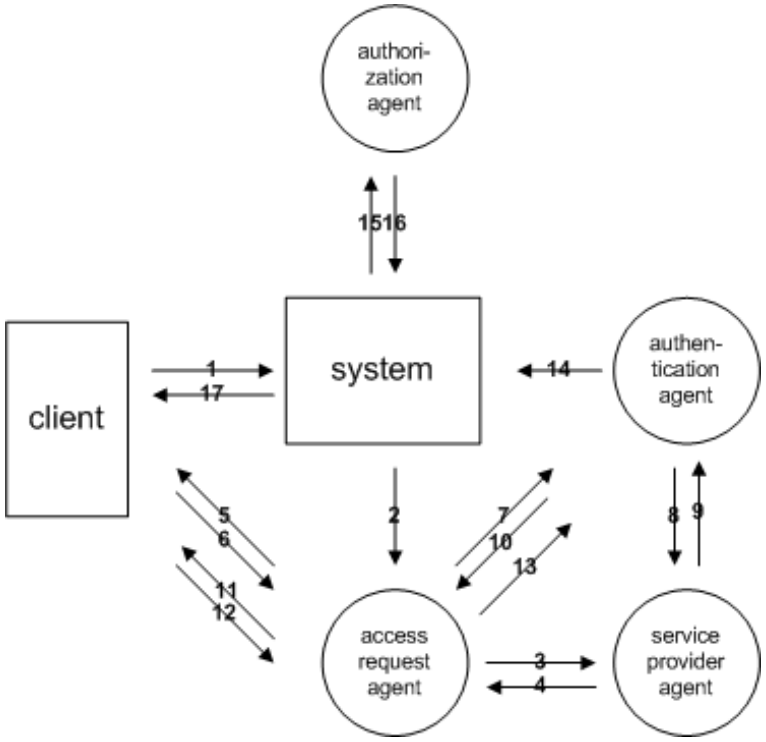

Figure 8.1: Architecture of authentication scheme

8. The authentication checks the validity of the service at the service provider

9. If the service is available, the service provider agent responds with authentication requirements

10. The authentication agent generates an authentication scheme for the requested service

11. The access request agent sends the authentication scheme to the client

12. The client sends its authentication request to the access request agent

13. The access request agent validates the authentication request at the authentication agent

14. The system is notified by the authentication agent about the result
15. The system queries the authorization agent about the authenticated client
16. The authorization agent gives a basic authorization to the authentication client
17. The system authenticates the client and provides it with the given authorization

The different steps in the scheme are recommended. Not all systems implementing this framework may necessarily have sharp separation of the different agents and some of the steps may melt together.

8.1.1 Authentication mechanisms

Whenever a user wants access to a chosen service, a proper authentication scheme is presented to him. The authentication scheme generated by the authentication agent depends on the security requirements of the requested service. It may take other factors into consideration, such as location and time. The authentication scheme supports basic one-factor authentication up to multi-factor authentication. The combination of factors is determined dynamically, so that the factors are not defined 'once for all' for the system. It means that if a service requires two-factor authentication, the two factors do not have to be defined exactly. The system will see if two authentication factors are given properly, and not necessarily which in details two. Of course the factors that can be selected are defined and known by the system.

There are two modes of dynamically generating the factor combination.

1. The authentication scheme can provide different possibilities (factors) in the authentication scheme sent to the user. Then the user can combine the factors of his own choice.

2. The system may also provide the user with a defined scheme of two-factors, but where the determination of the two factors are done by the system at the call time after taking in different factors, such as the time, client type (e.g. mobile device, personal computer), network connection type (e.g. wireless, LAN) etc.

In this way, the authentication scheme can be presented tailor-made to each case, increasing both the security and usability.

By doing so, a number of concerns with some of the factors are brushed aside. For example, a user wants to login to a service which requires two-factor authentication. In a static case, he is asked for a simple password and his finger print. As known, problems with biometric fingerprint sensors may occur (such as not recognizing the user's input) or maybe the user does not have access to a fingerprint sensor, for example, when logging on from a mobile device. In such case, the user would normally not be able to log in. Instead if the user can choose among factors, he will nearly always be able to find a proper combination for the given situation.

The backside is that the attacker also gets more factors to choose and can make a combination of the ones 'easiest' for him to break. However, although he can select the factors, the system will not be more vulnerable. This can be seen from the fact that two-factor authentication is deemed secure. When two-factor authentication is considered secure, it should not matter which two factors it is based on. If the attack was made easier by using a certain pair of two (real) factors instead of another, then how could two-factor authentication in itself be labelled secure, if it was only valid for some combinations of two factors? Hence, it may be concluded that offering the user the possibility to select, does not make the system more vulnerable, as long as the validation of each factor is done properly.

However, if it turns out, contrary to the above mentioned, that it is not reasonable to let the user select the combination, the other mode is available. There the authentication factors are well-defined, but created dynamically by the system at request time. In this way, it is still possible for the system to provide different factors depending on client information, location and time, but without giving the user the choice.

The system can determine that if the client is logging in from a known IP and in working hours, it should be asked for a certain number and type of factors, and another pair of factors, if the request comes from an unknown IP and/or outside the working hours. This is not the same as systems providing the same login scheme, no matter where, when and by whom the request comes.

The different techniques of authentication provided by the framework are listed below. In Chapter 6, the authentication mechanisms were discussed thoroughly along with benefits and drawbacks. Here they are only shown to get an overview of the mechanisms.

8.1.1.1 Passwords

This is the most common method of authentication for which the user knows a secret string, which may be a password, PIN code, pass phrase etc. The password should not be too small, and as far as possible, it should make use of both characters and numbers, along with being case sensitive.

8.1.1.2 Shared information (Challenge/Response system)

The user proves his identity by producing a response to a random challenge through the user's knowledge of a secret. In the simplest form a challenge/response system can be asking for password. However, usually the user must provide answers to a series of questions posed by the system in a challenge/response system. This is different from just sending a known password.

There are symmetric and asymmetric challenge/response schemes. An example of an asymmetric scheme is the digital signature on a random challenge message while a message authentication code on the challenge is an example of a symmetric scheme.

8.1.1.3 Digital Certificates using Public Key Infrastructure (PKI)

PKI enables users to authenticate their identity and private information exchange by means of public key cryptography. Each user has a key pair consisting of a public key and private key. The public key is given to those who want to verify the identity of the user. The private key is stored securely and only known to and used by the user. The private key is used to create a digital signature that uniquely identifies the holder of the private key. It can only be authenticated with the corresponding public key. For the issuance of keys along with digital certificates, a certification authority (CA) is needed. The certificate is used to verify that a public key belongs to a certain user.

8.1.1.4 Tokens

Tokens are (small) hardware devices through which the owner can prove his identity. The device may be in form of a smart card or integrated into, for example, key holders. Tokens are basically of three types which can be used for authentication.

Memory tokens contain authentication data in magnetic, electronic or optical form, such as USB, and do not have processing capacity.
Microprocessor tokens are like memory tokens, but extended with a microprocessor, where encryption algorithms can be implemented. Examples of such are smart cards.
Hand held password generators are used for generating one-time passwords or as challenge/response calculators allowing users to enter a challenge and calculate the response. One-time password generators can be time-synchronized, changing on time intervals, or event-based, providing a new password in response to a challenge or until some event has occurred.

Tokens represent the authentication mechanism being based on physical possession and it is often combined with other authentication forms, such as password or biometrics. Sometimes biometrics or passwords are used

on the tokens to activate it, for example on smart cards.

Private keys are often stored on tokens, as it prevents unauthorized parties from copying the keys if they get access to the computer.

8.1.1.5 Biometrics

Identification and authentication can be based upon unique biological characteristics. This is done through the use of technology for the recognition of individual human features, such as fingerprints, voices, eyes, faces. Biometric authentication mechanisms are based on either physiological features (eyes, hand, fingerprint) or behavioral features (such as signature, typing speed).

To carry out biometric authentication a scanner or device is needed which takes the biometric sample. It may for example be a video camera, microphone or fingerprint reader. Biometric authentication is used either for verification to prove the identity of a certain user, or to identify the user.

8.1.1.6 Location and time

Authentication may be based on the position of the user either using GPS devices to determine the geographical placement or based on IP addresses. An additional factor increasing the context-awareness of the authentication is time. Time intervals can determine if authentication is possible at all.

8.1.1.7 Out-of-band

In out-of-band authentication (OOBA), separate channels are used for transmitting the different authentication factors. Where the mentioned mechanisms are usually all transferred trough the common Internet pro-

tocols, OOBA may for example use the mobile network sending mobile messages, making calls etc.

A proper combination of all these authentication mechanisms can provide a very fine and context-aware authentication scheme uniquely attached to each service.

8.2 Three-tiered access control

In Chapter 7, the classical access control models were presented. None of these models alone can fully satisfy the requirements for an access control in a health care system. Medical data is often too complex and moreover exceptions occur from the standard access rights, for example in emergency situations. The complexity is shown in the fact that there are not just many roles, even the medical information, individually, requires differentiated access rights. Where granularity must be the finest possible, the speed also matters for such systems, which are often very large. Any identity based access control model could certainly give the best granularity, as permissions for each subject (user and/or the medical record) is set manually by the administrator, but in practical, it is impossible to control the access rights of every user in such a system. It would require almost as many administrators as there were users. Likewise both of the traditional access control mechanisms, DAC and MAC, are not applicable for medical systems.

The access control scheme in this framework is based on a three-tiered solution, with a combination of role-based access control, rule-based (or policy based) access control and view-based access control.

RBAC is unquestionably the best choice for a system where the users are divided into roles. In a medical system the roles would be such as patients, doctors, nurses, secretaries etc. RBAC is preferable because of its simplicity in administration, its flexibility and scalability:

Compared to the number of users there are much fewer roles than users. Many users would have exactly the same permission sets, which is easily

handled in a role-based access control model. When the rules for certain users (e.g. the patients) are updated, it would not be necessary to update every single user, but instead make the modification in the role once for all. New roles can easily be created, and assigned to the users. Users may have several roles, thus it is not necessary to remove one role before imposing another.

The emergency situations, in which exceptions occur, do not happen all the time, but are very rare compared to the normal (exception-free) usage of the system.

If there were no exceptions, RBAC alone would almost be sufficient as the access control model for medical systems. Therefore RBAC plays the central role in the three-tiered model proposed for the framework.

But the fact that exceptions occur cannot be neglected. Therefore RBAC is insufficient as the sole choice for the framework. However, by supplementing the RBAC model with the PBAC (or rule-based access control) model, these limitations can be eliminated.

RBAC assigns roles to the users. That is, it clarifies who the patients are, and who the doctors are. But not all doctors are 'doctors' for every patient (i.e., every patient is not being treated by every doctor). In the same way, not every patient is allowed to request a service, which may be available for other patients. The use of PBAC makes it possible to make these distinctions by setting up rules. Where RBAC provides a coarse granularity, PBAC fine-grains the access rights. For example, RBAC defines that here is a doctor, who theoretically should have the following rights over a certain patient for having the role 'patient'. But the PBAC model will then define if that very doctor does or does not have any relation to this patient. In other words, the RBAC tier decides which services any user can see, while the PBAC tier will decide whether and how the user can use the requested service.

Medical data itself needs to be classified and divided into categories according to its content and sensitivity. The role-and-rule-based approach makes it possible to define roles and rules, finally allowing the right persons to see/use the allowed information, but it does not allow content

classification. The same piece of medical data needs to be seen by different persons, in different ways, without having any influence on the data itself. That is, the data must not be duplicated or changed in order to make the content classification, and it should be possible to have the same part of the data in more views. For this reason the view-based access control is being used in the framework. The view-based access model makes it possible to make multiple virtual views of data, without making any changes to the original data. It is possible to make an emergency view, for the emergency cases, a patient view, only showing information the patient is allowed to see, a history view for reference purposes etc.

If CDAC or CBAC was used (or both were), it could give a content and context based authorization with an almost perfect granularity. Although, this is true in theory, in practise, they are not useful, and especially not for big systems. They would require so much labour which is simply impossible, in practise. Moreover, the system would be too slow. Hence, it is not realistic to base such systems on CDAC or CBAC.

8.3 Secure communication

As the framework is built upon the TCP/IP protocol suite, it uses the existing communications protocols. Chapter 5 explained the TCP/IP protocol suite and analyzed security at all layers. Here the communication protocols for the framework are mentioned with their secure replacements.

The protocols for e-mail messaging (IMAP, POP3, SMTP), file transfer (FTP), remote access using text communication (TELNET) and the web-protocol (HTTP) are all clear text protocols without security. When the data transfer does not require transfer security, these protocols are used. To fulfil the security requirements, each of these has a secure replacement.

The secure replacement for the FTP is the Secure FTP (SFTP). SFTP is an FTP connection using an SSL secured network connection. Thus, all data is sent secured. SSH has a sub-process to provide SFTP support. SCP (secure copy) achieves the same end result as SFTP, but it

uses a slightly different method. Most Linux systems have the SCP available and for Microsoft Windows many freeware clients are available, here among the excellent client called WinSCP.

The secure replacement for TELNET is SSH (secure shell). The connection here is secured using an SSL connection. The data can still be captured, but as it is encrypted it is of no use, and the decryption time is practically equal to infinity. Most Linux system have SSH built-in while there are many free and commercial clients Microsoft Windows, here among the famous free client Putty.

The e-mail messaging protocols (POP3, IMAP and SMTP) all have a secure replacement based on SSL (POP3S, IMAPS, SMTPS). Most servers already support the secure connection, and it is, therefore, only a question of changing the settings. Being based on an SSL connection the data in transit in encrypted. Hence it is not an end-to-end security solution. To get an end-to-end solution for e-mail, encryption software like PGP is used. As the messages are end-to-end encrypted, the encrypted data can be sent on insecure protocols.

For web-protocol HTTP, the secure replacement is the HTTPS, which is an HTTP connection using an SSL secured network.

Thus overall the SSL protocol plays the important role in securing the clear text protocols. It has the advantage that the encryption algorithms in SSL are 'automatically' updated and strengthened according to requirements and needs. This makes it a lasting solution, which does not have to be replaced often.

Secure remote access can be obtained using VPN technologies. There are two kinds of remote access needed. Either access to a specific application is wanted or access to the full remote system. The two major VPN technologies sufficient for this purpose are the SSL VPN and IPSec VPN.

SSL VPN can only be used for accessing a specific web-based application. The biggest advantage is that is clientless. IPSec VPN is used for remote access to the full system. It requires client software to be installed.

8.4 Case studies

In this chapter, application scenarios for the proposed framework will be presented.

Before looking at different cases, in which the different parts of the framework can be applied, a general example of a health care system using the framework is given.

The health care system in this example is a web-application. The web-based health care system offers a number of medical services here among access to the electronic patient journal.

The users of the systems are, for the purpose of simplicity, patients and doctors.

In order to use the system, the user directs a web-browser to the health care portal interface, i.e. the webpage of the system.

When the user enters the portal webpage, the system requests the available services and displays the service list for the user. (auth. framework, step 1-5)

The provided medical services are displayed at the portal interface (in form of a webpage, java applet etc.). The user selects the desired service, whereupon the authentication interface is displayed. (auth. framework, step 6-11)

If the authentication accomplishes successfully, the user is given access to the system. The user is given a basic authorization, enough to display the permitted services and actions inside the entered service. (auth. framework, step 12-17)

8.4.1 Authentication cases

A patient wants to get information about his doctor's vacation dates. He can enter the website and search for his doctor by himself in the overall register. This service does not require any authentication. He can call a service showing vacation information for his doctor only. This requires a simple password login, as some security is needed to protect the name of his doctor for the public.

A patient enters the service page from his home and wants to check his journal. This service requires a two-factor authentication. The user is presented for different "two-factor sets", i.e. he can select between a string password with either biometric authentication, digital certificate or a one-time password.

A doctor enters the service page and asks for a certain service which is only available in special wards at some hospitals and only during the working hours. First the authentication scheme validates the user's location from his IP along with the actual time configured at the server. If this passes the user is sent the appropriate authentication scheme if any needed, whether one or two-factor or more.

In all cases, all login information is sent through a secure connection using HTTPS.

8.4.2 Authorization cases

After passing the authentication and successfully being logged in to a service, it must be determined which parts of the service, (e.g. functions) or sub-services the authenticated user is allowed to view and use.

A doctor logs in to the system for browsing a user journal. As he has the role doctor, he can theoretically put in any personal number, to find the person's journal. When searching for person numbers he may see that a certain personal number exists in the database. To this part, only RBAC part is used. If he tries to access it, it must first be determined whether

or not he has the proper rights, (e.g. is the selected person a patient of that doctor, in actual treatment etc). Such rules are verified using rules which are set up in the PBAC mechanism. After getting access to the patients' journal, only the relevant part is shown to the doctor. This is determined according to the role, and is carried out using VBAC, as all journals are divided into several views.

The same doctor's secretary could do the same, but in the end, after getting access to the journal he would only be allowed to view the part of the journal relevant for the secretary. Thus the same journal is viewed by a doctor and his secretary, though the data shown is different.

In another application scenario a patient logs in to use some hospital service. As he is admitted for treatment a number of services are available. Some of them are available before the treatment and some after treatment. As being under treatment he can initially see the services, which is controlled using the RBAC mechanism. The management of which services the patient can see before, during and after the treatment is carried out using rules in the PBAC mechanism. In this case only the two upper tiers are used.

8.4.3 Secure connection

In the case where a patient enters the hospital webpage and searches for his doctor's vacation information, this is done anonymously. A normal HTTP connection is required. In the cases where a doctor logs in the service, the login information and the data afterwards must be transferred securely. This is done using HTTPS.

A patient wants to send his doctor an email. It is a requirement that all email messages from and to the doctor must be sent securely. The patient uses PGP for this purpose. He gets the doctors public key from the hospital webpage, and encrypts the message. In this way the message is end-to-end secured.

A patient logs into the hospital and submits personal data from a medical examination. The service must calculate some values and leave a receipt

in the clinical server. These are stored unencrypted at the clinic's mail server, but must be transferred securely. In these cases, SMTPS is used. Likewise if an authorized person wants to fetch a receipt, he needs to get it securely, which is done using POPS or IMAPS.

A home working doctor wants secure access to her office pc. She has a client installed on her home pc and creates a secure connection using IPSec VPN.

The same doctor is out for shopping and urgently needs to access some hospital service in order to help another doctor. She finds the nearest public Internet computer (e.g. library or internet bar etc.) and using SSL VPN from the hospitals site she can securely browse that application only, from the public machine.

8.5 Evaluation

The case studies illustrates how the framework can be applied in practical scenarios. To evaluate the usability of the framework in relation to EPJ, it must first be outlined which concrete requirements EPJ has. These were mentioned in Chapter 2 and they are quoted below.

The main points in [73] are the following:

- Proper user authentication.
- Providing different user rights and roles.
- Differentiated access control of both services and patient information.
- Being able to support emergency situations.
- Protection against data revealing to unauthorized parties.
- Security with data transfer, in form of integrity and confidentiality.

In [27] the same points are followed up with more details, especially the access control part is discussed.

There are three requirements for access to data that must be followed in any case.

1. Information can only be provided when it is necessary, e.g., when a patient is being treated.
2. Only the necessary and needed information must be read, if this is technically possible. This means that the part of a patient journal which is not relevant for the doctor should be kept hidden.
3. If the patient does not want his information or parts of it to be read, the system should be able to put such limitations.

For doctors and other parties, which the patient has approved, the above three rules apply. But for all others who can get access, point 1 and 2 are limited further, by adding that information should only be provided to the doctor from the treatment place, and that only information about the actual treatment is revealed.

However, besides the three main rules, there is a rule which allows the responsible party (e.g. the doctor) to get access to data in case of emergency, or if the patient is not able to take care of his affairs himself (e.g., mentally retarded persons).

Regarding the security, data must be encrypted at different levels. For sensitive data strong encryption is required. Moreover, stronger authentication for remote working places should be considered.

These are the main points of interest related to the security of EPJ. From the description of the framework developed in this thesis, it follows that the framework does fulfil these requirements. Therefore a longer discussion is not necessary at this point. It is explained briefly how the framework meets the conditions required.

The framework provides distinguished authentication mechanisms at sev-

eral layers. Thus, it can provide different authentication levels for remote working places. The authentication is not pre-defined, thus it is left to the actual implementation to define the number of and types of factors. Therefore the authentication scheme of the framework cannot be deemed insecure itself.

Regarding the access control scheme mentioned in [27], the framework supports user roles by RBAC, and by extending it with the PBAC all of the requirements (including the special rule) are met as explained earlier. Moreover differentiated access to the same information is possible with VBAC.

The transfer security is based on the standard protocols such as SSL, which are regarded secure by Datatilsynet, [24].

Hence, it follows that the framework complies with the requirements put up by [27] about EPJ.

Besides these, some other factors are considered relevant by stake holders in the definition of EPJ. However, a discussion of these are omitted as they do not have relevance for the framework itself, but rather the actual implementation. These are among others user-friendliness and system availability.

It should be noted, that traceability and event logging are not a part of the framework. These are minor things, and the structure of the framework (authorization scheme) easily allows adding central event logging. Though, this is left to the actual implementation.

Likewise, the discussion of the format and standard of the information stored in EPJ is not relevant for the framework.

8.5.1 Parallel studies

A number of papers and other frameworks considering different aspects of healthcare systems have been studied and analyzed to develope the proposed framework. In the following, these are briefly described, and

they can serve as parallel studies.

[16] describes a dynamic authorization framework which supports multiple authorization types. The framework uses a combination of Role-based Access Control and Dynamic Type Enforcement (DTE).

[38] presents a structured framework that considered many possible threats in wireless networks.

[10] describes a toolkit for analysis, specification, development and implementation of secure EPR systems.

[7] proposes a method for Internet-based exchange of patient data using XML, PKI and RBAC.

[65] presents a set of usage scenarios to explore the concept of patient controlled health records.

[84] discusses how to integrate many different authentication and access control policies with an existing group communication system.

[14] introduces a tailor-made challenge-response system for the WebBee environment which is a software framework supporting applications for mobile, handheld devices.

[8] analyzes security services and requirements, and discusses access control models.

[25] works out an example of an electronic health record system based on audit-based access control, and discusses the requirements and limitations of such access control.

[58] discusses security for medical databases, and presents a medical database development methodology that can help to overcome some of the problems.

[62] writes practical guidelines for how to protect one's computer and medical data transfer.

[69] proposes a two-tier access control for e-health portals.

[70] provides cost-efficient solutions to the issue of trust and systematic organization of users in web-based e-health portals.

[71] introduces a framework for authentication and authorization in e-health services, and highlights the importance of protecting the privacy of medical records.

[2] describes the design and implementation of certificate-based e-Health authentication and authorization architecture.

[4] identifies and discusses the two main information security technologies, PKI and biometrics. It presents a multi-layered framework based on these technologies.

[50] describes a contextual role-based access control authorization model for electronic patient records. It extends the RBAC model by the inclusion of contextual authorization mechanisms.

[34] proposes a secure identification scheme for accessing healthcare information systems. It secures the user's confidentiality by encrypting their ID (e.g., their social security number) into readable encrypted IDs.

[46] sketches a general data stream management structure and introduces security issues that need to be considered.

[33] describes the impact of privacy and security considerations on the Grid infrastructure design.

[52] analyzes the requirements that access control mechanisms must fulfil in the context of group communication, and defines a framework for supporting fine-grained access control in client-server group communication systems.

[40] discusses the legal basis and law requirements for implementing a framework, focusing on digital signature and encryption.

[18] gives an insight in the hospital security, crime and prevention.

Chapter 9

Conclusions

In this thesis, I have presented the project work made during the past months at DTU and the medical company SimeHealth. The first quarter of my time has been spent in fieldwork, primarily at SimeHealth.
Telemedicine and EPJ are very well-debated topics in Denmark as well as many other countries. At time being, the progress is so far only happening slightly, and unfortunately not according to the plans which even the Governmental publications admit [53] [78] [27]. The blame is not to be put at one place as there are many different parties involved. In this thesis several barriers have categorically been analysed. The fieldwork shows that the used applications, hardware as well as software, are often obsolete and should be substituted by applications especially designed and developed for the purpose. Many stakeholders have expressed their strong desire for renewing the systems. A common rallying point is needed, on which the parties can agree and start their development. The framework developed in this project can here serve as a good starting point.

The proposed framework
The framework for secure communication which has been proposed in this

thesis, is especially useful in systems dealing with sensitive information and having many users with different roles. Hence, it would be applicable in medical scenarios.

The framework covers three major aspects of communication, namely authentication, authorization and secure transfer. A secure scheme for each part has been presented.

Traditionally *authentication* is realized with predefined static authentication factors. Two-factor authentication is regarded secure today. This framework takes a different approach for authentication. It is not plausible to define the number of factors being used, just because two-factor authentication, at time being, is believed to be secure. Moreover, to predefine the number of factors to be used, is limiting the usefulness of the framework, as different systems need different authentication mechanisms. A specific system may require mechanisms depending on the actual situation and the service. The authentication scheme in the proposed framework provides several authentication mechanisms. It is suitable for systems with varying risk levels, caused by the sensitivity of the service or application, or by the user role, place/location, time etc. In this way, this authentication scheme distinguishes itself from the traditional schemes.

Classical *access control*, alone, insufficient for complicated systems and do not fully comply with the requirements of complex systems such as EPJ. RBAC (Role Based Access Control) is without doubt the best starting-point for systems with users and roles. However, even RBAC needs extensions in order to support special cases and emergency situations. By extending RBAC with PBAC (Policy Based Access Control or Rule Set Based Access Control), it is possible to add rules to cover situations that "do not follow the rules", e.g. emergency situations for health care systems. The framework presents a three-tier access control solution with RBAC and PBAC. By adding the VBAC (View Based Access Control) as a third layer, access differentiation which is necessary in any information system, is possible.

In theory, the three-tier model will suffice and can provide need-to-know access to some extent. Though for the practical implementation, the

question of granularity depends much on the labour. The finer granularity one wants, the more labour would be needed. At the same time it is a trade-off between obtaining a super-fine granularity level and the availability and speed of the system. The presented model is flexible. It provides the required functionality, while leaving the exact decision about granularity to the individual implementation. Even in one system, one section may have many exceptions, but without any need of information differentiation, while another would be using RBAC only.

Many basic *communication protocols* are somewhat insecure. However, extensions exist for almost all of them to make them secure. Among these, many are widely used. Hence it would not be sensible to leave these, and start writing one's own protocol while using these gives better compatibility and higher availability in systems. Concurrently with the security level being raised, these protocols are also updated and developed. Therefore, it is safe to base the communication on these. Especially, SSL plays a great role, as the large majority of the security extensions are based on this. SSL consists of several algorithms dealing with confidentiality, integrity and authentication.

The framework presents a structure on an abstract level which can be implemented in any operating system. As mentioned before, the compatibility is increased by using standard protocols and others that are well-defined in RFC documents. As there is no dependency to specific applications or hardware, different operative systems running on different hardware platforms do not become obstacles in the approach. This gives scalability and portability.

The framework is built in modules (in fact it is a 3-in-1 framework), which gives flexibility. This means that a module (e.g. for authentication) can be used alone or included in other systems without necessarily having to implement the other modules. Changes made in one module will not affect the other modules, and new modules can easily be added to the framework. Hence, it is fit for both add-ons and ensures security updates when security flaws are found, without having to alter the systems. The modular structure also paves the way for easy enhancements and modifications.

Telemedicine in future

A survey among doctors made in [85] shows that many (young) doctors are actually favourably disposed to the use of telemedicine and EPJ in their work, and believe that it will benefit them. At the same time, the fear of too complicated systems is also noticeable. The survey shows that the systems must be simple and not have too many complex functions. This is the general opinion among doctors and physicians observed in both the fieldwork and many surveys. Of course, doctors want the best for the patients and they would like having a stronger tool in form of telemedicine and EPJ. Their anxiety is also comprehensible as they do not want to jump into something they cannot control, i.e., complicated systems. Not being qualified enough, some may even risk their job in the long term. The doctors need to earn their living, and therefore their economical situation must, understandably, be secured. Especially the economical situation for private clinicians may be negatively affected by the implementation of telemedicine and EPJ under the current laws. This is due to the fact that the introduction of IT in the health sector is not just beneficial to the doctors, but also the citizens, as the citizens, to some extent, become self-propelled. According to the current governmental rules both hospitals and clinics are paid per visit per patient. The fact that patients can communicate with and get treated by the doctors from home, would at the time, not be considered in the doctors' payment. The county hospital in Vejle, which is the leading hospital in the field of telemedicine, has expressed similar economical worries. The hospital estimates that introducing telemedicine cause a yearly loss of 14.000 krones per patient, [28]. However, this discussion is beyond the scope of this thesis.

Besides this, implementing telemedicine will cause great costs due to development and design of proper software, providing necessary hardware, training of the personnel etc. Once these barriers are overcome, telemedicine can be predicted good chances. The use of Internet has become so widespread that it is not imaginable that the use of IT should be a barrier for the future propagation of telemedicine.

I definitely think that the barriers will be overcome in the near future. The topic is very hot, not only in Denmark, but many other countries, here among all the Scandinavian countries, UK and the United States.

The governments will, sooner or later, be compelled to deal more wholeheartedly with the case, especially the regulatory and economical problems. The recent Governmental publication [27] shows that steps are being taken to improve the situation, and *Sundhedsstyrelsen* has also admitted that something must be done [28], and that they are working on a solution.

Upon all this, telemedicine seems to have a bright future. During the project, I have definitely widened my horizon in this field, and added to my knowledge about security. Especially, the fieldwork has been beneficial, through which I have met and communicated with many different parties, here among hospitals, investors, professionals, researchers, and salesmen. I have seen many sides which would not be possible through theoretical studies only. I personally plan to go further in this field. This plan seems to be possible as I have been offered a job at SimeHealth as a result of my fieldwork during this project.

Appendix A

Abbreviations

3DES	Triple DES
ACL	Access Control List
AES	Algorithm Encryption Standard
AH	Authentication Header
CA	Certificate Authority
CCP	Compression Control Protocol
CHAP	Challenge Handshake Authentication Protocol
DES	Data Encryption Standard
DNS	Domain Name System
DoS	Denial of Service
DSN	Dedicated Secure Network
DTU	Denmark's Technical University
ECP	Encryption Control Protocol
EMR	Electronic medical records
EPJ	Electronic Patient Journal
ESP	Encapsulating Security Payload
FTP	File Transfer Protocol
HMAC	Keyed Hashing for Message Authentication
HTML	HyperText Markup Language

HTTP	HyperText Transfer Protocol
HTTPS	HTTP over SSL
ID	Identication
IEEE	Institute of Electrical and Electronic Engineers
IETF	Internet Engineering Task Force
IKE	Internet Key Exchange
IMAP	Internet Messaging Access Protocol
IP	Internet Protocol
IPSec	IP Security
LAN	Local Area Network
L2TP	Layer-2 Tunneling Protocol
MAC	Message Authentication Code
MD5	Message Digest algorithm 5
MIME	Multi-Purpose Internet Mail Extension
NAT	Network Address Translation
PDA	Personal Digital Assistant
PGP	Pretty Good Privacy
PKI	Public Key Infrastructure
POP3	Post Office Protocol version 3
PPP	Point-to-Point Protocol
PPTP	Point-to-Point Tunneling Protocol
RC4	Rivest Cipher
RFC	Request For Comments
RSA	Rivest Shamir & Adleman
SA	Security Association
SHTTP	Secure HTTP
SLIP	Serial Line Internet Protocol
S/MIME	Secure Multi-Purpose Internet Mail Extension
SMTP	Simple Mail Transfer Protocol
SRP	SSL Record Protocol
SSH	Secure Shell
SQL	Structured Query Language
SSL	Secure Sockets Layer
TCP	Transport Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
URL	Uniform Resource Locator
VPN	Virtual Private Network
WLAN	Wireless Local Area Network
WWW	Wireless Local Area Network

Appendix B

Security principles

Traditionally when discussing information security confidentiality, integrity and availability (also called the CIA-principles) are the most important principles. Short definitions of each follow below.

B.1 Confidentiality

When information is transferred from one place to another one often wants to secure the data as such that it is not readable by unauthorized instances. This is called confidentiality in terms of information security. The word also appears in real life where for instance doctors have professional secrecy and may not disclose information about their patients to a third (unwanted) person. A much used way for preserving confidentiality is cryptography. With cryptography information is encrypted and can only be decrypted by an authorized person that has a decryption key. The two major fields in cryptography are symmetric and asymmetric encryption respectively having shared and different keys. Cryptography is

system independent and no matter where the data is placed the information stays unreadable as long as it is not decrypted. There are also system-dependent mechanisms that may be even more secure but there are always a risk of data getting totally visible if the system is bypassed, turned off or fails.

Mostly confidentiality is about controlling access to information but sometimes a need for hiding information existence is just as important or even more.

B.2 Integrity

While confidential information transfer assures that data is not disclosed it does not say anything about whether data are authentic. That is whether the data has come unchanged from the sender and therefore can be trusted and relied upon or if it has been altered. The assurance of having authentic information is called integrity. Integrity concerns both reliability of the content and authentication which is reliability of the source.

Confidentiality and integrity respectively assure the information accessibility and trustworthiness. Both terms concern the information itself being transmitted but the difference between confidentially is important to note as integrity cannot just be deduced from confidentially as one might initially think.

Say, you want to send a string which you have encrypted. On account of the encryption the string is not accessible to unauthorized persons but an intruder may disturb the bits in the information stream and thus the string is not reliable anymore. When you transfer large files it may also be the case that bits get lost on the way. Therefore it is necessary to ensure that information is not compromised in any way, whether by falsification or data loss. The integrity is a much more complex principle then confidentiality as it often relies on a number of assumptions in practise.

B.3 Availability

Availability refers to the availability of the information resources when they are needed. For example, a firm is selling products through their website. If the server is down the website is unavailable to the customers thus bringing a financial loss to the firm and disturbance to the customers. Attacks with the purpose of blocking availability are called denial of service attacks. Such attempts are often hard to detect as deliberate attempts cannot always be distinguished from others easily.

Bibliography

[1] Carlisle Adams and Steve Lloyd. *Core PKI Services, http://technet.microsoft.com/en-us/library/cc700808.aspx.* Last visit: 30-10-08.

[2] Fahed Al-Nayadi and Jemal H. Abawajy. *An Authentication Framework for e-Health Systems.* School of Engineering and Information Technology, Deakin University, Australia.

[3] Amtsrådsforeningen. *Fælles arkitekturprincipper for EPJ.* Amtsrådsforeningen, 2005.

[4] Meletis A. Belsis Raouf N G. Naguib Peter Every Ashish Dwivedi, Rajeev K. Bali and Nahy S. Nassar. *Towards a practical healthcare information security model for healthcare institutions.* School of Mathematical and Information Sciences, Coventry University, UK.

[5] Phishing attack evades bank's two-factor authentication. *http://www.theregister.co.uk/2007/04/19/phishing_evades_two-factor_authentication.* Last visit: 28-10-08.

[6] Frederic Avolio. *Firewalls and Internet Security, http://www.cisco.com/web/about/ac123/ac147/ac174/ac200/about_cisco_ipj_archive_article09186a00800c85ae.html.* Last visit: 30-10-08.

[7] Web based secure access from multiple patient repositories. *Jun Choe, and Sun K. Yoo.* International Journal of Medical Informatics 77 (2008) 242-248.

[8] John Mike Davis Bernd Blobel, Ragnar Nordberg and Peter Pharow. *Modelling privilege management and access control.* International Journal of Medical Informatics 75 (2006) 597-623.

[9] Matt Bishop. *Computer Security - Art and Science.* Addison-Wesley, 2003.

[10] Bernd Blobel. *Advanced tool kits for EPR security.* International Journal of Medical Informatics 60 (2000) 169-175.

[11] Hannes Boesch and Gianluca Airaghi. *Secure Transfer of Medical Data over the Internet: From Regulatory Data Protection Jam to Framework-Based Requirements.* Burg G (ed): Telemedicine and Teledermatology 32 (2003) 71-75.

[12] Tom Bowers. *The insecurity of two-factor authentication, http://searchsecurity.techtarget.com/tip/0,289483,sid14_gci968784,00.html.* Last visit: 28-10-08.

[13] Dr. Levente Buttyan. *Internet Security Architectures and Protocols, http://www.hit.bme.hu/ buttyan/courses/BMEVIHI9367/.* Last visit: 28-10-08.

[14] Sugih Jamin Sarit Mukherjee Byung S. Yang, Soren Dreijer and Limin Wang. *Secure Communication Framework for Mobile Devices.* Department of Electrical Engineering and Computer Science, The University of Michigan, USA.

[15] Malaysia car thieves steal finger. *http://news.bbc.co.uk/2/hi/asia-pacific/4396831.stm.* Last visit: 28-10-08.

[16] Ramaswamy Chandramouli. *A Framework for Multiple Authorization Types in a Healthcare Application System.* Computer Security Division, ITL NIST, Gaithersburg, MD 20899.

[17] Cisco. *Evolution of the Firewall Industry, http://www.cisco.com/univercd/cc/td/doc/product/iaabu/centri4/user/scf4ch3.htm.* Last visit: 30-10-08.

[18] Terry Cocks. *A Brief Examination of Security and Crime Prevention in Hospitals*. Camden Police, London, UK.

[19] Douglas E. Comer. *Internetworking with TCP/IP*. Prentice Hall, 1995.

[20] Pretty Good Privacy Corporation. *http://www.pgp.com/about_pgp_corporation/corporate_overview/index.html*. Last visit: 30-10-08.

[21] Phishers crack two-factor authentication. *http://www.vnunet.com/vnunet/news/2160250/phishers-crack-two-factor*. Last visit: 28-10-08.

[22] Lorrie Faith Cranor and Simson Garfinkel. *Security and Usability*. O'Reilly, 2005.

[23] Camilla Daasnes. *Persondataloven, lægemiddelforsøg og datasikkerhed, http://www.ugeskriftet.dk/LF/UFL/2003/16/pdf/VP39965.pdf*. Last visit: 28-10-08.

[24] Datatilsynet. *http://www.datatilsynet.dk*.

[25] M. A. C. Dekker and S. Etalle. *Audit-Based Access Controlfor Electronic Health Records*. Electronic Notes in Theoretical Computer Science 168 (2007) 221-236.

[26] Connected: An Internet Encyclopedia. *IP Packet Structure, http://www.freesoft.org/CIE/Course/Section3/7.htm*. Last visit: 30-10-08.

[27] Jesper Fisker. *Informationssikkerhed - vejledning for sundhedsvæsenet 2008, http://www.sst.dk/publ/Publ2008/SDSD/Infosikkerhed_vejl08.pdf*. Sundhedsstyrelsen, 2008.

[28] Steffen Fog. *Kassetænkning står i vejen for telemedicin, http://www.computerworld.dk/art/41545?cid=2&q=EPJ&a=cid&i=2&o=21&pos=22*. Last visit: 28-10-08.

[29] Dansk Selskab for Klinisk Telemedicin. *http://www.dskt.dk/*.

[30] Center for Sundhedstematik. *http://www.cfst.dk/*.

[31] Simson Garfinkel. *Web Security & Commerce*. OŔeilly, 1997.

[32] Dieter Gollmann. *Computer Security*. John Wiley & Sons, Ltd, 2006.

[33] Silvia D. Olabarriage Guido J. van't Noordende and Matthjs R. Koot. *A Trusted Data Storage Infrastructure for Grid-Based Medical Applications*. Informatics Institute, University of Amsterdam, THe Netherlands.

[34] C. M.Hsu H. M. Chao, S. H Twu. *A Secure Identification Access Control Scheme for Accessing Healthcare Information Systems*. Department of Electrical Engineering, Chung Yuan University, Taiwan.

[35] Fran Howarth. *Biometrics: the legal challenge, http://www.theregister.co.uk/2005/02/09/biometric_legal_issues/*. Last visit: 28-10-08.

[36] Lars Hulbæk. *Hvad er telemedicin?, http://www.regionsyddanmark.dk/dwn45949*. Last visit: 28-10-08.

[37] Netilla Networks Inc. *A Functional and Cost Comparison of VPN Solutions: SSL vs. IPSec, http://www.netilla.com/downloads/netilla-ssl-vpn.pdf*. Last visit: 28-10-08.

[38] Bell Labs Innovations. *The Bell Labs Security Framework: Making the Case for End-to-End Wi-Fi Security*. Lucent Technolgies.

[39] Michael Daye Jr. *Virtual Private Networks: IPSec vs. SSL, http://www.infosecwriters.com/text_resources/pdf/VPN_MDaye.pdf*. Last visit: 28-10-08.

[40] Rosa Julia-Barcelo and Thomas Vinje. *Towards a Europen Framework for Digital Signatures and Encryption*. Computer Law and Security Report, Vol. 14, No. 2, 1998.

[41] Fredrik Kilemark. Secure working from home in an industrial context. Master's thesis, Technical University of Denmark, 2004.

[42] Andrew J. Klosterman and Gregory R. Ganger. *Secure Continous Biometric-Enhanced Authentication, http://www.pdl.cmu.edu/PDL-FTP/Secure/CMU-CS-00-134.pdf.* Last visit: 28-10-08.

[43] RSA Laboratories. *http://www.rsa.com/.*

[44] RSA Laboratories. *RC4, http://www.rsa.com/rsalabs/node.asp?id=2250.* Last visit: 29-10-08.

[45] RSA Laboratories. *What is Diffie-Hellman, http://www.rsa.com/rsalabs/node.asp?id=2248.* Last visit: 30-10-08.

[46] Wolfgang Lindner and Jorg Meier. *Towards a Secure Data Stream Management System.* MIT, Cambridge, USA.

[47] Camelot Information Technologies Ltd. *Differentiating Between Access Control Terms, http://www.windowsecurity.com/uplarticle/2/Access_Control_WP.pdf.* Last visit: 30-10-08.

[48] Tsutomu Matsumoto. *Gummy and Conductive Silicone Rubber Fingers.* 2003, Yokohama National University, Japan.

[49] Nagendra Modadugu and Eric Rescorla. *The Design and Implementation of DTLS, http://crypto.stanford.edu/ nagendra/papers/dtls.pdf.* Last visit: 28-10-08.

[50] Gustavo H. M. B. Motta and Sergio S. Furuie. *A Contextual Role-Based Access Control Authorization Model for Electronic Patient Records.* IEEE Transactions on Information Technology in Biomedicine Vol. 7, No. 3, Sep. 2003.

[51] Ari Niemi. *End-to-end web security - protocols overview.* Department of Computer Science, Science University of Helsinki, Finland.

[52] Cristina Nita-Rotaru and Ninghui Li. *A Framework for Role-Based Access Control in Group Communication Systems.* Department of Computer Sciences, Purdue University.

[53] Indenrigs og Sundhedsministeriet. *National IT-strategi for sundhedsvæsenet 2003-2007, http://www.sst.dk/upload/nat_itstrategi03_07.pdf.* 2003.

[54] IT og Telestyrelsen. *http://www.itst.dk/.*

[55] Karen Olsen. *The Impact of the FCC's Open Network Architecture on NS/EP Telecommunications, http://csrc.nist.gov/publications/nistpubs/800-11/titleona.html.* NIST, 1995.

[56] ROI Comparison: SSL VPNs or IPSec for Remote Access? *http://www.findwhitepapers.com/whitepaper56/.* Last visit: 28-10-08.

[57] DES Encryption Overview. *http://www.tropsoft.com/strongenc/des.htm.* Last visit: 29-10-08.

[58] Georges J. Pangalos. *Secure medical databases: design and operation.* International Journal of Bio-Medical Computing 43 (1996) 53-60.

[59] Compex White paper. *http://www.cpx.com/whitepapers/ Compex%20SPI%20Firewall.pdf.* Last visit: 30-10-08.

[60] Schlumberger White paper. *Virtual Private Networks Solutions for Remote Access - A comparison of IPSec and VPN, http://www.slb.com/media/services/consulting/infrastructure/ whitepaper_vpnsra.pdf.* Last visit: 28-10-08.

[61] SonicWall White paper. *IPSec vs. SSL VPN: Transition Criteria and Methodology, http://www.sonicwall.com/downloads/WP_SSLVPN_vs_IPSec_102907.pdf.* Last visit: 28-10-08.

[62] Marek Czubenko Piotr Kasztelowicz and Iwona Zieba. *Security of medical data transfer and storage in Internet, http://www.am.torun.pl/ pekasz/telemed-secur.pdf.* Last visit: 30-10-08.

[63] SSL protocol overview. *http://www2.rad.com/networks/2001/ssl/ over.htm.* Last visit: 28-10-08.

[64] Ronald L. Rivest. *The RC5 Encrpytion Algorithm, http://people.csail.mit.edu/rivest/Rivest-rc5rev.pdf.* MIT Laboratory for Computer Science, Cambridge. Last visit: 29-10-08.

[65] Lillian Røstad. *An Initial Model and a Discussion of Access Control in Patient Controlled Health Records.* Norwegian University of Science and Technology, Trondheim, Norway.

[66] Sachin Satthy. *Introduction to Spyware Keyloggers, http://www.securityfocus.com/infocus/1829.* Last visit: 28-10-08.

[67] Bruce Schneier. *The Failure of Two-Factor Authentication, http://www.schneier.com/blog/archives/2005/03/the_failure_of.html.* Last visit: 28-10-08.

[68] Six Biometric Devices Point The Finger At Security. *http://www.networkcomputing.com/910/910r1side1.html.* Last visit: 28-10-08.

[69] Qian Lu Shuo Lu, Yuan Hong. *Access Control in e-Health Portal Systems.* Department of Computer Science, Concordia University, Canada.

[70] Qian Lu Shuo Lu, Yuan Hong. *Securing Telehealth Applications in a Web-Based e-Health Portal.* Department of Computer Science, Concordia University, Canada.

[71] Vidyasagar Potdar Elizabeth Chang Song Han, Geoff Skinner and Chen Wu. *New Framework for Authentication and Authorization for e-Health Service Systems.* School of Information Systems, Curtin University of Technology, Australia.

[72] William Stallings. *SSL: Foundation for Web Security, http://www.cisco.com.* Last visit: 28-10-08.

[73] Sundhedsministeriet. *Redegørelse om patientrettigheder i forbindelse med indførelse af EPJ, http://www.im.dk/publikationer/epj2001/kap1.htm.* 2001.

[74] Sundhedsministeriet. *Telemedicin - Redegørelse vedrørende anvendelse af telemedicinske løsninger i det danske sundhedsvæsen, http://www.sum.dk/publikationer/telemedicin/index.htm.* 2001.

[75] Sundhedsstyrelsen. *http://www.sundhedsstyrelsen.dk/.*

[76] EPJ.dk Region Syddanmark. *http://www.epj.dk.*

[77] Andrew. S. Tanenbaum. *Computer Networks*. Prentice-Hall PTR, 1996.

[78] Teknologirådet. *Telemedicin - En vej til et bedre sundhedsvæsen, http://www.tekno.dk/pdf/projekter/974.pdf*. 1997.

[79] Playing the Remote Access Game: Will IPSec or SSL VPNs fit your needs? *http://www.indevis.de/dokumente/ipsec_vs_sslvpn_engl.pdf*. Last visit: 28-10-08.

[80] Iain Thomson. *Hackers crack two-factor security, http://www.vnunet.com/vnunet/news/2139253/two-factor-authentication*. Last visit: 28-10-08.

[81] Liam Tung. *SMS two-factor authentication dead in 3 years: NAB, http://www.zdnet.com.au/news/hardware/soa/SMS-two-factor-authentication-dead-in-3-years-NAB/0,130061702,339284387,00.htm*. Last visit: 28-10-08.

[82] Jeff Tyson. *How Virtual Private Networks Work, http://computer.howstuffworks.com/vpn.htm*. Last visit: 30-10-08.

[83] VeriSign. *http://www.verisign.com/*.

[84] Cristina Nita-Rotaru Yair Amir and Jonathan R. Stanton. *Framework for Authentication and Access Control of Client-Server Group Communication Systems*. Department of Computer Science, Johns Hopkins University, USA.

[85] Halvard Øysæd and Roy Otto Kleiv. Study of aspects in electronic patient journal relevant to the medical office agreement. Master's thesis, Agder University College, 2003.

[86] Melihat Zengin. *Nyt sikkerhedsfokus gør digital signatur meget bedre, http://www.computerworld.dk/art/43570?a=related&i=45197&bottom*. Last visit: 28-10-08.