HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Electrical and Communications Engineering

Juho Heikkilä

Secure Digital Pseudonyms for Privacy and Liability

Master's Thesis submitted for approval May 28th, 2002 for the degree of Master of Science

Supervisor Prof. Teemupekka Virtanen

Instructor Ursula Holmström, M.Sc.

HELSINKI UNIVERSITY OF TECHNOLOGY ABSTRACT OF THE MASTER'S THESIS

Author and name of the thesis:

Juho Heikkilä: Secure Digital Pseudonyms for Privacy and Liability

Date: 28.05.2002 Number of Pages: 56

Faculty: Dep. of Electrical and Communications Engineering

Professorship: Interactive Digital Media (T-111)

Supervisor: Prof. Teemupekka Virtanen

Instructor: Ursula Holmström, M.Sc.

The digital realm presents many new challenges to users' privacy as well as trustworthiness. An anonymous network lacks liability and trust, which are two important aspects that must be accounted for in order to build network communities or commerce that people can rely on.

In some cases trust and liability can be achieved using electronic ID cards, like FINEID, but the use of any single identifier in all online actions makes it extremely easy to merge databases and gather information about the users. In addition, the person's real identity is extremely hard to change if needed. Another problem of using the real identity is that revealing it makes it fairly easy for a network acquaintance to find out such attributes as physical address, phone number etc. Thus it would be conceivable that, in order to protect their privacy, not many people would like all their network activities easily traceable. However, people would like to be able to trust what others say about themselves, in other words, for them to prove these attributes.

I searched for an alternative identity model, and in particular, I have looked at the problem of anonymous chat rooms, where it is very difficult to know if the person at the other end is what he/she claims, or if he/she is the same person with whom one talked previously. One sad event during the writing of this thesis was a real case of such abuse, where an older man presented himself as much younger, persuaded young girls on a date and then raped them.

This thesis starts out by mapping the surrounding problems, namely how people see names and identifiers, and what actually is privacy and why it should be protected. Also, I present some existing technologies that are used to build an architecture that could solve at least some of the problems, namely that of a single identity, traceability and proving attributes of oneself.

I have built a model that allows the users to create multiple identifiers for themselves, certify their age and gender using a Trusted Third Party to write certificates to the identifier and then prove these attributes to other users who also trust the TTP in question. In addition, a simple subset is implemented in such a fashion as to make it easy to move the service on top of existing servers with minor modifications to the client.

Keywords: privacy, identity, recognition, identifier, authorization, certificate

TEKNILLINEN KORKEAKOULU DIPLOMITYÖN TIIVISTELMÄ

Tekijä ja työn nimi:

Juho Heikkilä: Secure Digital Pseudonyms for Privacy and Liability

Päivämäärä: 28.05.2002 Sivumäärä: 56

Osasto:
Sähkö- ja tietoliikennetekniikan osasto

Professuuri:
Vuorovaikutteinen Digitaalinen Media (T-111)

Valvoja: Prof. Teemupekka Virtanen

Ohjaaja: Ursula Holmström, DI

Digitaalinen maailma luo monia uusia haasteita niin käyttäjien yksityisyydelle kuin luotettavuudellekin. Anonyymissä verkossa ei ole helppoa luoda vastuullisuutta ja luottamusta, jotka kuitenkin ovat kaksi tärkeää käsitettä jotka on otettava huomioon kun rakennetaan verkkoyhteisöitä tai sähköistä kauppaa johon ihmiset voivat luottaa.

Joissain tapauksissa luottamusta ja vastuullisuutta voidaan luoda jo käyttämällä sähköistä tunnistusta kuten HST-korttia, mutta jos kaikessa verkkoasioinnissa käytetään yhtä ainoaa korttia ja tunnistetta, on hyvin helppoa yhdistää eri tietokantoja ja kerätä tietoja käyttäjistä. Lisäksi, tällöin käyttäjän identiteetti on hänen oikea identiteettinsä, jota on vaikea vaihtaa. Lisäksi oikean identiteetin käytössä on se ongelma että sen avulla on varsin helppo saada selville sellaisia tietoja kuin osoite tai puhelinnumero. Niinpä olisi oletettavissa, että suojatakseen yksityisyyttään kovin monet ihmiset eivät halua käyttää kaikessa asioinnissa helposti jäljitettävää tunnistetta. Kuitenkin monet haluaisivat voida luottaa siihen mitä toinen itsestään verkossa väittää, eli toisin sanoen että tietoja voisi saada varmennettua.

Etsin siis vaihtoehtoista tunnistemallia ja erityistapauksena olen tutkinut anonyymejä keskusteluhuoneita; näissä on hyvin hankala tietää onko toinen se tai sellainen kuin väittää olevansa. Työn aikana esille nousi tositapaus jossa vanhempi mies oli esiintynyt nuorena, houkutellut nuoria tyttöjä treffeille ja sitten raiskannut heidät.

Tässä työssä on aluksi katsasteltu ympäröiviä ongelmia, erityisesti sitä kuinka ihmiset näkevät nimet ja identiteetit, mitä oikeastaan on yksityisyys ja miksi sitä pitäisi suojata. Esittelen joitain olemassa olevia tekniikoita joita sitten käytän rakentaakseni arkkitehtuurin jolla ainakin osa esille tulleista ongelmista voitaisiin ratkaista, erityisesti ainoan tunnisteen ongelma, jäljitettävyys ja ominaisuuksien todistaminen.

Työssä on rakennettu malli ja koodattu sen pohjalta osittainen toteutus jolla käyttäjät voivat luoda itselleen useampia tunnisteita, varmentaa ikäänsä ja sukupuoltansa käyttäen luotettavia kolmansia osapuolia jotka kirjoittavat digitaalisia sertifikaatteja tunnisteille ja lopulta näitä käyttäen todistaa näitä ominaisuuksiaan muille sellaisille käyttäjille jotka luottavat kyseiseen kolmanteen osapuoleen. Toteutus on pyritty rakentamaan niin että palvelu voidaan siirtää olemassa olevien palvelinten päällä toimiviksi vain asiakasohjelmistoa hieman muuttamalla.

Avainsanat: yksityisyys, identiteetti, tunnistus, tunniste, pääsynvalvonta, sertifikaatti

TABLE OF CONTENTS

TABLE OF FIGURES

Acknowledgements

This master's thesis has been written while working as a research assistant in the Telecommunications and Multimedia Laboratory at Helsinki University of Technology and at the Helsinki Institute for Information Technology. The work was started in the TeSSA project and it continued later while I began working in the STAMI project at HIIT.

I want to thank the professor of my major, Tapio Takala for allowing me to write this thesis under another professor in the same laboratory. Originally this was Arto Karila whom I must thank for both starting this thesis with me and for originally inspiring me to join the laboratory and giving the seeds for this idea. The third professor associated with the thesis, Teemupekka Virtanen also deserves thanks for allowing and helping me to finish the thesis under his guidance.

My instructor, Ursula Holmström deserves thanks for giving ideas and for reading and commenting the work, especially at the early stages of the writing process.

Yki Kortesniemi deserves special thanks for both allowing me to continue the work under STAMI and for numerous comments and readings at the later stages of writing. His support really helped me to finish this thesis.

I also want to thank all the people from both projects who have been helpful and given new insights of the problem at hand, as well as helped me with programming or debugging problems.

Further, I want to thank the Data Protection Ombudsman Reijo Aarnio for insightful discussions, and Professor Jukka Kemppinen for introducing me to Mr. Aarnio. Prof. Kemppinen deserves further thanks for introducing me to Information Age trilogy by Manuel Castells and his inspirational and insightful visions.

Finally, thanks to all those, who have in many ways helped or supported me while I was working on this project. You are too numerous to mention, but all help was appreciated.

Helsinki, May 27th, 2002

Juho Heikkilä

Chapter 1

Introduction

Humans in a crowd are quite anonymous, but never totally. A face seen among others may be recognized, but it doesn't tell who that person is. Later that face can be recognized as familiar, human senses are well developed for such purposes. Computers and digital networks, however, lack the fuzzy pattern recognition skills that humans get in their genome and must rely on exact true/false data. Therefore, it is only logical that traditional digital systems have been built around the concept of identification rather than recognition. Computers have no feelings and they do not make mistakes, not in the sense that humans do anyway.

We humans rely in increasing amounts on our computers and digital networks. In the past decades many behind-the-scenes transactions such as in banking, taxation, insurances, public records etc. have mostly been digitized and are now maintained in a computer, even by a computer. At the turn of the millennia the penetration of computers into our society has reached an even higher level: many homes have an internet connection, people send electronic mail, surf web pages, call on mobile phones, even surf the web on their mobile phones. The latest trend seems to be location aware services; when ordering a taxi, for example, the mobile phone can tell the service the exact location from which the taxi is ordered. Should all such services be based on an assumption that the user is always identified, the user's privacy would be gravely threatened.

Anonymity on the network is often seen as troublesome. Some say that it is the devil, while others see it as a rightful protection of their privacy. Since anonymity makes it harder for troublemakers to be caught, it does promote opportunities. On the other hand, useful tools – like knives – are usually not outlawed just because they can be used for something illegal as well.

This thesis looks for an alternative to strong identification. Trying to go back to the recognition scheme it seeks to preserve the privacy of the user, while still maintaining a certain level of trustworthiness and liability. Since analog recognition is not easily accomplished using digital computers, I have tried to model recognition in this case with a system providing pseudonymous identification, in other words, hiding the true identity of the user behind a protective curtain of trusted third parties and service providers. These may certify things about the user, even guarantee some of them, yet unless certain conditions are met, never tell who the person really is. Such certificates could include information like age, student status, gender, licenses, etc. Basically anything that today is used with identifying digital certificates could be used with pseudonymous identifiers as well. The third party would act as a mediator and a liability guarantee. Using these attribute certificates users could alleviate the mistrust of the people they meet online since they could prove things about themselves without telling

who they are. Further, the pseudonyms could be used to automatically recognize those one has met before, providing they are using the same pseudonym. Finally, being cryptographically strong, the pseudonyms would help prevent identity theft in online communities like chat rooms.

The rest of this thesis is organized as follows: In the Problem Statement section I will give a general description of the field of problems related to identification in the information society and motivation for solving them. Criteria presents a set of criteria for such a system and in Existing Technologies the technologies on which the model is based on are presented. The Model section describes the generic model that could be used to solve the problem and in the Implementation section I present the limited functionality of the chat room demonstration. Then, in Analysis, the model is compared to the set of criteria and further problems for future work. Finally, Conclusions discusses the results, lessons learned, and presents the conclusions.

Chapter 2

Problem Statement

Internet is largely an anonymous and wild space. Anonymity is good for preserving one's privacy, but with it come the problems of total, or nearly total, lack of liability for those who do not behave themselves. One can not directly believe what others say about themselves in online environments like chats and other online services.

This thesis is about pseudonyms (quasi-anonymous identifiers) that could be used to recognize acquaintances as well as using digital certificates to verify attributes about the user behind the pseudonym. There is no perfect solution that would allow absolute privacy, yet certify a bulk of information about the user, but walking the tight rope, I try to bring trustworthiness into the anonymous world by using trusted third parties. In this section we shall explore the field to see why preserving privacy online is imperative for the user.

Information networks can bring a lot of information to the desktop of the end user, but at the same time they can be used to collect information about the user to the databases of service providers and network monitors. Every action, from email to web browsing, may leave a trace at least in some logs on the servers. If this data is collected and mined to find patterns in the logged data, the users' privacy can be breached.

Services that directly collect information about the user are among the most potential threats. Many such services require users to fill in a bulk of registration information in order to qualify for the service. Often the information gathered in such fashion is enough to make a unique identification on the person of the user. The threat of unique identification is further emphasized if we consider the fact that multiple sites can and will begin to combine their databases to gather even more information about the user. And since the services can collect information on how people use each of the services, complicated profiles of user's actions can be compiled without the user being aware of it. Taking the threat even further, it is possible that the services begin to sell these profiles to third parties with whom the user has not been in any contact [etoys]. Today users receive random unwarranted email from such third parties; in the future things might get more personal, or more disturbing, like receiving unwarranted SMS messages to one's mobile terminal.

From a legal point of view, many sites have a 'terms of use' document describing the regulatory aspects. However, in my experience these usually range from two to twenty pages (the longest was about 70) and contain statements some of which can be considered obvious. Examples of obviousness include that the site is not responsible for content on other sites even though the site may contain links to these, that content on the site may be copyrighted (most things are), that the service is provided as is (I've never heard of a site that promises to work), and that the service is not responsible for the

content posted by users. In addition, some contain statements that seem even outrageous, like giving the service a right to send the user email even after the service agreement has been terminated, or that user gives up (to the service) all copyrights to any content posted on the service and that service may use any posted content as it wills (which is obvious if it has already taken the copyright). This raises the question of whether the companies rely on the fact that people do not read the agreements, or at least do not care. After all, currently there is no way to negotiate on the agreement[1]. So a user wanting to use the service has no option but to agree and hope nothing dreadful happens. Legally, these statements may be quite binding, but from a moral point of view, their basis seems at least questionable.

Let's take a broader look, a look beyond the old society, at what might be called the e-society or the society in the information age, a society where the network is used to handle most affairs in an electronic way. Already, we might consider that payments are network actions, even though there is a cashier behind the desk to swipe the card through the reader. Difference between paying using the network and paying at a cashier are astonishingly small. The only difference is that the cashier can look at an ID card and verify the identity of the customer. Traditional cash does not need such identification and similar anonymous payment systems are technically possible today [Chaum85, Nordsec99].

The real threats are not the voluntary services used on the Internet today, but the future environment where all actions are taken through the Net. We are already seeing the first commercial services rising on the mobile networks, first in the form of WAP, but more sophisticated services will be around soon. Such services require higher levels of liability and billing.

On the web, there is currently no way for people to prove who they are. Smart Card based IDs are becoming available and making it possible to identify people over the network [HST]. However, they might actually create more privacy problems for the average end user than they really solve. If strong identification technology will become available to everyone, demand for it may also rise even in situation where it is not really necessary, leading to a situation where one can not use most web based services without first waving a electronic ID (EID). In the Orwellian world people's actions were followed, but people were not required to identify themselves at every instance. In common everyday situations, like meeting friends in a café, people do not need to identify their acquaintances, they need to recognize them as someone or thing they know and trust. Alice does not need Bob's social security number, but she needs to know this is the very Bob with whom she has been a friend for years.

The sad thing is that even the providers of smart card security infrastructure and PKIs seem to refuse to see alternatives to strong unique identification. Of course, there are many possible reasons for it: commercial gain from old technology, blind sightedness and deliberate Big Brother mentality among others. But the claim that strong identification is the only way of making e-commerce work [Dipl00] is almost as absurd as withdrawing all notes from the market and going back to cheques. Traceability is essential to recover abusers, but there is no need to automatically trace every action of every citizen just to catch the criminals. Optional tracing with a high enough cost to

[1] If successful, the P3P initiative [P3P] may slightly enhance the negotiation situation.

prevent automatic tracing of all transactions should be enough. Smart card security companies might actually find more profitable markets in selling people multiple secure pseudo-identities than in forcing them to use a single identifiable card and give up their privacy.

The very protection of privacy is the reason for users to not reveal too identifying information when meeting other people on different online services. Using EIDs to prove things about themselves is certainly technologically possible but hardly wise. One can never know what kind of a person is at the other end of the line, therefore revealing information that can be traced to one's address or telephone number is not always the thing most people want to do. On the other hand, since there is no telling about the person on the other end of the line, the level of trust on the things that he says about himself remains quite low, even to the level of generally mistrusting everybody. It would be nice to be able to prove some attributes about oneself without revealing too much information. Likewise, it would be nice to be able to recognize their acquaintances even in places where identifiers or nicks are not reserved. In chat rooms for example, the only rule is that multiple persons may not use the same nick at the same time. But the 'Jack' of today may be a very different person from the 'Jack' of yesterday.

Continuity is a central aspect of any long-term relationship; even when long term means anything with more than one uninterrupted session. For such continuity to take place the parties have to recognize each other. Recognition is a very natural thing between human beings; our brains are well developed for such purposes in conventional situations, using faces, voices, gestures etc. to recognize the other person [Ilm97, Oja97]. Despite the limitations of the online environment, people have the same fundamental need to recognize their acquaintances. In the online environment people are interacting through a complex system of computer programs. They rely on these systems to provide them with the signals from the other person. If technical means are to be used to recognize the person at the other end of the line, the system needs to be able to do it and provide the necessary information to the user in a human understandable form. Further, the recognition process must not burden the user; people are used to recognizing people at a glance, not through a series of clicks and queries.

2.1 Names

Names are something that people use to call things. Things, here, include humans, other living entities, dead objects, and just about anything that we want to relate or refer to. In such a way, it can be seen that names are in fact pointers or references. However, as people are likely to add feelings to their names, the names are more to people than file handles are to files, thus we could call them pointers with a little humanity on top.

Traditionally identification can be considered to mean the recognition of an entity in such a way that an individual can specify another individual object or entity. In such a schema, names are something used as an identifier, a tool used for identification. And certainly, this is what we use names for; to specify which person or object we are referring or calling to. However, plain names only work in a certain reference frame. *In this thesis identification is generally defined as identification in a (nearly) global reference frame*, rather than verifying possession or ownership of some identifier. In particular, it is opposed to recognition, an act of noting acquaintance.

Since humans tend to use simple and easy to remember identifiers – or names – for things that are important to them, the usable namespace is limited in a way that not all can have a usable unique name. This poses conflicts which in the physical world are usually fairly easily overcome, but pose more serious recognition problems in virtual environments where the number of people that the user comes in contact with is much higher than in the physical world. One reason for this is that the user needs an identifier for each person in order to relate to her, because there is no "you there, the lady in the blue dress". Further, since nicks do not usually have to be registered, there may be multiple people using the same identifier, say "Alice", at different times. Thus, one can never be certain whether that "Alice" is the same Alice as last time, without going into a discussion and finding out. This is one problem that the recognition architecture could be used for.

Besides the namespace size problem, it would appear that there is more to names than just specifying individuals. As humans, we seek meaning to a lot of things. In this sense the names we use are, to us, more than simple pointers or identifiers. Names are more because they are given to us (or taken by us). Behind at least well chosen names, there is some reason for that name to be linked to that individual. One good example of names having meaning comes from the Native Americans using highly descriptive names like Sitting Bull or Running Wind. Yet, those cultures are certainly not the only ones where the naming was an important event in a person's life; many tribal cultures still celebrate naming and coming of age. At first glance it might seem that the western culture has rid itself of such events, we do not uphold tests of manhood and coming of age as such. However, any Christian should be familiar with christening parties and young people certainly celebrate coming of age on their own. Likewise, one can have one or more nicknames given by friends in the physical world as well. What has this got to do with network identifiers and is this important all? I believe it is, because the connection is that this is the very basis how and why people form the nicknames they use in chat rooms and other virtual environments.

As a counter example, the social security number is certainly also an identifier, but it is not a name. People do not use their social security numbers in dealing with their acquaintances. In the famous British science fiction television show "*The Prisoner*", the main character, "*number 6*", was fighting for his right to be considered a person, not a number. Numbers are often seen as cold, since they are rarely seen to contain such contents that humans would find meaningful or insightful. For this reason identifiers like "john1234" are seen and felt to be more like a numbering scheme than names for a human. As we have a need to be part of something, the way we refer to ourselves is important as well. Besides, since we can bind meaning into words and strings, we are better at remembering those than at remembering arbitrary strings of numbers.

2.1.1 Nicks

In virtual environments like chat rooms, the users need to be differentiated from each other using some kind of identifiers. In IRC[2] and most chat rooms these names are called nicks, short for nickname. Such nicks are of more varieties than nicknames in the traditional sense or the ones used in the physical world. In some services, in particular

[2] Internet Relay Chat, one of the first and most versatile network chat implementations, see. [R1459] for more detailed information.

ones that also provide other services beside chat to the user, the identifiers have to be registered. These services require a userID or username, which can be similar to a nick, but there is a slight difference. Usernames are generally long-term, i.e. multiple sessions, while nicks can be session specific. Further, it would appear that people put a little more personality to nicks than to userIDs. This might reflect that nicks are considered more name-like than userIDs. Next, we take a look at some example nicks being used; I have divided these into categories mostly for convenience, this list is not intended to be comprehensive, just informative:

1. Nicks derived from a commonly used name. The source can be either the person's own or made up to maintain privacy. Examples of such nicks could be: Pete, Johnny, Jackie, JP…

2. Nicks derived from an adjective or other attribute describing the person behind the nick. Examples: Blond, lady76, tall-guy, boy13…

3. Nicks derived from some arbitrary concept, perhaps hobbies or what the person is seeking. Examples: driver, Loner, snoopy, ghostman…

4. Nicks constructed with special characters to convey some meaning that may be more or less understandable without knowing some specific piece of information about the person. Examples: ~♀@♥™~, "f…",M()…

5. Nicks that are totally random, i.e. have no internal meaning either. These seem to be very rarely used. Sometimes these are difficult to separate from the previous group without the required knowledge.

Of course, none of the nicks need to give a truthful impression on the person behind the nick. "Lady23" could as easily be a 13-year-old boy. But the important thing is that the person wanted to make an impression of some kind on people as they first see the nick. In other words the nick is the user's "face" in the virtual world, a face that can be easily masked. Reasons for masking are varied, from maintaining privacy to direct fraud.

2.2 Privacy

Throughout history and across sciences privacy has been defined and classified in multiple ways. One generally good definition comes from Patricia Newell: "*Privacy can be defined as voluntary and temporary condition of separation from the public domain.*" Burgoon divides privacy into four categories: physical, interactional, psychic, and information [Burgoon89, Laukka00]. Clarke divides privacy into four (or three) different categories: privacy of person, privacy of behavior, privacy of communication, and privacy of personal data, of which the last two can be combined into information privacy [Clarke97].

We will divide privacy into two classes and then mix those. Spatial privacy includes from above the physical/personal privacy, behavior and psychic aspects, while information privacy includes aspects related to personal data, information and communication. In cyberspace, however, space and data become mixed and therefore the divisions from the physical world may not be directly applicable. Yet according to [Jeff98] the physical privacy properties do get inherited into simulated physical environments.

2.2.1 Spatial Privacy

The extreme basis of physical privacy is that of the physical person. An unwanted touch can be a highly disliked experience. However, this is again subject to culture, contrast to the touch disliking cultures are the cultures where hugging and even kissing are common ways of greeting people. Likewise, the distance from which people interact to others, for example standing distance in a conversation, vary. At times this can lead to sadly humorous situations where a person liking longer distance is virtually running away from a close distance person. In addition to real physical persons, this behavior has also been found in virtual 3D environments [Jeff98].

Many animals keep a territory, an area where other members or only accepted others are allowed. We humans are much the same, we need to have our own space, a private space that is our domain. Such a territory is the extension of the privacy of our person. Home is perhaps the best example of such space, what we do there is no business for any outsider. To enforce the right for this privacy, laws of many countries support the right to keep unwanted people away from private property.

The preachers of strong identification schemes often make the claim that identifying people does no harm. Unless they try to hide something. They put it as if hiding things was something that had to be criminal. However, having privacy and being able to hide some things are essential for a human. That is exactly why letters are enclosed in an envelope (and opening letters of others is a crime) and why the police wanting to investigate a home needs to apply for a search warrant.

As our territory is extended from our person to the private territory like home, we allow others, especially family, on our territory. Such allowance is based on trust. And similarly to allowing people on our territory, we at least at times, let them know our secrets as well. Such exchange of information is usually done under circumstances where we believe nobody else can hear. Listening to such private conversations can be met quite coldly. While these conversational privacy methods can have spatial aspects, like using a closed meeting room, or the fact that tapping a phone line and listening to people is illegal, such cases are often in the grey area between spatial and informational privacy.

2.2.2 Informational privacy

Private conversations can take place in a public place, but the thing that is protected is the information that is passed between the persons. That is, something we do not want others to know. The same confidentiality is an essential part of a doctor-patient relationship, medical records are considered private enough to have legal protection.

Just like people do not want every passerby to know their medical records, they do not wish them to identify them either. Old acquaintances are recognized by for example their faces, but strangers should not be able identify us and we do not need to identify them. The idea of having ones credentials tattooed on one's forehead is the traditional example of losing one's privacy to the surveillance of the society. Continuous identification basically results in the same.

People are used to the fact that by using plain old cash, they can buy things without having to identify themselves. The whole concept of identification is due to lack of other methods of verifying the authorization to use tokens like bank and credit cards. In order to be certain of getting his money, the merchant needs to verify that the person

wielding the card is authorized to use it. Of course, even in the cash example the customer is not totally anonymous, the merchant might remember his face the next time he comes around or if someone asks questions about a person like that. In any real environment there are factors that make total anonymity essentially impossible. Even if it were possible, it would not likely be a worthy goal, a level of anonymity that allows people to maintain a decent level of privacy should suffice.

2.2.3 Network privacy

People are using the computer networks in increasing amounts. The network environment is something very different from the traditional physical world and therefore we discuss it separately. Though, depending on the type of online environment in question, the privacy environment can vary greatly across services, the network privacy is usually some sort of a mix of the traditional classes. If we consider that the network is cyberspace, therefore it should go under spatial privacy? On the other hand, the whole thing is nothing but information, therefore it should go under information privacy? It is a virtual environment, therefore it is a mathematical model that is not really true, and in this model actions are coded into information. Yet, in the human mind, the illusion becomes true and a mathematical model is treated similarly to a physical reality.

The network is not a space in the old fashioned sense; there is no volume or no (significant) distances. A bit of information on the other side of the globe can often be reachable practically as fast as another that is geographically closer, sometimes even faster.

On the network, most activity is transformed into information in forms of log files and other traces. Theoretically, anything done on the network leaves a trail, though in practice it may be impossible to follow the trails. Even so, this collection and recording of information about all activities makes the network a place very different from the traditional physical world.

In 1948, George Orwell presented in his book "1984" a world, or at least a society, in which a third of people were watching the rest, making sure they were politically correct and did not try to rebel. Individuality was forbidden [Orwell48]. What Orwell did not realize or depict was that even by 1984 the world would have developed powerful new machines called computers that were beginning to be used in every aspect of life. Or that soon after these powerful computers were linked together in a global network that allowed data to be transferred and used on multiple sites.

One of the major concerns of media industry today is the distribution of their works over the network by people. Peer-to-peer networks allow people to trade mp3 music, movies, and just about anything else over the network without the corporations having much control. This phenomenon certainly has some economic effect, though very likely most people downloading the media would not pay money for it even if they could not get it for free. Next we consider the other side of the coin: what people can do the government and corporations can also do, even more efficiently. The very same networks can be used to trade user records, profiles, even information classified as private. Normal log files can be sold for data mining organizations who further sell the refined product to other companies. And just like corporations have no control over music trading, normal people have little control over what the corporations do with the user accounts.

The latest trend is that individual people are using the Network for many different affairs, more or less personal. Corporations are providing services on the Network and often require people to give their personal, identifying information. Of all these actions and messages, the servers keep track of. Most of the data is just thrown away, but there is no guarantee of that. One of the main concepts of the so-called information society is that information becomes the main commercial product. In this sense the data describing the actions of these Jacks and Jills becomes valuable commodity. Who should own and gain the benefit of such data? Currently the global rules and laws have very little coherence on the subject.

2.3 Human Relations

Relations are connections or links between entities. In order to help understand the relations we have modeled, let us discuss recognition, trust and interaction types. The basic idea is to gain at least some understanding of how these concepts work. The basis must at least in some sense be the traditional physical world since that is the environment where people are used to relating to each other. In designing the virtual world, increasing amount of experience from design in the physical world should be used.

2.3.1 Recognition

Recognition is one of the primary requirements for human relationships. It is necessary for us to be able to associate the people we meet to people we have met before providing, of course, that they are the same person. Recognition takes many forms, any information we receive can be used for recognition purposes [Gleit91, Gold89]:

Appearance:	face, shape, hair, clothes, etc.
Sound:	voice, for a close person also footsteps etc.
Sayings:	Phrases or words often used by the known one, or rarely by others.
Behavior:	Especially if uncommon

Pattern recognition is one of the strongest fields of a human brain [Ilm97, Oja97]. We are especially good at visual patterns like a human face, and we train ourselves to quickly understand spoken languages we often use and to recognize and combine letters to read text [Oja97]. However, in the internet environment of today, we lose the visual and audio contacts to the other person. We cannot see the face or hear the voice. All we usually have is letters on a screen. This means that the only means the user has at recognizing an old acquaintance is through the nickname, which often can be used by somebody else, or by reading into the text and phrases used by the other person. Also the information the other person knows can be queried. This of course is quite agonizing since we are used to recognizing people on the fly.

One solution would of course be to require the users to register unique usernames, and require them to use those. However, as discussed in the section on names, since the namespace of commonly used names is limited, this leads to a situation where the users can not use names they are comfortable with.

2.3.2 Trust

The Internet has traditionally been a highly anonymous place, a fact that holds many good qualities, but unfortunately also allows for mischievous behavior. There is often no way to track the person behind a net name[3]. This of course helps preserve the privacy of the user, but it also makes it very hard for the other person to decide how much trust he or she should put to what the first person said.

When we consider just chatting on the net, such behavior does not pose any great damage, rarely more than annoyance. But if we consider that the same mechanisms that are used to recognize users in the chat rooms could be used to allow them to trade things or even make contracts, some more trust would certainly be required. Also it should be considered that people are increasingly meeting new people on the Internet and forming relationships with them.

Two important questions arise from the uncertainty that require solving. First: *Is this the person he/she claims to be?* This is important when meeting someone again, in personal meetings as well as in business situations where a customer relation is to be continuous. Secondly: *Is he what he claims to be?* This is also an important question for both individuals and businesses. If the other person seems interesting, one would like to know he or she is not a fraud as soon as possible. And businesses might be giving discounts to some people or might even have variant pricing to different groups or different locations. The latter is also a question of access control or authority, only certain kinds of people can do certain things, yet it is not usually necessary to single out the person in question.

2.3.3 Types of interaction

Let us list different types of interaction that may be required. There are several axis or dimensions in which each interaction is positioned. First is the number of recipients, is the interaction one-to-one or one-to-many? Secondly, does it take place in real time, i.e. does the recipient get the message without noticeable delay, or does he or she have to wait for it. Third, the other party may be an active service or another user. Using these axes, we can derive for example the following interaction categories:

One-to-one delayed interaction: *email*

One-to-many delayed interaction: *mailing lists, news*

One-to-one realtime interaction: *private chat, ICQ*

One-to-many realtime interaction with static environment: *chat*

One-to-service interaction with active environment: *single player games*

One-to-one interaction with active environment: *two-player games*

One-to-many realtime interaction with active environment: *Multiplayer games, MUD*

In a one-to-many environment, the service may even handle some of the 'players'. On IRC channels it is normal to keep 'bots' that maintain the channel, keep it open when no one else is present thus allowing the owner of the channel to maintain the owner status.

[3] Especially when talking about web based usage. Tracing people behind emails or IRC nicks is easier but still not trivial. Success in tracing is limited by the amount of information the system provides and the skills of the users.

(Because channels are dynamic, if everybody leaves, the channel is deleted and anyone can recreate it, thus gaining channel operator status.) In some cases these might even have a high level of artificial intelligence so that real players are not supposed to differentiate between these bots and real people. Intelligent bots or agents are rare these days but the level of intelligence is likely to rise and such bots may become more common in the future as artificial intelligence algorithms advance and become cheaper in relation to CPU load.

2.4 Traditional Identification

Identification is an important part of any information system dealing with people, because data management, whether taxation or personal relations, requires identifiers or handles to keep data relating to each person together. Using commonly known identifiers such as Inhabitant Identification Schemes or Social Security Numbers, pose its own problems, because they make database merging easy, and such merging is a threat to privacy. However, organizations managing information on self-deviced schemes face challenges in devising cost effective solutions to their particular cases. [Clarke94]

One essential question is whether there is need to identify the person fully, or if an unlinked identifier can be used. While there are certain situations where it is in the interest of the person in question to be identified, most everyday cases are not such. An example of the former case would be medical treatment, especially if the patient is unconscious or otherwise unable to tell of any special restrictions, allergies etc. On the other hand, buying groceries at the local market does not in itself pose any need for the merchant to identify the customer and an anonymous or pseudonymous payment system would suffice. The merchant could even keep giving bonus cards to customers who are willing to be profiled, but these cards could as well be anonymous, i.e. only contain a customer number, unless they are also a payment method like a credit account.

2.4.1 How secure is a traditional identity anyway?

In this section we will discuss identity as the identifier, in its various forms, that can be found in passports and other forms of ID. The important thing to realize is that infrastructure is not foolproof either, even if we are not talking about counterfeiting. Let us think for a while how such an identity is created. Providing the mother gives birth at some institute, i.e. not secretly, a birth certificate is almost certainly issued. A lot of other documentation is also produced. However, then the child will grow up. The next event he or she deals with the citizen registry might be more than ten years later when he or she will be old enough to require some sort of an identification certificate. At that point the child walks into an office providing access to the records and by providing enough identifying information is given a certificate. However, there is nothing to guarantee that any other person of about the same age and gender with access to the necessary information could not get a certificate of the former person. This certificate, a seed document, is used to gather up all sorts of identifying documents, such as identity cards, driver's licenses, etc. which in turn can be used to apply for passports, which are globally considered strong proofs of identity. And these documents can be used to set up bank accounts and the like. In this sense the conventional identity infrastructure is not at all of as high integrity as is usually thought. It simply works because very few people think of beginning to gather false proofs of identity at a low enough age for them not be expected to have previous ID cards. This is not to say that it would be impossible

to get such documents later. It might be more difficult since the officials might have old photographs of the individual in question, but especially in cases of visual similarity, false identifiers should be fairly easy to get by claiming all one's identifiers were stolen. The longer it is from the last known issuance, the more the person might have changed.

It is worthwhile to remember that though unique identifiers of citizens are used in many western countries today, such schemes are very young in the historical perspective. In most European countries such schemes appeared in the 1960's and the whole concept dates from late 19th or early 20th century. Also, Great Britain and some other "western" countries still do not have their citizens numbered. US has a social security number (SSN) but the system is of low integrity, there are an estimated 4.2 million Americans with two or more SSNs [Clarke94].

Therefore, it seems totally feasible for a state to manage itself (and its citizens) without a strong citizen numbering system.

2.5 The Problem

Anonymity on the network is problematic: it allows people to lie about themselves, but at the same time it allows them to maintain a decent level of privacy. Well-behaving users should be able to maintain their privacy, but at the same time the hoaxes should be detected.

The purpose of this study has been to research how we could create unbreakably strong pseudonyms that could be used to keep track of one's acquaintances. Further, it is necessary to be able to bind attributes to the pseudonyms so that they could be used to prove things about the user without them having to reveal their true identity. The goal is to provide privacy but at the same time bring liability and thus increase the level of trust both among users and between users and businesses.

Chapter 3

Criteria

What kind of requirements could be put on a system providing safe interaction between people, as well as between people and businesses? Since the people-to-business connection has to be made secure and there is no considerable extra cost of using the same security methods for people-to-people connections, there is no real reason to keep the people-to-people connections less secure. Therefore these cases can be considered fairly similar. The following set of 11 criteria was originally presented in [Nordsec00].

Anonymous to new Acquaintances

Anonymity here means that the person's real identity can not be directly seen or derived from the pseudonym.

Traceability in case of Crime or Misuse

Should the pseudonym be used for something bearing legal responsibility, the users need to be traceable.

Easy to Create new Identifiers

To allow people to use different identifiers on each of the services used, it is important that they can be easily created on the fly. Also, should an identifier be exposed, it is vital to be able to get rid of it and get a new one. In some services, like anonymous payment, getting new identifiers could be an important part of maintaining anonymity [ns98].

Secure Against Forging

The new identifiers should at least be more secure than conventional username-password pairs. And providing any financial action is to be performed, forging identities should be far more difficult. A level of security provided by electronic identity smart cards should be seeked.

Usable for Access Control

As the new identifiers are used to recognize the user, i.e. bind the user to a certain identifier cryptographically securely, they could also be used to bind the user to such authorizations that traditionally require the user to be identified. As the purpose is to create identifiers that can be used in a variety of services, granting authorities and access rights to the identifier would be a logical requirement.

Provider Independence

The identifiers must not come from a single source, any such source would find it easy to link them all together. Since we might want some of the identifiers to be next to anonymous and non-traceable, and ones used in financial transactions would require traceability, there must be different methods of providing traceability. Secondly, it would be difficult for any single entity to be certain of all the different kinds of attributes that might be required at different services.

Peer-to-Peer

If the users in a chat room only want to make certain that they recognize each other (from any imposter) the next time, there is no need for a third party provider. Using shared secrets could naturally suffice, but many people might feel awkward to come up with strange phrases to recognize their acquaintances. A technology based recognition service might be more comfortable. Also, people often want to form groups among themselves without external control. And associations might want to issue their own certificates, like electronic membership cards.

Proof without Identification

In real world people meet and see things about the other people they meet. They trust their eyes to tell things about them. In cyberspace this is not so, all that is seen is text on a screen or a computer rendered presentation of what the other person wants to show. In most cases this does not cause serious problems, but if we consider the case where people are looking for new acquaintances, any misinformation can lead to at least disappointment. If users could prove certain attributes about themselves, like gender, age, location (to a chosen accuracy), this would help keep out the imposters.

Low cost

Services on the Internet have been mostly free to end users, and there is no reason should change. The most successful way of spreading something around seems to be offering the basic package for free, but without warranty or support. Additionally, from a human right and equality point of view I feel that the infrastructure providing security and privacy should be everyone's right. Support and extra services, which also create more expenses to the provider, may need to come with a price tag attached.

Accessible on multiple Terminals

Users might want to – or need to – use multiple computers for online access. It would be highly cumbersome to need to carry the identifier key pairs around on a dedicated device, an encrypted floppy disk or the like, therefore it would be nice to be able to retrieve the keys from the network itself. However, this puts extra requirements on the secure storage.

Secure Storage

The access to the private keys used for the identifiers must be highly secure, even on a single user computer that is not continuously online. Especially in case such a store would be accessible from the network, it should be encrypted strongly enough to stand long-term scrutiny, even by organizations dedicated to breaking encryption.

Chapter 4

Existing Technologies

Networking and security have been actively researched in the last decades and this research has produced the technologies currently in use. We will present some of the most fundamental existing technologies available as these provide the basis for building the recognition architecture or infrastructure.

4.1 Virtual Environment Technologies

The environments where people would be likely to use – or wish to use – recognition schemes are different from those of traditional computer usage environments like office tools or information processing. Rather the environments are more likely social environments where people meet people and where they feel they can not trust the authenticity of the other person as they are more likely unfamiliar than in the office situation. Here we introduce three well-known social environments currently in use.

4.1.1 Chat Rooms

An Internet chat room is a virtual room to which people can connect to using their computers. These rooms are often also called channels. Messages typed on the keyboard are sent to a server that distributes the text to all the participants in that particular room. Today, many chat rooms are based on web browsers for ease of use, but some, especially more sophisticated, still require a specific client software.

The mother of all chat environments is Internet Relay Chat (IRC), developed by Jarkko Oikarinen in 1988 [irc]. It was originally an extension to the UNIX talk utility, which allowed two UNIX users to talk to each other. In the original talk utility the screen was divided into an upper and a lower half. Local typing appeared at the top screen and the remote text at the bottom. Oikarinen initially created IRC to allow multiple parties talking to each other, but IRC soon developed into a feature rich distributed network environment.

IRC is not just a server, it is a network of servers connected to each other. An IRC network is the collection of servers that work to provide a common environment. Such servers connect to each other in a spanning tree formation, i.e. in such a way that there are no loops, a new server only connects to a single server already on the network. This helps remove ambiguities in distributing messages. A client (user) can connect to any of the servers on the specific IRC network and talk with any user on that network.

Anyone can create new channels on IRC and channels are automatically removed if all participants leave. Access to channels can be restricted; invite only channels require an

invitation from a user already on the channel. A channel can have an operator that has larger control over the channel and can KICK or BAN users off the channel.

Users can be on multiple channels at the same time. In more advanced client software, users can actually be on multiple chat networks at the same time. In addition to talking on channels which is public users can send private messages to each other, talk privately beside a channel or form a new channel, the privacy of which can be set to desired level for example by making it invite-only.

IRC networks may or may not have registerable nicks.

4.1.2 Multi-User Dungeons

Multi-User Dungeons, or MUDs for short, are most often game like virtual environments where a character created by the user interacts with the surroundings and characters of other users. These can be seen as an extension to the IRC environment, while in fact they preceded IRC. However, the basic functionality of IRC, talking, is present in MUDs, and in addition there is the functioning side, actions and objects that the character can take, hold and use. [MUD, Bartle, Sempsey]

MUDs in general do have a recognizing authentication mechanism since there is a clear user account with user name and a password. Not all MUDs allow saving the character, so these would not require remembering usernames and passwords as the game is session specific. At least in my experience, MUDs do not ask for identifying information about the user, of course there may be exceptions. Unique identification is not required, it is sufficient that the user has the password to prove the ownership of the character, if necessary. Security is often very low as the passwords are mostly transmitted over a plain telnet connection. Since there is usually no financial connection, such lack of security is rarely seen as a threat and, as in any multi-user environment, processing power is at premium, making encryption seem hardly worthwhile for the maintainers and administrators.

Should someone want to create such a game with restricted user space, an attributed recognition system might be a solution. Such restrictions could be membership in a club, student membership at a university, participation on a course (some MUDs are used as a learning space), citizenship of a certain country, being of certain age or having a recommendation from a known user. None of this really requires unique identification of the person, providing the required information can be derived from a relatively trusted source. And even if it can not, unique identification does not always help.

4.1.3 Web Communities

With web communities in this context we mean a multitude of sites that provide different kinds of interaction for users. Certainly both IRC and MUDs can be considered communities, and individual channels on IRC can be considered tighter communities. There are those who hang on the channel every day, those who pop in occasionally and those random passers by. The web offers much more diverse possibilities: people can post articles (long term messages), put up photos of themselves or their activities, ask questions, sell or exchange items, etc. The web allows skilled people to create new functionality in their own virtual back yard and then call people to come and take a look. Less technically skilled people can use blocks created by others to build their own and even learn in the process.

Web communities might benefit from the same attributes as in restricted MUDs, above. In addition, in web communities people interact with real people and might want to be fairly certain of them being some specific type of persons before taking some action. In an auction they might just want to know that some trustworthy party has identified this person and if he or she does not fulfil the liabilities, they may be able to trace them or let the TTP trace them. The same applies for any situation with e-commerce, traceability adds a bit of certainty while preserving privacy for the well behaving.

4.2 Fundamentals of Security and Certification

Various technologies are in use today that allow for people to use network services more securely. Encryption is used to protect the confidentiality of information, hash-functions and digital signature algorithms allow for verifying the integrity and the source (and non-repudiation). Identity certificates are used to bind public keys to identities, or rather, enough personal information to uniquely identify the person in question. Authorization certificates are developed to allow authorization without identification and attribute certificates can be used to prove attributes of a certain entity. In this section we will walk through the concepts of cryptography and some other security technologies on a high level and in the next section we will introduce some more tangible architectures based on these concepts. [Singh99]

4.2.1 Cryptography

Cryptography is the art of making a message non-understandable so that only the intended recipients can read it. First examples include the use of non-standard hieroglyphs by Egyptian scripts and letter shifting in the Caesar cipher. Since then cryptography has developed first to more complex mechanical machines like the German Enigma during the Second World War and more recently into complex algorithms relying on computer calculations.

Cryptographic algorithms depend on a key, a piece of information that carries the values used for encryption and decryption. Later we shall see that the key can also be a more complex data set, like a key pair, but most basic algorithms rely on a single key that is used for both encryption and decryption. We refer to such algorithms as symmetric cryptography. In order to read the encrypted message, the recipient must know both the algorithm and the key used to encrypt the message. And of course, the security of the transfer depends on the strength of the algorithm and the key. Neither must have any weaknesses, though usually – especially in modern cryptography – only the key has to be kept secret. Secrecy of the algorithm is not considered to add security, more likely it is likely to diminish it. Since well known algorithms have gone through a much deeper scrutiny by the cryptographic society they are less likely to contain hidden weaknesses [Kahn67]. The algorithms have rarely been a problem recently, but one of the biggest problems is keeping the keys secret.

4.2.2 Key Distribution Problem

The symmetric – one key – cryptosystems discussed in the previous section are an effective way of securing the data, providing that the algorithm is secure, the key is long enough, or more specifically, has enough entropy and finally, both ends of the connection (but nobody else) have access to the key. This final condition has been the most difficult to achieve throughout history. Even when the algorithm itself has been

secure, gaining access to the key allows the crackers to read the message. Since both ends need to have the key, it is necessary to either share it in a meeting beforehand, or have it transported through a secure channel to the other party. The first case is impractical if numerous messages need to be communicated, or if such a meeting is impossible as in case of wanting to communicate with a new party. The latter case, a secure channel, is impractical as well; if one has access to a secure channel, why would one need the encryption to protect the message.

During the Second World War the number of messages encrypted with the German Enigma system was in millions of words a day. In that case the situation was still relatively easy, though the codes had to be distributed to thousands of stations, there was no need to hide the messages from the friendly stations, allowing for common daily keys. These could then be distributed as code books to the stations and be used to encrypt session keys. The military levels of security and trust, combined with the situation with a single united group of stations that were to communicate with each other helped the system to remain almost adequate. In fact, the Germans managed to keep the system secure and unbreached until nearly the end of the war. In particular, the Alliance did not manage to get their hands on the codebooks. [Singh99]

But when we look at the situation in the information age, where every person has need to secure his or her communications on the open network, everybody else becomes an enemy and every connection requires an independent secret key. It was this view of the future that in the early 1970's drove the young Whitfield Diffie to baffle with the problem of key distribution. In his view the commercial and military installations would find a way of dealing with it (with difficulty) but for the end users an easier solution was required. In 1974, Diffie teamed up with Professor Martin Hellman from Stanford University and they came up with the Diffie-Hellman key exchange, which allowed two communicating parties to create a shared secret key that no eavesdropper could catch. A few years later, Diffie came up with the idea of Public Key Cryptography (PKC), where, instead of a single key, two keys were used. One key was to be public and one private, linking the key pair to the owner of the private key.

4.2.3 Public Key Cryptography

But as recent revelations present, Diffie was after all not the first person to discover public key cryptography. Unknown to Diffie, just a couple of years earlier, in 1969, in the British GCHQ (Government Communications Headquarters), James Ellis had proven the existence of PKC. Unfortunately, GCHQ was a military installation so the results were not published. Further, since Ellis was not a mathematician, he could not find a solution to the concept and realization had to wait until 1973. It was at that time when a fresh-new young recruit at GCHQ, Clifford Cocks, was presented the puzzle and he quickly came up with the same solution that a few years later would be discovered again, that time in public research, and be published by Rivest, Shamir and Addleman as the RSA encryption [RSA]. Unlike the Diffie-Hellman key exchange, the RSA keys are specific to each person, not pair of persons, and the keys, or actually key pairs, can be generated individually, offline. Both Diffie-Hellman key exchange and RSA were patented, but those patents have recently expired.

The new idea of Public Key Cryptography is that instead of a single key, there are two, one that is secret, personal, and one that is public, for use by all the other parties. Using the public key, anyone can encrypt a message and only the person with the private key can decrypt it. By reversing the process, keeping the encryption key private and

publishing the decryption key, one can create digital signatures; anyone with access to the public key can now verify that the message was signed (encrypted) with the private key. Not all public key cryptosystems are reversible, but RSA has the remarkable property that either key can be used for the encryption and the other to decrypt the message. The sender can sign the message with his or her private key and then encrypt it using the recipient's public key. Now only the recipient can decrypt the message and also verify that the message truly came from the sender. At least, this is true as long as the private keys remain private.

Before assuming that the problem is completely solved, know that there are a few complications to public key cryptography. For one, most schemes are awkwardly slow. A public key scheme can be a thousand fold slower than a symmetric cipher and produce a ciphertext that is much longer than the plaintext. In addition to that, if long data sets are encrypted, the keys may become compromised. This of course is a problem with symmetric keys as well, but since those can be changed often, the problem is not as severe as with the public key scheme where the long lifetime of the public key is one key benefit. Therefore, public key cryptography is usually used for key exchange rather than encryption of the actual message. This is achieved in such a fashion that the actual message is encrypted with a random session specific key using a symmetric cipher and the session key is then encrypted with the public key system. Both of these are then sent to the recipient, who first decrypts the session key and then uses that to decrypt the actual message. This is much faster than using public key cryptography for encryting the actual message.

Further, digital signatures can be simplified using hash functions; these generate a short digest of the message, which is then signed. Providing the hash algorithm is cryptographically strong, the value is mathematically bound to the message and thus the signature is as well. Should the document itself be changed even the slightest, the hash function gives a different value and thus the signature is no longer bound to the changed document and forgeries can be detected.

The public key cryptography was to solve the problem of (session) key exchange. For long term relationships long term public keys are necessary. This, in turn, means that the involved parties need to be able to trust that the keys truly belong to the correct recipient and that no one else has access to the private keys. To manage the public key cryptographic keys another layer is needed to tell which key(s) belong to whom and which keys have been jeopardized. Such solutions are called Public Key Infrastructures.

4.2.4 Public Key Infrastructures

Public key encryption did not solve the problem of getting a secure channel to the desired party in its entirety. Let us consider the case of Alice and Bob where Alice wants to send a secure message to Bob, only she does not know Bob's public key. Bob could have his key on his web page, but there is still a chance that a cracker, let us call her Carol, who wants to intercept Bob's messages, has changed the key to one of her own. One simple way would be for Alice to ask Bob for the key, but if she does not really know Bob in any way, she can not be certain that the person giving the key is really Bob. It could instead be Carol (or her male friend) who wants to intercept the messages. Alice could end having a conversation with Carol. Or if Bob does not have any more knowledge of Alice's key than she had of his, Carol could intercept Alice's messages before passing them on to Bob, she would just open them with the key pair she showed Alice and give her own key to Bob. Now Alice thinks Carol is Bob and Bob

thinks Carol is Alice. If Carol keeps forwarding the messages, she could in fact even change them radically before passing them on. This is known as the man-in-the-middle attack.

This threat is one major problem that Public Key Infrastructures (PKI) were developed to solve. The idea is basically the same as that of a phone book. A trusted party keeps a catalog of phone numbers or public keys and makes them available. Now Alice could look Bob's public key up from a more trustworthy source. Of course, if the database connection is not digitally secured and Carol has access to the queries to the PKI database, she could forge those as well. Such case, however, is much less likely.

The next problem Alice faces is the same problem she faces with the phone book. Which Bob in the list is the one she wants to contact. If Bob did have his public key on his web page and Alice can find a Bob in the PKI with that public key, she can be fairly certain that the key is correct. But for her to look up the list of Bobs and try to decide, the database needs to have a lot of information about Bob, basically at least enough to uniquely identify him. Alice still might face the problem that she might not know that much about Bob and thus still not able to make the correct choice. In turn, Bob might not wish all that information to be more or less publicly available about him.

More problems remain. First, how can Alice be certain of the integrity of the information she retrieved from the database? The information is usually digitally signed, but does she have the correct public key to verify the signature? Back to square one? Not quite, providing the database is common knowledge or otherwise large enough to publish its public key periodically at large newspapers or some other means not practical for an individual.

Finally, even if Alice can find Bob from a database and can verify that the key she received truly came from the database, she still has things she might want to consider: does she trust the database to have been kept secure so that no cracker has corrupted it, and does she trust the database to contain correct information in the first place. The database could be certifying false information for various reasons: because somebody cracked it, because the information was not verified carefully enough when it was entered or even because the database owner wishes to distribute incorrect data.

4.2.5 Digital and Analog signatures

Digital signatures, with which we here mean cryptographic signatures as opposed to digitized (or scanned) signatures, are fundamentally different from the traditional handwritten (analog) signatures. First of all, unlike handwritten signatures that look the same independent of the document being signed, the form of a digital signature varies. Since the digital signature is not physically attached to the document, like ink attaches it to the paper, a digital signature is basically the hash-value of the document (practically uniquely specifying the document) "encrypted" with a private key. Encryption here refers to the process of producing cipher-text, even though the commonly known public key can be used to decipher it and verify that the correct private key was used to encrypt it, in other words, sign the document to which the hash-value refers.

Secondly, digital signatures are either right or wrong. The nature of digitality is 0/1, no/yes, there is no in-between. A digital signature can not be forged in the way of making it similar enough to the authentic signature to pass the examination. Unlike hand-written signatures that always vary slightly and which can scrutinized and

discussed, digital signatures have one[4] specific form, or number, which is correct, all others are plain wrong. This also means that if the protagonist actually manages to forge the digital signature, this forged signature is mathematically indistinguishable from an authentic one. If there are ways of deciding such a signature is fake, those are outside the field of cryptography.

4.2.6 Certificates

A digital certificate is often thought to be a digital identity certificate that binds a public key to an identity. This false definition is far too limited but lives on strongly since identity certification is the dominant form of certification today. The term certificate was first used in [Kohn78] to bind together a name and a key. In the more general terms a digital certificate is a document binding a public key or a name to an attribute, where the attribute may be an identity, an authorization or any other attribute. Though all these can be considered attributes of a kind, certificates are usually divided into three categories: identity certificates, authorization certificates and attribute certificates. In an even more liberal view, a digital certificate could be any kind of a document making any kind of statement that is signed.

4.3 Present-day Certification Technology

In this section we will take a look at the more practical applications of the cryptographic methods discussed in the previous section. First we will discuss identity certification, which was the first field where certificates were widely used, and which is still considered the only definition for certification by some. Then we move on to discuss Certification Authorities (CA), entities doing the certification. Third comes trust models, how the trust is transferred, and after that, delegation which relies on at least some form of trust transience. And finally we present a look at three PKIs (public key infrastructures), namely PGP, X.509 and SPKI.

4.3.1 Identity Certification

Since the first use and demand for public key cryptography was key distribution, the first application was identity certification. In other words, verifying that a certain key belongs to a certain person. Due to this historical demand and the fact that most certification today is identity certification, many people use the word certificate as a synonym for an identity certificate. If identifying entities is what is required, the system fulfils its purpose. In the FINEID[5] system, an electronic version of the traditional identity card was created to allow the citizens to handle some of their official business over the network. The system is based on a smart card that can also be used as a conventional ID, but with a computer equipped with a smart card reader and some additional software, electronic identification and signing can be performed. Among first uses are e.g. filling of EU subsidy applications for farmers, filling notices of moving and filing tax forms.

[4] The signature (number), on the other hand, can be the same for more than one document. The hash length and algorithm specify the likelihood to find another such document.

[5] FINnish Electronic IDentity, or Henkilön Sähköinen Tunnistaminen (HST) in Finnish

On the other hand, less official uses were also planned, like using the FINEID card for credit or bank cards. These fields, where identification would no longer actually be necessary if current state of the art technology would be used, would cause unnecessary risk to personal privacy. It is noteworthy that the need for identification with traditional bank and credit cards comes from assuring that the person in question is authorized to use the card in question. When that authority can be assured in other ways, identification should be discarded instead of keeping it up for the sake of tradition.

Another good use for identification certification is in certifying services like SSL-based web servers. A bank or some other reputable entity that wants to ensure that their identity is not used by some hoaxer, can get a certificate that proves to the client that he or she is really connected to the bank's server. In cases like these, personal privacy is not at risk and it is in the interest of the entity in question to allow it to be identified. Similarly, even though in most cases an individual's privacy should be protected, there are cases where identification serves the interests of the person in question. And also, there may be cases where services might want to stay anonymous, though I believe demand for such to be minimal.

As was discussed in the name section, it has to be remembered that identifying people is not 100% certain in itself. This follows from the simple fact that aside the human genotype, there is no truly unique key to which person actually is which. Therefore, identification is actually a set of attributes, such as a name or an identity number, that identify the person or entity uniquely only within a certain scope.

4.3.2 Certification Authorities and liability

A common misconception is that Certification Authorities certify things; they do not exactly do that. The most common problem in trusting a certification authority comes from the fact that the term in itself is confusing. Certificates are actually simple statements. Usually the meaning is more like that the issuer (CA) believes that the statement in the certificate is true about the subject. The level of assertion behind the certificate can vary significantly. To understand what each certificate actually means, one would need to read and understand a Certification Practise Statement (CPS) published by the issuer. The CPS is a document that states the policies and practices in issuing and maintaining the certificates that may be provided by the CA [DigiCer]. Generally, the CPS defines the procedure for issuance of certificates, but in practice the CAs take little responsibility in case the information actually is proven wrong. Considering the fact that a certificate really does not guarantee anything, such practise is understandable, especially in countries where even the smallest disputes end up in court.

Another problem in trusting the CA is that usually users have no pre-hand information about the CAs to know which are most trustworthy. To solve this problem, only trustworthy entities must be able to become a CA. The reasoning is that if anyone could be a CA – then anyone, i.e. also the hoaxes – could certify anything and fool the users. However, such restrictions easily lead to a reverse conclusion that anyone who is a CA is trustworthy, which in a real world is not always the case.

In addition to this, the average user rarely chooses which CAs to trust. Let us consider the web surfing user. He or she installs the browser software on the computer (or it comes pre-installed) and with that software comes a list of CAs chosen by the software provider. The CAs at least had a CPS stating their practices, but the software providers do not give any statements on which bases they have chosen the CAs to be included in

their distribution. In fact, the chain of trust now also includes the software provider. The user now "trusts" the software providers, not only to certify that these keys belong to the CA in question, but also to tell him or her which CAs are trustworthy.

Of course, most modern browsers allow the user to modify the list of CAs, but this is usually a difficult procedure that few not-highly-aware users dare even try. Most users do not likely even know that the browser has such a list. Further, if a CA is missing from the list, the average user has no way to verify the credibility of that particular CA. Most of this problem is caused by the fact that the users do not know anything about the CA companies; they are a new industry and unknown brands. The level of trust in such entities is not high. If the certification (or at least introduction to CAs) could be performed on a more local level, by entities the users already trust (such as banks, telecom operators, insurance companies etc.) the trust could actually be based on something to begin with. Of course, the user needs to start from some point, a root for his or her personal certification forest. We will discuss trust models in more detail in the next section.

Yet another problem in some cases is the lack of liability on behalf of the Certification Authorities. A CA that would take on a financial liability on the certificates it issues would most likely be preferable to one that does not. As more and more financial transactions are performed trusting the certification infrastructure, the question might become important to the average user who is finally beginning to use the web for commercial actions as well. In the future if money becomes electronic [Chaum85, FutMon], a liable introducer or certifier might become even more compelling than it is in the world of credit cards.

4.3.3 Trust models

Trust models are used to model the chains or paths of trust among different entities. The two most commonly known are the hierarchal model used in X.509 and the grassroots approach or a web of trust used in PGP. Neither of these achieves high accuracy in modeling human trust in a wide scope. The hierarchal model is too rigid and too bureaucratic to be usable among people themselves. On the other hand, the web of trust model, in its very basics, lacks the official hierarchies or at least discourages official structures.

We will discuss human trust in more detail later. Nevertheless, it could be seen that a combination of these models might achieve a better solution. People want to select who they trust and in which ways. No single root point exists, unless we make the users themselves such points. In the X.509 hierarchy the multiple root point problem is solved in a way that the relative roots certify each other, therefore allowing for any of those to be considered a root. This forms a forest of certification trees that are connected to each other.

To allow for a single root, the user himself needs to be the root. Thus every user has a differently formed tree. Or if we want to consider the web model, this could be considered in a way that the user is a node in a net and for each user the user's node is pulled up from the basic level. Each node that is connected to a raised point is also raised some amount and the further a point is from the central node, the less it is raised.

Considering a formation for modeling multiple types of attributes, i.e. more than just identity or public key binding, each link has a property that states the kind of trust it

denotes. Now the net or tree is not only different for each user, but also it is different for each property or attribute. Note also that the trust is different if we trust somebody to be something or trust them to make statements about others. In a trust formation, whether we consider it a tree or a net, each node has to be connected to other nodes for the formation to be useful. The user needs to be able to choose which entities or nodes he or she is going to trust and in which ways. That is the user needs to be able to choose which kind of a link he or she connects to each neighboring node with.

Such a model certainly does allow for more flexible modeling of trust, but using it may be too much of a burden for most users. Basically this means that setting up one's personal settings is much more work than it is the delivered-from-above model of X.509. Also, maintaining the model is likely to be more time consuming. On the other hand, nothing prevents the user from installing a delivered-from-above model that might be available from some provider. Such ready-made model could of course be modified for personal use.

4.3.4 Delegation

Trust is generally not transient. That is, if Alice trusts Bob not to lie, the fact that Bob believes Carol not to lie does not mean that Alice would automatically trust Carol not to lie. In fact, this is the same problem as with the certification authorities and certification. Statements made by different entities are those entities' beliefs, not absolute truths.

Another example of the non-transient nature of trust could be found in the PGP model. If Alice trusts Bob as an introducer (i.e. to bind a third person's name and public key), she trusts Bob to be certain that bindings signed by him are accurate. However, that would not, or at least should not, mean that the person whose key Bob signed is trustworthy as an introducer. At least the PGP model does not imply such a thing. It has to be remembered that Bob only states that the person brought him a public key to sign and he trusts the name he signed to that key belongs to that person. He makes no statement whatsoever about the trustworthiness of the third person.

In this way, authorization and attribute certification are very different. Authorizations can usually be delegated if allowed, it is like giving an employee access codes to the company's production line. Attributes, on the other hand, are generally not delegable, one can not delegate the fact that he is a male, or that he has dark hair. He may of course make statements about others that may say that they also are male and dark haired, but this is not the same as delegating those attributes to the third person.

However, a trust model where the leaf node would be different from the branch node might work. This is to say that if Alice trusts Bob to tell if another person, say Chris, is trustworthy as an introducer, then perhaps she could also trust that Diane, a fourth person vouched for by Chris, can be trusted as an introducer. However, Erin, introduced by Diane, is not per.se. trusted as an introducer, since Diane made no statement about her in that respect.

Basically, trust is not delegable in the fullest at least. Therefore, each delegation lessens the amount of trust by some amount. It might be easier to use an identification (or recognition) infrastructure to recognize a known entity, which is directly trusted, and consider attribute certification to be totally undelegable. In other words, only entities that have been chosen as trustworthy in some respect are trusted and in order to make

use of that trust, the entity has to be recognized to really be that entity. If such an entity should want to delegate the trust, it would have to introduce the new entity and tell the user to choose to trust it.

4.3.5 PGP – Pretty Good Privacy

Until the 1990's the problem with public key cryptography was computing power. As PKC is much more computing power hungry than symmetric crypto, like DES, only the government, military and large businesses had the power to use it. Therefore, the implementations were developed with such markets in mind. [Singh99]

Like Diffie, there was another man, Phil Zimmermann, who believed that people had the right to privacy in their communication. His goal was to speed up the RSA encryption and make PGP an encryption solution for the ordinary people and their computers [PGPg1]. PGP uses a symmetric cryptosystem, originally of Phil's own design but later IDEA [IDEA], to encrypt the main message, then encrypts only the symmetric key with the recipients RSA public key, adds that to the message and sends them together to the recipient. Now the recipient, who has kept his private key private, is the only person that can decrypt the RSA encryption and get the symmetric key that can be used to decrypt the main message. Thus, in PGP only the symmetric key, 128 bits in case of IDEA, must be encrypted with the power hungry RSA.

In addition, Phil's idea was to automate all the operation so that it would be easy to use to the end users. Of course, everything is relative. The original package was easy to use in relation to programs of that era. However, in the contemporary world of graphical interfaces and usability the later versions, developed by Network Associates, have been considered difficult to use by modern standards [Whitten99].

PGP's trust model, the web of trust, was a grassroots approach based on human networks. In this model people use their own key to sign the keys of others. In PGP this is also called Introducing. Each person also has a key ring, where they can store known keys. For each of these keys they can specify is that key is to be trusted as an Introducer, so that when coming across a new key they can trust that the name bound to that key is likely to be correct. So if Alice has Bob's key in her key ring and trusts it as an Introducer, she can trust that the key bearing Carol's name, if signed by Bob, is truly Carol's key.

4.3.6 X.509

For certificates to be widely usable there has to be a standard way of coding them so that all applications that try to understand them will have a similar understanding of what the issuer might have intended. Also, CAs (Certification Authorities) need to have a standard way of prescribing the validity period and revocation of the certificate.

X.509 is the ISO standard for (identity) certificates and is related to the X.500 directories. The first version was published in 1988, and it has been updated to allow new usages. The third, and latest, version was published in 1996 and it supports extension fields [DigiCer]. Though other attributes have later been added to the X.509 standard, identity certification is the original and dominant usage. The certificates used in web browsers that identify servers, issued by CAs like VeriSign Inc., are one example of X.509.

X.509 is a hierarchic system. Each node is part of a tree with a clearly defined root. Trust in any node of the tree implies trust in the root, which in turn implies trust on the whole tree. When we add the fact that multiple trees may certify each other, the implied trust is distributed in a manner totally uncontrollable by the end user. X.509 was developed for environments where the trust comes from above: official records, company employee records, etc. In these cases the assumption that certificates are always issued by a more or less official CA is acceptable, even practical. For such uses it is a rigid and functional tool, but for use among end users, it is quite inflexible.

4.3.7 SPKI

Simple Public Key Infrastructure (SPKI) is a less rigidly organized certification structure being developed by IETF (Internet Engineering Task Force) but currently not having too much enthusiasm behind it anymore. It was originally developed with authorization rather than identification in mind. SPKI certificates do not require the issuer to be an official CA or other predetermined entity. Further, SPKI certificates allow for delegation (see 4.3.4), so one can share the authority given to him to others. Of course, when issuing the certificate, such delegation may be denied. In other words, companies might not wish to allow their employees to delegate the rights to enter their building to others, while the administrators who share these accesses to the employees are given delegation rights.

The trust model behind SPKI is more distributed and more web of trust like, than the rigid tree of X.509. Since delegation is allowed, the general case in authorization is more likely a chain of certificates rather than a single certificate. In order to use or access the resource the user must present a chain of certificates with the required authorization, from the resource to herself. Of course, the other certificates in the chain could be fetched from other sources when needed so it is not mandatory for the user to keep all the necessary certificates with herself all the time. In fact, some certificates in the chain may even change during the validity period of the end certificate.

4.4 Trust

Trust is a very personal thing to humans. It is also a very complex thing that forms differently for each individual. Trust also comes in a variety of forms, we trust the bank to keep our money safe, our spouse to be loyal, home to be safe, our friends to not talk bad things about us, our car to work when it is needed, our computer not to crash when writing an important document, etc. None of these things is absolutely certain and some are even likely to backfire on us at times, but we trust things to go well most of the time so we do not need to worry all the time.

Trust is built on experiences and recommendations of trusted ones (among other things). If something trusted fails heavily, the level of trust comes crashing down. It is easier to lose trust than to gain it.

Trust is also relative. A person can say that he trusts somebody else, usually means it in general sense, but even such trust is not absolute. By digging deep enough, something can usually be found that that person does not trust in the other. As an example, not many would trust even their closest friends with their most private correspondence.

In the world of computers and databases, any generalizations of trust can be hazardous. Especially since the models used to model the trust are formal, while trust itself is not.

The formal nature of computers and the informal nature of humans often cause conflicts. As we are trying to build a model that could model human trust better than previous models, at the same time my greatest fear is that the model will fail miserably at some point I never thought of. This means I do not completely trust myself to be able to come up with a 100% proof system. Which on the other hand is a healthy thing.

Chapter 5

The virtual identity and attribute model

We shall discuss how to solve the problem on a general level. The model described in this chapter is not to be fully implemented as part of this thesis, only a partial implementation for demonstration purposes, (see next chapter).

We begin by reviewing what we need. First of all, we need pseudonyms, or rather, pseudonymous identifiers that are unforgeable. Second, we need a way to prove the ownership of the pseudonym. Third, we need a way to attach attributes to those pseudonyms in such a manner that a selected group of attributes can be presented and proven that these attributes belong to the person behind the pseudonym. Fourth, the linkage between the pseudonym and the person behind it must not be easily perceivable, at least, not unless the person behind the pseudonym desires to allow such. Fifth, if such a link needs to be made, to prosecute criminals, such linkage must be possible. However, we can settle for a case where this traceability is conditional, providing it is possible for the receiving side to verify whether such tracing is possible. This is to allow them to require the possibility, if desired. Yet, it must never be the choice of the receiver whether the tracing is performed, such a choice must be left to a party that can be considered objective enough by both the receiver and the person behind the pseudonym. Sixth, the pseudonyms and the attributes must come in a format that allows for open – non-proprietary – implementation, certification and interpretation.

Now, before we get more detailed, a quick intro to the model. People have pseudonyms, identifiers that are basically public keys of a PKC key pair. These allow people to verify that the person has corresponding private key and thus, on a next meeting, that this is the same person (or at least someone having access to the private key). Secondly, the keys, or pseudonyms, can have certificates issued to them. These certificates can be used to make statements about the owner of the pseudonym and if the issuer of the certificate is trusted, the user viewing the certificate can trust that the information is correct. We begin with a more detailed look at the trust model.

5.1 Generic trust model

Usage is where all this is aimed. So let us consider the case in figure 5.1, where the user Alice is communicating with an unknown entity Bob and needs to prove things about herself. She has a key pair to which her attribute certificate is linked. Her insurance company has identified her and granted her a certificate about some fact that they can be certain of, like her age. Now Alice can share the certificate with Bob, and prove that she has access to the private key that is paired with the public one in the certificate. Proving the possession of the private key is imperative to Bob trusting this key to really be

Alice's. If he does not make sure Alice has the corresponding private key, Alice could be giving her Carol's public key instead of her own and behave badly to give Bob a bad impression of Carol.

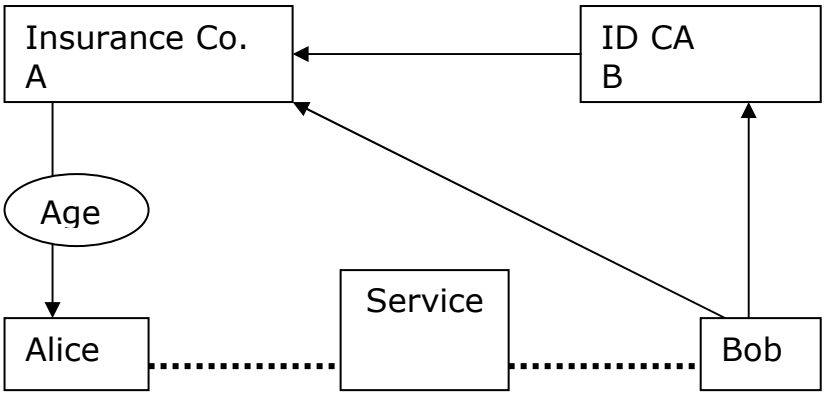

Figure 5.1: Chain of trust in age verification.

In the case that Alice and Bob only have the key pairs and no certificates, they can still make use of them. First of all, they can give the public keys to each other and prove that they also have the private keys. The practical reason for this is that the next time Alice and Bob meet on a channel, they can again prove the possession of the private keys and Alice can trust Bob to be the same guy she was chatting with couple of days earlier and vice versa. The nicks that Alice and Bob use can change from session to session, but using Pseudo Identities (PID) they can verify each other in a recognizing manner. So far no outsider parties or certifiers were needed, only the generation of plain simple key pairs, sharing the public keys, and verifying that the other party has the corresponding private key. But in order to make full use of our model, Alice and Bob should use the attribute certificates to prove thing about themselves to each other.

For Bob to be able to trust the certificate, there are some conditions. First, he must know the entity that has issued the certificate. If he's never heard of Insurance Co. A, why should he trust anything it says? It could be any hoax. Second, he must verify that the key used to sign the certificate really belongs to the very entity he trusts and that it is valid. For this, he usually needs the services of another entity, one that he also trusts, and one that verifies identities and keys of other entities. The ID Certification Authority B, is such a company. The service in itself is transparent and plays no role in this trust model. Note that this is not to say it could not attempt foul play. The important thing is that Bob can trust that the public key he has for Insurance Co. A is really the correct key. He might have gotten the key some other way, but an ID verifier service eases key management. Also, Bob needs to trust that the issuer (Insurance A) is capable of verifying the information, and that it has done so. Finally, he must verify that the private key is really used to sign the certificate. In other words, the requirements in list form:

- *Bob must know the certifying entity in order to have any kind of basis for trust in it*
- *Bob must be able to verify that the key really belongs to that entity and is valid*
- *Bob must trust that the entity is capable of verifying the information*
- *Bob must trust that the entity has verified the information*
- *Bob must verify that the key truly has been used to sign the certificate*

Providing these conditions are met, there is no more requirements that the party has to fulfil to be a trustworthy party in relation to Bob. Since real trust is relative, if Bob does not trust Insurance Co. A, it does not make much difference how trustworthy it is in general. Real trust cannot be forced on anyone. Note the difference in case Bob's employer forces Bob to use A's certificates, now it is the employer's resources that depend on A, not Bob's personally.

As can be seen, there is one major difference in this trust model to that used in authorization chains. Bob not only needs to trust the first part of the identity verification chain (ID CA B) but he also needs to trust the final entity that issued the certificate. There are two major reasons for this: first, since chain could be long, it would have to always certify all the same things which can be hard in practice, and second, since the root issuer does not guarantee the reliability of the subjects, it would make sense to only certify the identity. Thus Bob has first hand trust, he trusts an organization he knows directly, the chain only proves to him that the entity really is the one he trusts.

5.2 Recognition

Recognition in bit space is difficult for human senses. Nick names are natural to humans, but they are non-secure in themselves. A 1024 bit RSA key is secure, but is certainly not user friendly. The same would apply even to a 160 bit hash of the key. Represented in base64 encoded text these would be ca. 170 and 27 character long strings of random characters.

The obvious conclusion is that the computer must do the work. For the computer, keeping track of long strings and comparing them to each other is a piece of cake. In addition to the nick name, which can vary from session to session, the users would present their pseudonyms – the public key – for other users' computers to see, and if the computer finds the pseudonym in file, it can tell the user that a familiar pseudonym has been found. The computer should of course verify that the other user also has the corresponding private key rather than is just claiming to be somebody else.

Once the computer has verified a match for key in the local database of keys of acquaintances, it can show the user an identifier that the user can recognize. A nick used before, an identifier given by the local user, even a picture if available. An advanced software could allow the user to make notes of his or her acquaintances and show those either automatically or if requested. Since the nicks used by people may vary from day to day, it would be nice that the software could tell the user that this person is known to him or her as this and that person. Possibly the nick the person used in the first contact, or an identifier later given to the person by the user. Thus the user could know the person with an identifier best suited to him or her, this identifier would not be shown to the person in question.

5.3 Attributes

In idle chatting it does not make much difference whether the person in the other end is truly what he or she claims. But should the user be interested in making the relationship into something more, it would be nice to have further assurances that the other person is not a fraud claiming to be something totally different than she or he really is.

Similar questions rise when service providers provide services to the user. Sometimes it is important to know that the person has some required attribute. A service might be provided to paid customers only or to certain groups of people. A dictionary service might be available to the employees of a company that licensed the service, or it might be free for those studying at certain universities. Adult services may be required by law to verify that their customers are of age, or that the customer has paid the membership fee. Membership in some associations might give customers a discount at some stores or the store might want to keep track of how many purchases the customer has made so that at some point they can grant him or her a bonus.

The attributes mentioned above are generally attributes that a well-known authority could and would certify about the user. Or the service might give certificates that the customer can show them when he or she comes back to that service. But people also form groups among themselves and trust each other without any external CA. This kind of certification is not well supported by X.509 based systems. The membership in the association example above is closest to this, in small associations there rarely is a large infrastructure for certification and the people in charge would be the people writing the proofs of membership, whether traditional or digital. Similarly to the PGP web of trust model, people might wish to write statements about each other in the digital world. These could vary from specific certificates for attributes just like ones issued by well-known CAs to simple he's-a-good-fellow type of certificates. Such peer-to-peer certificates might be used to model access and other rights in digital communities.

5.4 Certification and Trusted Third Parties

We discussed the kinds of attributes that people might want to get certified and a little of who might issue the certificates. Next we shall provide a deeper discussion of who could certify what. By now it should be clear that there is a need for anyone to be able to certify things, or in better terms, make statements. The important thing to remember is that not all certificate issuers are trustworthy. This statement should be defaulted both for normal people as well as for more official Certification Authorities.

Also, all parties make mistakes. One good example is the Microsoft employee fraud in February 2001. A person managed to fraudulently claim to be an employee of Microsoft Corporation and get two certificates from VeriSign, one of the most reputable Certification Authorities on the Net. Using these certificates the fraudulent person could have signed code so that it would appear as if signed by Microsoft Corp [Ca0104]. This case is also a good example of why there has to be a way to revoke the certificates, one way or another. Such certificates as used for code signing, or server identification, are usually long term certificates with validity periods in the order of a year. VeriSign naturally revoked the certificates as soon as the fraud became public, and in general VeriSign is a trustworthy CA. However, all parties are not and there are fraudulent CAs as well as fraudulent users. More problems result from the fact that even though software, like browsers, should check the certificate revocation status, not all of them do. A notable exception of this is Microsoft's own Internet Explorer.

The important thing when a user gets a certificate is to verify that it is signed by a known and trusted party. When we talk about attributes it is also important to check exactly what the certificate states about the subject. A state register, a known telecom or an insurance company can usually be trusted to certify age and gender, but they might not be the first choice for a CA to certify a membership in a student union. And an

insurance company would be more trustworthy in cases of medical conditions than a telecom. On the other hand, any of these might be trusted to certify that a certain key belongs to the student union, and if the student union is trusted to certify its members, then such a chain of certificates could prove the student membership. However, forming such varied attribute chains is non-trivial and is not considered here. To keep things under control, it should suffice that the most complex chain is that of verifying identities of trusted parties and the final entity is the only one certifying other attributes than identity. See figure 5.2, below, where the optional ID chain is presented in grey. Note that Bob has to trust each TTP in the chain and have a trusted key for the first TTP.

Figure 5.2: Optional chain to verify the identity of the verifier.

It is important to make the distinction between identity and trustworthiness. A PGP key that is signed by a trustworthy friend means that the friend is (to a certain level) certain that this key belongs to this (third) person. It does *not* mean that the third person is trustworthy. In other words, it does not mean that any key signing by the third person could be trusted, unless of course, if he is known through some other means and known to be trustworthy.

5.5 Tracing

One of the requirements was the possibility to trace people that violated some treaty or contract, bringing liability to the anonymous world. As we have mentioned, a simple key pair is highly anonymous, there usually is no way to be sure to whom it belongs. This is especially true for user generated keys.

However, in order to have any basis for issuing the attribute certificates, the Trusted Third Parties must be sure of the identity of the person to whom they issue the certificate. In other words, the user most likely has to use some strong identification, like a FINEID card, to prove his or her identity, before being given a certificate. This of course means that the user is not anonymous to the trusted party.

Usually, since the trusted party is trusted by both the user waving the certificate and the user to whom it is presented, the third party is expected not to reveal the true identity of the certificate subject. However, under certain conditions, like a court order making clear that the user is being sued, or similar strongly objective demands, the trusted party could reveal the identity to the authorities.

We must note, however, that this is the most human link in the model and the most likely to be subjected to abuse. I would assume that the most likely abuse would be users trying to appear as somebody else to the TTP. The real trustworthiness of the third party is in it's ability to both verify to whom it issues the certificates, and how well it protects the privacy of its customers, or in other words, how easily the true identities are shared to demanding parties. A TTP should publish a clear and human understandable statement of the conditions under which it will reveal the identity and abide to it.

5.6 Access Control

Our model was not originally designed to handle access control, but it is an apparent, yet valuable addition, so I'll discuss it shortly. Consider that key pairs and username-password pairs have much in common, so it is not difficult to see that the same pseudonyms could be used for a more secure alternative for access control. Passwords are usually linked to a user name. Traditionally the user name is an identifying part and the password an authentication of that identification. The public key could be considered a similar ID, like the user name it is not secret in any way and can be distributed as widely as wanted. In fact, an email address often contains the user name. The private key is a secret part, like the password it should not be revealed to anyone. And like the password, it is linked to the public half of the pair so that the possession of the private part is considered adequate proof that the user is the one linked to the public part.

Besides the problem that passwords are often transported unprotected over the network, they have another security limitation. The real strength of a password that can be remembered is rarely comparable to more than 40bit key, usually much less. This makes them fairly easy to crack with time. Public key cryptography could provide two improvements. First, the authentication could be done in challenge-response pairs, where the challenge is a string encrypted with the public key so that only the private key can decrypt it and provide the reply. As the challenge string would be random, the response would always be different as well. Not the same, as in always sending the same password. Secondly, the strength of the key (or the entropy) could be much higher, in the class of over a 100bits for a 1024bit RSA key. Thus it would be much harder to crack using brute force attacks[6].

This reasoning would lead to a conclusion that the same keys that are used for recognition and message privacy could be used to replace the passwords, especially in many web services. The user could simply type the username, after which the service looks up the related public key. One reason to keep the user names rather than using the keys directly is that those are easier for humans to handle – we want to keep our names – and the service that had recorded the public key at sign-up would send a challenge, to which the user's software would have to respond; using a browser plug-in or the like.

It must be also noted, that certificates are also usable for access control, this was one reason SPKI was developed in the first place. SPKI certificates give much more options for managing access and user rights. One could, for example, grant a right to use the same account for multiple people (i.e. issue the right to multiple public keys), or even, grant different people different rights to the same resource. Likewise, the same key pair can be used in multiple services with less danger than using the same password in multiple places.

5.7 Using same pseudo-identities in various services

Once a pseudo-identity (PID) key pair is generated, it can be used in many places. Just like the conventional ID card, it is not restricted to a single use. Using different kinds of certificates, all kinds of functions can be bound to the PID.

[6] Brute force attack is an attack where the attacker goes through the number space systematically, trying each possible key until the correct one is found.

After chatting with Alice on a few occasions, Bob wants to have an email account to exchange messages with her, or with anyone else. As discussed in the Access Control subsection above, he could actually use the same PID key pair to replace the password on the account. The email service could register Bob's PID's public key as a replacement for his password or it could provide Bob with a certificate that allows him to access the account. The first is a very straightforward method, the latter allows the service to write different kinds of certificates and could possibly allow Bob to delegate some of the account rights to some other PID.

Delegating account rights on an email service is not very useful, but Bob could as well subscribe to some slightly less personal service, say a web page service to put up the web pages of an association he belongs to. Now he would be the main webmaster for the association, but he could delegate the right to modify the pages to a couple of other people to help him in the task. This way the password does not need to be distributed and remembered by a whole group of people. Also, providing the certificate infrastructure has a revocation scheme, Bob can cancel the certificates without having to change the password, distribute it to the other authorized persons and all of them to remember the new password, or more likely, write it down somewhere. Of course, for revocation to be of any use, it has to be checked.

Further, if Bob subscribes to a BBS or like derivatives on the web, he can use the same PID to 1) access the service and 2) be identified by that PID in his own posts. That is, in the event that Alice or some other acquaintance of Bob's would read the post by Bob, they could recognize that this is the very Bob that they know. Note that such recognition does not mean that Bob would have to sign his post, the service could post the public key of the poster along with the post. Of course, if Bob wanted to post something he does not want everybody to know was by him, he would create another PID and use that. So, he could be known in different societies by different PIDs. This could be handy should he belong to an AA group he does not want his employer to know about, or if he is a dissenter in a strict society.

Of course, none of the technical things can help prevent mishaps and should be not used to avoid moral or legal requirements. What the PIDs can provide him with is a tool that allows him some privacy from the automatic data mining robots. Were he required to always use his real ID, all such parts of his life, especially on the Net, would be at risk to be revealed to anyone having access to powerful enough search engines. It is the digital equivalent of keeping records on separate papers that are kept in the offices of the different societies or associations.

5.7.1 Symmetric and asymmetric keys

Different kinds of keys and cryptographic algorithms can be used for a variety of purposes. Secret keys, i.e. symmetric keys, can be used for session encryption. If there is only one other party, and that party is trusted not to leak the key to any outsider in a one-to-one communication session, the information encrypted by the key can be trusted to come from that party.

Such encryption offers that certainty only for real time communication, not for any lasting effect. Since the symmetric key is not tied to any identity, it must be considered non-confidential information after the session. If any lasting effects are desired, public key cryptography must be used.

Such lasting effects could include encryption and signing. Generally, one should never directly sign ones discussions. Remember what a signature stands for, that the entity that signed a document accepts the liabilities presented in that document. It would be highly frustrating if everything one blurted out would be legally binding, in the same sense, chat discussions should not be signed. Further, since chatting is usually performed a sentence at a time, signing those sentences separately means they can also be used out of context; such signatures serve no one. Signature schemes are used for identity authentication purposes in that a random string is given to be signed and if the signature matches the public key, the party can be trusted to be the one linked to the public key.

However, signing random strings can be hazardous. Since one can never be 100% certain that a certain bit string does not bear some meaning, singing it is like signing an empty document, or a document that has its contents covered. For this reason, encryption algorithms are a safer way to provide authentication. If only signature schemes are available, the signer should attach a prefix indicating this was signed as nonsense or for recognition purposes only.

In encryption based authentication the entity wishing to authenticate an entity uses the public key of the entity that the authenticated is supposed to be, encrypts a random challenge string with it and sends it to the party. If the other party can decrypt the challenge and thus return the unencrypted string, it can be trusted to have access to the corresponding private key and thus be authenticated. This is a safer method because nothing is signed.

5.7.2 Using public key cryptography for chat room security

Since we are using cryptography to protect the integrity of the correspondents, an obvious next question is why not protect the confidentiality of the message as well. And indeed, that is also easy to achieve, providing everybody is using the public key identifier system. Even better, a secure channel can be formed without posing the server with encryption load as well. If the clients take care of the encryption and decryption, the server does not need to bother with the task, and protection from eavesdropping from the server can also be achieved. However, should it be desirable to send the encrypted message to selected recipients only – whether a single one or a group – the server needs to know which ones are to receive the message. Thus the recipients can not be encrypted, and traffic analysis attacks are possible even if the payload itself is protected.

When a new channel is formed, a channel owner is needed. Using public key cryptography for anything longer than a hash or a symmetric key is unwise, therefore the channel owner (or the client of the channel owner) creates a symmetric encryption key that is used for the channel. This key is then encrypted with the public keys of the channel members to protect it in distribution. On an open channel, the client software of the channel owner can handle such distribution automatically, whereas on restricted or private channels the person could control the distribution. Of course, various other settings are also possible so that if no new members are allowed, all requests are denied or ignored. Yet a more sophisticated client could store the public keys of acceptable members and distribute the channel key to them when requested.

One important thing to remember is that the public keys are used to recognize the users, not to make them sign what they are saying. If non-repudiation is in some cases needed,

a special signing function could be used. Yet it is important that the users realize the difference so that nobody signs anything they are simply meaning for chatting.

However, the identifier keys can be used for various other things that have to do with the access to the channel. On a commercial channel the keys (possibly with certificates) can be used to automatically check that only those people who have paid, or who otherwise are members, gain access to a channel. Such events could be appearances of celebrities or some other events that otherwise would become crowded. Also, access to channels with a desired age profile could be restricted to those in that age range. This would allow the providers to keep adults away from teen channels, men away from women's channels and underage people away from adult channels. An important thing to realize is however, that especially if too restrictive rules are applied, people tend to social engineer around the problem. In other words, nothing prevents people from discussing adult things or acting maliciously in the open café channel or even forming a channel of their own, if such thing is supported.

5.8 Key Security

Securing the (private) keys used in the pseudonym is essential. Though key generation and storage are not on the same level as the rest of the model, without them a real implementation of the model has no basis; therefore I would like the reader to be familiar with at least some of the associated problems and discuss some possibilities.

5.8.1 Generating keys

The first requirement is the possession of a key pair, public and private key. The public key is the identifier and the private key can be used to authenticate it, i.e. to prove the ownership of that identifier. Cryptographic keys are generated algorithmically; this usually loads the CPU quite heavily.

There are numerous things to consider in generating the keys and also in storing them. The first issue is privacy, since PKC keys are generated in pairs, public and private key together, the private key has to be secured from the start. In many cases where key escrow[7] is desired, an external authority might be the entity generating the keys, alternatively the private key has to be shared when certificates are issued to that key. Secondly, the implementation of the key generation algorithm forms an integral part of security. Since computers (without specific hardware) cannot generate truly random numbers, the quality of the pseudorandom generator has essential consequences to the randomness, i.e. to the entropy, of the key and thus its resistance to attacks. Also it is important that the generation does not leak the key anywhere. Thus, we must have a trusted platform that both generates strong keys, and keeps them safe.

[7] Gaining access to the private key without having it. Can be implemented either centrally, i.e. storing the key somewhere, or distributed so that multiple parts of the key are stored in different locations. Key escrow is used to recover the key if it is lost, if the key holder has left a company or if government wants to access protected data that it normally would not have access to.

5.8.2 Storing keys

The private keys have to be kept private. Unfortunately, this is not as simple in the digital world as it might seem. The keys are vulnerable on many levels. First, the user must not inadvertently publish them, i.e. send them somewhere. Secondly, the user must keep them in some place (file or system) that is not accessible by others. This is a problem on multi-user computers, the storage files need to be encrypted. The problem is actually very similar to that of multiple passwords, except that no one can be expected to remember a single cryptographic key and thus the keys must be kept in storage and protected.

In the previous section on key generation we discussed the generation of key pairs on trusted platforms. These platforms can be designed so that they will not give out the private keys under any circumstances[8]. Smart cards are one such device. For example, the FINEID keys are generated in the factory and stored on the card. Users of the cards are assured that the private keys are not left in any records except on the card. FINEID cards do not create or store new keys on the fly, but such smart cards could be entering the market in the future.

Trusting the hardware platform is an assumption that must be made in every case of using cryptography. Or rather, there should be no feeling of distrust towards it. If one distrusts the platform, there is no sense in using it for anything sensitive. Even if in the real world the platform always has weaknesses that may be exploitable. And the more complex the platform, the more weaknesses, no matter how trustworthy the provider is.

Let us consider an unrealistic case of a securely encrypted file on a secure file system where no unauthorized person can gain access to it. When the data on that file must be used, the file is read into memory, then a cryptographic software decrypts the data and stores the unencrypted data in another part of the memory for the application to use. Any part of the memory may be accessible by malicious programs, either directly or when the operating system swaps the memory onto the hard drive. As can be seen, all the parts of the system must be secure. In this study we will not go any deeper into platform problems; that is a different field of countless problems.

However, the storage of the keys is still an important question. Let us consider a mobile user. If he or she wishes to use the keys on multiple computers like the campus computer classes, at work, or at a friends place, the securely encrypted file on his or her personal computer is not enough. We discussed smart cards because they are one possible platform that is easy to carry around which keeps its own operating environment along. There are other similar devices, like the iKey [Kingpin00] and other similar electronic "keys". Most of these devices still rely on an external interface to the user, the computer.

Another possibility would be to keep the encrypted keys in a publicly available database. In this case the faith in the security of the encryption has to be high. On the other hand, in most cases the assumption must anyways be that the antagonist has access to file and thus faith in the encryption has to be high. The gain from putting the

[8] This is the design goal. In all real systems there are surprising weaknesses that may allow the key to leak, for example in smart cards it may be possible to retrieve the key through analyzing the power consumption in the card's connectors.

file on the network is that the user can now fetch it from any computer that is connected to the network. In fact, the store could even be on the user's homepage, though it might be wise to avoid too much unnecessary exposure. Of course, an SSL [SSL] (or the like) based access control could be used to minimize unwanted access even to the encrypted file.

Chapter 6

Implementation

The demonstration implementation of the model is quite simplified. Of course, we have the three required parties: Service, Client and Trusted Third Party as presented in figure 6.1, but all except the client part have very basic functionality. Therefore the client could in theory be used with existing chat servers. The emphasis has been on the client software being able to demonstrate the use of pseudonyms to recognize others and certificates to verify two attributes, gender and age.

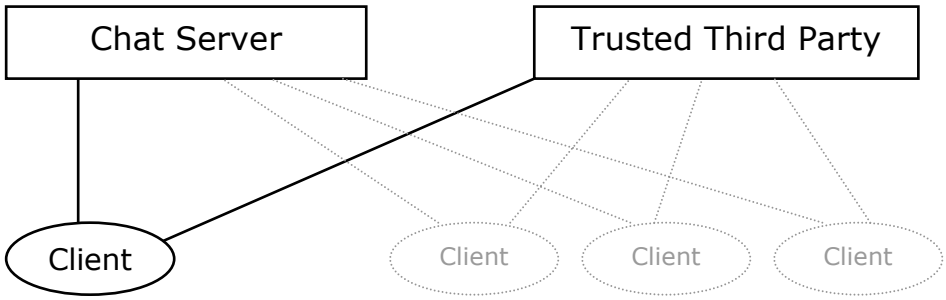

Figure 6.1: Implementation modules and their connections.

6.1 Chat application

The most central part of the demo application is the chat client. While the other parts are essential to the functioning of the system, they contain mostly trivial functions. The real new functionality is in the client, which can in fact be implemented in such a fashion that it could be used with existing servers. The CA side must of course be provided to support the attribute certification, but even that would not have to be online. We have implemented three commands for the server and of these only the one allowing a single recipient can be considered a requirement, since otherwise one would always be sending keys to the whole channel or room. The other two are used for quitting the session and querying who is in the room, which can be either left out or circumvented if necessary.

The architecture of the client can be seen in figure 6.2. Basically there are three major parts: the main program which is basically the GUI (Graphical User Interface), a parser that listens to incoming messages from the channel and decides what to do based on them and, finally, the pseudonym. Inside the pseudonym are also lists that contain acquaintances and trusted verifiers

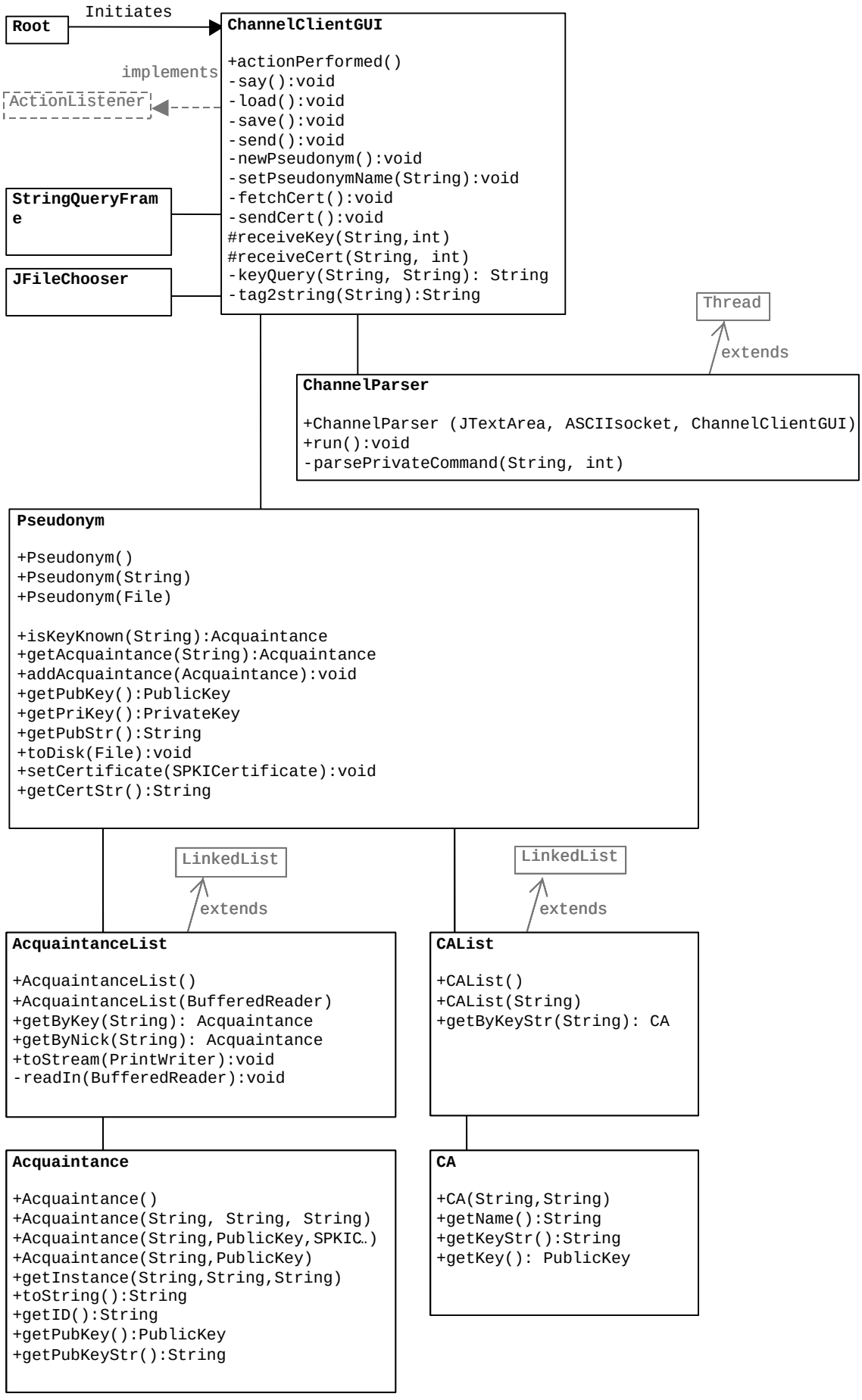

Figure 6.2: UML of the Client Architecture

In addition there are three minor classes: `Root` is used to run and initialise the client, then giving control to `ChannelClientGUI`, `StringQueryFrame` is used to query for a nick if it is not given as a parameter to `Root`, and `JfileChooser` is used to query the filenames for CA and acquaintance save files.

6.1.1 Main Class: `ChannelClientGUI`

This is the heart of the functionality of the client software, it draws the user interface and listens to the user's actions. Since the goal of the implementation is to demonstrate the use of pseudonyms and certification technology, the emphasis on the UI (user interface) was on visibility and ease of implementation, rather than on building a good GUI. Therefore we ended with the button-rich look-and-feel that can be seen in figure 6.3.

Each event, whether from the user or from the channel, ends up calling some method on the GUI. The only exception to the rule is an incoming message that is appended to the channel text area. This is because of the fact that method calls are CPU expensive and these are the most common events. Thus the `ChannelParser` class appends the incoming message directly to the `JTextArea` (the multi line text field) of the GUI.

Each of the buttons on the top of the screen has a method dedicated to it and these methods are private.

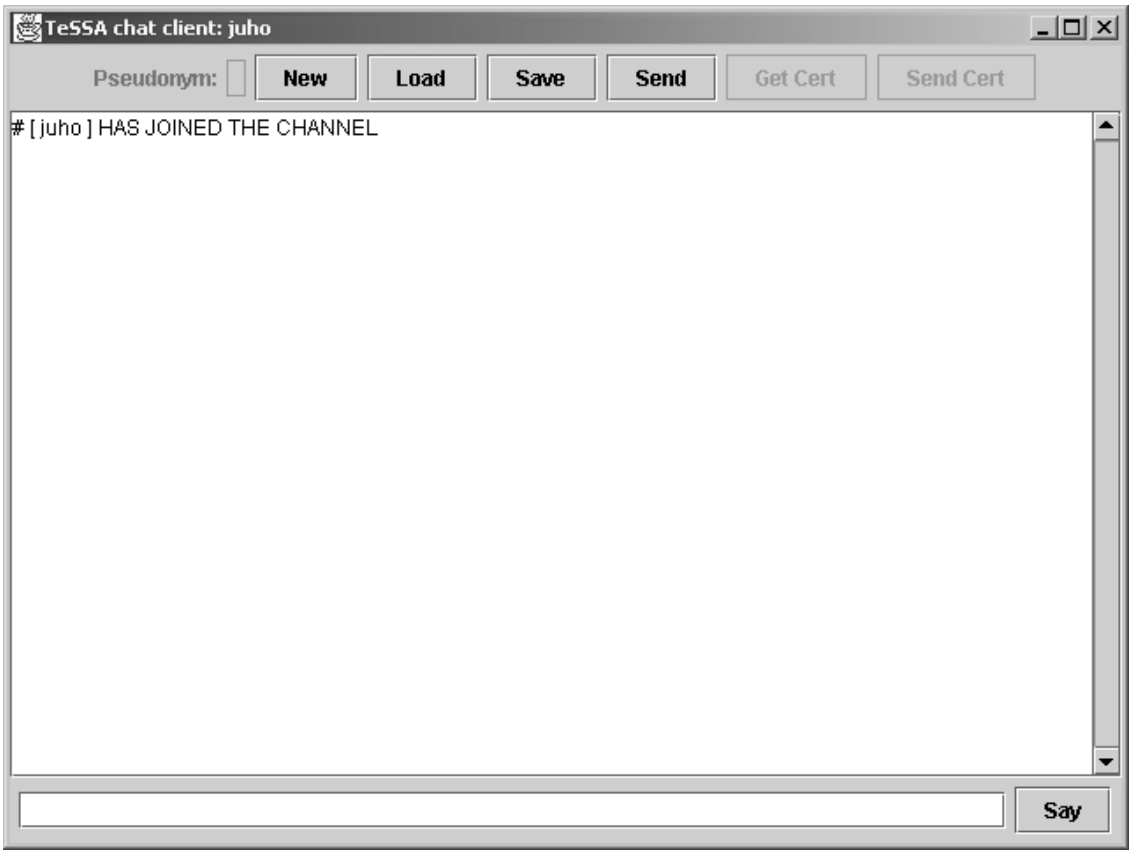

Figure 6.3: The GUI of the chat client window.

`say`	sends the message in the write area (single line text box at the bottom of the window) to the channel
`save`	Saves current `Pseudonym` to disk
`load`	Loads a `Pseudonym` from the disk
`send`	Sends your public key to a recipient (which it queries)
`newPseudonym`	Generates a new key pair and starts using it as a pseudonym
`fetchCert`	Contacts CA and fetches a new certificate. This also opens a query window and asks for your social security number.
`sendCert`	Sends your certificate to a chosen recipient. (Opens a query window.)
`receiveCert`	Is symmetric to the `sendCert` method but is not associated with a button. When the client receives a certificate from the server, the parser calls this method, which in turn shows it to the user in a pop-up window, rather than printing it to the log window. `Pseudonym` is consulted to verify whether the issuer is a known entity. A pop-up window shows the certified gender and age, and also who has certified it, if known.
`receiveKey`	Is likewise a symmetric method for receiving keys and showing them to the user. The `Pseudonym` object is consulted whether this is a known entity or not. If it is, the identifier by which the Pseudonym knows the key is shown. If not, the key is shown to the user and he is given a choice whether to save the key in the `AcquaintanceList` so that it can be recognized on further contacts.

The messages that the user types into the write field are send to the server as such. This allows for the user to use the server commands in a command line style and saved effort on the non-essential implementation. There are three commands that the server currently understands:

`/who` Which returns a list of members on the channel

`/quit` Which closes the connection and terminates the client. An additional leaving message can be written after the command itself.
`/quit` OR `/quit Going for lunch`

`/to` Which sends the message only to a single recipient (and a copy back). The recipient is specified as an (single word) argument after the command and the message is typed after it.
`/to Listener Hello my friend!`

6.1.2 Channel traffic interpreter: `ChannelParser`

This class has two basic functions: it listens to the stream coming from the server and it parses it to find out if there are any specific commands to catch that might require other actions than writing them as such for the user to see. For the object to be able to independently listen to the incoming data flow, it runs in its own thread.

First the parser checks if the message is a private message (all command messages in current implementation are such). This is indicated by the fact that the server inserts "_<<<_" before the author nick and "_>>>_" after it. (Underscore denotes <space>

character.) If this is further followed by a slash ('/') the parser tries to interpret it as a keyword. Current implementation has two recognized keywords: /key and /cert. If either of these is found, the next <space> is located and everything after it is passed on to the GUI either using its `receiveKey`-method or its `receiveCert`-method. The GUI will then, in collaboration with the `Pseudonym`, decide how these are shown to the user.

6.1.3 Pseudonym: `Pseudonym`

This class contains the virtual identity that the user is currently using. It has three major parts: the key pair, the list of known entities and the list of trusted entities. The `Pseudonym` acts as an interface between the lists and the GUI, coordinating all activity concerning the virtual identity. These include storing the certificate, if there is one. It also has the functionality to read and write its key pair and its acquaintance list to a file for later use. Current implementation also only uses a plaintext file, a real implementation should be more secure. Following format is used:

```
::SELF
MY-PUBLIC-KEY-IN-HEX-FORMAT-FOR-ASCII-COMPATIBILITY
MY-PRIVATE-KEY-IN-HEX-FORMAT-FOR-ASCII-COMPATIBILITY
::ACQUAINTAINCE
IDE:<identifier by which this entity is known by the user>
PUB:PUBLIC-KEY-OF-ACQUAINTANCE-IN-HEX-FORMAT
::ACQUAINTAINCE
IDE:<identifier by which this entity is known by the user>
PUB:PUBLIC-KEY-OF-ACQUAINTANCE-IN-HEX-FORMAT
```

The double colon `::` is used to denote a beginning of a new entity and is followed without space by declaration of what kind of an entity it is. SELF denotes the key pair of the pseudonym in question, first public then private key. ACQUAINTANCE has at least two attributes, an identifier by which the user knows the person (or other entity) denoted by IDE:, and his/her/its public key converted to hex format. In addition, it could have a CER: attribute to which a certificate could be stored. However, saving of certificates is not currently implemented.

Since the same user might have lots of trusted third parties and copying these to different pseudonyms would be hard, the CA list is in a different file and read in automatically when the client software is executed. This implementation works in the multi-user environment where each user has his/her own files, but is a bad idea in shared system environments since users should generally not pool their trusted entities.

6.1.4 Trusted parties: `CAList` and `CA`

These classes are used inside the `Pseudonym` class. Trusted entities are listed in `CAList`, which is a list object extending java's `LinkedList`. It has the functionality for reading the user's trusted parties from disk and for finding a given public key when a certificate is received. In a final product the number of methods would be greatly increased, as more functionality would be added to the product.

The CA object in itself is very simple, it simply binds together an identifier by which the user knows the entity in question and its public key and has get-methods for retrieving these.

6.1.5 Acquaintances: `Acquaintance` and `AcquaintanceList`

Similarly to the trusted entities in the previous sub section, acquaintances are listed in their own similar list, though these are more complex. The list can be searched using either the identifier or the public key and disk operations are implemented. Further, there are multiple constructors for the Acquaintance class to allow instantiation of objects from variable sets of data in variable formats.

6.2 Trusted party

The implementation of the Trusted Third Party server in this demo is very simple. Since the central concept is the use of certified attributes, not the issues behind granting and distributing the certificates, the TTP server's only function in the demo is to allow for the client software to fetch certificates for the pseudonyms. For a real implementation with liability issues to consider, the case is complex, but we settle for an entity that creates and signs the certificates.

The attributes used in this demonstration are gender and age. In Finland both of these attributes can be derived from the social security number (SSN). The number is of the form `ddmmyy#nnnc`, where dd is day, mm is month, yy is the last two digits of the year the person was born. # is a separation mark which also specifies the century. In the latter part, nnn is a serial number for babies born that day in such a fashion that males are only given odd numbers and females are only given even numbers. Finally c is a checksum over the previous numbers and it may be either a digit or a character (0..9, a..z). Therefore the code represents the date of birth (and thus age) and gender (even or odd serial number). Since we are not demonstrating security or trustworthiness of the TTP server, we shall simplify the implementation so that the server will write certificates based on a given SSN. In a real implementation the server could use something like the FINEID card to verify the SSN, but here we trust the user, for the sake of simplicity.

The TTP server will listen for connections on a given port, when a client connects, the server responds with "`HETU:_(ppkkyyOnnnN):_`" (underscores denote spaces), in which HETU is SSN in Finnish and the latter part describes the format. This allows for the server to be used from command line as well for ease of testing purposes. The client is next supposed to return the HETU, from which the server decodes the age and the gender. The client is then supposed to give its public key and the server writes a certificate of the given attributes to that public key. It should be noted that this protocol is definitely not secure in any way, its only purpose is to allow the clients to get certificates to send to each other, not to verify that the information is accurate. A more secure implementation can be achieved using existing cryptographic protocols and electronic identification methods.

6.3 Chat Server

As in the case of the TTP server, the chat server in itself does not contain any notable new functionality. It is a simple server that allows users to connect to it and send messages to the channel or each other. Some of these messages will be the users talking to each other and some will be the client software talking to each other (to exchange the recognition and certification data).

The original idea was to implement the server as simple as possible, distributing all messages to every member on the channel, thus it would not be required to parse the messages in any way and any transparent server could be used. However, three server commands were later added to make the system more functional. These are: `/quit`, `/who` and `/to`. The first terminates the session and may be complemented with are reason for quitting. The client is supposed to send a `/quit` before terminating, but the server recovers even lost sessions. `/who` is used to query for channel participants, when the server receives the query, it lists the nicks of the chatters and returns it to the querier only. Finally, `/to` implements functionality to send messages for one recipient only. The next word is considered the nick of the recipient, and it is searched for in the channel member list. If a match is found, the message is send to that person and a copy back to the sender. This functionality is the only one of the three that would really be necessary, because otherwise the users would end distributing their keys and certificates to all participants, which would not be acceptable.

In case the clients use encrypted messages, the server even cannot listen to the conversation. However, if the clients want the messages to be sent to one recipient only, the `/to` command and the recipient nick naturally has to be used unencrypted. Of course, messages could be sent to all recipients but only those with the key could decrypt them. In that scenario however, everyone would see that the person in question is sending encrypted messages and the approximate length of the message. This would also increase the network traffic. The server should in no case be loaded with requirements for cryptographic operations, since in any larger scale operation the server easily gets overloaded.

Chapter 7

Analysis

A user faces many problems when trying to maintain a desired level of privacy, no solution is fool-proof. Trust management is a very difficult task; for one, trust is too complex a phenomenon to be easily modelled by simple certificates that would automate the process. Human trust is needed as long as humans want to stay in control. However, certificates can be used to help people in areas where there is no trust at all without them.

7.1 Fulfilment of the Criteria

In this subsection we will go through the set of 11 criteria which we set in section 3 and compare the model with each of the criteria to see how well they have been fulfilled.

Anonymity to new acquaintances

The criterion was that the identifier could not in itself be directly linked to the user's identity. As a user can create the keys herself, there are initially no links between them and the user. Naturally, when the identifier is used, the link to the person behind it strengthens. Let us take a look at two of the most common events that strengthen it.

When the user goes to a third party to apply for a certificate they usually have to identify themselves using some trusted means. This could include a smart card based ID card etc. The third party may then record (if necessary) the public key and the users identity (i.e. identifiers and/or attributes that uniquely identify the user) in their database. Since the idea is that this linkage is only revealed after a set of criteria for the revelation are fulfilled (such as a court order) it is feasible to conclude that a reputable third party does not leak the information willingly. Crackers breaking into the database may, of course, be able to steal that information, but since the model assumes that users use TTPs which they consider trustworthy in this respect, these cases fall outside the scope of this particular model.

Further, as the user interacts with other users he reveals information about himself. If some party collects such information, eventually there will be enough to identify the person. This is one reason why the user should use different identifiers at different services (where the connection is not needed) and changes the identifier from time to time. As long as the amount of information that is publicly known does not exceed the level required for identification, the user maintains a certain level of anonymity.

Of course, in chat room use as in many other uses where the pseudonym are used for personal affairs rather than more official business, shopping, or video rental etc. the users are likely to grow fond of the pseudonyms, to build personas behind them and

gain a reputation for them. The same phenomenon can be seen on IRC and chat rooms where there are well known personas. Just like real life, if a persona becomes a celebrity, the curses of that status will follow. Cryptographic pseudonyms are, however, much harder to forge than plain nicks. Secondly, in case one wants to get rid of the celebrity status: when one's "face" is a self generated random string, it is much easier to generate a new key pair than to have to replace one's physical face. How users really would use their pseudonyms is open wide for future research, especially in the fields of psychology and sociology.

Traceability in case of Crime or Misuse

Traceability has a firm foundation in the third parties. A user creating a new key pair and using it can be quite impossible to trace if no one really knows the connection. Thus, the liability trustworthiness of a previously unknown pseudonym is relative to the certificates of the third parties. When providing service that requires traceability, the provider should check that the user has a certificate from a trusted party that provides such services. Also, the provider needs to trust the third party to really exist and really provide the service. Further, they might also want to check the conditions for the tracing.

To clarify, let us take a look at two examples:

1. A bank that issues payment certificates would likely guarantee the transaction, at least to a certain limit. In this case the third party is involved in the transaction and must be known and trusted anyway. Since the provider can trust that the bank will pay her the sum, she does not need to worry about who the customer really is. The bank will take care of the dept collection and will most likely suspend the certificate if the customer behaves badly.

2. A mischievous user might set up a CA service that appears to grant certificates and provide traceability services, but never verifies the attributes is certifies or does not store the information required for tracing. This is why an unknown CA should never be trusted. However, if the CAs provide cross certification among themselves with some kind of liability, a chain of trust might at some cases be enough for the provider. Naturally, there is much room for consideration.

Easy to create new identifiers

Since the user has software that generates new key pairs, and may even generate them in advance (and store them securely for later use), the user is not dependable on any third party to generate they keys.

Note, however, that the ease of generating new identifier keys does not imply the ease of certifying those keys as well. Since each key has to be certified individually (usually for each attribute as well) and each certification may require strong identification, automating the process of certifying multiple pseudonyms is not among the most likely initial services.

Secure against forging

This is very much a question of how strong cryptography is used. There are as many opinions among security experts on this as there are experts. Some believe in using as long keys as plausible compared with the computing power at disposal, while others

believe that simply reaching the level of annoyance to the cracker is usually enough to make them uninterested.

The real question really becomes, how unlikely one wants to make it that someone can crack the key and how much processing power does one have. Keys that are used in mobile terminals can not be as long as those used in a workstation computers, not as long as the terminals have no cryptoprocessors specially designed for long word operations. Also, the CAs and providers need to have more secure keys than the average users, since they are more likely targets for attacks. Similarly, the more exposed the user's key is, the stronger it might need to be. For example, a regular at a popular channel might want to have a stronger key since being well known makes her a more likely target. The strength of the key depends on the platform and choice of the user; implementations should support any standard key lengths.

Usable for access control

Asymmetric cryptographic keys can be used for challenge-response purposes to authenticate the user. Both encryption and signature keys can be used for this purpose but there are some issues concerning the signing of random strings that must be taken into account [HAC].

Since the cryptographic keys can be used to authenticate the user, they can be used for access control. The real question is whether it can be made convenient enough to replace the passwords. Since no one can be expected to remember the keys, they must be available at any time such authentication might be required. This in practice requires a physical token carried by the user. Ideally the tokens should be pseudonymous and small enough to carry at least a few of them around.

Provider independence

If the keys are transported in a format that allows any implementation to reconstruct the key, any standard types of keys can be used. With the certificates the format choices are even easier, since most commonly used certificate formats like SPKI and X.509 have standard presentations that any implementation should adhere to.

The contents of the certificates are a more complex matter though. Though SPKI standard specifies the formats for presenting the issuer, subject, validity and delegation, the format of the authorization field is open to allow its use for a variety of purposes. Therefore a number of keywords for common purposes should be standardized and the meaning of them should be well defined. In addition to these, however, it will always be possible to write any kind of statements to a SPKI certificate. This allows one to even write an essay and certify somebody with that. However, no one should expect such certificates to be processed by the computer.

Peer-to-peer[9]

Since the model is provider independent, it can also be used in peer (human) networks like in associations, among friends, etc. The open format of the authorization field in SPKI certificates allows such closed forums to write their own attributes, but it must be

[9] On the certificate level, not the network level.

remembered that those outside the community can never be expected to be understand or trust such attributes or certificates.

Proof without identification

Proof or trust is certainly possible, providing trusted entities begin to offer the certification services. These services were the original reason for the research so their demonstration was emphasized in the concept implementation of this thesis as well.

The concept implementation demonstrates how entities can make statements on the gender and age of a person behind a pseudonym. The user sees which entity has certified the attributes. If the certifier is an unknown entity, the given attributes are still given, but the user is told that this certifier is unknown, thus it should not be trusted.

Low cost

Possible costs to the end user come from three sources: client software, certification services and key storage hardware. In addition there is much work to be done, like in specifying the keywords for attribute use so that provider independence becomes reality, but the end user does not come in direct contact with such costs that are mainly one time initial costs.

Chat client software can be implemented as freeware, it is not very complex but for the cryptographic libraries. For even those, open source versions are available.

The cost for implementing the certification server, space for storing customer identities and pseudonym keys, and management of all such data is the likely source for highest costs in the model. Thus it is very unlikely that well-known large entities that would be trustworthy would provide the service for free. However, if the use of such certificates would be wide-spread, the cost per certificate would most likely be low. In addition to these costs, the initial cost for acquiring hardware for strong electronic identification like the FINEID system, are still fairly high, in the order of 100 euros.

Accessible on multiple terminals

With a cryptographically protected pseudonym file, which is available over the network, the pseudonym could be used in any computer with client software. And if the software comes in form of a java applet it could even be accessible through an ordinary browser without having to specifically install it.

Secure storage

The security of data storage can be implemented through either the file system or file specific encryption. The latter has the advantage of working in both multi-user systems as well as systems that do not support user differentiation. Many home computers are today do not (at least in practice) protect the files of one user from others. In case of pseudonyms it is necessary to allow only the one person who acquired the certificates to be able to use the pseudonym. Password protected encryption allows for such, but does not enforce it.

7.2 Concept implementation limitations

The implementation is naturally only a small subset of the model and has limited error handling capability and does not enforce restrictions on the use of special characters.

7.2.1 Key Authentication

Since implementing key authentication (challenge-response pairs) to a protocol that is basically stateless packet protocol would have made the design more complex without adding much to the demonstration of certification techniques, it was left out. In any real implementation authentication is naturally a fundamental requirement.

7.2.2 Identity verification

According to the model, online identification towards the certification authority is done using electronic identity cards like the FINEID. Implementing support for such cards would have required more effort than the complete concept demo otherwise. Thus we decided to simply ask (without verification) the user for his or her social security number, which in Finland includes information from which gender and age can be derived. Again, this would be a fundamental flaw in any real implementation.

7.2.3 Forbidden characters and message problems

The implementation does not check at all what kind of strings the user is sending to the server. Since the server and the clients check the incoming lines for the character '/' denoting commands, using these in the beginning of the message causes problems. Another specific string is '>>>' which is used to identify private messages and those are checked for keys and certificates again beginning with the '/' characters, this time inside the messages, not just the first character. And finally, the client does not check that nicks would be a single word, but private messages are delivered based on the first word after the `/to` tag. This leads to that multi word nicks can not receive private messages, including keys or certificates.

7.3 Remaining problems

While devising the model, it has been necessary to ignore many essential problems to keep the focus on the main idea. Also, many new problems arose that were not thought of at the beginning, some of them more important or more difficult than others.

7.3.1 IP addresses

If anonymity on Internet is desired, one needs to look at the whole picture. What one reveals about oneself is not all that can be gathered. Some of these problems are down to the basics of the way the Internet works.

Internet is a packet delivery network. In fact, it resembles the conventional mail service quite well. The client sends a packet (query) to a servers address, the routers deliver it there if they can, and the server sends a packet back to the client. For the system to work, every computer on the network must have a unique address and all the parties involved in transporting the package can see where it is coming from and where it is going. So, the server knows where you are, and if you are the sole user of that address, who you are.

We have ignored this problem in this thesis because it is on a very different level and there are many projects working on solving the problem. Most of these solutions are similar to the way the mail service solutions to the same problem, re-mailing. Some examples are: Onion Routing [Onion], LPWA [LPWA], Crowds [Crowds], Anonymizer [Ano] and the Freedom Network [Freedom].

7.3.2 Liability of the anonymous

One of the greatest fears of organizations in anonymity is the wild west mentality it allows for. If people can remain anonymous, if they feel they cannot be caught, they feel they can do anything to anyone. Examples could include the motion picture The Hollow Man, and the nuisance of the Net, spammers. Even when most ISPs have forbidden spamming, the act of sending loads of unconsolidated email, fortune seekers protected by forged sender addresses in emails keep bombarding people with unwanted mail.

Such lack of self regulation in humans is the biggest reason that any infrastructure that is used for any actions requiring any kind of liability must have a way of tracing back the user, should his abuse become unbearable.

7.3.3 Identity revelation conditions

Even when the certification authority or trusted third party has in its possession the identity of the user, it cannot reveal that identity on too loose grounds. Part of the trust in the TTP is due to the trust by the user, that his or her true identity will not be revealed on too loose grounds. That trust has to be maintained if the TTP wants to keep its customers and gain more of them. On the other hand, it must be capable of revealing the identity when it is truly necessary, because the other part of the trust comes from the other side, that the other users and entities can trust that those who abuse their anonymity too much can be brought to justice. If this trust is lost, then the customers again abandon the TTP since its certificates are not valued by anyone.

For this thin balance to be maintained, the TTP must have a clear and firm revelation policy that does not change much. Policies like in free web services that are due to change without notice and by continuing usage the user agrees to the changes, are quite unacceptable for use in circumstances where the private information of paying customers is at stake.

7.3.4 Willing Compromise of private key – ID sharing

On the user end the trust of the TTP can also be abused. If the user who has acquired some certificates from a trusted party decides to distribute his or her public key to some of his friends, this becomes a true problem for trust. First of all, such abuse can be extremely difficult to discover, especially if no crime is committed. Secondly, it undermines the trust of the certificates by the TTP.

One possible solution is that since electronic signatures can be made with the private key, those signatures could have legal significance. Knowing that sharing the private key allows the other person to make electronic signatures in one's name should lessen such crimes. People should also keep in mind that such sharing leads to the person being able to impersonate them.

This problem will also be quite difficult to prevent totally. In families computers and accounts are shared, members of the household are trusted and children use their parent's accounts etc. The problem is that many such things are not considered that personal today and it will take time for people to learn that the electronic personalities are not always just tools that can always be lent to one's friends.

7.3.5 Unwilling compromise of private key - Stolen ID

Very closely related problem is also the fact that the private key is compromised without the owner's consent. This problem can be just as difficult to discover. Further difficulties to the TTP are presented by the fact that it can be quite impossible to prove whether the legal owner of the PID consented the compromise or not.

Such identity theft is one of the most feared crimes of the electronic society. They are extremely difficult to spot and extremely difficult to clear the mess afterwards, finding and proving all the records that the actions of the hoaxer have corrupted.

7.3.6 Certificate validity period

How long should a certificate be valid? Depends on the application of course, but also on the certificate infrastructure. Does the infrastructure have good schemes for revoking certificates that have been compromised? If not, then the durations should be short.

7.3.7 Certificate revocation

Revocation is yet another difficult field that has no clear, easy and efficient solutions. Since certificates were designed to be independent document requiring no online access, requiring that access to verify certificate validity creates a serious conflict. On-going and future research may provide solutions that allow for alternative approaches to validity verification.

7.3.8 Negative recognition

Being sure that somebody is not some old acquaintance would often be useful, especially in places where some regular troublemakers keep harassing others. However, achieving this would not be simple. Simple usage of public key identifiers does exactly the opposite, allows one to hide his previous identities under the new one. Since that is part of the goal of the infrastructure (though against corporations keeping long term track of all the actions of their customers) the infrastructure itself cannot produce the same effect.

In fact, negative recognition could be feasible with certain restrictions. First, only one trusted party should exist, or all trusted parties should have a shared database. Secondly, all identifiers must be registered at the trusted database and the true identity of the customer be recorded there. With such a scheme the trusted party could make crosschecks by searching its database for all the identifiers of the true identity of the queried person. As can be seen, one more problem arises, either, the trusted party must return all the identities linked to that true identity which would seriously compromise the privacy of that person, or the one making the query must provide with a set of identifiers against which the query would be made. Note that in the latter case, unless the number of identifiers to be compared is limited, an attack could be formed against the identifiers of the queried person by always making queries using all known

identifiers. Further, multiple trusted parties are an essential part of the infrastructure. Therefore, it can be concluded, that negative recognition does not fit well into the modelled infrastructure.

7.3.9 Identification through attributes and actions

Even if the pseudonym in itself would be totally anonymous and untraceable, the person behind it could be traced by monitoring his or her actions and behaviour. In [CanPs] the authors present a system that reads a large amount of articles posted in a newsgroup and groups the articles by the authors using only the text written by the authors.

In a virtual environment where people are sharing information about themselves and views, total anonymity can not be achieved and neither should it be the real goal. The smaller the community, the better people know each other and less space there is for anonymity. Under normal circumstances in environments that we have considered, such "identification" should not be a serious problem, but there may be environments where it is.

Chapter 8

Conclusions

The goal of this thesis has been to enhance existing technology in order to add features like privacy and certification. In the early years of networking these were not considered important and in part were not even practical. We have worked on a model that could be applied to a wide field and implemented a very narrow study case for certifying two attributes in a single certificate. Future work would allow specification of keywords for a number of attributes, which in turn would allow for true wider usage.

We have shown that the technology can be used on fairly simple existing servers by adding a little intelligence to the clients. The recognition part of the technology can be used even without a single third party ever providing certification services, which are the only truly commercially oriented part of the model. Further, taking the grassroots approach of PGP, people could make use of the model among themselves by using peer-to-peer certificates and off-line certification.

For the wide field usage of the model, however, the trusted third parties are essential. These entities, which are well known enough to be trusted by masses of people, include large corporations like banks, insurance companies, telecom operators etc. Further, it should be remembered that all such known parties are not trustworthy and the level of trust given to them should depend on how well they verify the information that they certify. For international certification, these corporations would also need to be multinational, or otherwise known and trusted in the target countries. Future work might envisage a model that would allow known trusted parties to certify other trusted parties, but the non-transient nature of trust makes such practice difficult, especially in a formal digital language.

Privacy is a fundamental human right. Yet it is among the first that most governments are willing to break when threatened, if they can. Yes, most people want criminals and terrorists to be captured and their schemes prevented. And yes, most of the time these same people may be willing to say they do not care if anyone listens to their phone. But once the door is open, it is much harder to close it and there are the, often rare, situations in most people's lives that they do not want anyone to ever find out about. And even when people have nothing to hide, we need to remember that any surveillance equipment can and is likely to be used for unethical and corrupt purposes. Companies need to remember that these networks can also be used for industrial espionage and for example the ECHELON network of US is believed to spy on foreign trace secrets as well as national security issues.

Therefore, it is not surprising that there are people who find it hard to trust governments, especially those of power states. And in the community of security researchers the Privacy Enhancing Technologies (PETs) have been a hot topic lately.

New technologies are being developed that would fix the holes in existing infrastructures. The correct route would naturally be to design the infrastructure in such a fashion that these problems would disappear, but unfortunately the fundamentals of a widely used technology are hard to change, especially when interoperability is the expected norm.

In the current network it is easy to log people's activities, both connections and content. In order for people to have control over their privacy, they need control over publicity of both of these. Cryptography can be used to make it much harder to peek into the content, but the connection is still easily traced. With some new design into firewalls, proxies and addition of re-routing services, the connections could have more privacy. In this kind of location masking techniques and services, the ISPs and network operators would play a significant role, either as a privacy protector or a surveillance monitor. Meanwhile, legislators are concerned that any hiding of user activity turns the Net into a wild west, thus adding the possibility of tracing when justified might be required to balance the privacy. Perhaps using pseudonyms that are protected by independent organizations but that can be traced to the person behind it, might provide us with some of the necessary tools to achieve that.

References

[Adams99] *Users are not the enemy,* Anne Adams & Martina Angela Sasse, Communications of the ACM, vol. 42, no. 12, Dec. 1999

[Amoroso94] *Fundamentals of Computer Security Technology,* Edward G. Amoroso, Prentice Hall, 1994

[Ano] www.anonymizer.com [referenced: 7.8.2000]

[Bartle] *Early MUD history*, Richard Bartle, email message, http://www.apocalypse.org/pub/u/lpb/muddex/bartle.txt

[Burgoon89] *Maintaining and Restoring Privacy through Communication in Different Types of Relationships*, Burgoon, J.K., Parrot, R., Le Poire, B.A., Kelley, D.L., Walther, J.B., Perry, D., Journal of Social and Personal Relationships, 6, 31-158, 1989.

[CanPs] *Can Pseudonymity Really Guarantee Privacy?*, Josyula R. Rao, Pankaj Rohatgi, Proceedings of the 9th USENIX Security Symposium, USENIX Association, Aug. 2000

[Cast] *The Information Age: Economy, Society and Culture*, Volumes 1-3, Manuel Castells, Blackwell, 1996-1997-1998

[Ca0104] *CERT Advisory CA-2001-04 Unauthentic "Microsoft Corporation" Certificates*, cert-advisory@cert.org, 22 Mar 2001

[Chaum85] *Security without identification: Transaction systems to make big brother obsolete*, David Chaum, Communications of the ACM, Oct 1985, Vol. 28, Number 10

[Clarke87] *Information Technology and Dataveillance*, Roger Clarke, Communications of the ACM, May 1988, Vol.31, N. 5

[Clarke94] *Human Identification in Information Systems: Management Challenges and Public Policy Issues*, Roger Clarke, Information Technology & People 7,4, Dec. 94, http://www.anu.edu.au/people/Roger.Clarke/DV/HumanID.html, [referenced 6.9.2000]

[Clarke96] *Identification, Anonymity and Pseudonymity in Consumer Transactions: A vital systems design and public policy issue*, Roger Clarke, Invited presentation to the Conference on 'Smart Cards: The Issues', Sydney, 18. Oct. 1996, http://www.anu.edu.au/people/Roger.Clarke/DV/AnonPsPol.html, [referenced: 6.9.2000]

[Clarke97] *Privacy and Dataveillance, and Organizational Strategy*, Roger Clarke, Aug. 1997,

http://www.anu.edu.au/people/Roger.Clarke/DV/PStrat.html, [referenced: 6.9.2000]

[Clarke98] *Information Privacy On the Internet Cyberspace Invades Personal Space*, Roger Clarke, May 1998, http://www.anu.edu.au/people/Roger.Clarke/DV/Iprivacy.html, [referenced: 6.9.2000]

[Crowds] *Anonymous Web Transactions with Crowds*, Reiter, M., & Rubin, A., communications of the ACM, vol. 42, no. 2, Feb. 1999.

[DigiCer] *Digital Certificates, Applied Internet Security*, Jalal Fegghi, Jalil Fegghi, Peter Williams, Addison-Wesley 1999, ISBN 0-201-30980-7

[Dipl00] *Älykortti turvaa verkkopalvelut*, Diplomica 3/2000, Academic Communications Oy, Oct. 2000, ISSN: 1457-2001

[eToys] *Children's book publisher bids for eToys technology, customer list*, Computerworld, Lucas Mearian, 26 March 2001, http://www.security-informer.com/english/crd_customer_489222.html, [Referenced: 8.6.2001]

[Freedom] www.freedom.net [referenced: 7.8..2000]

[FutMon] *The Future of Money in the Information Age*, James A. Dorn (ed.), Cato Institute, 1997

[Gleit91] *Psychology*, 3rd ed., Gleitman, W. W. Norton Co., 1991

[Gold89] *Sensation and Perception*, Bruce E. Goldstein, Wadsworth, 1989

[HAC] *Handbook of Applied Cryptography*, Menezes, Oorschot, Vanstone, CRC Press, 1997

[HST] Henkilön sähköinen tunnistaminen eli sähköinen henkilökortti, http://www.sahkoinenhenkilokortti.fi/

[IDEA] *International Data Encryption Algorithm*, http://home.ecn.ab.ca/~jsavard/crypto/co0404.htm

[Ilm97] Lectures on Tfy-99-247, 'The Structure and operation of the Human Brain', Ilmoniemi, Risto, 1997

[Jeff98] *Personal Space in a Virtual Community*, Phillip Jeffrey, Proceedings of CHI 98

[Kahn67] *The Codebreakers: The Story of Secret Writing*, D. Kahn, New York: Macmillan Publishing Co., 1967

[Kingpin00] *Attacks on and Countermeasures for USB Hardware Token Devices*, Kingpin, Proceedings of NordSec2000, Reykjavik, October 12-13, 2000

[Kohn78] *Towards a practical public-key cryptosystem*, Loren Kohnfelder, Master's Thesis, MIT, 1978

[Laukka00] *Yksityisyys ja luottamus sähköisessä asioinnissa*, Markku Laukka, Pro Gradu, 2000

[Lpwa] *Consistent Yet Anonymous Web Access With LPWA*, Gabber, E., Gibbons, P., Kristol, D., Matias, Y., and Mayer, A., Communications of the ACM, vol. 42, no. 2, Feb 1999.

[Nordsec99] *SPKI based solution to anonymous payment and transaction authorization*, Juho Heikkilä and Markku Laukka, Proceedings of Nordsec 1999, Oct. 1999.

[Nordsec00] *Do I Know You? Recognition without Identification,* Juho Heikkilä, Proceedings of Nordsec 2000, Oct. 2000.

[MUD] *Concise Introduction to Multi User Dungeons*, Robert Snell, http://www.xania.demon.co.uk/mud.html

[Oja97] Lectures on T-61.231, 'Principles of Pattern Recognition', Erkki Oja, 1997

[Onion] *Onion Routing*, Goldschlag, D., Reed, M., and Syverson, P., Communications of the ACM, vol. 42, no. 2, Feb. 1999.

[Orwell48] *Nineteen Eighty-Four*, George Orwell, 1948

[P3P] *Platform for Privacy Preferences*, http://www.w3.org/P3P/

[PGPg1] *PGP User's Guide, Volume I: Essential Topics*, Philip Zimmermann, 11 October 1994

[PGPg2] *PGP User's Guide, Volume II: Special Topics*, Philip Zimmermann, 11 October 1994

[PGPt] *PGP Timeline*, Adam Back, http://www.cypherspace.org/~adam/timeline/ [referenced 10.4.2001]

[R1855] *Netiquette Guidelines*, S. Hambridge, IETF Request for Comments 1855, Oct. 1995

[R1459] *Internet Relay Chat Protocol*, J. Oikarinen, D. Reed, IETF Request of Comments 1459, May 1993

[RSA] *A method for obtaining digital signatures and public-key cryptosystems*, R.L. Rivest, A. Shamir, L. Adleman,

[Samuelson00] *Privacy as Intellectual Property*, Pamela Samuelson, 52 Stan. L. Rev. 1125 (2000)

[Schneier96] *Applied Cryptography* 2nd ed., Bruce Schneier, 1996, John Wiley & Sons, Inc.

[Schneier00] *Secrets and Lies, Digital security in a Networked World*, Bruce Schneier, John Wiley & Sons, 2000

[Sempsey] *The Psycho-Social Aspects of Multi User Dimensions in Cyberspace*, James Sempsey III, http://www.netaxs.com/~jamesiii/mud.htm

[Singh99] *The Code Book*, Simon Singh, Doubleday Books, 1999

[SPKI] *Simple Public Key Certificate*, C. Ellison, B. Franz, B. Lampson, R. Rivest, B. Thomas, T. Ylönen, Internet Draft, march 1998.

[SSL] Secure Socket Layer, http://home.netscape.com/eng/ssl3/

[Tav96] *Computer Matching and Personal Privacy: Can they be compatible?*, Herman T. Tavani, CQL '96, ACM 0-89791-827-4

[VeriS] *Verisign Certification Practice Statement version 1.2*, Verisign Inc. 1997

[Whitten99] *Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0*, Alma Whitten & J.D.Tygar, Usenix Security Symposium '99

Glossary

The terminology in this field is generally not well defined, especially when multiple aspects are to be considered. The language barrier is the first one on a collision course. For example, there are no different words for recognition and identification in Finnish. Suitable words can be found, but the implications are slightly – yet significantly – different. On top of that, different sciences define words differently. The concepts of trust and identity, for example, have very different definitions in technical, sociological, psychological, philosophical etc. fields.

Identity

The individual essence of an entity, usually a person. Especially defined not to be an identifier by which an individual is identified. See section on Traditional Identification for more discussion.

Identifier

A piece of information which is used to identify an entity or differentiate between entities. Identifiers do not need to be unique. Examples: A human name (John Smith), user code or number, or a public key with some link to an entity. Also called handle.

Unique Identifier

A unique identifier is something that is either absolutely unique or statistically unique. Absolutely unique is something that is checked to be unique, such as a social security number, while statistically unique is something that has extremely low probability of duplicates such as an RSA public key.

Linkage

Identifiers can either be linked or not. In real world the strength of the link usually varies, but the main point is to understand what is meant with the linkage. A linked identifier can be looked up in a database and connected to an identity. In practice this means that the link can lead to more identifying information or more information about the person in question.

An unlinked identifier is basically anonymous. However, for the case at hand the important definition is that there is no direct or relatively easily derivable link from the identifier to the true identity of its owner.

Further, the strength of the link can vary from strong to weak (in this case a totally unlinked identity becomes theoretical). A strong link would be a direct link like a social security number while a weak link would be something that is relatively easily traced. An example of such relative easiness in real world would be tracing a classmate's social security number.

As a real world side note, only unique identifiers can be strongly linked. In practice, however, non-unique identifiers such as names of celebrities can also become quite strongly linked.

Identification

Acquiring enough information to link a person or an identifier to the corresponding identity, i.e. enough to gather further information on the person.

Pseudonym

An identifier that is used to hide the real identity of the person behind the pseudonym. In the real world pseudonyms are used in classifieds, in Letters to the

Editor, etc. In this text pseudonym will also be a cryptographically secure identifier that is used for the recognition.

Recognition
The act of acknowledging that this is a familiar entity. In real world recognition is often based on faces, voices, etc.

Name
A name is usually what people call entities, whether persons or dead objects. The difference between a name and an identifier is that a name is used in speech to specify the target, while identifiers can be unusable for that purpose (e.g. an RSA key).

Handle
See identifier. Handle can also have a more human tone to it. In other words, human handles can be nick names. In this text handle refers more often to such handles than to technical handles, like file handles etc.

Nick name, nick
An identifier used to differentiate between users in an environment, especially a chat room. These are usually required to be unique in the specific space-time, i.e. no identical nicks in the same channel or room at the same time. The nick usually has a more human tone to it than a handle.

Channel
A virtual space where people can interact, for example an IRC channel. Usually refers to spaces where more than two people are interacting, or at least space that is capable of allowing interaction between more than two parties. At times called a room.

Room
See channel.

Meatspace
The opposite of cyberspace, i.e. the real world, where people are meat and bones rather than bits and bytes.