Brno University of Technology
Faculty of Information Technology

Department of Intelligent Systems

Contextual Information in Security and Privacy

Habilitation Thesis

Brno 2005 Daniel Cvrček

Contextual Information in Security and Privacy

by

Daniel Cvrček
Brno University of Technology, 2005

Submitted to *Brno University of Technology, Faculty of Information Technology* in partial fulfillment of the requirements for the title of Associate Professor at

Brno University of Technology
November, 2005

Signature of the Author .....

Faculty of Information Technology
Department of Intelligent Systems
November 18, 2005

Acknowledgements

I feel obliged to mention some of those who helped me to reach this point. First of all, I have to thank my wife Terezie as she is supporting me in what I am doing most of the time and forgiving me all the weekends and endless hours I am spending with a laptop and "weird" papers and books. Petr Hanáček and Vashek Matyáš who helped me to get where I am and made me realise that it is possible for me to write this thesis and apply for the title of associate professor. I also have to mention a number of people from the Computer Laboratory at Cambridge University where I spent a part of my life I really enjoyed as well as colleagues from Brno University of Tehchnology and Masaryk University.

. . . and of course my parents for not only bringing me up.

When I was thinking which book, film, theater piece, or song I remember most from the recent time one thing turned up instantly – BBC's series *Yes Minister* and *Yes Prime Minister*. In the end, I decided to use short quotations to amuse myself as well as those who are to read the text.
Any connection between quotations and the thesis is accidental.

Abstract

It is understood that if ministers want to know anything, it will be brought to their notice. If they go out looking for information, they might . . . -Find it? -Yes.

The emergence of the Internet has provided an unprecedented ability for people to browse and visit many different electronic places in an instant. However, this real-time connectivity is resulting in significant threats to individual privacy we can see in everyday life and whose theoretical potential has been demonstrated by researchers in recent five years.

The very same mechanisms underpinning the power of on-line services can also be used (sometimes without users' knowledge or consent) to collect sensitive information about an individual or his/her behaviour. Powerful data collection techniques, users' inability to find out what is being collected nor how to stop it, combined with press and TV exposures of revealed "bad actors" in privacy, have resulted in ever increasing lack of trust among on-line Internet users. Recent studies showed vigilance of Internet banking users and e.g. changes in their behaviour to decrease risks of online fraud.

The manner in which Internet on-line service providers behave and how they collect information about consumers stands at the root of privacy problems. The basic questions here are: how much information should be collected, to what extent should the information be used and for what purposes, and how, if at all, should the information be shared with other vendors and partners.

When we turn the table we will find out the other side of the game that is becoming important in mobile and ubiquitous computing. There is a strong research effort in the areas of large distributed systems, ubiquitous computing, and peer-to-peer networks with the main goal to make communication and computation as effective as possible. To reach this goal, we need substantial amount of information about system components as well as about users. Of course, it is a clear threat unquestionably deteriorating privacy of users beyond today's reality. Shortly, the surge in ubiquitous computing is bringing in new security challenges.

Technologies allowing communication in global environment become actual. Mobility is another paradigm for modern communication technologies – backed by growing numbers and computational power of mobile phones and other mobile computing platforms. Mobile phones are using communication networks of mobile operators to communicate with each other but new phone models are going to allow easy connection to Internet not only for data communication but for cheap voice connections in very near future. Mobile devices, not only mobile phones, communicating through the Internet have the potential to physically move over long distances and their access to the communication infrastructure is provided by mutually independent subjects (ISPs, mobile operators, nonprofit organizations).

What I have just described is a highly distributed environment with a very loose or missing hierarchical structure available for system administration. Security issues form an important part of administration and it implies necessity to solve security problems in distributed manner. This also fully applies to security mechanisms like authentication and authorization.

One of the available approaches is based on introduction of trust (or reputation). This approach does not require user enrollment – a process hardly feasible in the above mentioned distributed environments. All security decisions are then based on history of dealing with particular parties without knowing there real-world identity while using virtual identity systems instead. However, to be able to reason about someone's trustworthiness, we need to know quite a few information about the person's behaviour. In different wording, we need to know a substantial amount of contextual (personal) information.

This thesis, as well as papers it is based on, attempt to identify security properties needed for these news types of systems and find out the equilibrium between privacy and trustworthiness one will need to efficiently and securely access resources in future ubiquitous environments.

The final chapter and the first appendix deal with a special category of systems designed to provide anonymity for users. I try to introduce a possible and very interesting combination of anonymity and reputation systems where the reputation system is supposed to guard privacy of users.

Contents

Chapter 1

Introduction

-The minister's just left the office Sir Humphry, that's all.
-That's all? Do you mean he's loose in the building?

Identity, identification, profiling, big brother, anonymity, privacy, identity theft, online fraud – all these terms are used very often not only in research circles but in mass media as well. We are probably more aware about situation in Europe than in American, or Asian states as Europe is a special case in a way it treats privacy. Governments all over the Europe recognise the right for privacy as one of the basic fundamental freedoms and their policy reflects this fact. This is the main cause for a number of laws passed to protect privacy of citizens despite terrorism threats actual in the last four years.

United States are somewhat different as freedom of speech is just untouchable and as a result, anyone can publish personal information it possesses regardless who is subject of the information, even if this is a sensitive personal information. Well, it has been the case for until very recently and one can feel a light swing in perception of privacy caused by the alarming rise of identity theft in the United States. Asian states, on the other hand, recognise much stronger role of states as entities protecting citizens and as such they can use personal information (breach privacy) with much more freedom.

In general, common people only start realising that there is something like personal information and that it might be very uncomfortable to share this information with strange agencies. There has been several studies recently trying to estimate the price of personal information – surprisingly many people are willing to share their address or the way they create their passwords. What I am not so sure is the reason for this behaviour. One of possible causes may be that the interviewees had no time to think about potential consequences. Victims of identity fraud who already found out that it is very costly to recover own identity after it being exploited for any sort of crime. Some examples show that such a recovery effort may take *600 hours – to prove that their identity was stolen and clean credit reports* [38]. An interesting question here is what would be the results of

the mentioned studies if they interviewed these victims.

I am going to describe the data about users behaviour as contextual information as they describe behaviour of users of information systems, their habits, interests, social networks. In general, they specify contexts in which users interact with information systems. This thesis deals with two sides of contextual information. One of the sides related to processing of contextual information is privacy. As already mentioned, this is an aspect already widely recognised and its importance is further increasing. The other side is related to trustworthiness and reputation of a person. As it will be argued in more detail later on, we are particulary interested in the case when one cannot find out real identities of users in distributed systems as they are using digital pseudonyms. However, if it is possible to link actions of this party together (to the pseudonym), it is possible to use it to compute user's trustworthiness and/or reputation and use the resulting values for security and access control decisions.

1.1 Privacy

There are different definitions of privacy varying with context and environment. The term has been usually defined as the right of a person to be left alone, or to exercise control over one's personal information, or ability to protect individual dignity and autonomy. Basically, four basic types of privacy have been defined [61, 28].

Information privacy governing collection and handling of personal data. These data can be credit information (in countries using extensively credit history for people's trustworthiness evaluations), medical records, and generally any records containing information about one's way of life. It is also known as "data protection";

Bodily privacy concerns the physical protection against invasive procedures such as genetic tests, drug testing, biometric sampling;

Privacy of communications covers the security of data related to any kind of communication users exercise: mail, telephones, mobile phones, e-mail, and so on; and finally

Territorial privacy concerning the setting of limits on intrusion into the domestic and other environments of people like workplace or public space. This is the area covering for example warrants, legal house searches, ID checks, CCTV usage, surveillance.

From the computer scientist's point of view, the most important areas are those of information and communication privacy. Communication privacy is a part of a broader area of communication security and cryptography. This is covered by a massive number

of publications dating back to mid 70ies when systematic open research started, and much further back to history if assuming simple systems for protecting confidentiality of messages. We can say that we have got sufficient set of mechanisms allowing us to protect our communication and thus ensure communication privacy.

Very different story is with information privacy. The usual perception of the information privacy covers protection of data about persons that are stored in information systems or databases. The personal data in question contain names, addresses, social security numbers, national insurance numbers, identification numbers derived from birthday, political orientation, or medical information. This has changed in 1981 with the article of David Chaum – *Untraceable electronic mail, return addresses, and digital pseudonyms* [11]. The article shows how to use public key cryptography to hide communication partners for emails. This is one of the first examples when behaviour of users got under spot-light. This aspect is much more relevant today as people are using computers for different and various activities. From emails to voice over IP, from downloading music to grocery shopping.

This sort of information allows anyone to learn about a person of interest much more than a "static" list of personal information items. As we can find out from the Free Haven project website (www.freehaven.org), the literature covering this subject is very sparse until about 1998-2000. We can say that the year 2000 signs beginning of rigorous research into security of systems. The obvious target was anonymity of users in respect to their communication privacy – unlinkability to their communication partners. Location (or geographic) privacy is another important issue gaining importance with mobile devices and ubiquitous computing. Although there is a push to find geographical location even for Internet users

The information about users' behaviour is also used in reputation systems. This time, the information about behaviour is the basis for reputation computations. Reputation is subsequently exploited for security decisions. Reputation systems are connected with trust. Trust has been one of the main issues from the beginning of security research. It has been (and still is) crucial aspect of public-key infrastructures (based on PGP or X.509). Blaze, Feigenbaum, and Lacy [4] defined decentralized trust management as a distinct component of network security. This paper defines *PolicyMaker* system that is decentralized but still based on existence of public key systems delivering bedrock for the trust.

During the last ten years many reputation systems have been proposed but most of then did not attract much attention of security community. The problem is pretty straightforward. The trust-based or reputation-based mechanisms are not as reliable as conventional security mechanisms. They are based purely on former behaviour of users and users can exploit this to fool the system by honest behaviour until an adversary action

is worth to be launched. This is something that cannot be avoided and is an intrinsic property of all reputation systems.

As I am arguing further on, the research into reputation systems exists in two communities – security research and research in ubiquitous computing. The two groups also have slightly different objectives. Most of the papers published by security researchers tackle reputation as a possible enhancement of existing security mechanisms where existing results are not satisfiable while trust is the only way out of security problems in ubiquitous computing. I believe that while mobile computing is a relatively new area of research it will change the view on the importance of reputation systems.

1.2 Anonymity Systems

Research in the area of anonymity systems was started with [11] by David Chaum in 1981. The anonymity systems are supposed to ensure user anonymity when communicating with other users or when accessing various resources (primarily) in the Internet. Another goal of anonymity (privacy preserving) systems is to ensure uncensored and unpunishable publication of potentially controversial information. These systems must provide secure and distributed storage of documents so as nobody is able to alter the content nor shut down a server repositing the document and thus make the document inaccessible [17, 24, 29, 34, 50, 67, 68, 77].

I am primarily concerned with the former type of systems ensuring anonymity during communication. Obviously, these system cannot fulfill functionality goals without massive use of cryptography as security requirements are very high nowadays. Such system must be distributed, information are subsequently relayed by several randomly selected nodes and each node can see only its neighbours in the route and the rest of the nodes on the route must remain secret (anonymous). The systems should be resilient not only to outsider attacks but also to compromise of a certain number of their own nodes. When looking at the history of anonymity systems, there are three commonly distinguishable types of anonymity systems:

1. Cypherpunk remailers (Type I) – these remailers just simply strip off the sender's address. The message can be encrypted while being sent to the remailer and remailers (anonymizing nodes) can be chained. The remailers do not keep any logs about traffic.

2. Mixmaster remailers (Type II) – Mixmaster is a protocol (currently IETF draft in version 2 [57]) based on David Chaum's mix-net concept. One needs a special client to use this protocol for anonymous message sending.

3. Mixminion remailers (Type III) – this is the most complicated protocol for preserving privacy of its users. The protocol was designed by Danezis, Dingledine, and Mathewson

[17]. Unlike Mixmaster encrypting particular messages, Mixminion uses encryption on the link (TLS). There are also integrated directory servers that are synchronised and there is a simple policy for using dummy traffic.

Hand in hand with development of new technologies for remailer systems there has also been a progress in research of attacks on these systems. The original idea of anonymizing systems is to protect users from traffic analysis (as argued e.g. by Chaum in one of the first papers [12]). First paper breaking RSA implementation of MIXes appeared in 1990 [63]. Analyses of systems and successful attacks started to be regularly published from 2000 [2, 37, 66]. The complexity of the anonymity systems has been growing but new attacks followed. For a while, Tor (second generation onion routing system) seemed to be a very good implementation of remailer system, ensuring the required level of privacy. Mixmion (one of the newest) is a system whose design has taken into account all known attacks applicable on anonymity systems. However, it has been demonstrated that even a perfect system is not able to fully protect privacy of communicating users. There has been a series of works published by George Danezis, Andrei Serjantov, Paul Syverson, Claudia Diaz and others [18, 20, 59, 72] (to cite a few) showing that non-random (biased, predictable) behaviour of users is the key determining their privacy. Many previous works assumed that user behaviour is uniformly random and used this assumption for deriving properties of the anonymity systems. Unfortunately, habits and regularities in behaviour allowed to minimise anonymity set of users and effectively breach their privacy. The attacks then used properties of the systems (e.g. how much time does it take to tunnel a message) and user's behaviour.

Contextual information such as how many messages a week is one sending, at what daytimes, what is a probable number of recipients, and so on, gained a new dimension. They become the key to a successful attack on user's privacy.

1.3 Reputation Systems

The research into reputation systems is split into two rather disjoint communities. The first group consists of people primarily focused on security and cryptography, and anonymity systems as such comprise one of the interesting areas. The second group of scientists has been interested in distributed systems or ubiquitous computing and reputation has been just one of the ways out of security problems when commonplace security mechanisms cannot be used because of the distributed nature of the systems in question.

The two groups also have different objectives. Most of the papers published by security researchers tackle reputation as a possible enhancement of existing security mechanisms where existing results are not satisfiable. The situation is much more interesting in de-

centralised distributed environment. All currently deployed security mechanisms assume existence of a domain of trust encompassing a system we need to secure. This trust domain ensures authentication and authorisation mechanisms. We are able to manage security and deploy access control only within trust domain. The distributed environment lacking existence of such a domain of trust forbids use of security mechanisms requiring authentication and authorisation.

A similar situation has arisen with X.509 standard, when people started realising that one world-wide certification authority would never work. The original concept that was rather simple had to change. Public key certificates (the seeds of trust) had to contain much more information. The complexity of certification authorities grew up, as well as requirements on certificate owners and relying parties that have to be able to trustworthily verify validity of certificates. All the complications just to allow for mutual recognition of certificates issued by different certification authorities – and we still need external interference to establish trust between independent certification authorities (cross-certification).

There was another way of "doing business". X.509 technology is really complicated and there were those crypto export restrictions. This misery has become a virtue. The idea was very simple – what is a more secure way of exchanging public keys than physically meeting one's friend and exchange floppies, or business cards or any other "material form" of public keys. This approach introduced the concept of *web of trust* and PGP has arisen. The web of trust is an extension of mutual trust between two parties onto parties that are trusted by any of these two parties and subsequently by all parties "added into the web".

If Peter and Vashek are to exchange their public keys, Vashek may trust Peter in such a way that he accepts all public keys Peter has previously obtained from his friends. The fundamental question here is whether we can use the same paradigm in digital world – without physically meeting parties we are supposed and willing to trust.

1.3.1 Trust

It is just natural that from two terms – trust and reputation – the trust was elaborated first. Paper [4] of Matt Blaze, Joan Feigenbaum, and Jack Lacy is often cited as the one defining trust management as a distinct component of network security. They have defined a language, PolicyMaker, allowing expression and reasoning about trust relationships. The power of the language was demonstrated on public key validity verification.

Independently, trust was studied by sociologists and psychologists. An example can be [56], an extensive study trying to identify all possible meanings of trust and citing a huge number of sources. Conceptually, trust may be classified into six categories: disposition, structural, affect/attitude, belief/expectancy, intention, and behaviour. Orthogonally to this classification, trustee can be classified in another six categories: competence, benevo-

lence, integrity, predictability, openness/carefulness /..., and other trustees. As you can see such a complex definition is unrealistic to implement in information systems. The general conclusion, for our purposes, is that human trust is too complex notion and we shall follow the quotation by Robert Kaplan ([56]).

> ...researchers ...purposes may be better served ...if they focus on specific components of trust rather than the generalized case. – Robert Kaplan.

Grandison and Sloman define trust in [33] as: "...[trust] is a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action ...in a context in which it affects his own action.". Thus, trust can be again seen as a prediction of future behaviour. Reputation is by [54] defined as one of the factors influencing trust. Reputation is also context dependent as shown e.g in [9] or [58].

SECURE project [70] has introduced several mathematical definitions of trust. All are based on a function from a pair of principals (trusting principal and trustee) into a set of trust values. It is also important to mention that the trust here varies with context. Once trust is formed, it becomes subject of evolution and eventually propagation (here we are getting towards reputation). I am going to talk about it more later but the mathematical definitions constituted a base for interesting reasoning about trust. It is also true, from my viewpoint, that it would be very difficult to use these trust models in most real-world applications with autonomous security decision making.

The important fact is that trust as introduced by SECURE project deliverables and papers is based on evidence. It is essential leap from usual understanding of trust as a result of verifying a token issued by trusted third party (e.g. certificate issued by certification authority or ticket issued by Kerberos server). We begin to derive trust from partial and imprecise information obtained from possible untrustworthy party/principal.

1.3.2 Reputation

Difference between trust and reputation can be defined simply as follows:

Trust is an opinion of one party on many.

Reputation is an opinion of many on just one party.

One of the reputation definitions is "...a perception a party creates through past actions about its intentions and norms" [51]. This definition does not require existence of more than one party collecting evidence and participating in the reputation evaluation.

It means that we need a mechanism to transfer trust between principals so as one of them can compute reputation of the principle she/he is interested in. It is remarkable though that there is a long list of articles dealing with reputation systems but far less papers with "trust" as one of their keywords.

The notion of reputation system is generally used for systems using recommendations of users. These systems are centralised - there is one virtual server offering a set of services. Users then recommend services, objects, other users (as trustworthy for certain transactions). Examples of such systems are Google search engine or eBay on-line auction sites. Both systems work fine under stable and predicted conditions. This can change if an attack of large scale is launched and trustworthiness of villains is artificially improved. (There is a case of a trio of fraudsters who were sentenced in UK for a worldwide fraud via eBay worth at least 300,000 pounds. They used a dozen pseudonyms in eBay to boost their trustworthiness.)

There are two basic types of dishonest users in reputation systems. Passive, e.g. selfish users want to use the system without contributing to its quality. This is especially important when the system is dependant on active users by its nature. The most commonly known type here are free-riders, problem of P2P systems. The second type is an active malicious attacker. These attackers target the system itself, its part(s), or users with the goal to cause denial of service or to gain unfair advantage. [71] names five basic behaviours of malicious attackers.

Traitor Here is meant a principal behaving for a certain period and then instantly changing her/his behaviour. The result of this activity is that the principal gains sufficient reputation/trustworthiness to get wide access to the system and this trustworthiness is later-on exploited for obtaining unauthorised access to system's resources.

Collusion This is a situation when considerable number of principals cooperates in an attack. Douceur [26] shows that it is feasible for the attacker, under certain conditions, to create unlimited (or at least considerable) number of identities (virtual principals) able to take control over the whole system. The attacking principles are also called "cliques".

Front peer There also may exist principals gaining reputation through correct behaviour that is further boosted by help of other front peers in a clique. This reputation is then used to promote malicious peers and thus speed-up evolution of their trustworthiness (as you can see I am using terms reputation and trustworthiness as equivalent).

Whitewasher It does not have to be necessarily attack on the system. Principles, whitewhashers, are changing their identity when their reputation declines below certain threshold or to get rid of certain evidence.

Denial of service Users are using their reputation and force system to bring to bear extreme amount of resources to cover their request. This way, the system's service is denied to other authorised users.

A similar but shorter list has been defined in [54]:

Inactivity refers to free-riders activity – using resources but not offering anything in exchange;

Defame is an activity when attacker is giving recommendations lowering reputation of victim on purpose;

Collusion is a situation when multiple attackers are propagating good reputation to promote each other.

Utilisable reputation system should be able to detect and react to all of the above mentioned types of attacks. Naturally, it is not possible to cover all attacks for arbitrary number of attackers. However, we should still be able to define or estimate robustness of our system against all types of attacks. I can therefore define the following incomplete list of desired properties a good reputation system should have.

- Valid – system fulfills its basic goal, assigns reputations that reflect principles' behaviour and is able to discern honest from dishonest ones.
- Distributed – there is no trusted third party establishing domain of trust. Nor should there be any centralised storage for any data.
- Robust – system is able to detect and defend itself against attacks as listed above.
- Timely – changes in reputations should reflect recent behaviour of entities.
- Resource-saving – Computations of reputations and any other computations should take into account limited resources of nodes in the network (typically mobile devices).
- Flexible – if there are sharp changes in an entity behaviour, system is able to react quickly as well and spread information about this change throughout the system if possible.
- Scalable – it is easy to remove and add new entities into the system.
- Coherent – reputations of particular entities are coherent in the whole system.

These are however rather high-level requirements. It is also possible to identify basic threats on the level where it is possible to identify possible mechanisms preventing them.

The following lines contain potential threats related to exchange of recommendations (one of the possible instances of reputation instantiation).

- Lack of Privacy of Feedback Provider or Target – one of the issues I will be discussing later in the thesis. It is very hard to ensure privacy and accountability in distributed systems.
- Tampering of Feedback – to prevent this is relatively straightforward if we are able to establish a common key between the parties exchanging the feedback. However it is not so easy – see the next bullet.
- Masquerading of Identity – another problem of distributed systems. We can never be sure that who we are talking to is not a proxy masquerading as someone else. Either we are able to proof our identity – in this case we do need direct contact with our counterpart, or we need a trusted third party, or we have to take into account possibility of a proxy existence all the time.
- Intercept of Feedback – this is a problem very similar to tampering. The same solutions, the same obstacles.
- Repudiation of Feedback – the problem here can be viewed from two different angles. The first viewpoint is of the connection between digital identity being used in the reputation system and physical identity of the agent. Fortunately, we do not necessarily need to provide this but it is, indeed, advantageous to know that agents do not switch digital identities (pseudonyms). One of the solution here may lie in the economics. If the system is able to give obvious advantage to those agents with digital identity provably existing for a long time users will not be tempted to change their pseudonyms and we obtain a solid base for repudiation/non-repudiation of feedback's provider. The second point of view is repudiation of the data (assurance the data do not change during transmission). It is the question of authenticating data messages that can be easily done if we can make use a cryptography.
- Shilling Attack – is a particular attack based on deception of the feedback provider. It is a behaviour that should be penalised by decreasing trustworthiness of the provider. This, however, presumes existence of mechanisms able to detect fraudulent behaviour.

Chapter 2

Reputation Systems

-I'll bet the first thing he says is. "Any reports on my Washington speech?"
-How much? -A pound. -Done. He wont because he's already asked ...
In the car on the way back from Heathrow.

Trust and reputation are still relatively new notions and it is impossible to formulate definitions that would withstand new research results. As it was already mentioned, trust is rather complicated term and this chapter is discussing one facet of trust – computational trust. After all, although it is useful to have rich definitions for reasoning, we need something to grasp and describe in a way allowing implementation.

Dieter Gollmann said during ESORICS 2005 conference that one of the main problem of security is trust [32]. The reasoning is simple – once you trust someone, you tend to authorise her to access your resources. Any such authorisation can be abused and it is very hard to prevent the abuse – after all, you trusted her and authorised her.

When we start talking about trust we have to take risk into account as well. When we reason about reputation systems for a while, we will find out that what we want to solve is a problem that risk analysis, threat analysis, threat assessment have been solving for long time. The crucial difference is that we have to do it in real-time and automatically. However, there was a relatively long journey before we have proposed an architecture that is very similar to the processes commonly used for risk analysis.

Remark: Before I move forward, one note must be made. I am going to use adjectives trust-based and reputation when referencing systems I am discussing. The community, I was primarily talking with, is using the term trust-based systems. When I started looking for and studying available sources this was the buzzword I was using. It took me a little while to realise that what I am looking for is also called reputation systems. The latter term is common in the community doing research in the area of computer security while trust-based comes and is much more used by pervasive and ubicomp community. As I am trying to connect ideas from both of these communities I will be using both terms interchangeably.

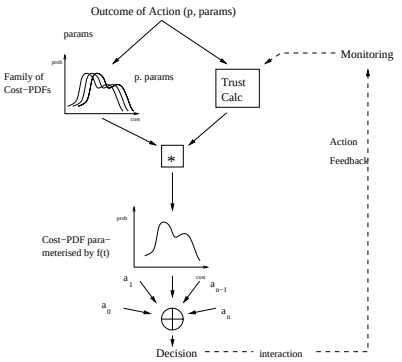

Figure 2.1: Schema of the framework data flows

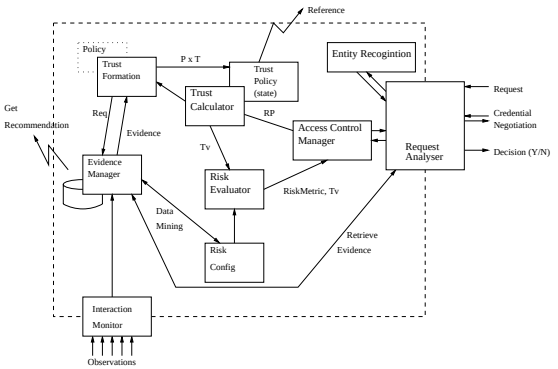

Figure 2.2: An overview of the SECURE framework

One of the goals of the SECURE project (the project I took part in during my stay at Cambridge University) was to devise a general architecture for trust-based systems. I believed that the first of the steps would be to draw a diagram with data flows and basic processing blocks. It turned up that it is not as simple as I (and not only me) thought in the beginning. Input and output data were more less stable: trust, risk, and evidence were on the inputs and access control decision was the output (fig. 2.1). However, the way risk and trust should be combined has been frequently changed – one of the versions is on fig. 2.2 [3].

You can see a pretty complicated internals of the model framework on the right hand figure. The bed rock of everything in the framework is evidence about behaviour of communication partners and evidence manager. Data mining processes influence configuration of Risk Evaluator. This block outputs RiskMetric and trust value (Tv) for Access Control Manager. Feedback from granted interactions is used to accommodate trust value of the particular principal. The most frequently changed part of the diagram is relation between Risk Evaluator and Trust Calculator. Which block should be primary and whose output is more important. A possible solution may lie in introducing a new notion into the framework – threat and promise (positive counterpart to threat).

There is a number of standards and procedures for evaluation of information systems on various abstraction levels ITSEC [73], TCSEC [10], Common Criteria [79], ISO17799 [41] (although it has got several versions), ISO13335 [40] (security management), ISO 18044 [39] (security incidents response), and so on. Although different definitions and notions are used the process consists of the following steps:

- Assets/resources enumeration – firstly, we need to know what resources are in the system and what is their importance or value. The resources include data, as well as applications, hardware components, network elements, and other valuable parts of the system.

- Identification of threats – when the system is mapped we can create a list of threats that are potentially applicable on its subparts.
- Risk assessment – each threat may be realised with certain probability. This probability is determined by several factors: security mechanisms deployed to prevent the threat, value of the resources open up by realising the threat, gain of the attacker, risks of the attacker in the case of her nicking.
- Damage estimate – using risks, threats, and value of assets affected, we can compute statistical damage (loss) over a period of time – e.g. a year.
- Improving the system – new security mechanisms are proposed, their implementation costs are compared with the influence on the overall losses, and the mechanisms are eventually implemented.

The idea I elaborated with Ken Moody is to introduce threats into the framework. The main problem is to adapt processes using threats (as introduced above) for digital environment and real-time processing.

2.1 Paper enclosed as C-1: Combining Trust and Risk to Reduce the Cost of Attacks

This paper was presented during iTrust conference held in Paris, France, in May 2005. The authors are Daniel Cvrcek and Ken Moody.

Abstract: There have been a number of proposals for trust and reputation-based systems. Some have been implemented, some have been analysed only by simulation. In this paper we first present a general architecture for a trust-based system, placing special emphasis on the management of context information. We investigate the effectiveness of our architecture by simulating distributed attacks on a network that uses trust/reputation as a basis for access control decisions.

Chapter 3

Dynamics of Trust

-No, Humphry, you haven't quite got my drift. I mean NOW.
-Oh ... you mean, NOW? -Got it in one, Humphry.
-Oh Minister, it takes time to do things NOW.

Dynamics of trust and reputation is an area that is very often omitted in papers targeting models of trust and reputation. We can name a few as examples: [46, 48, 52, 53, 54, 75]. For a long time, Dempster-Shafer belief theory was seen as one of the best methods for computing trust. As demonstrated in the enclosed paper, the behaviour of this theory is not suitable for digital environment when large number of evidence pieces is expected (see e.g. [36]) – for computational trust. Josang was using a similar approach [46, 47, 48] whose main purpose is to treat conflicting opinions (reputation recommendations).

3.1 Models for Computing Trust/Reputation

I am to introduce several of the models proposed in the above mentioned articles to give an overview of existing models and to create ground backing my work. Let us start with Liu and Issarny [54]. The reputation as they compute it is influenced by time elapsed from the moment a given behavioural evidence was obtained.

$$Rep_a(o)^t = Rep_a(o)^{t'} * \rho_e^{t-t'} + New\_behaviour * (1 - \rho_e^{t-t'})$$

The equation computes reputation of an agent o in the view of agent a in time t. ρ is a predefined constant such that $\rho \in [0..1]$. As written, the equation is recursive and its computation repeats until all evidence items "New_behaviour" are treated. Selection of ρ strongly influences the way evidence is fading as time passes.

The authors are using very similar approach (similar equations) for including contexts and thus adding a new dimension to reputations:

$$SRep_a(o,C)^t = \frac{\sum_{C' \in Tree_a} SRep_a(o,C')^t * \rho_c^{|C'-C|}}{\sum_{C' \in Tree_a} \rho_c^{|C'-C|}}$$

Cs stand for types and there is introduced a special construction for them. Inspired by contexts as used by e.g. mandatory access control models, a tree of contexts is created and the value of term $|C' - C|$ of two nodes in the tree is defined as the minimum number of intermediary nodes on the paths connecting C and C'. This way, all evidence is relevant to any contextual reputation computed for agent o. The level of the influence of a particular evidence piece is given by the distance of the two contexts (the context of the evidence and the context of the computed reputation) in the graph of contexts. The flaw here is that the results are fundamentally connected to the graphs that may be drawn very differently each time. We can mention as an example, organisational structure of British public administration that is changing "almost" continuously, although the duties are "almost" constant.

The constants $New\_behaviour$ are derived from the satisfaction of an agent – her satisfaction based on her expectations. This satisfaction is computed as a vector of contextual values. Interestingly, this can be computed in one of three possible ways: (i) as a ratio of results to expectations, (ii) a ratio of expectations to results, or (iii) by multiplying expectations and results.

The whole system is even more complicated as there are experience and recommendation managers whose opinions are combined by reputation manager to get final reputation, but the idea has been in principle described.

Very interesting are plots from the paper [54] – fig. 3.1 and 3.2. It is very clear that the dynamics of reputation – the ability to change reputation of an agent switching from honest to dishonest behaviour (and vice versa) is very low.

The second example comes from [64] by Qu et al. featuring a method based on Fuzzy Sets to compute reputation for grid entities. The reputation is assessed from a series of previous reputations R_i's in times $t_i's$: $\{R_0/t_0, R_1/t_1, \ldots, R_n/t_n\}_{\langle e_i, e_j \rangle}$. e_i is the entity assessing reputation of e_j. The same effect of time (evidence is loosing value with its age) is described here as decaying described with a generic function D. Before the sole reputation is assessed, three precursors must be computed.

Behaviour Coherence Factor – CF characterises the way a given entity cooperates with other entities and whether this behaviour is coherent. The factor has got two facets: time coherence (TCF) and entity coherence (ECF). (The following equation is for Hamming approximation.)

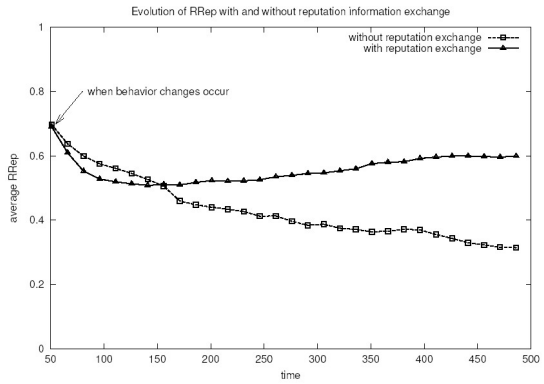

Figure 3.1: Changes of RRep with and w/o reputation exchange

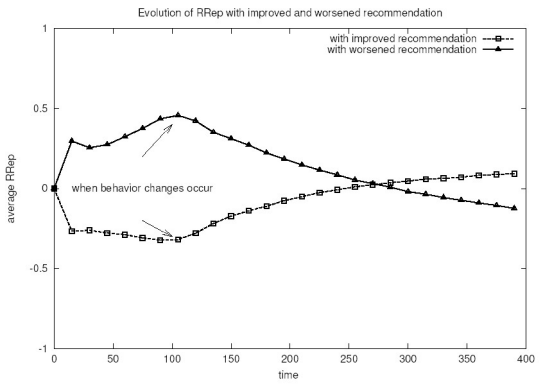

Figure 3.2: Changes of RRep for different trustworthiness of RRep

$$TCF = N(E, E_0) \triangleq 1 - \frac{1}{n+1} \sum_{i=0}^{n} |R_i - R_n|$$

Behaviour Inertia – reflects a trend in the entity's behaviour. There are again two factors here: positive (PID) and negative (NID). Both are in the interval [0..1]. PID shows whether the behaviour is getting more satisfying with time, the higher the value the stronger the trend is. NID reflects exactly the opposite trend. The following simple formulae use counts of previous reputations (n), number of instances where successive reputation has higher value than the previous one (m), and a count of instances with successive reputation is having value lower than the previous one (l).

$$PID = \frac{m}{n} \qquad NID = \frac{l}{n}$$

Behaviour Deviation – is based on the fluctuations (variations) in the behaviour and the value is again non-negative, lower or equal 1. What follows is therefore an equation for deviation.

$$BD = \frac{|\sum_{i=0}^{n-1} R_n - R_i|}{n}$$

Finally, the reputation is computed from $BE = (TCF, PID, NID, BD)$, an Eigenvector, and R_{wavg} denoting time-decaying weighted average of previous reputations as follows:

$$R_{\langle e_i, e_j \rangle} = \begin{cases} TCF * R_{wavg} + min(BD, 1 - R_{wavg}), & PID > NID; \\ TCF * R_{wavg}, & PID = NID; \\ max(TCF * R_{wavg} - BD, 0), & PID < NID. \end{cases}$$

Although the paper does not contain any experimental data nor plots, it is relatively easy to find out that dynamic properties of $R_{\langle e_i,e_j\rangle}$ won't be excessive as the conditions for switching conditions are based on counts of good and bad behaviours.

Another example is the EigenTrust algorithm of Kamwar's, Schlosser's, and Garcia-Molina's paper [49] targeting management in P2P networks. This is a very highly cited paper, describing a way for isolating malicious nodes from P2P networks.The authors stated several design criteria.

1. System is self-policing, there is no central authority.
2. System maintains anonymity and peers are identified through a random, opaque identifier.
3. There is no advantage assigned to newcomers, i.e. the worst possible reputation is equal to the one being assigned to new users.
4. Computations are simple and computationally effective.
5. It is still possible to defend against whole groups of malicious users.

The idea in one sentence would sound like this: Each peer is assigned one trust value derived from trust values given by all other peers weighted by these peers' global reputation.

The cornerstone of the approach is normalisation of reputation/trust values. Normalised local trust value of peer j by peer i would be computed from a number of "satisfaction aggregates" s_{ij} of peer i towards peer j:

$$c_{ij} = \frac{max(s_{ij}, 0)}{\sum_j max(s_{ij}, 0)}$$

The highlight of this normalisation is that values are kept in the interval [0..1], the low point is that values are relative. It means that if $c_{ij} = c_(ik)$ we know that j and k behave the same way but we do not know how. It is also worth noticing that the normalised values are non-negative and changing identity thus does not bring anything to attackers. The model allows to aggregate local trust values (by asking friends for their opinions on peer k):

$$t_{ik} = \sum_j c_{ij}\ c_{jk}$$

This sort of asking may continue – friends' friends, and so on. The result is denoted as $\overrightarrow{t} = (C^T)^n \overrightarrow{c_i}$, where C^T is transitive closure of the matrix $C = [c_{ij}]$ values and as the resulting value converge to the left principal eigenvector, a vector $\overrightarrow{e}$ of size m is used, where $e_i = 1/m$. As demonstrated, the algorithm converges after very low number of iterations. The paper also proposes several practical security mechanisms: there is several pre-trusted peers (the ones creating the network), reputation of peer i is not stored by a different peer, and to be able to defend against cliques, it is stored by more peers (score managers). Distributed hash function (DHT) such as CAN or Chord [65, 74] are used to find all score managers.

The trust model is able to decrease number of downloads of corrupt files to ten percents in the networks with up to 70 % of malicious nodes. The problem of dynamics – ability to change trustworthiness of particular nodes is however similar to the previous ones as counts of positive and negative experiences form basis of the model.

The approach I came up is to combine two trust values: long term trust and actual trust. The long term trust is computed in a way similar to Josang, and in fact, it is a weighted arithmetic mean of all the observations related to particular node. The second part is actual trustworthiness of the node that has very short memory but is able to dramatically change actual trustworthiness of a given node/peer. This short-term trust is defined with a help of Dirac impulse that is later used for normalising responses of the system.

3.2 Paper enclosed as C-2: Dynamics of Reputation

Paper is authored by me and it was presented during Nordsec'04 conference.

Abstract: To enforce security without user enrollment, trust (or reputation) systems were proposed to use experience as crucial information to support cooperation as well as for security enforcement mechanisms. However, use of trust introduces very difficult problems that still discourage from exploitation of trust for security mechanisms. The ability to change trust quickly and react effectively to changes in environment and user behaviour is profound for usability of mechanisms built on top of trust. Dempster-Shafer theory was proposed as a suitable theoretical model for trust computation. Here, we define general requirements for reputation dynamics and demonstrate that Dempster-Shafer theory properties are not as good as is widely thought. On the contrary, simple formulae work.

Chapter 4

Evidence

-What's the difference?
-Well, 'under consideration' means we've lost the file,
'under active consideration' means we're trying to find it.

The evidence, reputation systems are able to process, is key to correct and intended security decisions. Introduction of this chapter consists of two parts: a short overview of definitions from existing sources and a definition of evidence I have stated for use in our own system framework.

4.1 Literature survey

A nice overview of evidence types is given in [60]. Evidence here is defined as a "non-repudiable token that may be arbitrarily transferred". This is an assumption that is very hard to fulfill in fully distributed systems without trusted third party. (However, we can still assume that every principal or agent is able to issue and verify validity of evidence while keeping the issue of non-repudiation aside.) There is also another definition targeting recommendations that do not have to be necessarily verifiable. The authors also assume that recommendation may contain an evidence or a set of evidence pieces.

In the context of simple evidence, one of the most important problems is atomicity. We would love to have atomic evidence as they offer the highest possible freedom for subsequent processing. Unfortunately, atomic evidence implies enormous load of data that is hard to analyse – this sort of problems is subject to research in the area of intrusion-detection systems. The authors of [60] present four "high-order" types of evidence: receipt, affidavit, bond, contract. Receipt is a confirmation of a transaction that took place and ended up with a result (positive or negative). Receipt can be issued by any of the parties taking part in a transaction and it describes the transaction. Bond is a promise to provide a service in the future – it is not exactly evidence as we may understand the notion. The sole existence of a bond, however, signals that a certain type of action was carried out.

Affidavit is a general recommendation about long-term behaviour of an agent. Finally, contract is an evidence bounding both parties of a transaction to provide a service or action in a future.

This high-level evidence is easy to process as the number of evidence items will be low but it also implies that a lot of data about information system will be lost. You can see the evidence types and their semantic is rather rich implying complicated algorithms for effective usage of the evidence. It is also worth noticing, that the evidence types reflect more homan world than digital world with hundreds or thousands of decisions per second. A slightly more simplistic definitions and more oriented towards automatic systems come from [78].

The evidence here is divided into two groups – direct (directly witnessed by an agent) and indirect (a third party information about transactions or behaviour). The distinction is very important as indirect evidence requires much more complex processing where trustworthiness (or reputation) of the third party must be reflected and as we can also presume that indirect evidence may be an aggregate of a number of direct evidence pieces. There are three basic types of evidence here:

- Observation – direct evidence, gathered while interacting with a peer.
- Recommendation – is an indirect evidence about behaviour of a subject peer S, passed by witness peer W to receiver R. If we wanted to keep the model simple we would assume that recommendation is just an observation of a third party, however, the semantics may by much complicated.
- Reputation – is na indirect evidence measuring overall trustworthiness of a subject S by peer population $\mathcal{P} = \{P_i\}$: $r(S) = \sum_{\mathcal{P}} m(P_i)(s)$.

This evidence classification is used in [27] for a design of a processing engine.

A similar distinction between direct and indirect evidence can be found in other sources as well. [84], for example, divides evidence A_i can use on (i) data supplied by A_j and (ii) data supplied by other agents, when A_i evaluates trustworthiness of A_j.

4.2 Formal Definition of Direct Evidence

The following lines are focusing only on direct evidence and as we will see, it is still a non-trivial task to do get it right. I set out a few basic definitions to make the notion of evidence more precise and at the same time, I am trying to set some invariants for any evidence that can exist. One of the most evident, for the beginning, is monotonically decreasing weight of evidence with time (see chapter 3 for examples of trust models). Having spent some time

on the issue of invariants, I found out that even this statement does not have to be true everywhere. As the result I was not able to find a single invariant for evidence processing but defined the notion of evidence for use in computations instead. I very briefly define the following basic notions: evidence data, contexts, and weight functions. I reference a model instance as $\mathcal{M}$. Instance $\mathcal{M}$ expresses a context of one physical place and one application (or policy) and forms a context in which the evidence is being processed.

Definition 4.1. (Evidence) Evidence is an encoding of interaction outcomes related to $\mathcal{M}$. Any *evidence* is an element of set containing all types of evidence available in $\mathcal{M}$: $\mathcal{E} = \{E_1, E_2, \ldots, E_n\}$, where indexes run through $I_E = \{1, \ldots, n\}$ representing all sources of data available in $\mathcal{M}$. It also holds that $E_j = \langle 0, 1\rangle \cup \varepsilon_j$, ε_j is undefined value.

There are three specific values in E_i: value 1 expresses absolute success of interaction while 0 is the opposite. Value ε stands for undefined value. The set of indexes of evidence I_E can be stated in advance but it is not a necessary condition and it is only up to the implementation whether it allows to add new sources or remove existing sources of evidence in the runtime. The set may express set of evidence sources (firewall, anti-virus, network applications like ssh, telnet, and so on). The value ε_j is a technical value used when a value is needed but nothing had been gathered. Undefined values are also indexed as (in)equality of undefined values between evidence types can be treated in several different ways.

Definition 4.2. (Context) Context is any information specifying evidence in the model $\mathcal{M}$. There is a tuple $\mathcal{C} = \langle C_1 \times C_2 \times \ldots \times C_m\rangle$ defined, where $I_C = \{1, .., m\}$ is a set of indexes for all possible contexts of a given instance $\mathcal{M}$.
Domains of contexts are specific and we only require that $\forall\ i \in I_C : \epsilon \in C_i$, where ϵ represents undefined value for a given context. We shall call $\mathcal{C}$ as *full context*.

Context domains are not known in advance. It means that we do not have to define them when configuring the system but it also implies that encoding of evidence should be recomputed each time the domain has changed.

Each piece of data stored in $\mathcal{M}$ is associated a context that is a subset of full context $\mathcal{C}$. The context characterizes the data. We can bring in *principal, time, computer port, service that it is associated to, bank operation* as examples of such contexts.

I wanted to define evidence data in as a general way as possible. That is why fully qualified evidence is defined.

Definition 4.3. (Fully qualified evidence) (FQE) is a record of the following form: $\theta \in E_i \times \mathcal{C}$, or $\theta = e_i \times \langle c_1 \times \ldots \times c_m\rangle$.

FQE is the elementary form of evidence specification. The set of contexts may change

in time as new sources of evidence are added to the system (external change), or new patterns in existing contexts are identified as important for trust computations.

The main reason why contexts are used is that we need to express varying impact on values of E_i element of FQE. There are three ways how to use particular context for particular computation. We can just compute an aggregate value from e^i_j values (with fixed j - i.e. one type of evidence) over the context. We can fix context by some value or interval, and we can also define a weight function for the context.

Definition 4.4. (Context weight functions) These functions may be explicitly defined for some contexts. A set of weight functions is defined as $\Phi = \{\phi_k : k \in I\}$, where $\phi_k : C_k \rightarrow \langle 0, 1\rangle$.
Contexts with weight function ϕ_k are called bounded – $\mathcal{C}^B = \{C_k : \phi_k \text{ exists}\}$.
Contexts from set $\mathcal{C}^L = \mathcal{C} - \mathcal{C}^B$ are referred to as loose contexts.

The weight functions might be as well computed dynamically as an output of risk analysis.

There is nothing else on what we can base risk, trust, or any other real-time computations than evidence. To initiate such a computation a request must be handed over. Request defines what we want to get – it is selective and it defines subset of interesting evidence by parameters – full context. All contexts that have defined value $c_i \in C_i \wedge c_i \neq \epsilon$ are *fixed* for the following computation. Functions ϕ_k are applied on contexts that are not fixed for a given request.

There are some really important contexts. We can start with principals. When there is a possibility for external entities to request access to a system resources, there must be a context that allows to identify them. Let us call the context *principal*. When a system grants access to system resources, it is usually through its services that mediate such an access. Context identifying those services is called *action*. Those are the basic contexts. It is possible to name some other contexts with specific names with precisely defined semantics. One of examples could be age of evidence.

Now, when a request is issued with principal as a fixed context, the result of computations is related to a particular principal. The precise meaning of computation corresponds with a subset of hypotheses that we are interested in (e.g. trust, risk, time of response, ...). The combination with trust related hypotheses outputs *trustworthiness* of a given principal.

Another important request is the one with action as a fixed context and risk related hypotheses used for aggregation of evidence. The value obtained from such a computation expresses *risk* associated with a given action. We can also fix e.g. time with some intervals and compare results. We obtain an evolution of the risk. When we try and do those computations for all actions we get two-dimensional map of risk that can be used for

decision whether to adjust security properties in a model-wide or only action-wide scope.

4.3 Article enclosed as C-4: Evidence processing and privacy issues in evidence-based reputation systems

This article written by Vashek Matyas, Ahmed Patel, and me was published in Security Standards & Interfaces journal in the beginning of 2005 and it summarises our view on privacy in reputation systems.

Abstract: Issues related to processing of evidence in evidence-based reputation systems, with a particular concern for user privacy, are discussed in our paper. The novel idea of evidence-based reputation (or trust) systems is that such systems do not rely on an objective knowledge of user identity. One has instead to consider possible privacy infringements based on the use of data (evidence) about previous behaviour of entities in the systems. We provide a brief introduction to evidence-based trust/reputation systems, as well as to the privacy issues, addressing the common problem of many papers that narrow the considerations of privacy to anonymity only. We elaborate on the concept of pseudonymity through aspects of evidence storing and processing. This, together with a consideration of current work on trust models, leads to our specification of requirements for the trust model for evidence-based systems supporting pseudonymity.

4.4 Paper enclosed as C-3: Using Evidence for Trust Computation

Presented during Santa's Crypto Get-together 2003 as the first output of the author's work on SECURE project.

Abstract: Trust can have its life-cycle and we can model it and utilize it for establishing secure environment for mobile environments. We assume that entities in the collaborating environment are mobile. It is not possible to perform entity enrollment. There is no globally trusted third party. Usual authentication mechanisms can not be used. We propose use of trust based on principal behaviour observations. The overall model has been devised with the SECURE project. The article makes a brief overview of the model and proposes specific approach for computation of trust values from observations. The method introduced in the article is based on Dempster-Shafer theory of confirmation that is enriched to fit needs of the SECURE project.

Chapter 5

Privacy Model

-You simply cannot go around speaking to people in the department. -Why not? -Minister, how can I advise you properly if I don't know who's saying what to whom?

This is the most difficult part for me to write. The reason is not as much the difficulty of the topic but scarcity of work done in the area. What I am interested in are not social models of privacy but formal privacy models allowing evaluation of privacy. The privacy for digital environment is best described in Common Criteria [79] where it is divided into four subgroups. Although the definitions are meant for a centralised system containing a trusted part ensuring certain security functionalities the principles can be used more generally.

- Unobservability – is provided when it is not possible to detect any interaction of users with system.
- Anonymity – ensures that while it is possible to see activities in the system, it is impossible to identify users. It also implies that one cannot discern users from each other.
- Pseudonymity – with this privacy property, one is able to discern users and identify them but only in a way allowing accountability. The property still forbids "identification" of users.
- Unlinkability – is the most complicated privacy property, even in the sense of defining it. One can see actions of users in the system but it is impossible to link actions of particular users.

More detailed discussion of the notions is introduced in the papers enclosed to this chapter.

Regarding formal models, I have found not many of them. The most interesting model is discussed in the enclosed papers: FLASCHE (Freiburg location addressing scheme) [85]. The model introduces Freiburg Privacy Diamond model where vertices represent user, service, location, and device. Device can be understood as a pseudonym for user and location is one of existing contextual information.

I have recently discovered another model that is cited more widely and that is worth a short discussion – k-anonymity model. It represents usual perception of anonymity that is relatively simple. One of the papers applies the model on location privacy.

5.1 k-anonymity Model

k-anonymity model [76] took the name from its property that each message is always indistinguishable in a set of minimum size k. It is based on the same idea as anonymity measurement techniques introduced in section 6.3. Privacy is defined and perceived as an equivalent of anonymity in a set of indistinguishable entities. The model as applied for location anonymity [31] where existence of two types of information is assumed. The data identified as private and should be therefore anonymized contain spatial and temporal identification of agents. The authors define the system in such a way that messages from mobile nodes are transformed by a location anonymity server and then safely exported to LBS (location-based service) provider. You can see that it is a centralised architecture allowing to enforce certain policies – such as the data flow just described.

The anonymity is achieved by either "bluring" position of mobile nodes or by postponing data sent by a mobile node until sufficient number of nodes transmitted from the given spot has been obtained. As you can anonymize nodes in two dimensions (space and time) the algorithm ensuring privacy should work in such a way that the requested anonymity is satisfied and information about time and location of the node is as precise as possible.

The model is able to use certain contextual information but the treatment of this data is rather simplistic and the whole model is kept very simple and easy for potential implementations.

The articles enclosed to this chapter introduce basic ideas of a model I and Vashek Matyas have developed to model level of anonymity. The model is based on graph theory and is able to describe all sorts of contextual information available for nodes/users in the system. I am currently developing with Marek Kumpost a practical implementation of the model using clustering techniques as one of the methodologies for evaluation of privacy level.

5.2 Paper enclosed as C-6: Privacy - what do you mean?

This paper was presented during Ubicomp Privacy Workshop: Current Status and Future Directions, Ubicom 2004.

It is the first attempt to argue possibilities of ensuring privacy in digital environment. Revised and extended version has been published as the paper below.

5.3 Paper enclosed as C-5: On the role of contextual information on privacy attacks

This paper written by me and Vashek Matyas has been firstly presented during Workshop on Privacy and Security in Data-mining (Brighton, UK, 2004). A version with minor corrections was also presented at Security and Embedded Systems workshop (Greece, Aug 2005) and is going to appear in IOS Press/Kluwer. It was also submitted for International Scientific Journal of Computing.

Abstract: Many papers and articles attempt to define or even quantify privacy, typically with a major focus on anonymity. A related research exercise in the area of evidence-based trust models for ubiquitous computing environments has given us an impulse to take a closer look at the definition(s) of privacy in the Common Criteria, which we then transcribed in a bit more formal manner. This lead us to a further review of unlinkability, and revision of another semi-formal model allowing for expression of anonymity and unlinkability – the Freiburg Privacy Diamond. We propose new means of describing (obviously only observable) characteristics of a system to reflect the role of contexts for profiling – and linking – users with actions in a system. We believe this approach should allow for evaluating privacy in large data sets.

Chapter 6

Contextual Information and Privacy Attacks

-Apparently, the Employment Secretary, he's going to get kicked upstairs.
-How do they know? -His driver's been re-assigned.

This chapter does not bring any final results nor closes any of the fundamental problems waiting for their solutions. The main reason is that there is no straightforward solution for countering attacks using contextual information. I begin by recapitulating principles of several anonymity systems and attacks that have been developed. I am using examples of attacks based on traffic analysis as this is the realm of the most intensive research. The second area of privacy attacks relates to data stored in databases as there has been introduced relatively large number of mechanisms for protection of sensitive data. The problem of limiting access to sensitive information in databases has been extensively studied since 70's. The general approach is to return only statistical aggregates of the raw data. There are two basic methods for hiding sensitive data in databases

- Query restriction – queries are required to follow a predefined structure preventing queries on particular data items.
- Data/output perturbation – the content of the database is changed in such a way that the original data items are replaced with new ones reflecting statistical properties of the database content but hiding original values.

The problem here is that the results have not been subject of as thorough security research yet and I believe that principles of traffic analysis, namely exploitation of non-uniform behaviour of users can be used on database data sets with the same success. Although the published results of simulations with random data argue sufficient security.

6.1 Mixes

Basic list of important issues for traffic analysis can be found in [66] and it inspired content of this section. Traffic analysis is trivial until a specially designed system for communication is used. The first architecture of such a system proposed for anonymous message broadcasting were dining-cryptographers networks [12] by Chaum.

The network functionality is based on revealing specially created messages by all nodes. The node willing to send a data combines the data with keys it possesses while all other nodes create their message only from the keys. When all messages are combined, data is revealed. A bit more formally, let us assume that there is a set of participants $P = \{P_1, P_2, \ldots, P_n\}$ and there is also a finite Abelian group $(F, \oplus)$. Network must be firstly initialised so that participants share common pairwise keys $K_{y,z}$ (key between P_y and P_z).

When the network is initialised, we can start transmitting messages. When P_i wants to broadcast a message M the procedure to compute the "encrypted" message is as follows:

$$C_i = M \oplus \sum_{\forall j s.t. \{P_j, P_i\} \in G} sign(i-j).K_{i,j}$$

where $sign(x) = 1$ if $x > 1$ and -1 otherwise. All other participants transmit noise C_j which is composed of the second term in the previous equation. All those interested in the message M may obtain it by cancelling out noise by adding (exor-ing) all broadcasted messages $C_i, \forall\ i \in P$.

This protocol has several drawbacks. It does not protect against active adversaries, jamming of the channel, and limited number of messages that can be broadcasted, as well as necessary participation of all participants in every broadcast.

Special processors have been designed for allowing more flexible designs of anonymity systems – mixes. A mix node receives messages that are modified – usually split onto blocks of the same length, encrypted or/and decrypted – and sent in random order to another mix node or final recipient of the message. As there are usually several mix nodes on the route, it is necessary to determine along which route the message will be sent. There are three basic constructions of mix network.

- Cascade – routes in this architecture are constant for each given pair sender-recipient. The same entry points, exits, as well as intermediate nodes. It is relatively easy-to-analyse the architecture and perform traffic analysis.

- Random order – route is random, i.e. any node can be the next hop in the route. Actual node decides the next hop. The low point is that when a message hits an active

dishonest node belonging to a clique, the message is kept inside the clique for the rest of the route.

- Variations – routes can be partially fixed and partially randomly generated, the route may be defined partially by sender and the rest computed by the mix network, and so on.

When the mix network architecture is chosen, we need to define behaviour of its nodes. On this level of abstraction level we are most interested in the way the nodes flush messages. The decisive parameter is usually the number of messages currently held by the node:

- Threshold – the mix node waits until a certain number of messages is received. At this moment, the messages are processed, mixed and all sent to the next hop in the route.
- Pool – the node has got an internal buffer containing a pool of n messages. When the pool is filled up, each message is flushed with a probability p. When $p = 1$ we get the threshold approach. This procedure is repeated each time the pool fills up.
- Stop-and-go – each message is assigned a randomly generated interval during which it remains at the node. The message is sent when its time is up. It means that if there is only one message in the mix and no other message enters node during this delay interval of the message, the attacker can directly eavesdrop the next recipient and cancel the given node from further traffic analysis computations.

6.2 Attacks

There is an extensive list of papers describing attacks on anonymity systems (see web *www.freehaven.net*). There are at least four basic dimensions classifying attacks:

1. Internal-external – the adversary is either part of the anonymity (mixing) network and controls certain part of the network or she can only follow inputs and outputs of the network nodes.
2. Passive-active – quite understandable classification. The adversary is either only keeping eye on the traffic or is also actively manipulating messages of traffic flow.
3. Static-adaptive – depends on the moment when the attacker selects set of nodes in the network she will be observing. Adaptive attack allows for change of the set during the attack, e.g. by following possible paths of particular messages and identifying nodes promising best attack results.

4. Global-local – in the former case, the adversary is able to follow traffic in the whole mixing network (all nodes, all messages), while the latter case significantly reduces strength in this respect.

There is a number of attacks. As it is not my purpose here to write down detailed descriptions of particular attacks, I just list some of them with brief descriptions.

Brute force attack requires to follow all possible routes the message can be possibly sent along. A set of possible recipients is created with the gathered knowledge. This method can be repeated for more messages and intersections of sets eventually identifies recipients.

Flushing attack tries to flush mix nodes as soon as an interesting message is accepted by a mix node. The attacker actively generates her own dummy messages to reach thresholds triggering flushing of the messages by the node.

Timing attack exploits detailed knowledge of functionality of the mix nodes and, particularly delays introduced by the nodes on possible message routes.

Contextual attacks are the attacks we are interested most as the use information about users' behaviour.

Others like denial of service, "Sting", "Send n'Seek", message delaying, message tagging attacks, and so on.

6.3 Anonymity Measuring

There are two papers that are very interesting for the theme of this thesis. [72, 20] define contextual attacks and use results of the attacks for measuring anonymity (privacy) of the anonymity system users. Diaz et al measure anonymity of the system by relative decrease in the anonymity after an attack. The basis for the computation is again anonymity set – set of users who can possibly be recipients of messages). The relativeness is computed towards the maximum entropy the system is able to provide. The degree of anonymity is defined as

$$d = 1 - \frac{H_M - H(X)}{H_M} = \frac{H(X)}{H_M},$$

where $H(X) = \sum_{i=1..N} p_i log_2(p_i)$ is the anonymity after the attack (p_i are probabilities of users i to belong to the anonymity set), while $H_M = log_2(N)$ is maximum possible

$N \setminus p_f$	0.9	0.75	0.5	0.25
5	0.99	0.94	0.76	0.48
10	0.98	0.92	0.73	0.45
15	0.98	0.91	0.71	0.43
20	0.98	0.90	0.70	0.42
100	0.96	0.86	0.64	0.37
1.000	0.95	0.83	0.60	0.33
100.000	0.93	0.80	0.56	0.30

Table 6.1: Anonymizing power of Crowds system without any attackers

entropy. This method has been used to evaluate anonymity provided by system Crowds [67] in a scenario where the adversary controls certain part of nodes of a Crowds system.

Let us assume that the total number of nodes in the Crowds system is N and the number of corrupted nodes is C. $H_M = log_2(N - C)$. Each node decides whether the request/message will be sent to another node (probability p_f) or directed towards the receiver. After working out some computations it is shown that probabilities assigned to non-collaborators are:

$$p_i = \frac{p_f}{N}, \;\; i = C + 2 \ldots N$$

the entropy of the whole system after attack is defined as

$$H(X) = \frac{N - p_f(N - C - 1)}{N} log_2\Big[\frac{N}{N - p_f(N - C - 1)}\Big] + p_f \frac{N - C - 1}{N} log_2\Big[\frac{N}{p_f}\Big]$$

This function is on plots very close to a linear function. An interesting point is the anonymity level for zero collaborating nodes (no attacker). This level depends on probability p_f and slightly on the number of nodes in the network (it is slowly decreasing with the number). The anonymity level is 0.8 for 5 nodes and $p_f = 0.75$ and 0.6 for $p_f = 0.5$ (see table 6.1). One can see that we can shorten delay caused by Crowds with decreasing number or intermediate nodes but we are at the same time lowering anonymizing power of the system.

Andrei Serjantov and George Danezis published a paper [72] at the same time. They used entropy measure to compute ideal anonymity of composed mix networks. Let us say that we have l mixes with effective sender anonymity size S_i. Let us further assume that each of these mixes sends messages to a new mix *sec* and probability of a particular message coming from S_i is p_i and $\sum p_i = 1$. The anonymity size of messages delivered through the new system is:

$$S_{total} = S_{sec} + \sum_{0<i\leq l} p_i S_i$$

This equation allows us to compute ideal anonymizing strength of any systems if we are able to obtain properties of "atomic" blocks.

6.4 Users

As you can see, it is possible to compute anonymizing strength of systems. Unfortunately, user behaviour may compromise privacy much easier than the strength of the system itself would suggest. We can use intersection attacks [18] as an example. This sort of attacks takes advantage of repeated communications between pairs of users. Naturally, you would not be sending emails randomly to anyone in the network but most probably to your friends or known users. In different wording, there can be found patterns in your behaviour that differentiates your messages from random traffic.

Authors of [18] exploited this fact and introduced a method allowing for identification of recipients of messages. The core idea is that behaviour of each particular user is non-random while aggregated behaviour of all other users in the system can be perceived as uniformly random. They demonstrated the attack on pool mixes with a surprisingly good results and efficiency.

6.5 Paper enclosed as C-7: Pseudonymity in the light of evidence-based trust

This paper was presented during Security Protocols Workshop 2004 (Cambridge, UK). The authors are Daniel Cvrcek and Vashek Matyas.

Abstract: This position paper discusses the relation of privacy, namely pseudonymity, to evidence-based trust (or rather reputation). Critical concepts of evidence-based trust/reputation systems are outlined first, followed by an introduction to the four families of the Common Criteria (for security evaluation) Privacy Class: Unobservability, Anonymity, Unlinkability, and Pseudonymity. The paper then discusses the common problem of many papers that narrow the considerations of privacy to anonymity only, and elaborates on the concept of pseudonymity through aspects of evidence storing, attacks and some of their implications, together with other related issues like use of mixes.

Chapter 7

Anonymity Systems

-Of course in the [civil] service, CMG stands for Call Me God.
And KCMG for Kindly Call Me God.
-What does GCMG stand for? -God Calls Me God.

Anonymity seems to be a forbidden word for most politicians recently. There is always the dilemma between rights of individuals and public interests. The latter is currently being more stressed. I believe that it is a profound requirement to preserve privacy of users in information society we live in. The problem is how to find out whether we have enough privacy or too little.

Several computational metrics to gauge anonymity offered by dedicated anonymizing networks (mix networks, remailers) has been devised during last couple of years. The metrics are based on properties of the networks as well as on patterns in behaviour of their clients. It seems that the latter is becoming to be the main reason for deterioration of privacy related to long-term usage of any system providing some kind of nymity (unlinkability, pseudonymity, anonymity). This chapter is based on an unpublished paper and it focuses on possible use of reputation systems to perform self-evaluation of mix networks. It seems feasible to estimate reputation of network nodes, reputation of the network as a whole, as well as reputation (predictability) of clients. All such data can be used to improve estimates of anonymity provided by the mix network.

7.1 Introduction

Reputation (trust-based) systems are usually meant for environments where no other information about users but certain observations about their behaviour is available. It is, in fact, the only information available in most mobile environments. Furthermore, there is a special class of applications that lack identity information – applications providing some level of privacy for their users through anonymity or pseudonymity. Anonymity networks or systems like Dining Cryptographers' networks (DC networks) [12], Stop and Go Mixes

[50], flash MIXes [42], or Pool Mixes [14] constitute infrastructure providing unlinkability between senders and receivers (for emails) or browsers and www servers (for on-line connections).

Reputation systems use reputation or trust of entities to manage their access rights. Trust was deeply analysed from social and psychology viewpoints in [56]. I just pick several of the characteristics to explain complexity of the notion. Basically, trust is treated conceptually in categories like disposition (properties of trusting party), structural (based on institutional structures), affect/attitude, belief/expectancy, intention, and behaviour. A definition of trustee is orthogonal to this in categories like competence, benevolence, integrity, predictability, ... The categories come mostly from social sciences and can not be directly applied in digital environment. What we can take from the definitions is the necessity to categorise (evidence-based) trust according to context of gathered evidence of behaviour. It also gives us a hint that it is possible to compute trust in different ways to express different subjective viewpoints.

7.2 Risk and Trust - Semantics

Some say that risk is complement to trust, in fact most of current work done in the area of reputation (or trust based) systems uses this definition [44, 45, 48, 80] or ignore risk completely [81, 83, 82]. Such a perception makes risk as a category more or less redundant. I try to present an entirely different meaning of risk that potentially allows self-regulation of reputation systems.

Although there is only one set of evidence that can be used for computations, the results express various properties of the system and may differ substantially. Orthogonality of results can be used for self risk–assessment of the system. I am essentially defining trust as information about particular users in the system and risk as information describing threats to the system associated with actions, processes, as well as with special contexts.

Let us define evidence as an n-tuple $\theta_i = o_i \times c_i^{i_1} \times \ldots \times c_i^{i_k}$, where o_i is encoding of an interaction outcome (number between 0 and 1) and c_i-s are values of contexts. The outcomes encode success or failure of interaction while contexts specify further information about actual conditions. Contexts may contain time, names of clients, IP addresses, services used, and so on. This is however just a simple example as only one source of evidence exists. Otherwise, o_i-s would have to be double-indexed to identify evidence source – meaning of the record.

Trust is worth mentioning in such contexts where predictions of behaviour are needed and we are not certain about their correctness. The basic scenario is to compute trust for active entities – users. The same can be done for services that are vulnerable to attacks, or that are not reliable from one reason or another. As users act through services we can

get an orthogonal information and perform a kind of correlation analysis to find out more specific dependencies.

Where is the risk? The answer is in the way we are computing trust. Trust is an aggregated value of interaction outcomes, let us say that it is again value between 0 and 1 and a particular user, e.g *Alice*, is trusted with value 0.5. We are to decide her request for a service but the required trustworthiness is just about the value. Shall we strictly compare the two arithmetical values or allow certain benevolence? We may improve the decision process when we identify contexts that are risky with particular users or in general, regardless user identity. We can then say whether contexts of the actual Alice's request are the risky ones and reject the request or whether we can increase benevolence and authorise the request.

Risk is an attribute of threats. The set of known threats is very limited (when compared to all the threats existing) after initialization of a system. Furthermore, known threats are basically derived from a list of general threats. There are two main reason for this: (1) there is a limited explicit knowledge about the system behaviour, (2) the system may be so complicated that it is not possible to precisely describe its properties.

The most important task of the risk assessment is to identify correlations among contexts and interaction outcomes. Such a search may be particularly time consuming but it can be run continuously and performed in a distributed manner. Very interesting is also a possibility to apply principles of artificial immune systems [35] where combinations of contexts are generated randomly and when a strong correlation (definition of a threat) is found, the information can be spread throughout the network (analogy of antibodies). We can also run a more thorough search risk assessment in contexts close to the one already identified as risky.

The semantic difference between trust and risk is initially given by difference between evidence sets used for their computation. Trust is derived from all available evidence relevant to particular user (one of the contexts must be *user* identifying context). Risk (and threat it is associated with) is derived from negative outcomes of interactions, i.e. when access was granted and subsequently abused or when results (benefit of the interaction) were not as good as expected, as well as from sequences of events preceding such outcomes.

An interesting problem is whether we can improve security of initially insecure system through a reputation system. We see the main problem in finding the moment when the system becomes too restrictive and denies requests that should be granted – the decision criteria saturate.

Second point that is not entirely clear is whether and how to assign weights to interaction outcomes. There are two ways to adjust weight of a given evidence piece in our system. The basic weight can be set to determine relative importance of different evidence

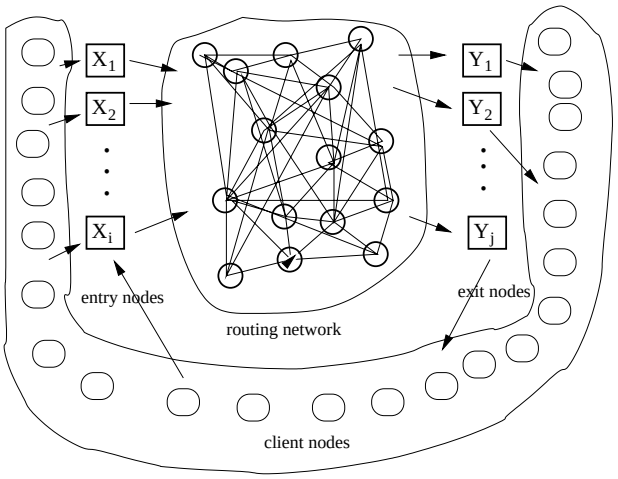

Figure 7.1: Schematic picture of our mix network

sources. One can also fiddle with the value of an observation according to its specific context. The resulting cost/benefit value can be computed with the formula:

$$V(\theta_i) = weight(o_i).\prod value(c_i^{i_1})$$

An example might be a configuration of SpamAssassin program to evaluate probability of email messages to be spam. SpamAssassin computes number of partial information about message contents. The final evaluation is achieved when those partial data are combined into one final number – to do that, one has to set weights to the partial information.

7.3 Anonymizing Networks

There is a number of systems for Internet traffic anonymization that were designed and implemented [5, 6, 24, 29, 67, 68]. While this is supposed to be a conceptual proposition, I do not define any sophisticated architecture but select a basic but widely used scheme of anonymity system. Let us define the concept and pick up several properties of the network important for implementation of a reputation system, see [17] for details.

Anonymity networks have become an interesting research area during last twenty years from the first definition of D. Chaum [12]. A mix network consists of a network of servers (MIXes) with associated public keys. Each node (server) receives encrypted messages that are decrypted, their order permuted, and then forwarded to a next routing node.

First generation of mix networks were produced by cypherpunks mailing lists – Type I anonymous remailers. They were followed by the second generation in the middle of nineties and the third generation recently [17]. There is a number of implementation

problems related to cryptographic properties that are required. The practical implication is that only low lengths of routing paths are possible because of computational complexity related to cryptographic operations.

A simple scheme of a mix network is used to allow short and clear description of ideas (see fig. 7.1). There are sets of entry and exit points that mediate communication between the network and clients. The mixing network between sets of entry nodes $X_1, \ldots, X_i$ and exit nodes $Y_1, \ldots, Y_j$ consists of a cluster of routing nodes composing a dense graph. The entry/exit nodes can be part of the routing network or be placed aside e.g. as SMTP/POP3 servers. Each routing node may be addressed directly from any entry node and it may address any exit node. I am not concerned with computational properties of nodes as they may vary according to actual scenario.

7.3.1 What to Measure

Anonymity networks are very interesting from the viewpoint of reputation systems. As such it seems to be very useful to measure reputation (or trustworthiness) of nodes in the anonymity network to detect active adversaries, reputation of and risks in the network to get information about anonymizing properties of the network. The most important information, however, is reputation of network's clients. The reputation expresses predictability of behaviour – a characteristic that must be as low as possible in the context of privacy or anonymity.

Papers of Díaz [19, 20] or Danezis and Serjantov [16, 18, 72] clearly state importance of clients' behaviour on level of their anonymity. It was shown that certain patterns in selection of recipients or amount of traffic may make attacks considerably easier as long term observation may filter uninteresting traffic (seen as random noise) out. The final impact of client's behaviour on his anonymity is obtained when it's combined with actual traffic in the anonymizing network (mix network), or more precisely, in the chosen routing nodes.

Reputation systems have been used in anonymity systems by Dingledine et al. in [21, 22, 23, 25] but their idea was to use principles of reputation systems to improve reliability of MIX-network. I am going to introduce a different application.

A goal of this text is to informally discuss possibility to use metrics developed for attacking mix networks as benchmarks for self-evaluation of anonymity properties. Current analyses assume usually powerful attackers that can watch all traffic outside of (between entry/exit nodes and clients) as well as inside the network. I want to achieve similar results with one restriction.. I do not want to concentrate information necessary for the reputation system as the very same information can be used for attacks and centralisation would decrease requirements for a successful attacks – it implies use of distributed and

statistical computations.

Obviously, it is not possible to enhance network with self-evaluation capabilities without strengthening power of nodes to acquire necessary information. However, we can use a reputation system to measure inconsistencies between real traffic and number of requests from nodes to evaluate their trustworthiness so the gain overweighs new risks. Cryptography can be used in some cases to limit the threat.

7.4 Definition of Reputation System

Our system is based on metric introduced in [72]. It is built on entropy contained in information about directional traffic between senders and receivers. Let us cite definition of anonymous communication model from [16].

Definition 7.1. Given a model of the attacker and a finite set of all users Ψ, let $r \in \mathcal{R}$ be a role for the user ($\mathcal{R}$={sender, recipient}) with respect to a message $\mathcal{M}$. Let $\mathcal{U}$ be the attacker's a-posteriori probability distribution of users $u \in \Psi$ having the role r with respect to $\mathcal{M}$.

The idea is that all computations are in probabilistic manner. It is completely in line with nature of reputation systems and with our effort not to disclose unnecessary information eventually threatening anonymity of clients. Effective level of anonymity is defined as follows.

Definition 7.2. We define the effective size $\mathcal{S}$ of an r anonymity probability distribution $\mathcal{U}$ to be equal to the entropy of the distribution. In other words

$$\mathcal{S} = -\sum_{u \in \Psi} p_u \; log_2(p_u)$$

where $p_u = \mathcal{U}(u, r)$.

The value of $\mathcal{S}$ is between 0 (no anonymity) and $log_2|\Psi|$ (perfect anonymity).

This is our basic metric. We can do such a computation easily on one node. The task of the reputation system is to estimate $\mathcal{S}$ for the whole anonymity network (and given client) with sufficient precision.

I have already mentioned crucial information that should be included in computations of our reputation system. The following list is a summary:

1. Reputation of the network – what level of anonymity is the anonymizing network able to provide. Computation uses data about all or randomly selected messages.

2. Reputation of network nodes – describes what is the effective size of anonymity sets on particular routing nodes. It may be more effective to track messages through the network than watch only input/output messages.

3. Reputation of clients – estimates predictability of clients' behaviour. The values are based on messages already sent by the clients.

4. Risk associated with a message – we can estimate risk coming from insufficient level of sender anonymity for a particular message (or particular context) by comparing previous observations about mix network properties with similar sets of messages in a given time window, failures of mix network's nodes, length of route, requirements on delivery time, and so on.

7.4.1 Implementation of Metrics

All reputations are worked out from results of the basic formula defined in def. 7.2. Computations of the reputation system must be implemented as distributed and we thus need a formula to combine anonymity provided by a set of nodes.

The model of anonymity network is supposed to deliver messages (e.g. emails) and it comprises of three tasks. There is a set of entry nodes accepting messages from clients, set of exit nodes delivering messages to recipients. There is a network of routing nodes in the middle of these two sets. Each node has got information about messages containing only ID of a previous node, ID of a following node, and times of message delivery and dispatch. It means that entry nodes can identify clients–senders, and exit nodes can share information about client IDs who are final recipients of messages.

Based on these facts, we can make a simple definition of a perfect anonymity network.

Definition 7.3. (Perfect mix network) ensures unlinkability of input and output messages to such extent that any attacker, able to follow all communication between routing nodes, with any amount of logged traffic data, cannot link entry and exit nodes of any message with probability higher than $\frac{1}{n}$ where n is the number of exit nodes.

Note: It is possible to analyse mix networks in a very similar way to symmetric ciphers. We can calculate anonymity properties of a routing node and determine minimum length of message routes to ensure perfect mixing according to the size of the network and the number of exit nodes.

There are basically two sources of entropy that determine level of anonymity for clients.

1. Entropy of the anonymizing network – this entropy is provided by perfect mixing of each single message when using a *perfect mix network*.

2. Unpredictability (randomness) of sender behaviour. Unfortunately, no one's behaviour is uniformly random. Each user has got usually small number of addresses she sends most messages to. There may be other patterns in behaviour, like regular times of sending emails, regular repetitions of messages to certain addresses, and so on.

We expect that perfect mix network does not allow an attacker to gain any further information useful for elimination of nodes from anonymity sets [62].

7.4.2 Entropy of the MIX

Reputation systems should utilise risks connected with two basic threats to anonymity as described above as each of them, network functionality as well as client behaviour, may completely ruin anonymity of the client. Actual architectures assume that clients are the party most interested in preserving anonymity. This is the reason why e.g. source routing (sender selects whole route of the message through the network) is usually used. The reverse side of this approach is that clients may not be fully aware of importance of each single decision that has to be made. Randomness in routing could be an example.

Each mix network can be assigned with theoretical mixing properties calculated under assumption that all nodes behave properly. However, active adversary may control some of the routing nodes in the network and use them for active attacks against client's privacy. These active attacks may be based on delaying/removing some messages [69], repeated sending of a message, flooding the network or cooperation of malicious nodes to uncover routing paths of significant number of messages and thus reduce mixing properties of the network [2].

A reputation system can detect improper behaviour by comparing trustworthiness (reputation) aggregated from nodes encircling (directly communicate with) malicious nodes. This is very important even when senders select routes for their messages – the results may indicate attacks on senders (e.g. viruses, or incorrect implementations). One can devise a variant of flooding attack based on predetermined routing of messages by considerable set of senders caused by some kind of virus.

Most general gouge of improper behaviour is distorted distribution of messages on exit nodes when compared with entry nodes. To get this information, one needs distribution of time delays on routing nodes $D(T_{node})$, route lengths L_{route} or their distribution $D(L_{route})$, number C_{msg} and time distribution of messages $D(T_{entry})$ on entry nodes.

Algorithm 1: Any exit node queries entry nodes for C_{msg} and $D(T_{entry})$. The exit node then simply reconstructs actual state of the mix in time interval t_i:

$$D^{t_i}(S_{mix}) = D^{t_i}(T_{entry}).L_{route}.D(T_{node})$$

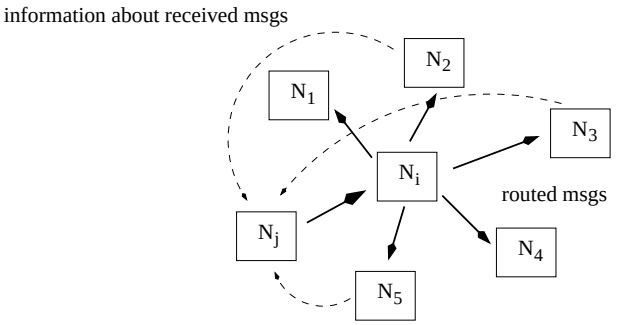

Figure 7.2: Nodes do not use information from all nodes with actual traffic but only a representative sample.

or

$$D^{t_i}(S_{mix}) = D^{t_i}(T_{entry}).D(L_{route}).D(T_{node})$$

and calculates correlation with number $C^{t_i}_{exit}$ and time distribution ($D^{t_i}(S_{exit})$) of messages on itself

$$\rho\Big(\frac{C^{t_i}_{msg}}{n}.D^{t_i}(S_{mix}),\ C^{t_i}_{exit}.D^{t_i}(S_{exit})\Big)$$

or alternatively compares means and deviations from the expected ones.

Algorithm 1 may be performed by any exit node as number and statistical distribution of messages over exit nodes should be uniform. Differences among exit nodes may indicate further improper behaviour in the network.

This computation uncovers very little information about the traffic in the anonymizing network and it therefore imposes very low threat for anonymity of clients. Very similar computation may be utilized by routing nodes – correlations are again computed with traffic on input nodes. This time we need to uncover more information about routing in particular spot of the network. On the other side, we can get much better information about statistical distributions of nodes in message routes.

The first variant protects against attackers able to observe only communication outside of the network, while the latter should be successful in protection against attackers observing all traffic inside the mix network or against distributed attacks based on diverting fraction of messages from certain parts of the network.

Algorithm 2: A routing node N_i (see fig. 7.2) starts evaluation of a node N_j. It randomly selects subset of routing nodes and asks for distribution of messages sent to and received from N_i during certain time interval. N_i then uses the information (a sort of reputation) and aggregates it into a sufficiently testifying information. Substantial

differences between distributions of incoming and outgoing messages of N_j are reflected in a reputation of the node.

This computation is more precise and the data concentrated on nodes are more useful for potential attacker in this case. We have to realize that such computations would take place continuously in all routing nodes. The reputations can be sent to entry nodes and used either as a basis for administrative solution like removing untrustworthy node from the network or warn clients of potential risk associated with certain routes.

The algorithm allows detection of clusters of malicious nodes until their number is lower than $\frac{1}{2}$ of nodes in the network. As long as exchange of reputations throughout the network is ensured.

7.4.3 Clients' Behaviour

In contrary to the reputation of a network from previous subsection, reputation of clients can be evaluated only on entry nodes. Exit nodes can be used to determine portion of messages delivered to particular clients but there is no direct connection to senders.

The profound goal is to measure regularity (bias) in client's behaviour. The idea is very simple and it was described in [72]. I assume behaviour of all network clients to be random noise. It means that continuing observation of a certain client will inevitably reveal any regular patterns in his behaviour. The problem is that messages are not being delivered to all receivers in the same amount and with the same timing distributions. All one needs for a successful attack is a track of messages exiting mix network and a set of exact times when analyzed client sent her messages.

Algorithm 3: Entry node collects times of all messages sent from a sender of a current message. She randomly selects one (or maybe several to decrease bias) exit node and ask him for anonymity level (computed as entropy) or statistical properties of receivers distribution. The entry node compares required anonymity level with the one achievable and she eventually bounces the message back.

The content received from an exit node during *Algorithm 3* determines its possible abuse by an attacker. Number of requests may be monitored and used during reputation evaluations (higher number of requests could be suspicious). More powerful is a cryptographic solution when senders accompany their messages with a special part used as a ticket to request this kind of information from exit nodes.

The protection here is based on the network performing attacks and signaling when level of anonymity drops below certain level (the attack was successful) back to a client. The new threat lies in a consequences of an attacker gaining control of any of the entry nodes. She does not need to follow all traffic then, but she may just ask the information

within regular operation. The implemented algorithm must therefore release only minimum necessary information that can not be used for more powerful analysis. The second precaution lies in careful setting of threshold between deficient and sufficient anonymity level.

7.4.4 Path Coupling and Markov Chains

ESORICS'03 conference introduced a paper with a new methodology that can be used to estimate mixing properties of anonymity networks [55]. The method exploits fast mixing property of certain Markov chains [8] and can be used to prove mixing of not only single messages but also mutual mixing among messages. It can measure impact of dependencies among messages on the input of an anonymizing network. Although the application of path coupling lemma in [55] is weak and we believe that the results are not correct, a way of mix network analysis was presented (see Appendix A for detailed analysis).

The general idea is to statistically estimate difference of outputs for two inputs that are identical but one of their element (difference of the inputs is 1). Rapid mixing denotes situation when estimated outputs' difference is lower than 1. We can then determine minimal necessary length of route to ensure mixing with probability distribution close enough to uniform distribution.

There is an optimistic and a defensive application of path coupling in *reputation enhanced* mix network. The optimistic is that the routing nodes are honest and they can determine quality of their mixing algorithm and share it with the nodes further on the message path. The defensive one assumes that some nodes are under attacker's control. Mixing properties of nodes are computed by their neighbors. The mixing estimates can be always sent to entry nodes or bounced back to senders when mixing of a message was not sufficient.

7.5 Privacy Issues

There is a couple of layers of privacy issues in the introduced scenarios. The first layer is related to implementation of reputation systems themselves. This scope is more thoroughly analyzed in [15]. Even more important are issues related to concentration of traffic information on all nodes in the anonymizing network.

We commented some of the problems related to new information accessible on routing or entry/exit nodes. Unfortunately, it is not clear whether it is possible to achieve a state when the information available locally for an attacker can not be used for more powerful analysis than the network itself is performing. We cannot be sure that it can not used in a different way decreasing anonymity more than the network estimate. There is no proof

and we do not know whether it is possible to deliver a proof stating minimum level of anonymity preservable with a certain set of traffic information.

7.6 Conclusions

It is very hard to make any firm conclusions. I introduced very interesting application of reputation systems on privacy preserving systems. This deployment is very different from most of current application scenarios for reputation systems.

There are, however, serious questions about the implications of such a system on privacy of clients. So far, very powerful attacker was considered when analyses of anonymizing networks were conducted. Application of a reputation system decreases requirements for successful launch of an attack because considerable part of necessary information is available on any node in the anonymizing network.

On the other side, we know that privacy of clients deteriorates and it is important to have an instrument capable to measure actual level of anonymity – privacy. We have recalled several methods that are based just on bias implied by clients behaviour.

Chapter 8

Conclusions

-I was just ... passing. -Passing? -Yes, passing.
-Oh, passing. And where were you going? -I was just going ... past.

The goal of the thesis was to present a relatively new area in the research of information system security. We are able to solve problems related to privacy of users through various policy settings and by using trusted systems enforcing our policy. The problem is that new information systems are more likely to be designed as distributed and mobile. P2P systems are just the first example of such systems whereas ubiquitous computing, roaming entities, large cooperating networks characterise systems that are currently subject of intensive research and probable to become widely used in near future.

This sort of systems does not offer the comfort of trusted platforms allowing for global enforcement of our policies. We have to accommodate to new conditions and come up with new security paradigms suitable for these new environments. The research is lead by P2P systems that are already widely used for sharing huge amounts of data. It seems that basic functional problems are already solved and we can make use of very powerful routing protocols and middleware technologies. These systems should ensure certain properties so as the honest users are guaranteed a certain level of quality of services. And security properties are one of the prime research goals today.

The preferable cornerstone for solutions of security problems seems to be reputation or trust – as this thesis demonstrated. This approach assumes processing of large amounts of data describing user's behaviour. This data can be used to derive sufficient information about user's trustworthiness but it can also be used to breach his/her privacy – one of the basic human rights.

An extensive research has been carried out in the area of attacks on anonymity systems through traffic analysis. It proves, sofar, that it is not possible to build an anonymity system able to completely hide users in anonymity sets. The problem is that user behaviour is not random but contains regularities that have the potential to reveal identity of par-

ticular users. One can compare this problem to a task of building a hash function hiding relations between inputs (respective outputs are random) while sequences of the inputs contain repeating patterns and repetitions that remain unchanged in the outputs.

The first step in the research here is to define the notion of privacy. It is possible to find number of definitions but they are useless for computational tasks. We have to be able to state requirements on privacy so we can measure the level of privacy and identify breaches of privacy that are unacceptable for users. In that moment we can use the measures to warn users about deterioration of their privacy and react.

As we need a lot of contextual information for security mechanisms in ubiquitous systems, the second goal in this area is to find the sweet point. The situation when enough data needed for security mechanisms is available while still ensuring sufficient level of privacy.

A very specific problem whose solution would essentially change our ability to preserve privacy while obtaining enough information for enforcing system security is keeping control over evidence life-cycle. The existence of a solution here is an open question.

Bibliography

[1] Aldous, D.: Random walks on finite groups and rapidly mixing markov chains. In Seminaire de Probabilites XVII 1981/1982, vol. 986 of *Lecture Notes in Mathematics*, Springer-Verlag, 1983, pp. 243–297.

[2] Back, A., Möller, U., Stiglic, A.: Traffic analysis attacks and trade-offs in anonymity providing systems. In Proceedings of Information Hiding Workshop (IH 2001), I. S. Moskowitz, Ed., no. 2137 in LNCS, Springer-Verlag, 2001, pp. 245–257.

[3] Bacon, J., Belokosztolszki, A., Cvrcek, D., Dimmock, N., Eyers, D., Ingram, D., Moody, K.: Preliminary definition of a trust-based access control model. Project deliverable D3.2, 2004.

[4] Blaze, M., Feigenbaum, J., Lacy, J.: Decentralized trust management. In Proc. IEEE Conference on Security and Privacy, AT&T, 1996.

[5] Boucher, P., Shostack, A., Goldberg, I.: Freedom systems 2.0 architecture. White paper, Zero Knowledge Systems, Inc., 2000.

[6] Brown, Z.: Cebolla: Pragmatic IP Anonymity. In Proceedings of the 2002 Ottawa Linux Symposium, 2002.

[7] Bubley, R., Dryer, M.: Faster random generation of linear extensions. In Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms, ACM Press, 1998, pp. 350–354.

[8] Bubley, R., Dyer, M.: Path coupling: A technique for proving rapid mixing in markov chains. In Proceedings of the 38th Symposium on Foundations of Computer Science, FOCS'97, IEEE Computer Society Press, 1997, pp. 223–231.

[9] Cahill, V., et al.: Using trust for secure collaboration in uncertain environments. *IEEE Pervasive Computing Magazine*, July-September, 2003.

[10] Center, N. C. S.: *Trusted Computer System Evaluation Criteria*. No. DOD 5200.28-STD in DoD standards. Department of Defence, 1985.

[11] Chaum, D.: Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 4(2), 1981.

[12] Chaum, D.: The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology: the journal of the IACR*, 1(1):65–75, 1988.

[13] Chaum, D.: Secret-ballot receipts and transaprent integrity: Better and less-costly electronic voting at polling places. http://www.vreceipt.com/article.pdf, unknown.

[14] Cottrell, L.: Mixmaster and remailer attacks. Electronic Document, 1994.

[15] Cvrček, D., Matyáš, V.: Pseudonymity in the light of evidence-based trust. In Proc. of the 12th Workshop on Security Protocols, LNCS (forthcoming), Springer-Verlag, 2004.

[16] Danezis, G.: Mix-networks with restricted routes. In Proceedings of Privacy Enhancing Technologies workshop (PET 2003), R. Dingledine, Ed., no. 2760 in LNCS, Springer-Verlag, 2003.

[17] Danezis, G., Dingledine, R., Mathewson, N.: Mixminion: Design of a Type III Anonymous Remailer Protocol. In Proceedings of the 2003 IEEE Symposium on Security and Privacy, 2003.

[18] Danezis, G., Serjantov, A.: Statistical disclosure or intersection attacks on anonymity systems. In Proceedings of 6th Information Hiding Workshop (IH 2004), LNCS, forthcoming.

[19] Diaz, C., Serjantov, A.: Generalising mixes. In Proceedings of Privacy Enhancing Technologies workshop (PET 2003), R. Dingledine, Ed., no. 2760 in LNCS, Springer-Verlag, 2003.

[20] Diaz, C., Seys, S., Claessens, J., Preneel, B.: Towards measuring anonymity. In Proceedings of Privacy Enhancing Technologies Workshop (PET 2002), R. Dingledine and P. Syverson, Eds., no. 2482 in LNCS, Springer-Verlag, 2002.

[21] Dingledine, R., Freedman, M. J., Hopwood, D., Molnar, D.: A reputation system to increase mix-net reliability. In Proceedings of the 4th International Workshop on Information Hiding, no. 2137 in LNCS, Springer-Verlag, 2001, pp. 126–141.

[22] Dingledine, R., Mathewson, N., Syverson, P.: Reputation in privacy enhancing technologies. In Proceedings of the 12th annual conference on Computers, freedom and privacy, 2002.

[23] Dingledine, R., Mathewson, N., Syverson, P.: Reputation in p2p anonymity systems. In Workshop on Economics in P2P Systems, 2003.

[24] Dingledine, R., Mathewson, N., Syverson, P.: Tor: The second-generation onion router. In Proceedings of the 13th USENIX Security Symposium, 2004.

[25] Dingledine, R., Syverson, P.: Reliable mix cascade networks through reputatio. In Proceedings of Financial Cryptography (FC'02), 2002.

[26] Douceur, J.: The sybil attack. In 1st International Workshop on Peer-to-Peer Systems (IPTPS'02), no. 2429 in LNCS, Springer-Verlag, 2002, pp. 251–260.

[27] English, C., Wagealla, W., Nixon, P., Terzis, S., Lowe, H., McGettrick, A.: Trusting collaboration in global computing systems. In iTrust 2003, P. Nixon and S. Terzis, Eds., no. 2692 in LNCS, Springer-Verlag, 2003, pp. 136–149.

[28] (EPIC), E. P. I. C.: *Privacy and Human Rights 2003*, 1st ed. EPIC, 2003.

[29] Freedman, M. J., Morris, R.: Tarzan: A peer-to-peer anonymizing network layer. In Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS 2002), 2002.

[30] Frieze, A. M., Luczak, T.: On the independence and chromatic numbers of random regular graphs. *Journal of Combinatorial Theory Series B*, 54(1):123–132, 1992.

[31] Gedik, B., Liu, L.: Location privacy in mobile systems: A personalized anonymization model. In 25th IEEE International Conference on Distributed Computing Systems (ICDCS'05), IEEE Press, 2005, pp. 620–629.

[32] Gollmann, D.: Why trust is bad for security. In First International Workshop on Security and Trust Management, S. M. Cas Cremers, Valeria Issarny, Ed., Electronic Notes in Theoretical Computer Science, 2005, pp. 3–9.

[33] Grandison, T., Sloman, M.: A survey of trust in internet applications. *IEEE Communications Surveys & Tutorials*, 3(4), 2000.

[34] Gülcü, C., Tsudik, G.: Mixing E-mail with Babel. In Proceedings of the Network and Distributed Security Symposium - NDSS '96, IEEE, 1996, pp. 2–16.

[35] Harmer, P. K., Williams, P. D., Gunsch, G. H., Lamont, G. B.: An artificial immune system architecture for computer security applications. *IEEE Transactions On Evolutionary Computation*, 6(3):252–280, 2002.

[36] Helpern, J. Y., Pucella, R.: A logic for reasoning about evidence. In Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence (UAI'03), 2003, pp. 297–304.

[37] Hintz, A.: Fingerprinting websites using traffic analysis. In Proceedings of Privacy Enhancing Technologies workshop (PET 2002), R. Dingledine and P. Syverson, Eds., no. 2482 in LNCS, Springer-Verlag, 2002.

[38] Hulme, G. V.: Data breaches: Turn back the tide. http://www.it-observer.com/articles.php?id=806, 2005.

[39] ISO JTC 1/SC 27: *Information technology – Security techniques – Information security incident management.* No. 18044:2004 in ISO/IEC. ISO, 2004.

[40] ISO JTC 1/SC 27: *IT security techniques – Management of information and communications technology security – Part 2: Information security risk management.* No. 13335-2 in ISO/IEC. ISO, 2004.

[41] ISO JTC 1/SC 27: *Information technology – Security techniques – Code of practice for information security management.* No. 17799:2005 in ISO/IEC. ISO, 2005.

[42] Jakobsson, M.: Flash Mixing. In Proceedings of Principles of Distributed Computing - PODC '99, ACM Press, 1999.

[43] Jakobsson, M., Juels, A., Rivest, R. L.: Making mix nets robust for electronic voting by randomized partial checking. In Proceedings of the 11th USENIX Security Symposium, 2002.

[44] Jøsang, A.: The right type of trust for distributed systems. In Proceedings of the 1996 New Security Paradigms Workshop, C. Meadows, Ed., ACM, 1996.

[45] Jøsang, A.: A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(9), 2001.

[46] Jøsang, A.: The consensus operator for combining beliefs. *Artificial Intelligence Journal*, 141(1–2):157–170, 2002.

[47] Jøsang, A.: Subjective evidential reasoning. In Proc. of the 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2002), July, 2002.

[48] Jøsang, A., Daniel, M., Vannoorenberghe, P.: Strategies for combining conflicting dogmatic beliefs. In Proc. of the 6th International Conference on Information Fusion, 2003, pp. 1133–1140.

[49] Kamwar, S. D., Schlosser, M. T., Garcia-Molina, H.: The eigentrust algorithm for reputation management in p2p networks. In WWW 2003, 2003.

[50] Kesdogan, D., Egner, J., Büschkes, R.: Stop-and-go MIXes: Providing probabilistic anonymity in an open system. In Proceedings of Information Hiding Workshop (IH 1998), no. 1525 in LNCS, Springer-Verlag, 1998.

[51] Lik Mui, Mojdeh Mohtashemi, A. H.: A computational model of trust and reputation. In 35th Hawaii International Conference on System Sciences (HICSS'02), vol. 7, 2002, p. 188.

[52] Lik Mui, Ari Halberstadt, M. M.: Evaluating reputation in multi-agents systems. In AAMAS 2002 Ws Trust, Reputation, ..., R. F. et al., Ed., vol. 2631 of *LNAI*, Springer-Verlag Berlin Heidelberg, 2003, pp. 123–137.

[53] Lin, W., Yang, Y., Zhang, S.: A computational reputation model in p2p networks based on trust and distrust. In ICCNMC 2005, W. Z. X. Lu, Ed., no. 3619 in LNCS, Springer-Verlag, 2005, pp. 501–508.

[54] Liu, J., Issarny, V.: Enhanced reputation mechanism for mobile ad hoc networks. In iTrust 2004, C. D. J. et al., Ed., no. 2995 in LNCS, 2004, pp. 48–62.

[55] M. Gomułkiewicz, Klonowski, M., M. Kutyłowski: Rapid mixing and security of chaum's visual electronic voting. In ESORICS 2003, E. Snekkenes and D. Gollmann, Eds., no. 2808 in LNCS, Springer–Verlag, 2003, pp. 132–145.

[56] McKnight, D. H., Chervany, N. L.: The meanings of trust. Working Paper 96-04, University of Minnesota, 1996 (updated 2000).

[57] Moeller, U., Cottrell, L., Palfrader, P., Sassaman, L.: Mixmaster protocol version 2. http://www.ietf.org/internet-drafts/draft-sassaman-mixmaster-03.txt, 2004. IETF Draft.

[58] Mui, L., Halberstadt, A.: Notions of reputation in multi-agents systems: A review. In Proceedings of AAMAS-02, ?, Ed., 2002, pp. 280–287.

[59] Newman, R., Moskowitz, I., Syverson, P., Serjantov, A.: Metrics for traffic analysis prevention. In Proceedings of Privacy Enhancing Technologies workshop (PET 2003), R. Dingledine, Ed., no. 2760 in LNCS, Springer-Verlag, 2003.

[60] Obreiter, P.: A case for evidence-aware distributed reputation systems. In iTrust 2004, C. D. J. et al., Ed., no. 2995 in LNCS, Springer-Verlag, 2004, pp. 33–47.

[61] of Parliamant, H.: Privacy and personal information protection act. New South Wales Consolidated Acts, http://www.austlii.edu.au/au/legis/nsw/consol_act/papipa1998464/, 1998.

[62] Pfitzmann, A., Köhntopp, M.: Anonymity, unobservability and pseudonymity – a proposal for terminology. In Designing Privacy Enhancing Technologies: Proceedings of the International Workshop on the Design Issues in Anonymity and Observability, vol. 2009 of *LNCS*, Springer-Verlag, 2000, pp. 1–9.

[63] Pfitzmann, B., Pfitzmann, A.: How to break the direct RSA-implementation of MIXes. In Proceedings of EUROCRYPT 1989, no. 434 in LNCS, Springer-Verlag, 1990.

[64] Qu, X., Yang, X., Tang, Y., Zhou, H.: A behaviour charactersitcs-based reputation evaluation method for grid entities. In EGC 2005, P. M. A. S. et al., Ed., no. 3470 in LNCS, Springer-Verlag, 2005, pp. 567–577.

[65] Ratnasamy, S., Francis, P., Handley, M., Karp, R.: A scalable content-addressable network. *ACM SIGCOMM Computer Communication Review*, 31(4, Proceedings of the 2001 SIGCOMM conference):161–172, 2001.

[66] Raymond, J.-F.: Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems. In Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability, H. Federrath, Ed., no. 2009 in LNCS, Springer-Verlag, 2000, pp. 10–29.

[67] Reiter, M., Rubin, A.: Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security*, 1(1), 1998.

[68] Rennhard, M., Plattner, B.: Introducing MorphMix: Peer-to-Peer based Anonymous Internet Usage with Collusion Detection. In Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2002), 2002.

[69] RProcess: Selective denial of service attacks. Usenet post, 1999.

[70] SECURE Consortium: RTD Proposal - SECURE: Secure Environments for Collaboration among Ubiquitous Roaming Entities, 2001.

[71] Sergio Marti, H. G.-M.: Taxonomy of trust: Categorizing p2p reputation systems. *submitted to Elsevier Science*, 2005.

[72] Serjantov, A., Danezis, G.: Towards an information theoretic metric for anonymity. In Privacy Enhancing Technologies (PET), no. 2482 in LNCS, Springer-Verlag, 2002, pp. 41–53.

[73] SOG-IS: *Information Technology Security Evaluation Criteria (ITSEC)*. Department of Trade and Industry, London, 1991.

[74] Stoica, I., Morris, R., Karger, D., Kaashoek, M. F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications. *ACM SIGCOMM Computer Communication Review*, 31(4, Proceedings of the 2001 SIGCOMM conference):149–160, 2001.

[75] Suzanne, K., Kim, J.: Belief revision process based on trust: Agents evaluating reputation of information sources. In Trust in Cyber-societies, R. Falcone, M. Singh, and Y.-H. Tan, Eds., no. 2246 in LNAI, Springer-Verlag, 2001, pp. 73–82.

[76] Sweeney, L.: k-anonymity: A model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):557–570, 2002.

[77] Syverson, P., Tsudik, G., Reed, M., Landwehr, C.: Towards an Analysis of Onion Routing Security. In Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability, H. Federrath, Ed., no. 2009 in LNCS, Springer-Verlag, 2000, pp. 96–114.

[78] Terzis, S., Wagealla, W., English, C., Nixon, P., McGettrick, A.: Preliminary trust formation model. Project deliverable D2.1, 2003.

[79] The Common Criteria Project Sponsoring Organisations: *Common Criteria for Information Technology Security Evaluation - part 2, version 2.2.* CC Project, 2004.

[80] Twigg, A., Dimmock, N.: Attack-resistance of computational trust models. In IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003.

[81] Weeks, S.: Understanding trust management systems. In IEEE Symposium on Security and Privacy, 2001, pp. 94–105.

[82] Winsborough, W. H., Li, N.: Towards practical automated trust negotiation. In Proceedings of the Third International Workshop on Policies for Distributed Systems and Networks (Policy 2002), IEEE Computer Society Press, 2002, pp. 92–103.

[83] Xiong, L., Liu, L.: Building trust in decentralized peer-to-peer electronic communities. In The 5th International Conference on Electronic Commerce Research(ICECR-5), 2002.

[84] Yu, B., Singh, M. P.: An evidential model of distributed reputation management. In 1st International Joint Conference on Autonomous Agents and MultiAgent Systems, ACM, 2002.

[85] Zugenmaier, A., Kreutzer, M., Müller, G.: The freiburg privacy diamond: An attacker model for a mobile computing environment. In Kommunikation in Verteilten Systemen (KiVS) '03, 2003.

Appendix A

Rapid Mixing in Anonymity Networks

-...figures we'll be nonsense. -Why -They'll be incomplete.
-Government figures are a nonsense, anyway.
-I think Sir Humphrey wants to ensure they are a complete nonsense.

This chapter is based on an unpublished text related to properties of anonymity systems. It is an analysis of a voting scheme (more precisely, of an approach for estimating security of mixing network) me and George Danezis carried out during my stay at Cambridge University. The subject of our analysis is a paper describing usage of path coupling to determine minimum length of paths in the mixing network to achieve sufficient anonymity of, in this case, votes. The purpose of including this text in the thesis is to demonstrate how difficult it is to compute theoretical privacy properties of systems.

A.1 Rapid Mixing in Anonymity Networks

I am to analyze results of Gomulkiewicz et al. [55] about the size of mix-cascade and corrected results are presented. This analysis leads to a more general approach for describing properties of anonymizing networks based on path coupling.

A.1.1 Introduction

The paper [55] of Gomułkiewicz, Klonowski, and Kutyłowski presented at ESORICS 2003 seemed to be a break-through in theoretical description of anonymizing networks (the paper was also nominated for the award of the best work in privacy). Although there are several methods to describe quality of mixers, the approach presented in the paper seemed to be very general and applicable on many different types of anonymizing networks.

However, we have discovered a number of inconsistencies while studying the paper.

We show that the conclusion of the paper is not correct. Particularly the statement that the necessary size of mix network for achieving ideal mixing is constant in the terms of number of messages to be mixed. We base our result on detailed analysis of usage of path coupling used in the original paper.

Path Coupling

I should start with definition of path coupling. Coupling methodologies were used particulary to solve problem of graph colourings [30] modelled as Markov Chains. Coupling uses a metric $D(M)$ defined over finite Markov chain M. Maximum value of $D(M)$ over all pairs of states of Markov chain $(X, Y \in \Omega)$ is called *diameter* of M. The following lemma is taken from [1, 7].

Lemma A.1. ***(Coupling)*** *Suppose (X, Y) is a random process (the coupling) such that marginally, X and Y are both copies of M. Moreover, suppose Y_0 is chosen from π, and μ_t is the distribution of X_t, then*

$$d_{TV}(\mu_t, \pi) \leq P(X_t \neq Y_t)$$

where d_{TV} is the (total) variation distance metric on measures, π is the stationary distribution and μ_t is M's distribution after t steps.

X and Y are coupled when $X_t = Y_t$. Unfortunately, to receive correct value for d_{TV} we have to analyze all possible pairs of states of M.

Practicability of coupling has changed with *path coupling* presented in [8]. They defined a path – sequence of states of Markov chains – and considered pairs of states that are adjacent on the path. It is then sufficient to show that path-wise adjacent states of the two Markov chains will come closer for all pairs of these states.

Theorem A.2. ***(General Path Coupling [dyer])*** *Let Ω be a set of all Markov states, and β is expected distance of states X_t, Y_t after performing a step t.*
Then, if $\beta < 1$ and $t \geq \left\lceil \frac{ln(n\varepsilon^{-1})}{ln\beta^{-1}} \right\rceil$, we have $d_{TV}(\mu_t, \pi) \leq \varepsilon$.

Parameter β is defined for graph colouring in [8] as

$$\beta = max_{X,Y \in \Omega, i \in V} \Big\{ 1 - J(i) + \sum_{j \in V} J(j)\ d_{TV}\ (\kappa_{X,j}, \kappa_{Y,j}) \Big| Y = K_{i \to c}\ \ for\ some\ \ c \in C \wedge Y \neq X \Big\},$$

where $J(i)$ is a probability of choosing i-th vertex and $d_{TV}(\kappa_{X,j}, \kappa_{Y,j})$ is a distance of colourings when states X_j and Y_j are selected.

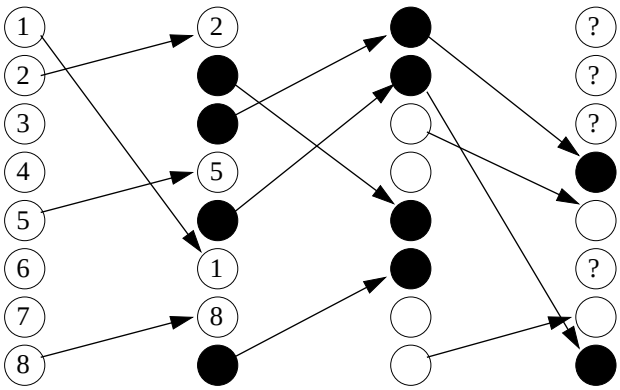

Figure A.1: An example of vote mixing.

A.1.2 Correct Mixing for Chaum's Electronic Voting

We are dealing with Chaum's Voting Procedure [13] enhanced with *randomized partial checking* [43]. The basic scheme assumes that there is a set of trustees that ensure unlinkability of a vote with a particular voter. Each vote is encrypted with keys of all trustees so the vote cannot be counted until decrypted by all trustees:

$$T = D_k(D_{k-1}(\dots D_1(C)\dots))$$

Each trustee (1) decrypts votes with her key, (2) randomly permutes the results, and (3) sends the permuted list to the next trustee. The trustees do not trust each other and therefore another operation (4) (randomized partial checking) is added. E.g. trustee $\mathcal{C}_i$ asks trustee $\mathcal{C}_{i-1}$ to uncover keys used to decrypt randomly selected half of the votes. This randomized partial checking ensures, with very high probability, that no trustee replaces genuine votes (fig. A.1).

The problem of the scheme is that there is a small chance the whole path of the decrypted vote was uncovered and privacy (impossibility to link voter with his vote) will be broken. Each trustee therefore performs operations (1) and (2) twice (steps i_1 and i_2). When she is asked to uncover transformations for a set $\mathcal{A}$ of votes she shows keys used for votes in $\mathcal{A}$ in step i_1 and keys for the rest of votes used in step i_2.

Problem Statement

We take the problem as stated in [55]. We need a strong result determining amount of information about voting preferences leaked in the revealing process for any outcome of elections.

Definition Let Π denotes permutation of votes such that $\Pi(i) = j$ describes situation when i-th ciphertext processed by the first trustee $\mathcal{C}_1$ corresponds to j-th vote as published by the last trustee $\mathcal{C}_k$. The voting scheme is secure when probability distribution of Π is indistinguishable from uniform distribution, i.e. $d_{TV}(\Pi, \pi) < \varepsilon$, where $\varepsilon \leq \frac{1}{n^k}$ for a constant k and number of votes n.

The goal of the following text is to determine minimum number of trustees needed for obtaining the uniform distribution of Π.

Definition of a Stochastic Process

Behaviour of trustees is independent. Assuming perfectly secure decryption, the decryption and permutation can be seen as a purely random function.

Lemma The operations (stage) performed by each trustee as defined for electronic voting with randomized partial checking are indistinguishable from process defined with the following steps [55]:

1. Randomly select half of votes and put their indexes into set A_i;
2. The items on positions $j \notin A_i$ are permuted in random;
3. The items on positions $j \in A_i$ are permuted in random;
4. All items are permuted in public.

Proof Let us assume that N is a set of all votes such that $N = A \cup \overline{A}, A \cap \overline{A} = \emptyset, |A| = |\overline{A}|$. We have also defined permutations π_1, π_2 and sets $B, \overline{B}$ such that $B = \pi_1(A), \overline{B} = \pi_1(\overline{A})$ that represent original definition of a trustee operations. The whole state is then:

$$\pi_2 \circ \pi_1(N) = \left(\pi_2^B \cup \pi_2^{\overline{B}}\right) \circ \left(\pi_1(A) \cup \pi_1(\overline{A})\right) = \left(\pi_2^B \circ \underline{\pi_1(A)}\right) \cup \left(\underline{\pi_2^{\overline{B}}} \circ \pi_1(\overline{A})\right)$$

$$= \pi_2(B) \cup \pi_2(\overline{B})$$

The equation holds because associativity of permutations. By definition of electronic voting the permutations and $\pi_2(\overline{B})$ are public. We have two compositions of permutations. It is not important which of the permutations in the chain is made public from the attackers viewpoint. When π_2^B is made public instead of $\pi_1(A)$ we get

$$\pi_2 \circ \pi_1(N) = \left(\underline{\pi_2^B} \cup \underline{\pi_2^{\overline{B}}}\right) \circ \left(\pi_1(A) \cup \pi_1(\overline{A})\right) = \underline{\pi_2} \circ \left(\pi_1(A) \cup \pi_1(\overline{A})\right)$$

QED

Power of Electronic Voting Stage

When we look at the anatomy of one stage we can quickly find out that a passive external adversary can not discern two votes when both belong to A (or $\overline{A}$). We can therefore model votes just with two colours: black (belonging to A) and white (elements of $\overline{A}$). It means that our primary target – to determine moment when distribution of votes is indistinguishable from a uniform one – can be reduced to uniform distribution of black/white coloured votes.

We would like to use path coupling to compute coupling time. We will use a metric function $\Delta : \mathcal{P}_n \times \mathcal{P}_n \rightarrow \{0, 1, .., n\}$. Defined for all pairs $p_1, p_2 \in \mathcal{P}_n$ as minimum number of transpositions necessary to make p_1 and p_2 identical.

Construction of Path Coupling

For any two states X, Y of Markov chain (any two permutations of vote colouring) we can construct a path $X = Z_1\ Z_2\ \ldots\ Z_k = Y$ of length $k \leq n$ such that $\Delta(Z_i, Z_{i+1}) = 1$. The last essential assumption for the path coupling is value of β. According to definition β is computed according to the following rule: from initial value of 1 is deducted probability of the paths to become identical and a sum of possible differences is added back.

There are four basic cases that may happen for two states p_1, p_2 with $\Delta(p_1, p_2) = 1$:

1. $p_1 \in A_i \wedge p_2 \in \overline{A_i}$
2. $p_1 \in \overline{A_i} \wedge p_2 \in A_i$
3. $p_1, p_2 \in A_i$
4. $p_1, p_2 \in \overline{A_i}$

The probability of each of the four cases is 25 % because of random selection and equal size of of sets A_i and $\overline{A_i}$.

Case 1 $\Delta(p_1, p_2) = 1$ on the input to the stage i. Processing of the stage does not change distance of states of the two Markov chains at all. We still know that one two votes has reversed colors.

Case 2 $\Delta(p_1, p_2) = 1$ alike Case 1.

Case 3 $\Delta(p_1, p_2) = 1$. The random permutation inside the halves of votes imply coupling of the two chains.

Case 4 $\Delta(p_1, p_2) = 1$. Alike Case 3.

This results in $\beta = 1 - 0.5 + 1 * 0,25 + 1 * 0.25 = 0.5$ – this result is much simpler that the one presented in [55]. The minimum coupling time is then

$$\tau(\varepsilon) \leq \left\lceil \frac{\ln(D.\varepsilon^{-1}}{\ln\ \beta^{-1}} \right\rceil = \left\lceil \frac{\ln(n.\varepsilon^{-1})}{\ln\ 2} \right\rceil$$

The original paper stated $\varepsilon = \frac{1}{n}$ as sufficient level of mixing, the time would be:

$$\tau\left(\frac{1}{n}\right) \leq \left\lceil \frac{\ln(n.\frac{n}{1})}{\ln\ 2} \right\rceil = \frac{2.\ln n}{2} = \ln n$$

A.1.3 Conclusions

No particular conclusions here. We have shown through a correct application of path coupling the the number of steps is logarithmic in number of votes. A result that is also much more corresponding to reality and intuition – when compared to the original result stating the minimum number of steps as $O(1)$.

Appendix B

Publications

B.1 List of Publications Constituting This Thesis

1. Cvrček D., Moody K.: Combining Trust and Risk to Reduce the Cost of Attacks, In: iTrust 2005, Berlin, DE, Springer, 2005, pp. 372–383, ISSN 0302-9743.

2. Cvrček D.: Dynamics of Reputation, In: NordSec'04, Helsinki, FI, HUT, 2004, pp. 1–7, ISSN 1455-9749.

3. Cvrček D.: Using Evidence for Trust Computation, In: Mikulášská kryptobesídka, Brno, CZ, ECOM, 2003, pp. 12–20.

4. Cvrček D., Matyáš V. ml., Patel Ahmed: Evidence processing and privacy issues in evidence-based reputation systems, In: Computer Standards & Interfaces, Vol. 27, No. 5, NL, pp. 533–545, ISSN 0920-5489.

5. Cvrček D., Matyáš V. ml.: On the role of contextual information for privacy attacks and classification, In: Workshop on Privacy and Security Aspects of Data Mining, Brighton, GB, 2004, pp. 31–39.

6. Cvrček D., Matyáš V. ml.: Privacy - what do you mean?, In: UBICOMP Privacy Workshop, Nottingham, GB, 2004, pp. 12–19.

7. Cvrček D., Matyáš V. ml.: Pseudonymity in the light of evidence-based trust, In: Authentic Privacy, Berlin, DE, Springer, 2004, pp. 109–116, ISSN 0302-9743.

B.2 List of Other Publications of the Author

1. Bond M., Cvrček D.: Penetration to Secure Hardware, Invited lecture, In: SPI 2005, Brno, CZ, 2005.
2. Cvrček D., Latislav R.: TCP - resetovací útok, In: DATAKON 2005, Proceedings of the Annual Database Conference, Brno, CZ, MUNI, 2005, p. 301-310, ISBN 8021038136
3. Cvrček D., Matyáš V. ml.: PIN (&Chip) or signature - beating or cheating?, In: SPW 05 Proceedings, Berlin, DE, Springer, 2005, p. 5, ISSN 0302-9743
4. Cvrček D., Švenda P.: Smart dust security - key infection revisited, In: STM 2005, Milano, 2005, p. 10-23, ISSN 1571-0661
5. Cvrček D.: RFID - přeceněné ambice?, In: SmartWorld 2005 - soubor prezentací, Zlín, CZ, 2005, p. 7
6. Bond M., Cvrček D., Murdoch S. J.: Reverse-engineering kryptografického modulu, In: Crypto-world, Vol. 2004, No. 9, Praha, CZ, p. 8-14, ISSN 1801-2140
7. Bond M., Cvrček D., Murdoch S. J.: Unwrapping the Chrysalis, In: Technical report, No. 592, Cambridge, GB, p. 15, ISSN 1476-2986
8. Cvrček D., Krhovják J., Matyáš V. ml.: Hardwarové bezpečnostní moduly - API a útoky, In: Europen, XXV. konference, sborník příspěvků, Plzeň, CZ, ECOM, 2004, p. 91-114, ISBN 80-86583-07-4
9. Cvrček D., Krhovják J., Matyáš V. ml.: Útoky a kryptografie v hardwarovém provedení, In: DSM Data Security Management, Vol. 2004, No. 5, CZ, p. 16-19, ISSN 1211-8737
10. Cvrček D., Matyáš V. ml.: Informační soukromí a jeho měřitelnost, In: DSM Data Security Management, Vol. 2004, No. 6, CZ, p. 10-14, ISSN 1211-8737
11. Cvrček D., Hanáček P.: Metodika pro hodnocení HW bezpečnosti čipových karet, CZ, 2003, p. 8
12. Cvrček D., Masařík K.: Private Key Infrastructure, In: Proceedings of Security and Protection of Information 2003, Brno, CZ, VABO, 2003, p. 15-24, ISBN 80-85960-50-8
13. Cvrček D., Šopík B., Špidla M.: Úvod do útoku na informační systémy, Brno, CZ, 2003, p. 1-85
14. Cvrček D.: Using Evidence for Trust Computation, In: Mikulášská kryptobesídka, Brno, CZ, ECOM, 2003, p. 12-20

15. Cvrček D., Hanáček P., Švéda P.: Návrh a realizace zařízení pro testování hardwarové bezpečnosti čipových karet, CZ, 2002, p. 46

16. Cvrček D.: Vytvoření lokální klíčové infrastruktury, In: Mikulášská kryptobesídka - sborník přednášek, Brno, CZ, ECOM, 2002, p. 19-25, ISBN 80-903083-2-5

17. Cvrček D., Matyáš V. ml.: PKI není všelék, In: DSM Data Security Management, Vol. 2001, No. 6, CZ, p. 26-29, ISSN 1211-8737

18. Cvrček D., Švéda P.: Hardwarové a softwarové řešení bezpečnosti, In: Mikulášská kryptobesídka, Brno, CZ, ECOM, 2001, p. 87-97, ISBN 80-903083-0-9

19. Cvrček D.: Active Authorization as High-level Control, In: Data And Applications Security, Amsterdam, NL, Kluwer, 2001, p. 339-345, ISBN 0-7923-7514-9

20. Cvrček D.: Authorization Model for Strongly Distributed Information Systems, Brno, CZ, 2001, p. 125

21. Cvrček D.: Real-World Problems of PKI Hierarchy, In: Proceedings of the SPI Conference, Brno, CZ, VABO, 2001, p. 39-46, ISBN 80-85960-28-1

22. Cvrček D.: Mandatory Access Control in Workflow Systems, In: Knowledge-based Software Engineering, Brno, CZ, IOS, 2000, p. 247-254, ISBN 1-58603-060-4

23. Cvrček D.: Access Control in Workflow Systems, In: MOSIS'99 Proceedings, Rožnov pod Radhoštěm, CZ, MARQ, 1999, p. 93-100, ISBN 80-85988-31-3

24. Cvrček D.: Active Authorization Model for Workflow System, Sborník prací studentů a doktorandů, Brno, CZ, FEI VUT, 1999, p. 73-74

25. Cvrček D.: Aktivní autorizační model, In: Sborník z letní školy Informační systémy a jejich aplikace 1999, Ruprechtov, CZ, FAST VUT, 1999, p. 46-51

26. Cvrček D.: Elektronická komunikace - hrozba nebo šance?, In: IT System, Vol. 1, No. 2, Brno, CZ, p. 15-18, ISSN 1212-4567

27. Cvrček D.: Elektronický obchod, In: IT System, Vol. 1, No. 5, Brno, CZ, p. 6-9, ISSN 1212-4567

28. Cvrček D.: Zákon o elektronickém obchodu, In: IT System, Vol. 1, No. 6, Brno, CZ, p. 47-51, ISSN 1212-4567

29. Cvrček D.: Access Control in Database Management Systems, In: DATASEM '98 sborník přednášek, Brno, CZ, 1998, p. 153-163

30. Cvrček D.: Problems of Modelling Access Control for Object Oriented Databases, In: ASIS 1998 proceedings, Krnov, CZ, MARQ, 1998, p. 21-26, ISBN 80-85988-27-5

B.3 List of Publications of Students Lead by the Author

1. Špidla, M.: Distribuované útoky na IS. Diploma thesis, Brno University of Technology, 2003.
2. Čončka, P.: Rozšíření systému bezpečné komunikace pro workflow systém. Diploma thesis, Brno University of Technology, 2002.
3. Burda, Z.: Grafické rozhraní pro konfiguraci firewallu. Diploma thesis, Brno University of Technology, 2003.
4. Doseděl, T.: Možnosti použití modelu secure v bezpečnostních aplikacích. Diploma thesis, Brno University of Technology, 2004.
5. Horák, V.: Analýza počítačové sítě a návrh optimální topologie. Diploma thesis, Brno University of Technology, 1999.
6. Hrabovský, D.: Grafické rozhraní pro konfiguraci bezpečné pošty. Diploma thesis, Brno University of Technology, 2003.
7. Šopík, B.: Bezpečnost na úrovni IPSec. Diploma thesis, Brno University of Technology, 2004.
8. Jalůvka, P.: Anonymizující sítě. Bachelor thesis, Brno University of Technology, 2004.
9. Král, R.: Vlastnosti prahových podpisových schémat. Diploma thesis, Brno University of Technology, 2002.
10. Kupčík, J.: Hw bezpečnost čipových karet - epp pro zařízení scsat02. Bachelor thesis, Brno University of Technology, 2004.
11. Kyánek, D.: Rozšíření autorizačního modelu workflow systému. Diploma thesis, Brno University of Technology, 2003.
12. Lodrová, D.: Kryptografie na začátku osmdesátých let. Bachelor thesis, Brno University of Technology, 2004.
13. Michel, R.: Distribuované útoky - teardrop. Diploma thesis, Brno University of Technology, 2004.
14. Miško, M.: Firewall s reputačním systémem. Bachelor thesis, Brno University of Technology, 2004.
15. Očenášek, P.: Verifikace bezpečnostních protokolů. Diploma thesis, Brno University of Technology, 2002.

16. Petráš, T.: Praktická bezpečnostní analýza DNS. Diploma thesis, Brno University of Technology, 2004.

17. Peňás, P.: Zvýšení důvěryhodnosti certifikátů veřejných klíčů. Diploma thesis, Brno University of Technology, 1998.

18. Popelka, T.: Řešení autentizace v lékařském informačním systému. Diploma thesis, Brno University of Technology, 2001.

19. Vejnárek, P.: Autentizace pomocí čipových karet v OS Unix. Diploma thesis, Brno University of Technology, 2003.

20. Černocký, P.: Rozšíření jádra workflow systému. Diploma thesis, Brno University of Technology, 2002.

Appendix C

Enclosed Papers and Articles

C-1 Combining Trust and Risk to Reduce the Cost of Attacks

C-2 Dynamics of Reputation

C-3 Using Evidence for Trust Computation

C-4 Evidence processing and privacy issues in evidence-based reputation systems

C-5 On the role of contextual information for privacy attacks and classification

C-6 Privacy - what do you mean?

C-7 Pseudonymity in the light of evidence-based trust