NORDSEC 2003

Proceedings of the Seventh Nordic Workshop on Secure IT Systems – Encouraging Cooperation

15th – 17th October 2003, Gjøvik , Norway

Sponsors

Microsoft Research

Morgenlandet

Norges forskningsråd
The Research Council of Norway

Gjøvik kommune

telenor

GJØVIK KUNNSKAPSPARK

KONTOR & DATA
inventum
IT-Senteret AS

NORDSEC 2003

Proceedings of the Seventh Nordic Workshop on Secure IT Systems – Encouraging Cooperation

15th – 17th October 2003, Gjøvik , Norway

Edited by:
Svein J. Knapskog
NTNU, Department of Telematics and Q2S, Trondheim, Norway

Published by:
Department of Telematics, O. S. Bragstadsplass 2, Trondheim, Norway.
Phone: +47 735 94324

ISBN 82-993980-4-5

Contents

	Page
Preface	v
NORDSEC2003 – Conference Committees	vi
Conference Programme	viii
Privacy and Unsolicited Commercial e-mail Andreas Jacobsson, Bengt Carlsson	1
A Framework for Enforcement of Privacy Policies Ragni R. Arnesen, Jerker Danielson	13
NetRAM: A Novel Approach for Network Security Risk Management Mohamed Hamdi, Jihene Krichene, Noureddine Boudriga, Mahmoud Tounsi	25
Trust Evaluation Based Security Solutions in Ad Hoc Networks Zheng Yan, Peng Zhang, Teemupekka Virtanen	37
Investigating Spyware on the Internet Johan Wieslander, Martin Boldt, Bengt Carlsson	51
Consolidation and Evaluation of IDS Taxonomies Magnus Almgren, Emilie Lundin, Erland Jonsson	57
A Wavelet Based Approach to Detect Computer Network Attacks Mohamed Hamdi, Noureddine Boudriga	71
On Visualising Intrusions Henrik Almegren, Ola Söderström, Erland Jonsson	83
Extending EMV to support Murabaha transactions Mansour Al Meaither	95
Implementing elliptic curve cryptosystems using Hesse curves over prime fields Terje Gjøsæter, Kjetil Haslum	109
Secure Storage for Mobile Terminals Jani Suomalainen, Aarne Rantala, Markku Kylänpää,Jarkko Tolvanen, Janne Mäntylä	117

SRBAC: A Spatial Role-Based Access Control Model for Mobile Systems Frode Hansen, Vladimir Oleschuk	129
An Access Control System for Business Processes for Web Services Hristo Koshutanski, Fabio Massacci	143
Decentralized Credentials Marius Gjerde, Stig F. Mjølsnes, Aslak Buan	151
Privacy-Preserving Spatially Aware Authentication Protocols: Analysis and Solutions Geir Køien, Vladimir Oleschuk	161
TECP-Tutorial Environment for Cryptographic Protocols Jelena Zaitseva, Jan Willemson, Jaanus Pöial	175
Improving the Gnutella protocol against poisoning Meelis Roos, Jan Willemson, Peeter Laud	185
On Server-Aided Computation for RSA Protocols with Private Key Splitting Anne-Maria Ernvall, Kaisa Nyberg	195
Author Index	207

NORDSEC2003

Preface

The NORDSEC Workshops started in 1996 with the aim to bring together researchers and practitioners within IT security in the Nordic countries. The first workshop was organized by Chalmers University of Technology, and since then, all Nordic countries have been hosting the workshop one or more times.

NORDSEC 2003 is organized by Gjøvik University College in Gjøvik Norway. The workshop is held back-to-back with the European security conference ESORICS2003, making it a viable option to attend both a major European scientific conference in the field of IT security and the NORDSEC Workshop in one and the same week.

In response to the call for papers for this year's workshop, 39 proposals were received, of which 19 were accepted for presentation at the workshop. The acceptance process is based on anonymous refereeing and subsequent ranking of the papers based on the grading given by normally three independent assessments. All proposals were received from the authors and submitted to the referees in an electronic format, and the referee process was conducted by e-mail, coordinated by the Chairman of the Programme Committee.

Unfortunately, one accepted paper has been withdrawn due to lack of funding for the authors of the paper – nevertheless, it is the view of this year's Chairman of the Programme Committee that the remaining 18 papers, together with two invited talks, will provide a tantalising menu for you who attend the workshop, giving you new insight and inspiration to probe further into this exciting field of research and development on your own in the years to come. Presentations at the workshop are from such a variation of areas as user privacy, surveillance on the Internet, security relevant risk management, security solutions for Ad Hoc networks, Intrusion Detection and Prevention, security support for Murabaha transactions, elliptic curve cryptography, cryptographic protocols, secure storage and role based access control. As always, the working language of the workshop is English, and researchers from 9 countries have contributed to the programme, making it even more diversified than a pure Nordic approach might have been.

It has been an honour and a privilege to organize the scientific programme for the NORDSEC2003 Workshop, and all members of the Programme Committee deserve a hearty thanks for the effort in reviewing papers and thus contributing to a successful event in Gjøvik. A special recognition goes to Tønnes Brekne, Erland Jonsson and Simone Fischer-Hübner who all have given me pivotal support in the final selection of papers for the workshop.

NTNU September 2003

Svein Johan Knapskog, Programme Committee Chairman 2003.

NORDSEC 2003 - Conference Committees

Steering Committee:

Mads Dam
Swedish Institute of Computer Science, Sweden

Ùlfar Erlingsson
Green Border Technologies, USA

Simone Fischer-Huebner
Karlstad University, Sweden

Viiveke Fåk
Linköpings universitet, Sweden

Erland Jonsson
Chalmers University of Technology, Sweden

Arto Karila
StratosVentures, Finland

Svein Knapskog
NTNU, Norway

Helger Lipmaa
Helsinki University of Technology, Finland

Hanne Riis Nielson
Technical University of Denmark, Denmark

Kaisa Nyberg
Nokia Research Center, Finland

Nahid Shahmehri
Linköpings universitet, Sweden

Louise Yngström
Stockholm University, Sweden

Teemupekka Virtanen
Helsinki University of Technology, Finland

Programme Committee

Mads Dam
Swedish Institute of Computer Science, Sweden

Ùlfar Erlingsson
Green Border Technologies, USA

Simone Fischer-Huebner
Karlstad University, Sweden

Viiveke Fåk
Linköpings universitet, Sweden

Erland Jonsson
Chalmers University of Technology, Sweden

Arto Karila
StratosVentures, Finland

Svein J. Knapskog
NTNU, Norway

Helger Lipmaa
Helsinki University of Technology , Finland

Hanne Riis Nielson
Technical University of Denmark, Denmark

Kaisa Nyberg
Nokia Research Center, Finland

Nahid Shahmehri
Linköpings universitet, Sweden

Louise Yngström
Stockholm University, Sweden

Teemupekka Virtanen
Helsinki University of Techn., Finland

Jon Ølnes
PKI Consulting Services AS

Knut Johannesen
Telenor

Tønnes Brekne
Center of Excellence - Quantifiable Quality of Service, NTNU

Stig F. Mjolsnes
NTNU, Institutt for telematikk

Chair: Svein J. Knapskog, NTNU, Norway

V. Oleshchuk
Høgskolen i Agder, Avdeling for teknologi

Einar Snekkenes
Høgskolen i Gjøvik

Jon Bing
Universitetet i Oslo, Institutt for informatikk

Audun Jøsang
DSTC, Queensland University of Technology, Brisbane

Anniken Seip
Statskonsult

Tage Stabell-Kulø
Universitetet i Tromsø, Institutt for informatikk

Øyvind Eilertsen
SIS, Trondheim

Johs Hansen Hammer
Skatteetaten, Oslo

Organizing Committee

Birgith Børthus, HiG
Hilding Sponberg, HiG
Rigmor Øvstetun, HiG
Fred Johansen, HiG
Svein Pettersen, HiG

Chair : Birgith Børthus, HiG

NORDSEC 2003

Programme

Wednesday, October 15th 2003:		
12:40 - 13:40	Lunch (optional)	
13:40 - 14:45	Registration, Coffee	
14:45 - 15:00	Opening speech	
Session 1: Invited talk Session chair: Svein J. Knapskog, NTNU		
15:00 - 15:45	Invited speaker: Eivind Jahren, Avdelingsdirektør, seksjon for IT-sikkerhet og infrastruktur, NHD	Information Security – Initiatives and Challenges
15:45 - 16:00	**Coffee break**	
Session 2: Privacy Session chair: Erland Jonsson, Chalmers University of Technology		
16:00 - 16:30	Andreas Jacobsson, Bengt Carlsson	Privacy and Unsolicited Commercial e-mail
16:30 - 17:00	Ragni R. Arnesen, Jerker Danielson	A Framework for Enforcement of Privacy Policies
Thursday, October 16th 2003:		
Session 3: Risk and Evaluation Session chair: Ulfar Erlingsson, Green Border Technologies		
09:00 - 09:30	Mohamed Hamdi, Jihene Krichene, Noureddine Boudriga, Mahmoud Tounsi	NetRAM: A Novel Approach for Network Security Risk Management
09:30 - 10:00	Zheng Yan, Peng Zhang, Teemupekka Virtanen	Trust Evaluation Based Security Solutions in Ad Hoc Networks
10:00 - 10:30	Johan Wieslander, Martin Boldt, Bengt Carlsson	Investigating Spyware on the Internet
10:30 - 11:00	**Coffee break**	

Session 4: Intrusions and Intrusion detection Session chair: Karin Sallhammar, NTNU/Q2S		
11:00 - 11:30	Magnus Almgren, Emilie Lundin, Erland Jonsson	Consolidation and Evaluation of IDS Taxonomies
11:30 - 12:00	Mohamed Hamdi, Noureddine Boudriga	A Wavelet Based Approach to Detect Computer Network Attacks
12:00 - 12:30	Henrik Almegren, Ola Söderström, Erland Jonsson	On Visualising Intrusions
12:30 - 13:30	**Lunch**	
Session 5: Miscellaneous Session chair: Ursula Holmstrøm, HUT		
13:30 - 14:00	Mansour Al Meaither, Chris J. Mitchell	Extending EMV to support Murabaha transactions
14:00 - 14:30	Terje Gjøsæter, Kjetil Haslum	Implementing elliptic curve cryptosystems using Hesse curves over prime fields
14:30 - 15:00	**Coffee break**	
Session 6: Access Control Session chair: Emilie Lundin, Chalmers University of Technology		
15:00 - 15:30	Jani Suomalainen, Aarne Rantala, Markku Kylänpää, Jarkko Tolvanen, Janne Mäntylä	Secure Storage for Mobile Terminals
15:30 - 16:00	Frode Hansen, Vladimir Oleschuk	SRBAC: A Spatial Role-Based Access Control Model for Mobile Systems
16:00 - 6:30	Hristo Koshutanski, Fabio Massacci	An Access Control System for Business Processes for Web Services
19:00 - ?	**Dinner in a cavern on Ice**	

Friday, October 17th 2003:		
Session 7: Authentication Session chair: Kaisa Nyberg, Nokia Research		
09:00 - 09:30	Marius Gjerde, Stig F. Mjølsnes, Aslak Buan	Decentralized Credentials
09:30 - 10:00	Geir Køien, Vladimir Oleschuk	Privacy-Preserving Spatially Aware Authentication Protocols: Analysis and Solutions
10:00 - 10:45	Invited speaker: Håkon Styri	"Are real life stories unreal?"
10:45 - 11:15	**Coffee break**	
Session 8: Protocols and Protocol Aids Session chair: Helger Lipmaa, HUT		
11:15 - 11:45	Jelena Zaitseva, Jan Willemson, Jaanus Pöial	TECP-Tutorial Environment for Cryptographic Protocols
11:45 - 12:15	Meelis Roos, Jan Willemson, Peeter Laud	Improving the Gnutella protocol against poisoning
12:15 - 12:45	Anne-Maria Ernvall, Kaisa Nyberg	On Server-Aided Computation for RSA Protocols with Private Key Splitting
12:45 - 13:00	**Closing**	
13:00 - 14:00	**Lunch**	

Privacy and Unsolicited Commercial E-Mail

Andreas Jacobsson & Bengt Carlsson

Department of Software Engineering and Computer Science
Blekinge Institute of Technology, S-372 25 Ronneby, SWEDEN

{andreas.jacobsson;bengt.carlsson}@bth.se

Abstract. In our society, the Internet becomes more and more indispensable, and the issue of personal information between consumers and businesses is recognised as critically important in building secure and efficient systems on the Internet. With respect to handling personal information, consumers generally want their privacy to be protected, but businesses need reliable personal information and an access channel to reach consumers for e-commerce. Undoubtedly, these demands must be satisfied to establish sound e-commerce. However, with the technologies available today it is reasonably easy for companies to gather information about consumers in order to make personalised offers through e-mail. There is a fine line between collecting personal information to make customised offers that users or customers regard as useful information and what is an intrusion to personal privacy. In this paper we discuss how consumer privacy is affected by unsolicited e-mail messages sent with a commercial purpose (spam). We found that, albeit most of the investigated web sites behaved well, a small fraction generated a large number of spam. This *tragedy of the commons* problem is discussed from an economical, ethical and legislative point of view.

Keywords: privacy, spam, e-commerce, EU-Directive

1 Introduction

Today, many companies provide personalised services and offers to their customers by utilising customer preference information and/or behavioural information. Web sites can collect each user's browsing log, and by way of, for example cookies, they can display personalised pages for their customers. These customised services certainly make life more convenient. Therefore, it is reasonable to allow a business to provide such services. However, whether a company is successful or not, often depends on having more information about consumers than the competitors do [15]. In effect, this means that a business must gather as much information about consumers as possible, which on the other hand increases the risk of unintentional, or intentional, infringement of consumers' privacy. A consequence of continued abuse of consumers' privacy is that customers may be uneasy about sharing personal information with companies, and possibly also sceptical to the idea of e-commerce. Human society on the Internet will only thrive if the privacy rights of individuals are balanced with the benefits associated with the flow of personal information [6] [15].

We use the definition of privacy that was first proposed by Samuel D. Warren and Louis D. Brandeis in their article "The Right to Privacy" [12], and define privacy as "*the right to be let alone*". The advantage of this definition is that it is widely accepted in society, and thus easily can be adopted to the Internet setting. The extraction of the definition is that users can specify what information should be disclosed to whom, when and for what purpose. Also, that they are guaranteed that the information will be treated accordingly. In effect, this correlates well with the most commonly used definition of privacy by Alan Westin:

> "Privacy is the claim of individuals, groups and institutions to determine for themselves, when, how and to what extent information about them is communicated to others." [13]

In general, the explicit demands from consumers and businesses regarding commerce based on the customers' personal information do not always fit well with having the right to be let alone. Consumers want to be able to control their privacy and still get the best personalised services available. Businesses, on the other hand, need reliable personal information about customers, and also an access channel to bring the best service possible to the appropriate consumer. From this discussion we can extract three requirements for sound e-commerce based on personal information [15]:

1. **Privacy control:** Consumers should be provided with a means to decide what, when, and for what purpose their personal information is used.
2. **Data reliability:** Businesses should be provided with reliable consumer information.
3. **Consumer accessibility:** Businesses should be provided with a means to access targeted consumers directly.

As can be seen later in this paper, these requirements are not fully met when it comes to e-businesses informing users about offers. In the next sections, we will conduct a discussion regarding privacy requirements and unsolicited commercial e-mail messages. Further on, there will be an exploration in the legal aspects concerning the European Directive on Privacy and Electronic Communications [8]. We also add some empirical perspectives where we investigate whether the mentioned requirements are met or not.

2 E-Mail Marketing

2.1 Basic Idea of e-Commerce

In order to set the scene for the occurrence for e-mail marketing, a summary of the ideas behind e-commerce is hereby given. There can be a number of reasons explaining commerce over the Internet. Although these are not new, some important aspects are [4]:

- **Availability and supply:** In theory, Internet shops are independent of time and place, and stay open 24 hours a day, every day. Also, due to, for example reduction of costs, a much broader selection of goods and services can be offered, than by way of a traditional physical store.
- **Reduction of intermediaries:** The product or service can be offered directly to the customer. This reduces the number of intermediates in the distribution chain, which results in a lower total transaction cost and a lower price than in a traditional, physical store.
- **Customer relationships:** Based on the information collected about customers, it is possible to tailor personalised information to the customers in a much larger range

than with the physical commerce medium. A common method to distribute offers is by way of e-mail messages.

As stated, one of the foundations of e-commerce is customer information, and in principle, the company that has the most information about its customers is generally the most successful one [15]. There are two ways for a company to get hold of information about customers, either the customers freely give out information about themselves, or the companies gather information. Often, though, companies use a combination. With respect to the three requirements for sound e-commerce, there is a bit of a dilemma. As companies need reliable information about customers, the customers must agree to give out personal information. On the other hand, this is often misused, and thus resulting in reluctance from customers to provide such information. One example is if users' inboxes are flooded with unsolicited offers.

In order to ensure sound e-commerce, both consumer privacy and business needs must be taken into consideration. On the requirements for sound e-commerce, the first requirement, privacy control, concerns the use and control of personal information. Not only should consumers be able to decide what, when, and for what purpose their personal information is used, they should also be provided, by the company, with a means to make such decisions. The second requirement, on the other hand, is directed to the customer, and states that if the first requirement is met, then the customers should provide companies with reliable consumer information. Also, regarding the third requirement, businesses should get an access-channel (e.g., e-mail messaging) to reach targeted consumers directly. Although some e-companies might prefer to access customers in other ways than via e-mail, marketing and preserving customer relations by way of the e-mail media is the most used form [6].

2.2 Economical Aspects of Spam

Unsolicited commercial e-mail, also known as spam or junk e-mail, has increased dramatically in number over the last years. In the beginning of 2002, estimations showed that one out of twelve e-mail messages fit the description for spam. During that year the number of spam rose and reached an average frequency where one out of every third e-mail was a junk message. An assessment made this year by Ferris Research estimated that it takes the average Internet user about 4.4 seconds to handle a spam, and also that approximately 20 billion such e-mail messages are sent every day from databases holding up to 200 million e-mail addresses [1]. The accumulated time for handling spam messages approaches 25 million hours per day. On a personal level, at least a few minutes daily is spent on deleting unsolicited e-mail.

There are a number of reasons trying to explain the vast increase of spam. However, one main aspect argued here is that marketing by spam is cheap (in comparison to other mass marketing methods).

For an e-commerce company, there are various costs for dealing with computer based information. One useful and general model is dividing production and distribution of digital information into different costs [3]. This model can also be applied on spam. When it comes to spamming or e-mail marketing, there are, typically, four costs that can be associated with spam [3]. The first one is *first-copy sunk cost*, which is the cost for producing the first, original e-mail copy of the offer. This cost cannot be recovered if the offer flops, however, the cost for the first e-mail copy is generally insignificant. Then, there is the *reproduction cost*,

or *variable cost*, which refers to the cost of producing an additional copy. Normally, this does not increase, even if a great many copies are made. The *distribution cost* refers to the cost of distributing a message from the sender to the recipient. Being that the e-mail medium is quick, technically available and economically favourable, the distribution cost per sent e-mail is very low. Finally, the fourth cost is the *transaction cost*, that is the amount of money the sender might have to pay to get hold of e-mail addresses (for example, e-mail lists or software packages programmed to sniff e-mail addresses), which is also relatively low.

On the sender side, the costs for distributing spam are almost non-existing. However, on the recipient side, the cost of receiving a single spam is small, but the cost of receiving many spam messages can be considerable. Even though users are not paying per-message or per-minute fees (unless they are connected to a modem), spam may be expensive in terms of time. Users may waste minutes or hours transferring unwanted messages from their ISPs to their personal computers, sorting through the messages, and deleting them.

Furthermore, spam places a burden on ISPs, requiring them to spend time and money on implementing filters, responding to subscriber complaints, and increasing their e-mail system capacity more frequently than would otherwise be necessary. Also, large amounts of junk e-mail messages, will in turn lead to an increase of network load and ultimately traffic jams over the Internet. In addition, ISPs are burdened by spam not destined for their own subscribers, but relayed through their system by spammers who are attempting to hide the true origin of their messages [9].

2.3 The Contents of Spam

In two experiments, performed during the spring of 2003, the occurrence and contents of spam, as well as their impact on personal privacy were investigated.

In a master's thesis experiment, 30 different well-known and highly visited web sites were selected as a test sample [2]. The Internet sites were chosen equally from the United States and the European Union. Two fake user accounts, Adam and Bill, representing average Internet users from the US and EU respectively, were set up to test the spam message results from signing up to the web sites. By signing up (i.e., giving out first and second name as well as e-mail address) for the web sites e-mailing lists, the intention was to examine if personal information was spread to third parties and/or if it generated spam. During each session Adam and Bill registered numbered e-mail addresses, i.e., adam1@ourdomain, adam2@ourdomain, etc., to clarify which e-mail address was added to which web site. It turned out that only one site, a respected music newsletter service (Music.com), out of 30 web pages that were visited, generated spam. Furthermore, it was only Adam, the pretended American visitor, who received spam.

By the second day of the test, Adam got his first spam, and after that it increased during the forthcoming two weeks. In all, 468 spam messages were received over a five-week period of analysis (see Figure 1). The first spam offered an insecure Gold Card and was sent from Arbango.com. The company claimed that Adam had requested to receive special promotional messages from Arbango.com, however such a request was never made. This kind of claim has also occurred frequently in other spam received, as well as statements that the e-mail address was passed to the spammer by an alleged friend. In some of the e-mail messages, Adam was greeted by his full name (Adam Smith), though the e-mail address did not

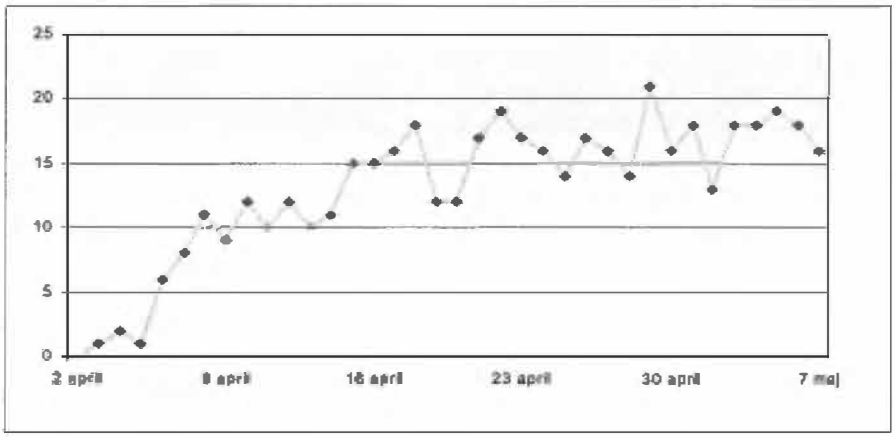


Figure 1 Number of spam day by day [2].

reveal his last name. So this information must have come from the Music.com database. However, there was no correlation between the contents of spam and the original music site when examining the received spam messages. In all, spam messages from 15 different categories were received, see Figure 2 for results.

The categories that generated the most spam were *free offers* and *financial advertisements*, such as offers to *loan* and *make money*. This finding somewhat correlates with what Cranor and LaMacchia found in their article "Spam!", which was published in the journal *Communications of the ACM*. Based on studies of 400 spam messages, they concluded that *money making opportunities* was the most common advertisement of spam, on second place was a category called *other products and services* [9]. This category included phone services, vacation packages, nutritional supplements, weight loss products and on-line newsletters. In the experiment, the category, which is called *free offers* also matches the result from Cranor and LaMacchia (here called *other products and services*), because the advertisements in both categories are of the same kind.

In a second experiment, we investigated if unsubscribing to spam e-mail lists generated new spam. In all, 219 unique spam messages were analysed and 182 of them, or 83%, allowed the user to unsubscribe. By transporting the spam messages to a newly configured e-mail account with a "clean" environment it was possible to investigate the impact of unsubscribing. During a period of four weeks, we did not receive a single spam in return.

Cranor and LaMacchia also noted that only 36% of the spam messages contained instructions for how to be removed from the mailing list [9]. What perhaps revealed the nature of these messages the most was the fact that less than 10 percent supplied name, postal address, phone number, and e-mail address of the sender. Also, most of the spam messages offer the recipient the possibility to delete his or her e-mail address from the mailing list using opt-out lists. Even though it is commonly known that the recipient should not reply to the sender of the spam or to sign any opt-out lists (then, the spammer will know that the e-mail address is active and possibly spam it even harder). However, in our investigation we could not find such a correlation, that is; the unsubscription of spam did not result in getting new spam.

There might be difficulties in making generalisations from the results in the experiments, due to a limited number in sample. Therefore, we present an additional perspective on empirical work about spam.

In the summer of 2002, the Center for Democracy & Technology (CDT), embarked on a project to attempt to determine the source of spam [14]. CDT set up hundreds of different e-mail addresses, used them for a single purpose, and then waited half a year to see what kind of e-mail messages those addresses were receiving. During six months, 250 e-mail addresses received 10.000 e-mail messages, of which nearly 9.000 were unsolicited commercial e-mail offers, that is, spam. However, this study's main finding was not the number of spam attracted, but the investigation of the different ways that e-mail addresses got spammed. CDT's view is that this mainly depends on where the e-mail addresses have been used. They found that e-mail addresses posted in clear text on public web sites or in news groups attracted the most spam. In contrast, they also noted that companies that offered users a choice about receiving commercial e-mail messages respected the decision of the user, and accepted their request for privacy. Only a very small fraction of the Internet sites, possibly those who did not protect the e-mail addresses, resulted in spam activity. CDT's conclusion was not that the company itself leaked the information, but that spammers took hold of addresses that were available in clear text (i.e., not adequately protected) on the public Internet or in news groups. CDT discovered that spammers have software programs (so called robots or spiders) that are programmed to search for e-mail addresses. In effect, 97% of the total amount of spam, were received in this way.

In light of the CDT results, a conclusion from our experiment is that one site in thirty rendering in spam activity might be a representative finding. CDT arguing that most Internet services respect the privacy choices of the users and leave them alone, support this view. In our experiment there can be two perspectives on explaining how spammers got hold of Adam's e-mail address and personal information. The first explanation is in accordance with the CDT-survey, and suggests that the spam messages generated were not a result of commercial intent from the original music site, but one of poor security measures on their behalf. Being reluctant in protecting users' personal information might, such as in this case,

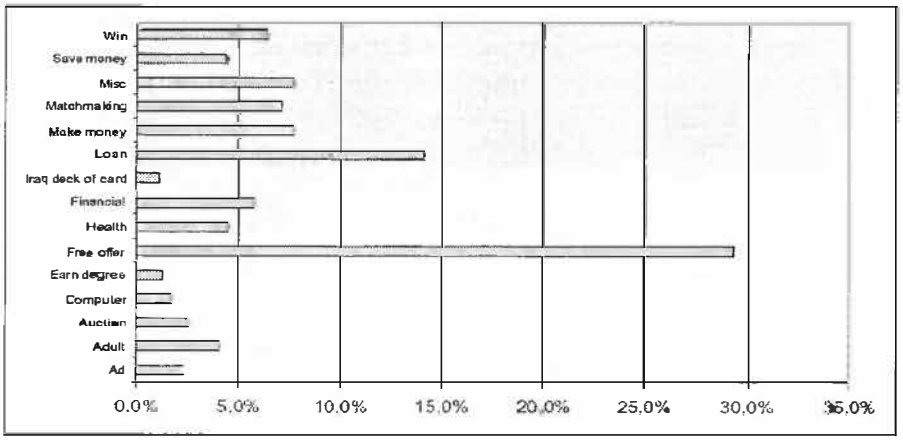


Figure 2 Spam categories [2].

have led to spammers taking hold of Adam's user information, and thus spammed him with 468 e-mail messages in five weeks, containing offers of anything but music. The second explanation has to do with the original Music.com site. Due to the nature of that web service, it seems that the site is merely a "fly paper", set up with one single purpose, namely to attract the e-mail addresses of gullible consumers. There are mainly four aspects supporting this view; (1.) the only service the site supplies is a poorly written newsletter arriving at arbitrary times, (2.) this only service is for free, meaning that the site owners must somehow gain a return (user e-mail addresses) on their investment (web site and newsletter service), (3.) the user interface very much has the resemblance of a database interface, and (4.) the URL name (Music.com) is probably one of the first web address suggestions that users type into the address bar in search of music on the Internet. Thus, Music.com easily attracts users, and in return of access to user information they offer a music newsletter.

Independently of what explanation one believes in, the empirical investigations have shown that not all companies respect privacy, because they do not provide users with a means to decide what, when and for which purpose their personal information is being used. One company in 30 that ignores personal privacy rights may be sufficient enough to damage the whole Internet community.

3 Privacy and Spam

3.1 On the Principles of Privacy

Privacy principles, such as the ones below, are needed in order to guarantee privacy when personal data is collected or processed. These principles are frequently used when designing national privacy laws, codes of conduct, codes of ethics of computer societies, as well as international privacy guidelines or directives. Most western data protection acts, e.g., the EU-Directive, and the OECD-Guidelines, are founded on these privacy requirements [7]:

- **Principle of lawfulness and fairness:** Personal data should be collected and processed in a fair and lawful way;
- **Principle of the purpose specification and purpose binding (also called purpose limitation):** The purposes for which personal data is collected and processed should be specified and legitimate. The subsequent use of personal data is limited to those specified purposes, unless there is an informed consent by the data subject;
- **Principle of necessity of data collection and processing:** The collection and processing of personal data should only be allowed, if it is necessary for the tasks falling within the responsibility of the data processing agency;
- **Information, notification and access rights of the data subjects:** Data subjects have the right to information, to notification and the right to correction, erasure or blocking of incorrect or illegally stored data. These rights should not be excluded or restricted by a legal transaction. Information and notification rights help to provide transparency of data processing;
- **Principle of security and accuracy:** Appropriate technical and organisational security mechanisms have to be taken to guarantee the confidentiality, integrity, and availability of personal data. Personal data has to be kept accurate, relevant and up to date;

- **Supervision and sanctions:** An independent data protection authority (also called supervisory authority, data protection commissioner or ombudsman) has to be designated and should be responsible for supervising the observance of privacy provisions. In the event of violation of the provisions of privacy legislation, criminal or other penalties should be envisaged.

These principles are generally formulated and therefore hard to apply to a narrow topic such as spam and spamming. Thus, an exploration in other legal frameworks is needed in order to fit the description of spam. In Article 13 of the European Union's Directive concerning the processing of personal data and the protection of privacy in the electronic communications sector, relevant privacy principles are addressed and stated. Also, by way of the empirical investigations, we will be able to see how they are interpreted and followed.

3.2 EU-Directive and Spam

When it comes to the processing of personal data, the EU-Directive ensures that the rights and freedoms of natural persons are preserved [8]. Article 13 of the Directive is devoted to unsolicited communications, such as spam messaging. Below we present a discussion concerning that article and the concept of spam.

3.2.1 Permission Marketing

In Article 13, paragraph 1, of the EU-Directive it is stated that unsolicited communications such as e-mail with the purpose of direct marketing may only be allowed in respect of subscribers who have given their prior consent [8].

E-mail marketing is a highly debated topic. Typically, a sharp line divides spam from, so called, real or serious e-mail marketing. Solicited e-mail can be defined as a message that the customer asked for, or agreed to receive [6]. Spam is unsolicited, which means that the recipient never asked for it. This is different from when you agree to receive commercial e-mail messages (also called opt-in to e-mail), according to authors Sterne and Priore, who also state that serious e-marketers nowadays apply opt-in when sending commercial e-mail messages [6].

In the spam experiment presented in Section 2, five weeks of ordinary Internet browsing produced 468 spam messages from one Internet site alone. The personal information that was registered on this web page ended up in a database managed by another site, which in turn resulted in that this particular e-mail address landed on several opt-in mailing lists. Hence 468 unsolicited messages clogged the inbox in five weeks.

The EU-Directive only allows direct marketing when subscribers have given their prior consent. In the spam experiment, such was the case in one respect. There was a consent to opt-in on the first Internet site, and therefore an acceptance to receive offers via e-mail from that particular company. On the other hand, there was no consent to having the e-mail addresses forwarded to other sites, that thereafter redistributed them to spam-related services.

In the EU-Directive it is explicitly stated that the sender must have permission from the receiver in order to convey offers and news. Given the background to e-mail marketing (see Section 2) and the spam experiment, we can conclude that not every company operating on the Internet manage personal information about customers according to the Directive. One argument is that the Directive only applies for companies within the European Union, since it is a European Directive. On the other hand, similar, but not identical, directives exist for

Canada and the US [11]. Also, it is contradicted by Sterne and Priore, who state that opt-in, or permission e-mail marketing is practised by all serious companies [6]. Thus, companies that want to appear as serious should not send unsolicited messages.

In light of this, one severe risk a spammer is facing is getting a bad reputation. Good reputation is the result of the fulfilling of customers' expectations in quality of product and/or service over time [10]. Often, that leads to customers returning to the company. This is something that e-commerce companies cannot overlook. However, according to Choi et al., a reputation is important only if a company aims at staying on the market for a long time [10]. One addition to explain the vast increase of spam is that for some companies (and/or persons) sending spam is their very core business (which most likely is the case in the CDT-survey, for instance). Another explanation might be that the companies behind spam really have no interest in remaining on the market for a long time, their primary object is to make a quick profit by selling a certain product or service. Given the economical analysis of spam messaging (see Section 2.2), it is fair to conclude that spammers get value for money even if only a very small percentage of a great many spam messages result in closed deals, or if their marketing campaigns and e-mail offers render maximum exposure at minimum cost.

3.2.2 The Right to Be Let Alone

In the EU-Directive it is evident that customers have a right to be left alone by companies that collect personal information about them. The second paragraph of Article 13, states that:

> "... legal person may use these electronic contact details for direct marketing of its own similar products or services provided that customers clearly and distinctly are given the opportunity to object, free of charge and in an easy manner, to such use of electronic contact details when they are collected and on the occasion of each message in case the customer has not initially refused such use." [8]

When it comes to using contact details for direct marketing purposes, it is allowed only if it concerns the company's own products or services. Obviously, so was not the case in the spam experiment. The 468 spam messages that were received contained a broad spectrum of offers, such as financial offers, adult entertainment, health care products, etc. Though, it was a music company that got the e-mail address. Of course, we cannot be certain that the music site was actively contributing to the spamming of Adam's address, however, they are still responsible for maintaining Adam's personal information in a secure manner (something they failed to do).

A second requirement noted in the Directive is that customers must be given the opportunity to object, free at charge, to the use of electronic contact details. Also, in the spam experiment, there was little chance to object to the use of the e-mail address and other personal information. In reality, this is no surprise, every day most people receive several spam messages that they have not agreed to subscribe to. Should one try to object, a common argument is that the spammers may take the e-mail address off one list, just to put it onto another. Although, one of our investigations showed that no such correlation could be found.

3.2.3 Controlling Spammers

As can be noted in the EU-Directive, member states have an explicit responsibility in the matter of spam. In the third paragraph of Article 13 it is stated that:

> "Member states shall take appropriate measures to ensure that, free of charge, unsolicited communications for purposes of direct marketing,..., are not allowed either without the consent of the subscribers concerned or in respect of subscribers who do not wish to receive these communications, the choice between these options to be determined by national legislation." [8]

In this respect it is up to the member states to find and take appropriate measures towards spamming. On the other hand, as can be seen both in our spam experiment (see Section 2) and the CDT survey [14], as well as in the spam analysis by Cranor and LaMacchia [9], this paragraph of the Directive has not yet had any effect on the Internet community. However, the EU-Directive is quite new (2002) and has perhaps not been around long enough to be carried into effect.

4 Aspects Concerning Privacy and Spam

4.1 The Effects on e-Commerce and Privacy

In the above discussion we can conclude that there are three requirements for sound e-commerce, of which the first one is critically important. Consumers should be provided with a means to decide what, when, and for what purpose their personal information is being used. Without such a control of personal privacy, consumers will hardly provide companies with reliable personal data, or not provide them with any data at all. If so, companies will most likely try to get consumer information elsewhere, so that they can continue to direct-access customers with personalised offers (e.g., via e-mail).

The experiments discussed in this paper have showed that it is difficult to tell to what extent consumers' personal privacy is ensured. There are some efforts on the regulatory area (e.g., the EU-Directive), but even though it is carried out and in progress it has not yet had any real effect on the occurrence of spam. However, it might be a bit too early to assume that the Directive has no impact, given that it has only been in use since 2002. In addition, our investigations suggest that most serious companies actually do not send offers unless they have the users' permission. Research indicates that most e-companies cease to send commercial messages when being unsubscribed to. Given this, the notion of the right to be let alone seems to be not so far away. On the other hand, statistics concerning the occurrence of spam say otherwise. Over the last couple of years the amount of spam has augmented, and a statistical institute [1] predicts that they will continue to increase in number over the next years.

In the spam experiment, privacy policies for every visited site were collected for a later evaluation. From this point of view the information collection was mentioned in the fine print of the privacy policy. The spam list generated was selective, an American address was included but not a Swedish one. The spam messages received have general contents, not connected to the music site.

There is a fine line between what users or customers regard as useful information and what is intrusion to personal privacy. One thought is that the more personalised the offers are,

the more likely users are to regard them as privacy invaders. If so, what happens when offers arrive to end users in such an extent that they hardly are able to distinguish personal messages, and possibly serious offers, from all the offers. So, there is a great risk for the success of e-commerce if the volume of unsolicited e-mail messages continue to grow without discrimination.

Even though empirical result suggests that only one Internet site in thirty render in spam, this might be enough to have a negative impact on the Internet and e-commerce society as a whole. Our view is that this is a case of the tragedy of the commons.

4.2 The Tragedy of the Commons

If and when there is a conflict of interest between a single actor and the whole community there may be a tragedy of the commons situation [5]. This situation arises when the benefit of a single actor exceeds the benefit of belonging to a community [5].

In our case, if a single company benefits from "littering" the Internet with spam, this actor will do so independently of the consequences for the whole system. Especially, considering that the costs for producing and sending spam messages are low (as are the initial investments in computers and software). A spammer can send millions of messages a day with minimal work. With such low expenses, mass mailers can recoup their costs even if only a tiny fraction of the messages they send result in purchases.

From an e-commerce perspective, this tragedy of the commons situation might render in customers and users stop paying attention to advertisements and personalised offers sent over the Internet, which would be somewhat of a disaster for the idea of e-commerce. Also, should companies neglect or overlook user privacy that might lead to the failure of an efficient and secure Internet. Privacy rights of individuals must be balanced with the benefits from the flow of personal information if human activity is to thrive on the Internet.

The legislative control of spamming occurs as a tragedy of the commons problem, when the benefit for any individual to break the rules exceeds the legal punishment imposed by the community. This is characterised by the difficulties to control spammers. With respect to different national legislation, and in combination with a judicial "grey area", this aggravates the tragedy of the commons.

Albeit, the investigations show that most companies behave well, no spam are generated after giving away personal information to commercial web sites and no new spam are generated after unsubscription. The only exception, a respected music site, alone generated a lot of spam each day. This is the tragedy of the commons.

5 Conclusions

This paper investigates and discusses how consumer privacy is affected by unsolicited e-mail messages sent with a commercial purpose, and how e-commerce companies' access to consumers may decrease depending on treatment of privacy issues when it comes to unsolicited commercial e-mailing. The empirical surveys show that most companies behave well; no spam messages are generated after giving away personal information to commercial web sites and no new spam are generated after unsubscription. The only exception, accidentally or by purpose, generated a lot of spam messages each day.

The conflict of interest between a single actor, benefiting from the results of a whole community, and the e-commerce society is described as the tragedy of the commons. Here, this sole actor may risk consumer accessibility, for considerable parts of the e-commerce society, to be able to mass-market commercial offers. These problems are discussed from an economical, ethical and legislative point of view.

References

[1] Ferris Research, http://www.ferris.com/, 2003-09-25

[2] Gunnarsson, A. and Ekberg, S., "Invasion of Privacy", Master Thesis, Blekinge Institute of Technology, 2003

[3] Shapiro, C., and Varian, H., "Information Rules", HBS Press, USA, 1999

[4] Kelly, K., "New Rules for the New Economy", Viking Penguin, USA, 1998

[5] Hardin, G., "The tragedy of the commons", Science vol. 162 pp. 1243-1248, 1968

[6] Sterne, J., and Priore A., "E-Mail Marketing - Using E-Mail to Reach Your Target Audience and Build Customer Relationships", John Wiley & Sons, Inc., USA, 2000

[7] Fischer-Hübner, S., "IT-Security and Privacy - Design and Use of Privacy-Enhancing Security Mechanisms", Springer-Verlag, Lecture Notes in Computer Science, vol. 1958, Germany, 2001

[8] Directive 2002/58/EC of the European Parliament and of the council of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector (Directive on privacy and electronic communications), 2002, http://www.cdt.org/privacy/guide/protect/telecom-priv02.pdf, 2003-09-25

[9] Cranor, L.F., and LaMAcchia, B.A., "Spam!", 1998, Communications of the ACM, Vol. 41, No. 8 (Aug. 1998), Pages 74-83, 1998, Definitive version: http://lorrie.cranor.org/pubs/spam/spam.html, 2003-09-25

[10] Choi S-Y., Stahl D.O., and Winston A.B., "The Economics of Electronic Commerce", Macmillan Technical Publishing, USA, 1997

[11] Rotenberg, M., "The Privacy Law Source Book 2002", EPIC Publications, USA, 2002

[12] Warren, Samuel D., and Brandeis, Louis D., "The Right to Privacy", Harvard Law Review, 1890-91, No.5, pp.193-220

[13] Westin, Alan, "Privacy and Freedom", 1987, New York

[14] Center for Democracy & Technology, "Why Am I Getting All This Spam? Unsolicited Commercial E-Mail Research Six Month Report", March 2003, http://www.cdt.org/speech/spam/030319spamreport.shtml, 2003-09-25

[15] Otsuka, Takuya, and Onozawa, Akira, "Personal Information Market: Toward a Secure and Efficient Trade of Privacy", Springer-Verlag, Lecture Notes in Computer Science, vol. 2105, Germany, 2001

A Framework for Enforcement of Privacy Policies[1]

Ragni Ryvold Arnesen and Jerker Danielsson
Norsk Regnesentral / Norwegian Computing Center
{Ragni.Ryvold.Arnesen, Jerker.Danielsson}@nr.no

Abstract

This paper presents an ongoing work on a framework for enforcement of privacy promises, policies and regulations. The aim is to develop an open framework that can form a basis for discussion of such enforcement, and deployable components that enable integration with legacy systems as well as state-of-art development environments.

Keywords: Privacy framework, policy, enforcement, privacy legislation.

1 Introduction

There are mainly two approaches to the implementation of privacy-enhancing or privacy-assuring technologies and processes. One is to minimize the amount of personally identifiable data through pseudonymisation or anonymisation, or by simply not collecting any data at all. The other approach is to assure that the privacy agreement, e.g. codified in P3P [22], that both data subject and data collector have consented to is enforced. There is no conflict between these approaches. Both are important.

There are circumstances where processing of personal data is useful or necessary. In some cases personal data must be collected due to legislation, or because it is necessary in order to provide some public service. In other cases the collection of personal data may be of benefit to both the data collector and the data subject. An example of such a case is the possibility for the data collector to customize offers to the data subject based on her/his interests, history and current context, e.g. location.

The data subject, i.e. the person whose identity is, or may be, connected to the data, usually has little control over information collected and stored. The notion of privacy when personal data is collected implies some form of trust in the data collecting entity. Systems for mandatory and automated enforcement will contribute to the establishment of such trust, as will a conceived high level of information security. There is in our view a need for an open framework for enforcement of privacy regulations and the privacy promises made by data collectors. Such a framework can form a basis for discussion of technology and processes, and a basis for development and deployment of enforcement functionality.

This paper describes our ongoing work on designing and implementing such a privacy enforcement framework, which comprises functionality necessary for adherence to privacy agreements pertaining to collected data, as well as applicable privacy regulations.

---

[1] The work presented in this paper is fully funded by the Norwegian Research Council through the research project "Personalized Internet-Based Services and Privacy Protection."

In addition to privacy enforcement functionality, there is a need for processes that guide the integration of privacy protecting functionality with legacy systems and existing business processes, and that guide the design of new privacy-enabled applications and business processes. We acknowledge this need, but it is not addressed further in this paper.

The remainder of this paper is organised as follows. Chapter 2 gives a brief account of related work in privacy frameworks. Chapter 3 gives an overview of the framework and some of the design principles and rationale behind it, and chapter 4 describes the framework elements in more detail. Finally, our current activities and future plans are described in the conclusion.

2 Related work

There are other ongoing efforts in defining privacy frameworks. The most noticeable being the Privacy Framework [11] developed by the International Security, Trust & Privacy Alliance (ISTPA) and the Enterprise Privacy Architecture (EPA) [15] developed by IBM Research.

The ISTPA Privacy Framework defines a number of services and capabilities that implement the fair information practices, see [18]. A capability is implemented through the invocation of multiple services. The services and capabilities provide functionality that supports both the data subject (e.g. preference definition and validation of preference) and the data collector (e.g. auditing).

EPA is a methodology for introducing privacy awareness, and privacy services and processes into enterprises. It consists of four building blocks: Privacy regulation analysis, management reference model, privacy agreement framework, and technical reference architecture. The privacy regulation analysis identifies and structures applicable regulations in a unified terminology and relates these regulations to the personal data held by the organisation. The management reference model defines processes necessary for a comprehensive privacy management program. The privacy agreements framework is a methodology for privacy enabling business processes. It results in a model of the personal data used in the process, the privacy-relevant players and operations of the process, as well as the rules that govern these operations. Finally, the technical reference architecture is a model of a system for the enforcement of privacy promises. It defines a management system, an audit console and a reference monitor.

3 The privacy framework

The framework presented here is inspired by the life cycle of personal data. That is, collection, various forms of processing (e.g. disclosure to third parties), and finally deletion or depersonalisation. The framework is intended to function as a layer of control between personal data on the one hand, and services accessing and collecting personal data on the other hand.

The framework consists of framework elements (e.g. Access) that together provide the functionality necessary for enforcement of applicable regulations and privacy agreements reached in connection with data collection. Each framework element is composed of components that support the implementation of its functionality (e.g. Reference Monitor).

The framework elements form a basis for discussion of the functionality of a general framework for enforcement of privacy policies. Moreover, the components of the framework elements should be deployable, meaning that they should enable integration with legacy systems and state-of-art development environments. Achieving these two properties of the framework, i.e. basis for discussion and deployable components, is the main objective of our work.

A basic requirement for the framework is that there must be a clear separation of functionality and responsibilities between the framework elements. Further, the framework should be complete in the sense that it should address all functionalities necessary to enforce local privacy policies, legal requirements, and agreements made between data subjects and data collector.

In addition, the framework must have support for traditional security mechanisms, such as authentication of users, and protection of confidentiality and integrity of information. How these mechanisms are integrated into the framework is not addressed in this paper.

4 Framework elements

The framework manages Personal Data Bundles that contain personal data, and the Agreement and access history pertaining to the personal data. IBM Research calls bundling of data and policy the "sticky policy paradigm" [14].

Personal Data Bundles can be introduced into the framework in two different ways: Personal data with pertaining Agreements can be imported from a third party (handled by the Data Import Manager of the Communication element), or personal data can be collected directly from the data subject. From the view of the framework, collected data is assumed to be packaged in Personal Data Bundles. It is further assumed that the Agreement pertaining to the collected personal data is derived from the privacy promise of the data collector and the privacy preference of the data subject. If the privacy promise and the privacy preference are codified in machine-readable formats, such as P3P [22] and APPEL [21], software agents can be used to automatically negotiate and consent to the Agreement on behalf of the parties.

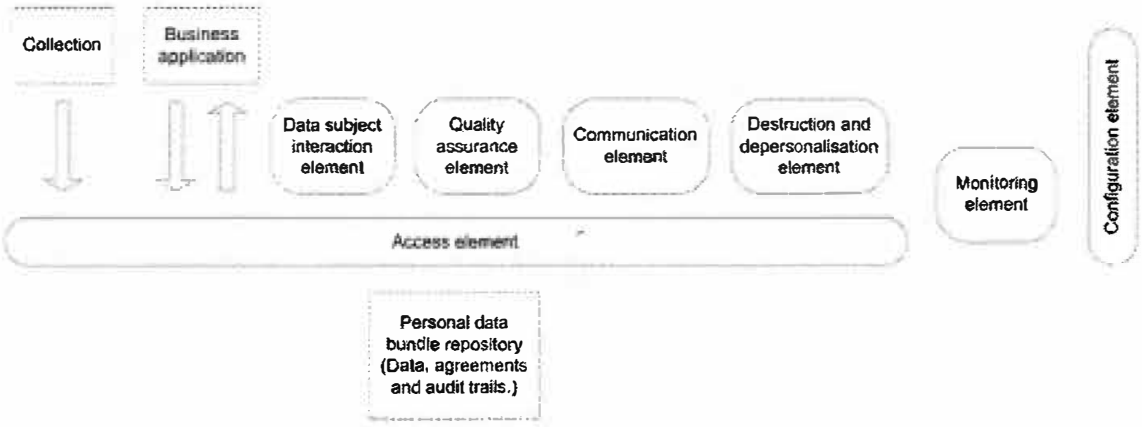


Figure 1 Overview of the framework.

Figure 1 illustrates an overview of the framework. The Access element controls the flow of personal data in its position between the personal data bundle repository, and collection modules, business applications and the other components of the framework. The audit trails that all components generate are analysed by the Monitoring element.

The Configuration element is responsible for assuring that the configuration of the framework elements complies with applicable privacy regulations, that it is consistent with the local privacy policy and that the published privacy promises are consistent with the configuration of the framework.

The following sections describe the Agreement and the Personal Data Bundle concepts, and the framework elements and their components, in more detail.

4.1 Agreement

An Agreement is a set of rules that determine how the personal data the Agreement pertains to can and should be used, and that both the data subject and data collector have consented to. An Agreement is derived from the privacy promise of the data collector and the privacy preference of the data subject.

The data collector's privacy promise forms an important base for all Agreements that the data collector makes. The privacy promise is based on an analysis of the need for personal data to

conduct the business processes of the data collector. The data subject's privacy preference defines the Agreements that the data subject is willing to consent to.

The attributes of the Agreement can be divided into two categories: attributes that can be suggested by both the data subject and data collector (they can negotiate the attribute) and those that can only be determined by one party. For example only the data collector can determine the purpose of collection.

Attributes that may be part of an Agreement include:

- Purpose – Why is the data collected? The collected data must only be used for the stated purpose.
- Subject access – Can the data subject access its personal data and the access/usage history of its personal data?
- Disputes – How are disputes solved?
- Remedies – How is a breach of agreement handled?
- Obligations – When performing certain actions, the data processor may be required to take further steps. E.g. if the data is accessed the data subject of the data must be notified.
- Retention – How long will the data be retained? Will it be destructed or depersonalised?
- Disclosure – To which third parties will the collected data be disclosed?

4.2 Personal Data Bundle

A Personal Data Bundle contains personal data and the Agreement regulating how the personal data can and should be used. The access/usage history of the data is also included in the Personal Data Bundle. The Personal Data Bundle may include signatures and the credentials of the data subject and the data collector, to bind the Agreement to the two parties. The credentials of the data subject may also be used in the implementation of subject access (see section 4.6.1).

4.3 Configuration

The Configuration element encompasses functionality for generation of the other framework elements' configuration and functionality for generation of privacy promises. This functionality is automated or semi-automated, in the form of consistency checking, or a combination of both. For example a privacy promise may be generated automatically from the configuration of the framework (see 4.3.3) or it may be constructed more or less manually with the support of consistency checking (see 4.3.4), verifying that the constructed privacy promise is consistent with the configuration.

The automated and/or semi-automated generation of the framework's configuration is based on the local privacy policy and applicable regulations, see Figure 2. The local privacy policy is based on an analysis of the processes of the organisation, their need for collecting and processing personal data, the players involved in the tasks of the processes, etc.

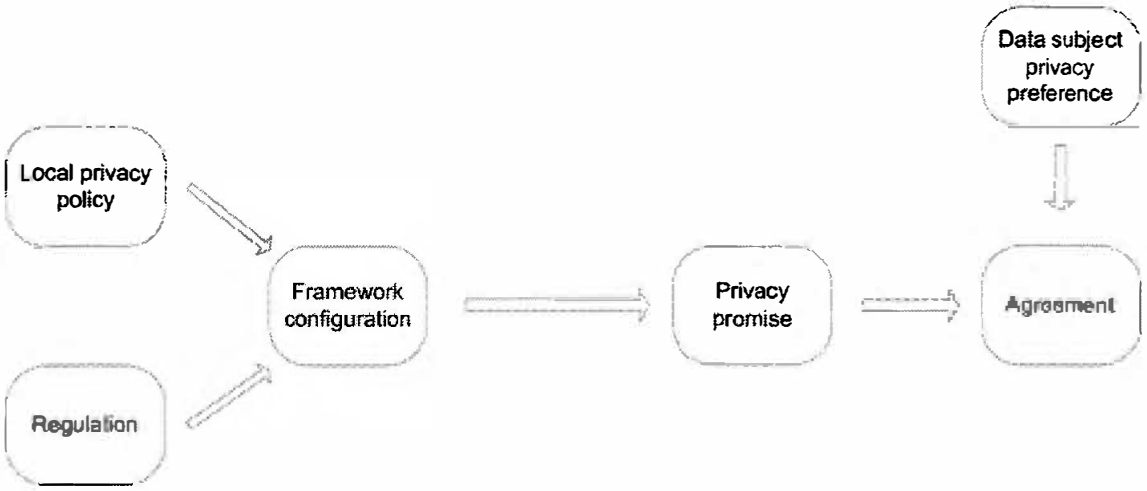


Figure 2 Overview of the configuration process and the making of an Agreement.

In Europe, most national privacy legislations are based on EU directives [5, 6], which in turn are based on the OECD Guidelines [17]. The privacy laws and regulations give rules as to how different types of information can or must be handled, including conditions for, and obligations following from, such handling. To enable automated or semi-automated integration of regulations into the configuration of the framework, the regulations must be codified by a human interpreting them. However, the laws are usually complex and difficult to comprehend, and the process must be repeated whenever the laws change. Hence, this task presents a great challenge. But such an encoding is portable and resource saving, in the sense that once it is defined, it may be used by any system that understands the language the rules are encoded in.

We are currently working on such an encoding of the Norwegian privacy law [16]. This is not an easy task due to the frequently intricate formulation of the paragraphs. Some rules say what you are not allowed to do, some say what you may do, and some say what you must do. There is a multitude of special cases, exceptions to the rules (positive or negative), and exceptions to the exceptions. There are listings of conditions and obligations, but which ones are required often depends on which other conditions are fulfilled and/or which are not.

In addition to the configuration of the framework, the flow of personal data is regulated by the Agreements in the Personal Data Bundles. The Agreements supplement the configuration in the sense that an Agreement must not contradict the configuration.

4.3.1 Configuration Generator

The Configuration Generator provides automated and semi-automated support for the generation of the framework configuration, according to the local privacy policy and codified regulations. The functionality of the Configuration Generator overlaps the functionality of the Legal Compliance Analyser, in the sense that if an ideal Configuration Generator exists there is no need for a Legal Compliance Analyser.

4.3.2 Legal Compliance Analyser

The Legal Compliance Analyser controls that all framework elements are configured in accordance with applicable regulations. The Legal Compliance Analyser is not necessarily an automated tool that analyses everything and outputs an "ok" or identifies were the problems/inconsistencies are. It can also be more of a questionnaire that assures that all relevant checkpoints are gone through so that the configuration complies with regulations and that there are no inconsistencies between regulations and configuration.

4.3.3 Privacy Promise Generator

The Privacy Promise Generator generates a privacy promise based on the framework configuration. The privacy promises define the set of Agreements that the data collector is willing to accept. That is, the privacy promise defines the base Agreement that the data subject may modify through opt-ins and opt-outs defined in the privacy promise.

The functionality of the Privacy Promise Generator overlaps the functionality of the Privacy Promise Analyser. An example of such a generator can be found in [13].

4.3.4 Privacy Promise Analyser

The Privacy Promise Analyser checks that the privacy promises of the organisation and the resulting set of possible Agreements are consistent with the configuration. Moreover, before personal data is imported (by the Data Import Manager, see section 4.8.1), the Privacy Promise Analyser may be used to control that the set of possible Agreements of the imported data is consistent with the local configuration.

4.4 Access

The Access element is responsible for keeping track of all personal data that is held by the organisation and for regulating the access to this data in accordance with the configuration and the Agreements pertaining to the data.

4.4.1 Reference Monitor

The Reference Monitor denies or accepts access to operations on personal data to requestors internal to the organisation, or internal to the domain if the framework is implemented as a security barrier between domains. External requestors or third parties request access through the Disclosure Controller (see 4.8.2).

The Reference Monitor is essentially an access control mechanism, but the type of access control necessary to enforce privacy policies is different from other access control models, such as the well-known Bell LaPadula model. This is mainly because the *purposes* of the data processing, as well as other context information, are important in the privacy case. These differences are discussed in [7], which also presents a formal access control model for the enforcement of privacy policies. Other examples of work on privacy enabled access control are the control service in the ISTPA framework [11] and the privacy policy model presented in [12].

The Reference Monitor bases its decisions on authorisation rules written in a machine-readable policy specification language, e.g. EPAL. EPAL is a formal language for writing "privacy authorization rules that allow or deny actions on data-categories by user-categories for certain purposes under certain conditions while mandating certain obligations." [2]

Vocabularies need to be built to encompass the specific data- and user-categories, actions, purposes, conditions and obligations pertaining to the system and policies in question. For instance, codifying the Norwegian privacy legislation will require a vocabulary containing the condition "informed consent from the data subject". Defining useful vocabularies for actions and purposes, and mapping the applicable policies to these, will require analysis of the operations performed by the applications accessing data through the framework. Suitable vocabularies will facilitate efficient and fine-grained access control.

To prevent aggregation or inference of data, the Reference Monitor may also base its decisions on special context conditions like the access history of the data and/or data requestor. In many cases the application does not really need access to the actual personal data; access to the relationships between data may be enough. In such cases, the application should only get access to pseudonymised data. The Reference Monitor may also implement functionality to

reduce the accuracy of data, e.g. the granularity of location data, based on the authorisation rules.

4.4.2 Personal Data Broker

The Personal Data Broker acts as a librarian, i.e. handles requests for personal data and locates the requested data. Based on the Reference Monitor's access decisions, it delivers the personal data requested from the different data repositories where personal data is stored. In addition, it assures that documentation is maintained over the personal data held by the organization.

The Personal Data Broker may additionally implement the identity protector concept presented in [10]. The identity protector creates one or several pseudo-domains in which data subjects are known under pseudo-identities. Only the identity protector knows the mapping between identities and pseudo-identities, and the mapping between pseudo-identities, and is thus able to reverse the process and retrieve identities from pseudo-identities.

The Personal Data Broker also triggers the events needed to ensure that any obligations following from the requested type of access are fulfilled.

4.5 Monitoring

The Monitoring element monitors and analyses the audit trails generated by the other elements. Other elements, in particular the Access and Communication elements, implement *proactive* mechanisms whose purposes are to prevent users from doing what they are not allowed to do according to the framework configuration and the Agreements of the accessed data. The monitoring mechanisms, on the other hand, are *reactive* in the sense that they may enable detection of a policy breach and cause some reaction after the breach happened. The proactive mechanisms are the front line of security mechanisms, but the reactive mechanisms are also important, particularly to build and maintain the users' trust in the system.

Monitoring mechanisms are important parts of an internal control system, which is mandated by Norwegian legislation ([16], §14).

4.5.1 Audit Manager

The Audit Manager supports auditing (manual and semi-automated) of the audit trail that the components generate. It implements functionality for searching and reviewing the audit trail.

4.5.2 Privacy Violation Detector

The Privacy Violation Detector continually monitors access to personal data and detects misuse and/or anomaly behaviour. Anomalies can be detected using e.g. data mining methods, see [1] for a survey of such methods.

4.5.3 Remote Privacy Audit Manager

The Remote Privacy Audit Manager provides seal issuing authorities and/or official authorities (e.g. The Data Inspectorate in Norway) with the possibility to remotely monitor and review the site.

4.6 Data Subject Interaction

The Data Subject Interaction element provides access to personal data, access/usage history and Agreements to data subjects. It also provides mechanisms for data subjects to submit complaints, and support for resolving these complaints.

4.6.1 Subject Access Manager

The Subject Access Manager manages data subjects' requests for reviewing and updating their personal data. It may also provide access to the access/usage history of the subjects' personal data. In addition, the data subjects may have the possibility to modify the Agreements that pertain to their personal data.

Subject access might improve the quality of the personal data. If the data subject has access to its personal data he/she might assure that it's correct, especially if there is some sort of incentive for the data subject to do so. Additionally, subject access may provide a powerful tool for detecting agreement violations. The probability of detecting violations increases if data subjects review the access/usage history of their personal data.

Subject access can be managed through electronic means (e.g. the Internet) or through traditional mail delivery. In any case, subject access sets authentication requirements. If the authentication is not strong enough subject access will instead contribute to the impairment of the privacy of the data subjects.

Norwegian legislation gives the data subjects rights to information about the nature of the personal data processing and what information pertaining to the data subject is stored ([16], §18). Data subjects also have a right to demand correction of incorrect or incomplete data, or in some cases also blocking or complete erasure of data (§27).

4.6.2 Dispute Manager

The Dispute Manger offers support in resolving disputes. It offers different mechanisms (e.g. web, email) for submitting complaints to the data collector and/or some other relevant authority. In addition, it may provide support for semi-automatic processing of complaints and compilation of reports on the use of the personal data pertaining to the complaining data subject.

4.7 Quality Assurance

The Quality Assurance element encompasses functionality that aims at upholding the correctness of the stored personal data. Quality assurance is also provided by the data subjects through the Subject Access Manager, but the data collector also has a responsibility and interest in maintaining data quality.

Norwegian legislation demands that the data controller (i.e. the entity storing and processing the personal data) ensures that personal data processed are accurate and up-to-date, and also adequate, relevant and not excessive in relation to the purpose of the processing ([16], §11). Internal control procedures must be implemented to ensure quality of data (§14). If incorrect, incomplete or unauthorised data have been processed, the data controller shall to the extent possible ensure that the error does not have any effect on the data subject, for instance by notifying recipients of disclosed data (§27).

4.7.1 Subject Preference Register Monitor

This component monitors customer preference registers and assures that the framework is compliant with the information in such registers. One example of such a register is the Norwegian reservation register against direct marketing [19]. Here the users may request that their address is removed form address lists used in direct marketing (with a few exceptions), and companies performing such marketing must update their address lists at least every three months.

4.7.2 Validator

The Validator component checks the consistency of incoming data from data subjects or third parties against defined bounds and heuristics. It can also check input data against data col-

lected previously and external sources. The bounds and heuristics should be defined when the data collection or communication is defined.

4.8 Communication

The Communication element provides functionality for importing and exporting personal data in and out of the domain controlled by an instance of the framework. Automatic export and import of data is dependent on standardised exchange protocols. A proposal of such a standard is the Customer Profile Exchange (CPExchange) standard [3], but its adoption has been limited.

4.8.1 Data Import Manager

The Data Import Manager controls the import of personal data and possibly linking and matching of locally controlled personal data with the imported personal data. During import it controls the Agreements of the imported data to verify that the import is allowed. It guarantees the preservation of the Agreements pertaining to the imported data. In addition, it controls that any linking and matching is conducted according to the Agreements pertaining to the data involved in the operation, and that the resulting data is bundled in new Personal Data Bundles with updated Agreements.

Mergers, acquisitions and internationalisation can create a wish to link and match or integrate databases containing personal data. The trend towards one-stop-shop services in the public sector also actualizes the issue of linking and matching. Proper control of linking and matching is essential since separation of data repositories is fundamental to privacy protection.

Norwegian legislation states that data subjects have a right to be notified when data is collected from other parties ([16], §20). Also, if the data controller contacts the data subject or makes decisions regarding the data subject on the basis of personal profiles, the controller must inform the data subject of the sources of the data (§21).

4.8.2 Disclosure Controller

The Disclosure Controller controls the disclosure of personal data to third parties outside the framework's domain. It determines to whom personal data may be passed and under what conditions based on the Agreements of the exported data.

Internationalisation and outsourcing of functions, like Customer Relationship Management (CRM), are two trends that contribute to the transfers of consumer and employee data by businesses. Disclosure is complicated by the fact that different countries or regions have different privacy legislation and some have none. According to Norwegian legislation ([16], §§29-30), personal data may in general only be transferred to countries that ensure an adequate level of protection of the data.

4.9 Destruction and Depersonalisation

The Destruction and Depersonalisation element is responsible for the last step of the life cycle of personal data. After this step the data should no longer be considered personal data, with the exception of pseudonymised data where the depersonalisation can be reverted.

An important principle in Norwegian legislation is that personal data may not be stored longer than necessary for the purpose ([16], §11, §28).

4.9.1 Destruction Controller

The Destruction Controller is responsible for assuring that any commitment to destruction of personal data is fulfilled in time. The data should be destructed in such a way as it is made irretrievable and unreadable.

4.9.2 Depersonalisation Controller

The Depersonalisation Controller is responsible for assuring that any commitment to depersonalisation of personal data is fulfilled in time.

Depersonalisation can be reversible (pseudonymisation) or non-reversible (anonymisation). See [8] for a definition of anonymity and pseudonymity.

How data should be depersonalised is not always straightforward. For example if the record contains name, employer and year of birth it may not be enough to delete or pseudonymize the name field. It may still be possible to identify the person that the record pertains to, especially if one has access to supplementary information that could be linked and matched with the "depersonalised" record. That is, the risk of reidentification depends on the size of the dataset and the entropy of the remaining attributes [9].

5 Conclusion

In this paper we have presented our proposal for an open framework for enforcement of privacy policies. The framework comprises functionality to enforce local privacy policies, privacy legislation and agreements reached between data subject and data collector.

We are currently working on a prototype implementation of some of the components of the Configuration, Access and Monitoring elements in the form of a Java framework and plan to experiment with implementations of other components as well. We are also working on codifying the Norwegian privacy legislation into a machine-readable format. Meanwhile, we will continue to refine and develop the framework, and we are also looking into future possibilities for realising stronger enforcement mechanisms. For example, the technology proposed by the Trusted Computing Platform Alliance (TCPA) [20] may provide possibilities for forcing the recipient of personal data to act in accordance with the agreement bundled with the data.

6 References

[1] Aas K., Huseby R., and Thune, M., *Data Mining - A Survey*. Report No. 942, June, 1999. ISBN 82-539-0426-6

[2] Ashley, P., Hada, S., Karjoth, G., Powers, C., Schunter, M. (ed.), *Enterprise Privacy Authorisation Language (EPAL)*, IBM, 2003. Available via http://www.zurich.ibm.com/security/enterprise-privacy/epal/

[3] *Customer Profile Exchange (CPExchange) Specification*, version 1.0, October 2000, Available via http://www.cpexchange.org/standard/

[4] Datatilsynet (The Data Inspectorate), *Veiledning i informasjonssikkerhet i kommuner og fylker*, 1999, http://www.datatilsynet.no/dtweb/attachment/783/Kommuneveiledning.pdf

[5] *Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data.* Official Journal L 281, 23/11/1995, pp. 31-50. Available via http://europa.eu.int/eur-lex/en/index.html

[6] Directive 2002/58/EC of the European Parliament and of the Council of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector (Directive on privacy and electronic communications). Official Journal L 201, 31/07/2002, pp. 37-47. Available via http://europa.eu.int/eur-lex/en/index.html

[7] Fisher-Hübner S., Ott, A., *From a Formal Privacy Policy Model to its Implementation*, National Information Systems Security Conference (NISSC 98), 1998. Available at http://www.rsbac.org/niss98.htm

[8] Köhntopp M. and Pfitzmann A., *Anonymity, Unobservability and Pseudonymity – A Proposal for Terminology*, Draft v0.12, Available at http://123.koehntopp.de/marit/pub/anon/Anon_Terminology.pdf

[9] Fisher-Hübner S., *Privacy-Enhancing Technologies*, Karlstad University, Department of Computer Science, PhD course, Slides session 2, Available via http://www.cs.kau.se/~simone/kau-phd-course.htm

[10] Hes, R. and Borking, J. (eds.), *Privacy-enhancing technologies: The path to anonymity. Revised edition.* ISBN: 90-74087-12-4. Registratiekamer, The Hague, August 2000

[11] International Security, Trust & Privacy Alliance (ISTPA), *Privacy Framework*, Version 1.1, ISBN: 0-9721484-1-8, 2002

[12] Karjoth G., Schunter M., *A Privacy Policy Model for Enterprises*, 15th IEE Computer Security Foundations Workshop, June 2002

[13] Karjoth G., Schunter M. and Van Herreweghen E., *Enterprise Privacy Practices vs. Privacy Promises - How to Promise What You Can Keep*, 4th IEEE International Workshop on Policies for Distributed Systems and Networks (Policy '03), Lake Como, Italy, June 2003

[14] Karjoth G., Schunter M. and Waidner M., *Platform for Enterprise Privacy Practices: Privacy-enabled Management of Customer Data*, 2nd Workshop on Privacy Enhancing Technologies, 2002

[15] Karjoth G., Schunter M. and Waidner M., *Privacy-enabled Services for Enterprises*, IBM Research, Zürich Research Laboratory, Switzerland, January 2002

[16] *LOV 2000-04-14 nr 31: Lov om behandling av personopplysninger (personopplysningsloven)*. Available at http://www.lovdata.no/all/hl-20000414-031.html. Norwegian privacy law. An unofficial English translation is available at http://www.datatilsynet.no/lov/loven/poleng.html

[17] *OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data.* Available at http://www1.oecd.org/publications/e-book/9302011E.PDF

[18] *Privacy Online: A Report to Congress.* Federal Trade Commission, June 1998. Available at http://www.ftc.gov/reports/privacy3/priv-23a.pdf

[19] Reservasjonsregistret i Brønnøysund (The Brønnøysund reservation register), http://www3.brreg.no/oppslag/reservasjon/om_resreg.jsp

[20] The Trusted Computing Platform Alliance (TCPA), http://www.trustedcomputing.org/

[21] W3C, *A P3P Preference Exchange Language 1.0 (APPEL1.0)*, W3C Working Draft, April 2002, Available at http://www.w3.org/TR/P3P-preferences/

[22] W3C, *The Platform for Privacy Preferences 1.0 (P3P1.0) Specification*, W3C Recommendation, April 2002, Available at http://www.w3.org/TR/P3P/

NetRAM : A Novel Approach for Network Security Risk Management

Mohamed Hamdi, Jihène Krichène, Noureddine Boudriga, Mahmoud Tounsi

Higher School of Communications, Tunisia

This paper addresses the risk management in organizations networks and provides a 10-process approach to monitor security and prevent attacks. Our approach develops various formal techniques that are needed to guarantee the efficiency, correctness and generality of risk management.

Key words : Networked environment, risk management, quantitative risk assessment, theoretical design.

1 Introduction

Over the last years, companies' dependence on information technology has grown tremendously. In many domains of activity, organizations rely totally on their networked environment and could not survive without it. However, the increasing occurrence of harmful attacks against networks and computing infrastructures has shown that security holes can lead to large damages. Therefore, security is becoming a crucial issue that has to be addressed seriously by network administrators and companies managers.

A statistical analysis of network attacks have revealed that most of the targets were protected by means of various security mechanisms and techniques, and that, in the presence of protection, the success of an attack is due to the misuse of the security resources by the administrators. In fact, many of the security administrators still rely on intuition to take strategic decisions related to their information system security.

Attempts to develop fundamental quantitative methods to avoid (or mitigate) network security risks arose recently ([1]). In addition, many governmental departments and organizations have published guides to assist companies in taking rational decisions to secure their systems security ([2, 3, 4]). Despite the increasing interest of many researchers in developing networks security technologies, and strengthening the existing approaches, security mechanisms are still non-sufficiently applied to the real world, because of many factors. In fact, quantitative risk analysis models present a high computational complexity, and their application in large environments, without the use of automated tools, is quasi unfeasible.

On the other hand, the available software tools do not permit to perform a complete risk management cycle as specified in the published methods. None of the available products automates all the steps of the risk management cycle. This makes risk management application hard for system administrators. Furthermore, most of the risk management products do not keep up with the evolution of risk analysis theoretical models.

To overcome these problems, we present an automated multi-process approach that assists information technology systems administrators and managers in their attempts to reduce security risks that threaten their assets. We also develop three major concepts: the composite data structure, which allows a better representation of the analyzed system, the optimization problem for risk analysis, and the security state monitoring of a network. We demonstrate that NetRAM

covers the most important limits of the existing methods. From an architectural point of view, it introduces two extra processes that instill a continuity to the risk management process. The other differences reside mainly at the risk analysis level where a new formalism is proposed to avoid problems caused by asset-driven approaches. In addition, NetRAM relies on quantitative features to assess security risks unlike the most used methods that are based on qualitative criteria.

In the following sections, we give the key issues that characterize this approach. In Section 2, we present the main processes of our approach. In Section 3, we describe a set of theoretical tools that the proposed model uses to process specific data structures and we present a model for the decision making process. Furthermore, we address the problem of gathering and updating data necessary for conducting risk assessment. Section 4 addresses the definition and management of security network state. Section 5 concludes the paper.

2 NetRAM: architecture and features

NetRAM (Network Risk Analysis Method) has been developed and designed at the National Digital Certification Agency[1] (NDCA, Tunisia) within the framework of the project "Risk Management in a Networked Environment". It consists of ten processes depicted in Figure 1 and discussed below. The reader would notice that this method is inspired from Riskit [5] (a risk management method for software development projects) and OCTAVE [1] (developed by the Computer Emergency Response Team). However, our method palliates the deficiencies discussed in the previous section, offers a continuous control of the state of the system as it includes a central monitoring process, and develops various models that help reasoning about and monitoring the security state of a network.

1. Initialization: A prediction step is first performed to estimate the time, the effort and the budget that would be needed for the whole risk management project. This implies a quantification of the complexity of the target system using input parameters that are defined according to an estimation method. This method enables security teams to easily recalibrate, customize and extend the cost model that the estimation can produce.

Complexity estimation, however, can not be achieved without identifying the tasks that constitute the risk management project. While the processes of NetRAM are known and fixed, their constituency depends on the context of the conducted project. For example, *"Security Document Review"*, which makes part of the vulnerability analysis process, has not to be done if security documentation is absent in the enterprise that owns the analyzed system.

The resulting data are very useful in the case of an outsourced or an in-house information security risk assessment, as they provide a good basis for the mission chiefs to schedule the activities and to take decisions such as those concerning the required manpower that will conduct the mission or the amount of money that should be allocated for it.

2. Asset analysis: This process is designed to collect detailed information about the assets that make part of the analyzed system. Indeed, an inventory containing all the resources must be established including some parameters such as the criticality of each asset or the objects that are authorized to access it. Furthermore, knowing that the components of the analyzed system are - in most cases - interrelated, the interaction between the resources given in the inventory is a point of interest. For instance, issues as data or information flow between the different entities may be focused. Also, physical interaction has to be analyzed as the security of an asset can depend on the security of an other asset that contains it physically (e.g., the security of an equipment put in a room depends on the security level given by the walls, the doors and the windows of the room itself). Thus, dependency trees can be built to show the interrelation between the resources of the system to facilitate the risk analysis process. This will be discussed later.

It is worth to mention that the documents related to security (e.g., security strategy, security policy) should be considered as special assets.

[1] NetRAM was designed and developed with NDCA by the Communication Networks and Security Research Lab, University of Carthage, Tunisia.

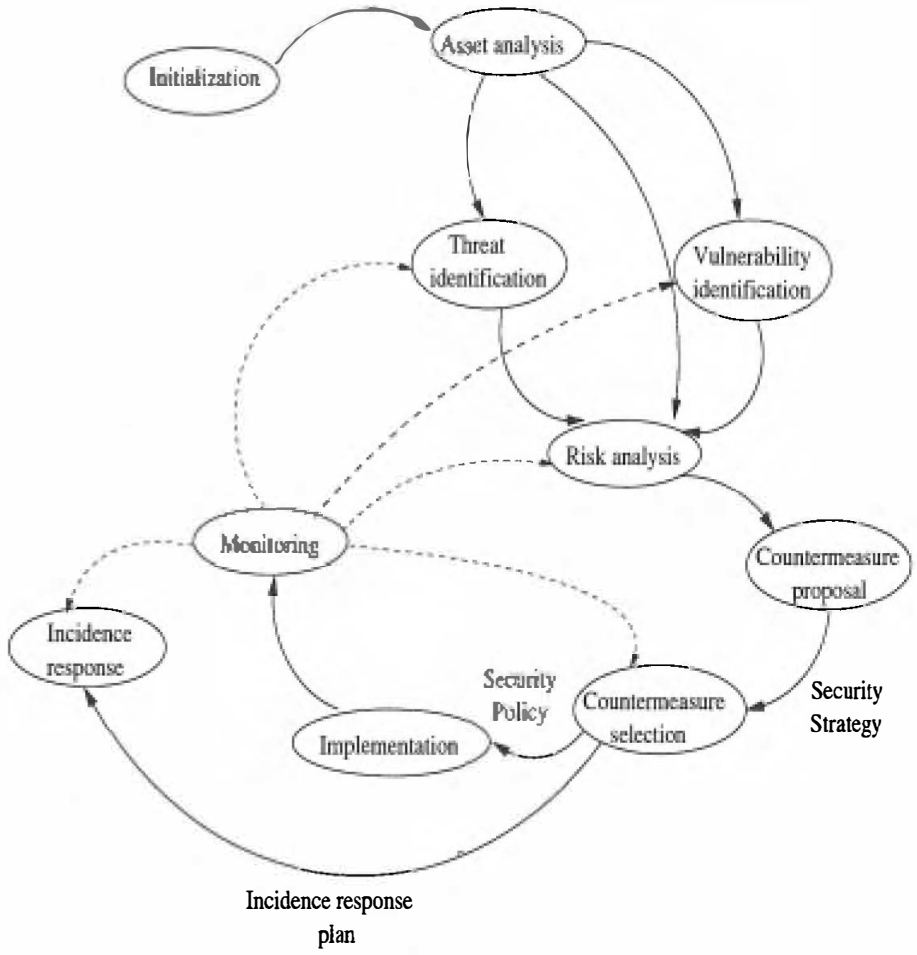


Figure 1: NetRAM processes.

3. Vulnerability identification: The purpose of this process is to identify the weaknesses of the system described in the former step. The recommended approach is to have a vulnerability library pre-built and to check, for every vulnerability, whether it is present or not in the studied case. Many types of vulnerabilities can be considered, including:

- Bugs: Most of the software used in networked environment contain security holes, which can be exploited by malicious entities.
- Mis-configurations: Lack of experience or insufficient training of the personnel opens many breaches in system security. For instance, a mistake in the security policy of a firewall can allow unauthorized users to gain access to a private network.
- Physical vulnerabilities: Computers, communication equipments and media must be located in a secure facility. For example, if an attacker has physical access to a router, he can break it (denial of service) or try to modify its configuration from the console port (information leakage) if he is more clever.
- Conceptual vulnerabilities: The theoretical specifications of widely used communication and security protocols contain vulnerabilities that have to be addressed when analyzing the target system. One of the most famous vulnerabilities of this class is related to Simple Mail Transfer Protocol (SMTP), which does not authenticate the source of an e-mail.

- Procedural vulnerabilities: Inadequate procedures or inappropriate security measures can be exploited to realize malicious acts. An example of this type of vulnerability could be the absence of a backup policy, which can result in an irrecoverable loss of data.

4. Threat identification: Threats are potential events that can affect the system under analysis. They can result from malicious actions, accidents, natural disasters, etc. Unlike vulnerabilities, threats are measurable as each of them can be represented by its frequency and severity. Obviously, threat rates and impacts depend on the environment in which the system is situated. Factors as geographical position, political stance, or activity domain have to be taken in consideration when allocating probabilities of occurrence to threats.

In our approach, the frequency and the severity of a threat are dynamically updated according to the values of several metrics measured during the monitoring process.

5. Risk analysis: We define an attack as a combination of a threat and a set of corresponding vulnerabilities. A risk can then be seen as a weighted attack. Many weights can be allocated to a single attack where each weight corresponds to a criterion. The probability of the attack, its technical difficulty or the amount of money needed to carry it out may constitute good criteria. Then, as an extension to this reasoning, we introduce the concept of attack scenarios which may be viewed as attack chains where the last link is called the main attack, the first link is an elementary attack while the other links are intermediate attacks. This approach expresses accurately the occurrence of real attacks where a malicious user performs a set of intermediate attacks in order to achieve his major goal : the main attack. Furthermore, as a weight can be assigned to each attack, we can conclude that a global weight may be allocated to an attack scenario by combining the elementary weights corresponding to each attack of the scenario. Weighted attacks are then called risk scenarios.

The aim of the risk analysis process is to define the main risks corresponding to each asset and to establish the risk scenarios leading to every main risk. Indeed, this process can be divided into the following steps :

- Identify the global (main) attacks corresponding to the asset;
- Build the attack scenarios for every main attack;
- Determine the risk coefficients (weights) for each scenario. A weight corresponding to a scenario is computed by combining the weights of the elementary attacks belonging to it.

New attack scenarios should be automatically appended to the existing ones whenever the occurrence of a new attack chain is detected during the monitoring step.

6. Counter-measure proposal: Possible risk scenarios issued from the former step can be ranked and prioritized. Then, for every scenario, a set of security rules are defined to minimize its priority. If the scenarios are ranked according to their probability of occurrence, the goal of the rule should be to minimize this probability. These rules must be general and must not include issues related to the effective implementation (e.g., practices to follow, products to acquire, technical standards to comply with, etc.).

The obtained set of rules constitutes the security strategy of the analyzed system.

7. Counter-measure selection: A set of candidate risk control techniques are proposed to implement the rules of the security strategy. Then, according to several criteria, the actions that would be taken to protect the system are selected and clearly described in a document that is called the security policy. These criteria may include: (1) the efficiency or the degree of protection given by the technique, (2) the cost of the technique, (3) the criticality of the asset concerned by the risk, and (4) the feasibility.

Moreover, a plan defining the prioritization of the retained actions with respect to the corresponding level of risk and to the allocated budget must be done during this process. Several counter-measures can even be omitted if the security level they allow to attain is not proportionate to their costs.

An incident response plan is also performed at this level. It defines the actions that might be taken when a security incident occurs. These actions should include procedures for the notification and the documentation of security incidents as well as a description of the recovery mechanisms.
8. Implementation: During this phase, the actions defined in the previous process are effectively implemented. The key operations of this process are:

- Acquiring the needed hardware and software specified in the security policy;
- Defining the teams that will be in charge of securing and monitoring the system;
- Promote security awareness;
- Enforce the application of the security practices;
- Implement the technical operations so that the system becomes compliant with the security strategy and with the security policy.

9. Monitoring: Monitoring is the central process of NetRAM. It can be seen as the detection of events that change several properties of the system under control. Events include security incidents, the addition of a new asset, or the appearance of a new vulnerability.

The monitoring module has two main features: it is continuous and retrospective. In fact, monitoring must be done, from the moment of the implementation of the first action of risk control, in a continuous way to ensure an efficient detection of the interesting events. In addition, it can be seen as a trigger that launches other processes which were already performed such as vulnerability identification in the case of the appearance of a new vulnerability or counter-measure selection if a technological innovation allows the implementation of a solution that were impossible at the time of the initial application of the framework.

Moreover, several measurements are done at this level to compute the values of parameters used in the previous processes including the probability and the impact of an attack. This issue is discussed in more details in Section 3.

10. Incident response: This process is performed whenever an anomaly is detected at the "monitoring level".

The incident response plan defines the constituency and the role of the incident response teams as well as the actions they have to take if an anomaly do occur. These actions cover essentially four topics:

- Notification: This includes the determination of who should be not notified and which mean of communication should be used.
- Impact attenuation: This defines what should be done in order to reduce the impact of the incident and to stop its propagation through the analyzed system.
- Recovery: In case of damage, what should be done to ensure a recovery of the essential functions of the target system has to be clearly decided and achieved.
- Documentation: Part(s) of the information related to the incident has to be archived.

The two latter processes are not present in most of the existing methods such as OCTAVE, which has basically three phases consisting at building asset-based threat profiles, identifying infrastructure vulnerabilities and developing security strategy and plans. The main benefit of the addition of the monitoring and the incident response processes are to guarantee a continuous control of the system state and a better survivability meaning that the critical components can be recovered in an optimal time after the occurrence of an attack.

The major advantages of our method include the following four capabilities:
NetRAM is a structured method. The risk management task is organized among distributed collaborative processes. Any extension, addition, or modification within a process is transparent to the other processes,

NetRAM is a general purpose method. NetRAM is completely integrated to the activity of an enterprise in the sense that libraries, rules, metrics, and process localization take into consideration the enterprise specificity. Therefore, it can be applied in-house or outsourced without applying fundamental changes on its structure,

NetRAM is adaptive. Several processes are dynamic and adaptive in order to adapt cost estimation and security analysis to the evolution of the security technology,

NetRAM is model based. Theoretical developments support the process design in order to guarantee coherent representation of the enterprises' systems and consistent formal validation when they are needed. This allows NetRAM to be automated to a high rate because of the theoretical tools it specifies and the multi-objective decision making it integrates,

3 NetRAM design

A variety of theoretical tools can be used in order to automate NetRAM's steps and increase their efficiency. In this section we give a brief description of the activities associated with NetRAM most important processes.

3.1 Composite data structures

The basic ingredients of NetRAM design are four sets (R_b, V_b, A_b and D_b) representing respectively assets, vulnerabilities, attacks and decisions (counter-measures). A key feature of these sets is that they contain basic and composite elements. This allows an efficient modeling of the global situation of the system under analysis. Table 1 gives the meaning of basic and composite elements for each of the considered entities.

Table 1: Significance of basic and composite elements.

	Basic element	Composite element
Assets	Network node	Set of network nodes
Vulnerabilities	Single vulnerability	Set of vulnerabilities
Attacks	Simple attack	Attack scenario
Decisions	Single decision	Set of decisions

The reader would notice that composite assets, vulnerabilities and decisions have a different nature from composite network attacks. Indeed, by changing the order of the elements belonging to a composite attack, a different attack scenario is obtained. However, to represent a composite asset, vulnerability or decision, the order of the elements constituting it is not considered. For this reason, sets can be used to model the latter composite entities while uplets would represent attack scenarios. This means that, given the sets R_b, V_b, A_b and D_b that contain respectively the basic resources, vulnerabilities, attacks and decisions, the global sets representing these components have the following expressions:

- $R = R_b^*$,
- $V = V_b^*$,
- $A = \bigcup_{n \in \mathbf{N}} A_b^n$,
- $D = D_b^*$,

where $(.)^*$ denotes the set of partitions of a given set.

Using this notation, the application of a set of security decisions to the analyzed system can be modeled by a counter-measure matrix, denoted C, having the size $card(D) \times card(R)$ where each element can be equal to 0 or 1 according to the following rule:

$$\forall (i,j) \in \{1,..,card(D)\} \times \{1,..,card(R)\}$$

$$\begin{cases} C_{ij} = 1 & \text{if the decision } d_i \text{ is applied to the asset } r_j, \\ C_{ij} = 0 & \text{if not.} \end{cases}$$

Based on this notation, the objective of the risk analyst is to find the matrix C^* that is most appropriate to the system. This can be performed by allocating quantitative values to security attacks and decisions.

3.2 The risk analysis problem

Let $\mathbf{I}$ and Π be two $card(A) \times card(R)$ matrices denoting the impact and the probability of the success of each attack against each asset. Similarly, consider two matrices $\mathbf{I_I}$ and $\mathbf{I_\Pi}$ where $\mathbf{I}_{\mathbf{I}_{i,k}}$ (resp. $\mathbf{I}_{\mathbf{\Pi}_{ik}}$) corresponds to the influence of the application of the decision d_i on the impact (resp. the probability) of the attack a_k, for every (i,k) in $\{1,..,card(D)\} \times \{1,..,card(A)\}$. For instance, $\mathbf{I}_{\mathbf{I}_{2,3}} = 0.9$ means that if decision d_2 is made, than the impact of the success of the attack a_3 on any asset is reduced to a rate of 10% of the original impact.

In order to perform a balance between the efficiency of a set of counter-measures (modeled by the matrices defined above) and its cost, a $card(D) \times card(R)$-size matrix Γ should be introduced. For every $(i,j) \in \{1,..,card(D)\} \times \{1,..,card(R)\}$, Γ_{ij} denotes the cost of the implementation of the decision d_i to protect the resource r_j.

These matrices are particularly useful to quantify security counter-measures. In fact, using the operators $*$ and . denoting the term-by-term and classical multiplication, the functions that the risk analysis process aims at optimizing are represented as follows:

$$f_1(C) = \|\mathbf{I}_\pi \boxtimes_{\mathbf{C}} (\mathbf{I} \circledast \mathbf{M})\|,$$

$$f_2(C) = \|\mathbf{I}_\iota \boxtimes_{\mathbf{C}} (\mathbf{I} \circledast \mathbf{M})\|,$$

$$f_3(C) = \|C \circledast \Gamma\|,$$

where

- $\mathbf{M}$ is a $card(A) \times card(R)$-size matrix such that

$$\forall (k,j) \in \{1,..,card(A)\} \times \{1,..,card(R)\}$$

$$\begin{cases} \mathbf{M}_{kj} = 1 & \text{if the attack } a_k \text{ is possible to carry out against the asset } r_j, \\ \mathbf{M}_{kj} = 0 & \text{if not.} \end{cases}$$

- $\circledast$ denotes term-by-term multiplication of matrices,
- $\boxtimes_C$ is an operator such that for every $card(D) \times card(A)$ and $card(A) \times card(R)$ matrices $\mathbf{M_1}$ and $\mathbf{M_2}$,

$$\forall (i,j) \in \{1,..,card(D)\} \times \{1,..,card(R)\}$$

$$\begin{cases} (\mathbf{M_1} \boxtimes_{\mathbf{C}} \mathbf{M_2})_{ij} = \sum_{k=1}^{card(A)} (\mathbf{M_1})_{ik} (\mathbf{M_2})_{kj} & \text{if } C_{ij} = 1, \\ (\mathbf{M_1} \boxtimes_{\mathbf{C}} \mathbf{M_2})_{ij} = \sum_{k=1}^{card(A)} (\mathbf{M_2})_{kj} & \text{if } C_{ij} = 0. \end{cases}$$

- $\|.\|$is a norm on the $card(A) \times card(R)$-size matrices space.

The matrix **M** ensures that only possible attacks are taken into consideration when evaluating the efficiency of a set of counter-measures. In fact, even if a decision mitigates the effect of a given attack, it has not a positive impact on the system if there is not an asset such that this attack is possible to perform against it.

The functions f_1 and f_2 allow to evaluate the efficiency of a matrix C while f_3 represents the cost of the decisions enclosed within this matrix. In practice, the objective is to minimize these former functions which can not be directly achieved due to the lack of natural order on the set of vectors. We propose heuristic-based evolutionary algorithms to address this multi-objective decision problem as they have been widely used in this context since several years.

3.3 Data collection

In the above analysis, we assumed that the basic components, sets and matrices, are known. In practical situations, gathering this data requires specific procedures and equipments which are described in this subsection.

- The set R_b consists in an inventory of the assets and can be determined by doing on-site visits or surveys,
- The set V_b is the vulnerability library that can be built through the use of databases provided with known vulnerability scanners (e.g., Nessus, etc.). Expert opinion can also be used, particularly for human behavior vulnerabilities,
- The sets of attacks A_b are obtained from Intrusion Detection Systems (IDSs). Furthermore, attack scenarios are deduced by the mean of attack trees that were first introduced by B. Schneier in [6] and then discussed in [7]. Our purpose is to determine and assess the possible sequences of events that would lead to the occurrence of main attacks.

The root of a tree represents the main objective of an attacker while the subordinate nodes are elementary attacks that are necessary to perform for achieving the global goal. A *qualitative analysis* yields to a logical representation of the tree by reducing it to the form:

$$t_0 = S_1 \vee S_2 \vee .. \vee S_N, \tag{1}$$

where t_0 is the root of the tree and S_i, for $i \in \{1,..,N\}$, is the i^{th} attack scenario corresponding to t_0 and having the following structure:

$$S_i = t_1^{S_i} \wedge t_2^{S_i} \wedge .. \wedge t_{N_{S_i}}^{S_i}, \tag{2}$$

where $(t_j^{S_i})_{j \in \{1,..,N_{S_i}\}}$ are the elementary threats belonging to S_i.
For instance, the tree shown in figure 2 can be reduced to the following expression:

$$a = (b \wedge e \wedge f) \vee c \vee (d \wedge g) \vee (d \wedge h). \tag{3}$$

- The impact of performing an attack on a given resource is assessed by the risk analysis team members. This task is among the hardest in the risk analysis process as different kinds of potential effects should be translated to monetary values (e.g. loss of popularity, etc.),
- Determining the probability of an attack is a crucial issue that has an important influence on the performance of the risk analysis method. In NetRAM, probabilities are determined on the basis of expert opinions. Statistics provided by renowned institutes can be helpful in this context,
- Similarly, the evaluation of the influence of counter-measures on attacks, represented by the matrices $\mathbf{I_I}$ and $\mathbf{I_{II}}$, relies on human intervention.

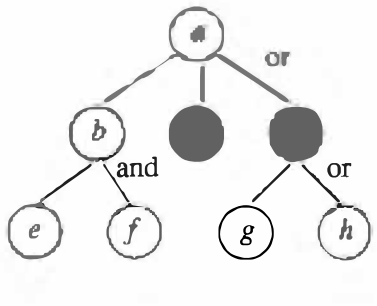


Figure 2: Typical attack-tree.

The three latest points show that even NetRAM processes can be automated to high rate, the human factor remains a basic component of the risk management cycle. To this end, the risk analysis team members should be chosen appropriately to guarantee the efficiency of the method. For example, technical skills should be a good selection factor.

Concerning the matrix $\mathbf{M}$, it can be seen a the product of two matrices $\mathbf{E}$ ($card(A) \times card(V)$-size) and $\mathbf{P}$ ($card(V) \times card(R)$-size) containing binary elements according on the following rules.

$$\forall (k,l) \in \{1,..,card(A)\} \times \{1,..,card(V)\}$$

$$\begin{cases} \mathbf{E}_{kl} = 1 & \text{if the attack } a_k \text{ exploits the vulnerability } v_l, \\ \mathbf{E}_{kl} = 0 & \text{if not.} \end{cases} \tag{4}$$

$$\forall (l,j) \in \{1,..,card(V)\} \times \{1,..,card(R)\}$$

$$\begin{cases} \mathbf{P}_{lj} = 1 & \text{if the vulnerability } v_l \text{ is present in the asset } r_j, \\ \mathbf{P}_{lj} = 0 & \text{if not.} \end{cases} \tag{5}$$

$\mathbf{E}$ can be built using public attack databases such as Mitre's CVE [10] (Common Vulnerabilities and Exposures) or the NIST I-cat project [11]. $\mathbf{P}$ can be filled using various vulnerability detection mechanisms. In our case, automated scanners and questionnaires are used.

3.4 Data update

Since NetRAM is a continuous process that involves a periodical re-assessment of security risks, its quantitative parameters must be updated depending on the changes that occur in the studied environment. For instance, the probability of an attack can be assimilated to its frequency of occurrence. Indeed, supposing two occurrences of an attack are independent events, it can be stated, according to the central limit theorem, that is a good estimation of the probability of the attack.

Moreover, attack and vulnerability databases (A and V) should be maintained in order to take into consideration the appearance of new threats and exploits. This is addressed by implementing a learning system based on neural networks [12]. Discussing this mechanism is beyond the scope of this paper and will be addressed in a future work.

This shows that putting and implementing an efficient update policy assumes the use of a set of sensors as well as several event collectors and analyzers. A distributed hybrid (host-based and network-based) IDS presents a good alternative to achieve this objective, if enriched with an appropriate learning mechanism.

4 Network state definition and management

Several metrics that help expressing the state of the analyzed network should be measured in a continuous way to permit an efficient detection of various anomaly. [13] offers a good list of such

metrics. In particular, a warning system has to be implemented to generate alerts if a metric (input signal) exceeds a given threshold value.

Define X to be a random variable that represents the actual percentage of use of a resource (e.g., CPU, memory, and disk space) and Y a random variable that represents the estimated percentage of use of the same resource. Let $p(x)$ be the probability density function (pdf) of X and $p(y|x)$ the conditional pdf of Y, given X. Then, the posterior pdf $p(x|y)$ can be expressed as:

$$p(x|y) = \frac{p(y|x)p(x)}{k(y)}, \tag{6}$$

where $k(y) = \int_0^\infty p(y|x)p(x)dx$, according to Bayes rule.

The warning threshold y^* is defined by the fact that the domain for issuing an alert is $\{y \geq y^*\}$. If c denotes the capacity of the considered resource available in the system under analysis (CPU speed, amount of memory, etc.), the probability that this capacity will be crossed, conditioned by the estimation y is $P(x \geq c|y)$. The decision system is represented by four cost functions:

- $D_{00}(y, c)$: no exceed, no alert,
- $D_{10}(y, c)$: alert present, no exceeds,
- $D_{01}(y, c)$: no alert, exceeding noticed,
- $D_{11}(y, c)$: alert and exceeding noticed.

Thus, the expected property loss without a warning system is

$$R_1(c) = \int_0^\infty q(y, c)D_{01}(y, c)k(y)dy, \tag{7}$$

$q(y, c)$ is the exceeding probability $P(x > c|y)$.

The expected property loss with a warning system is equal to

$$R_2(c) = \int_0^{y^*} (q(y, c)D_{01}(y, c) + (1 - q(y, c))D_{01}(y, c))k(y)dy$$

$$+ \int_{y^*}^\infty (q(y, c)D_{11}(y, c) + (1 - q(y, c))D_{10}(y, c))k(y)dy.$$

The warning threshold is then determined by solving the problem

$$\mathbf{max}(R_1(c) - R_2(c)).$$

5 Conclusion and Perspectives

In this paper, we have exposed a risk management method, called NetRAM. We began by defining the main processes of this method and detailing the constituency of each of them. We attempted to palliate the theoretical lacks of the commonly used methods. Therefore, a substantial portion of the paper was devoted to outlining the theoretical developments related to the proposed method. In our future work, we plan to extend the formalisms introduced in this paper in order to obtain a global mathematical framework that represents NetRAM's risk management cycle.

References

[1] Christopher J. Alberts, Audrey J. Dorofee, "*Managing Information Security Risks: The OCTAVE Approach,*" Addison Wesley Professional, ISBN: 0321118863, *July 2002*.

[2] G. Stonebumer, A. Grogen, A. Fering, "*Risk Management Guide for Information Technology Systems,*" National Institute for Standards and Technology, special publication 800-30.

[3] "*A Guide to Risk Management and Safeguard Selection for IT Systems,*" Government of Canada, Communications Security Establishment, January 1996.

[4] "*Information Security Risk Assessment: Practices of Leading Organizations*," United States General Accounting Office, GAO/AIMD-00-33, November 1999.

[5] J. Kontio, G. Getto, D. Landes, "*Experiences in Improving Risk Management Processes Using the Concepts of the Riskit Method,*" SIGSOFT98, Sixth Symposium of the Foundations of Software Engineering, November 98.

[6] Bruce Schneier, "*Secrets and Lies: Digital Security in a Networked World,*" John Wiley & Sons,ISBN: 0471253111, 2001.

[7] A. P. Moore, R. J. Ellison, R. C. Linger, "*Attack Modeling for Information Security and Survivability*", Carnegie Mellon University, Technical Note, CMU/SEI-2001-TN-001.

[8] R. E. Rosenthal, "*Concepts, Theory, and Techniques: Principles of Multi-objective Optimization*", Decision Sciences, vol. 16, pp. 133-152, 1985.

[9] K. Deb, "Evolutionary Algorithms in Engineering and Computer Design," John Wiley &Sons, pp. 135-161, 1999.

[10] FedCIRC, U.S. General Services Administration, "Common Vulnerabilities and Exposures," http://www.cve.mitre.org.

[11] National Institute for Standards and Technology, Computer Security Division"The ICAT Project," http://icat.nist.gov.

[12] C. M. Bishop, "Neural Networks for Pattern Recognition," Oxford University Press, Oxford, 1995.

[13] "*Detecting signs of intrusions*", Carnegie Mellon University/Software Engineering Institute, Security Improvement Modules (m-09), December 2000.

[14] D. J. Marchette, "Computer intrusion Detection and Network Monitoring: A Statistical Viewpoint," Springer-Verlag, ISBN: 0387952810,2001.

Trust Evaluation Based Security Solution in Ad Hoc Networks

Zheng Yan[1], Peng Zhang[2], Teemupekka Virtanen[3]

[1] Nokia Research Center, Nokia Group, Helsinki, Finland
[2] Nokia Venture Organization, Nokia Group, Helsinki, Finland
[3] Helsinki University of Technology, Finland
{zheng.z.yan, peng.p.zhang}@nokia.com,
teemupekka.virtanen@hut.fi

Abstract. Ad hoc networks are new paradigm of networks offering unrestricted mobility without any underlying infrastructure. The ad hoc networks have salient characteristics that are totally different from conventional networks. These cause extra challenges on security. In an ad hoc network, each node should not trust any peer. However, traditional cryptographic solution is useless against threats from internal compromised nodes. Thus, new mechanisms are needed to provide effective security solution for the ad hoc networks. In this paper, a trust evaluation based security solution is proposed to provide effective security decision on data protection, secure routing and other network activities. Logical and computational trust analysis and evaluation are deployed among network nodes. Each node's evaluation of trust on other nodes should be based on serious study and inference from such trust factors as experience statistics, data value, intrusion detection result, and references of other nodes, as well as node owner's preference and policy. In order to prove the applicability of the proposed solution, authors further present a routing protocol and analyze its security over several active attacks.

KEYWORDS: trust, security, ad hoc networks

1 Introduction

Ad hoc networks are new paradigm of networks offering unrestricted mobility without any underlying infrastructure. An ad hoc network is a collection of autonomous nodes or terminals that communicate with each other by forming a multi-hop radio network and maintaining connectivity in a decentralized manner. Each node functions as both a host and a router. More critically, the network topology is in general dynamic, because the connectivity among the nodes may vary with time due to node departures, new node arrivals, and the possibility of having mobile nodes. There are two major types of wireless ad hoc networks: Mobile Ad Hoc Networks (MANETs) and Smart Sensor Networks (SSNs) [1]. In this paper, our discussion will

mainly focus on the MANETs. Significant applications of MANETs include establishing survivable, efficient, dynamic communication for emergency/rescue operations, disaster relief efforts, and military networks that cannot rely on centralized and organized connectivity.

Operation in an ad hoc network introduces new security problems. The ad hoc networks are generally more prone to physical security threats. The possibility of eavesdropping, spoofing, denial-of-service, and impersonation attacks increases [1]. Similar to fixed networks, security of the ad hoc networks is considered from the attributes such as availability, confidentiality, integrity, authentication, non-repudiation, access control and usage control [2, 3]. But security approaches used for the fixed networks are not feasible due to the salient characteristics of the ad hoc networks. New threats, such as attacks raised from internal malicious nodes, are hard to defend [4]. New security mechanisms are needed to adapt the special characteristics of the ad hoc networks.

Trust is an important aspect in the design and analysis of secure distribution systems [5]. It is also one of the most important concepts guiding decision-making [6]. Trust is a critical part of the process by which relationships develop [7]. It is a before-security issue in the ad hoc networks. By clarifying the trust relationship, it will be much easier to take proper security measures, and make correct decision on any security issues. A trust model specifies, evaluates and sets up trust relationship among entities. Trust modeling is a technical approach to represent trust for digital processing. Recently, trust modeling is paid more and more attention in electronic systems. Current trust academic work covers such aspects as analyzing the problems of current secure systems [8, 9], proposing models for achieving trust in digital systems [10-12] and quantifying or specifying trust in digital systems [13,14].

In this paper, the authors study the security problems in the ad hoc networks and propose a trust evaluation based security solution. The rest of the paper is organized as follows. Section two discusses the security problems in the ad hoc networks. Section three presents the current security schemes in the literature. In section four, a trust evaluation based solution for the ad hoc networks is proposed. In the next section, the solution is illustrated by a routing protocol and proved by analyzing its security against several active attacks. In section six, the authors further discuss the solution and present its characteristics. Finally, the conclusions and directions of future work are given in the last section.

2 Security Problems in Ad Hoc Networks

The salient characteristics of the ad hoc networks pose challenges to security [2-4].

First of all, the use of wireless link renders an ad hoc network susceptible to link attacks ranging from passive eavesdropping to active interfering. Unlike fixed hardwired networks with physical defense at firewalls and gateways, attacks on an ad hoc network can come from all directions and target at any node. Damage includes leaking secret information, interfering message and impersonating nodes, thus

violating the basic security requirements. All these mean that every node must be prepared for encounter with an adversary directly or indirectly.

Secondly, autonomous nodes in an ad hoc network have inadequate physical protection, and therefore more easily to be captured, compromised, and hijacked. Malicious attacks could be launched from both outside and inside the network. Because it is difficult to track down a particular mobile node in a large scale of ad hoc network, attacks from a compromised node are more dangerous and much harder to detect. All these indicate that any node must be prepared to operate in a mode that should not immediately trust on any peer.

Thirdly, any security solution with static configuration would not be sufficient because of the dynamic topology of the networks. In order to achieve high availability, distributed architecture without central entities should be applied. This is because introducing any central entity into security solution may cause fatal attack on the entire network once the centralized entity is compromised. Generally, decision making in the ad hoc networks is decentralized and many ad hoc network algorithms rely on the cooperation of all nodes or partial nodes. But new type of attacks can be designed to break the cooperative algorithm. Malicious nodes could simply block or modify the data traffic traversing them by refusing the cooperation or hacking the cooperation. As can be seen from the above, no matter what security measures are deployed, there is always some vulnerability that can be exploited to break in.

It seems difficult to provide a general security solution for the ad hoc networks. Traditional cryptographic solution is not adapted for the new paradigm of the networks. As can be seen from the above analysis, what is lacked in the ad hoc networks is trust since each node must not trust any other node immediately. If the trust relationship among the network nodes is available for every node, it will be much easier to select proper security measure to establish the required protection. It will be wiser to avoid the un-trusted nodes as routers. Moreover, it will be more sensible to reject or ignore hostile service requests. Therefore, the trust evaluation becomes a before-security issue in the ad hoc networks. Its security solution should be dynamic based on the changed trust relationship.

3 Related Work

Current security study for the ad hoc networks is scattered on special topics such as intrusion detection, secure routing, and key management.

3.1 Intrusion detection

The ad hoc networks have inherent vulnerabilities that are not easily preventable. Intrusion prevention measures, such as encryption and authentication, are required to protect network operation. But these measures cannot defend compromised nodes, which carry their private keys. Intrusion detection presents a second wall of defense.

It is a necessity in the ad hoc networks to find compromised nodes promptly and take corresponding actions to against. A distributed and cooperative architecture for better intrusion detection was proposed in [3]. Based on the proposed architecture, a statistical anomaly detection approach is used. The detection is done locally in each node and possibly through cooperation with all nodes in the network. But how to define the anomaly models based on which trace data is still a main challenge.

3.2 Secure routing

In the ad hoc networks, routing protocol should be robust against topology update and any kinds of attacks. Unlike fixed networks, routing information in an ad hoc network could become a target for adversaries to bring down the network. There are two types of threats. The first one comes from external attackers. The attacks include injecting erroneous routing information, replaying old routing information, and distorting routing information. With these ways, the attackers can successfully partition a network or introduce excessive traffic load into the network, thus cause retransmission and ineffective routing. Using cryptographic schemes, such as encryption and digital signature can defend against the external attacks. The second threat comes from compromised nodes, which might send malicious routing information to other nodes. Typical attacks fallen into this category are black hole attacks, routing table overflow attacks, impersonation and information disclosure, etc. [4]. The internal attacks from malicious nodes are more severe because it is very difficult to detect because the compromised nodes can also generate valid signature. Existing routing protocols cope well with the dynamic topology, but usually offer little or no security measures [2].

In [18], a set of design techniques for intrusion resistant ad hoc routing algorithm (TIARA) was presented mainly to against denial-of-service attacks. Secure aware ad hoc routing (SAR) in [15] uses security properties (e.g. time stamp, sequence number, authentication password or certificate, integrity, confidentiality, and non-repudiation) as a negotiable metric to discover secure routes in an ad hoc network. The SAR can be implemented based on any on-demand ad hoc routing protocol with suitable modification. But it only considers the effect of security properties on the trust. In [4], a secure routing solution is proposed for the black hole problem. But unfortunately, this solution does not solve the problem caused by cooperation of multiple malicious nodes.

3.3 Key management

Traditional cryptographic mechanisms, such as digital signature and public key encryption, still play vital roles for the security of the ad hoc networks. All these mechanisms require a key management service to keep track of key and node binding and assist the establishment of mutual authentication between communication nodes. Traditionally, the key management service is based on a trusted entity called a certificate authority (CA) to issue public key certificate of every node. The trusted CA

is required to be online in many cases to support public key revocation and renewal. But it is dangerous to set up a key management service using a single CA in an ad hoc network. It will be the vulnerable point of the network. If the CA is compromised, the security of the entire network is crashed. In [2] and [16], a threshold cryptography is used to provide robust and ubiquitous security support for the ad hoc networks. The CA functions are distributed through a threshold secret sharing mechanism. This approach is very complicated to implement. It is also hard to survive from multiple hijacked nodes that have secret shares.

The security for the ad hoc networks is still in its infancy. Existing solutions cannot solve this issue well. What is missed is an effective mechanism that can provide reasonable inference based on available knowledge, such as intrusion detection result, past experience, communication data value, and preferences, to evaluate trust relationship among network nodes. With the evaluation result, it is possible to make correct decision or close-correct decision on security protection. New mechanisms are expected to adapt the special characteristics of the new network paradigm.

4 Trust Evaluation Based Security Solution

In this section, the authors propose a trust evaluation based security solution for the ad hoc networks. It introduces a fair and rational security mechanism into the ad hoc networks by simulating human being's decision-making procedure. The perfect security may not be reached, but the average security level should be satisfied based on accumulated knowledge and experience, as well as trust relationship established and adjusted. The decision-making on data protection approach, secure route selection, and any other activities related to security should be based on trust analysis and evaluation.

4.1 Trust modeling

Trust modeling is a technical approach to represent trust for digital processing. Herein, two trust models are proposed based on two ad hoc system models. One is an independent model that represents independent ad hoc networks without any connection to the fixed networks, as shown in Figure 1. The other is a cross model that represents ad hoc networks with few connections to the fixed networks, as shown in Figure 2. In both models, the basic unit that represents an ad hoc node is a Personal Trusted Bubble (PTB). In the bubble, the owner of the ad hoc device has illogically full trust on the device, which is responsible for the ad hoc communication and organization. Among bubbles and between the bubbles and the fixed networks, logical and rational trust relationship should be evaluated computationally. The above trust evaluation is conducted digitally ahead of any communication and the evaluation result should be considered for better security decision.

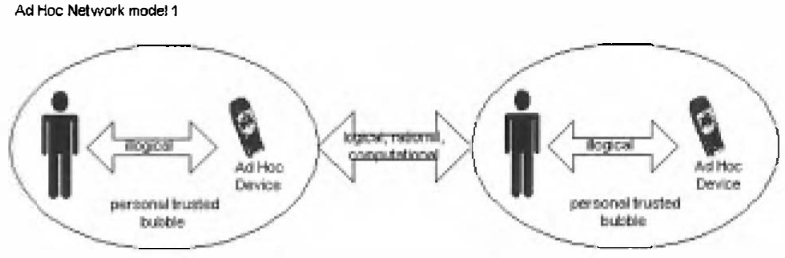


Fig. 1. Independent model (without connection to fixed networks)

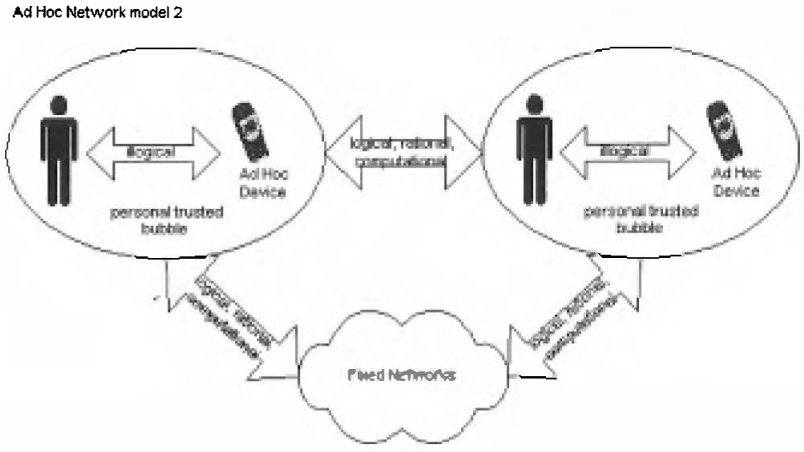


Fig. 2. Cross model (with few connections to fixed networks)

4.2 Trust evaluation mechanism

Based on the study of the trust definition in [17], it is understood that trust is a concept hard to define because it is itself a vague term. The trust defined herein is the confidence of an entity (PTB) on another entity (PTB) based on the expectation that the other entity will perform a particular action important to the trustor, irrespective of the ability to monitor or control that other entity. The level of trust considered sufficient may be different for different individuals (ad hoc device owners in PTB – PTB owners). It is also dynamic because it is affected by many changeable factors.

In the modeled ad hoc networks, the trust evaluation mechanism is introduced into each PTB. The trust relationship between the host bubble and other bubbles is evaluated digitally according to the knowledge accumulated and subjective factors of the bubble owner. In each bubble, there is a trust matrix which stores the knowledge

used for trust evaluation on every other bubble, as show in Figure 3. The factors that may affect the trust are as follows. Note B(i) stands for a PTB (i.e. a node in the ad hoc networks described below).

B(i)	experience statistics	data value	reference	personal preference	PTB policy	intrusion black list	..... others
B(1)	Ves(i,1)	Vd(i,1)	Vr(i,1)	rx(i, 1, a)	Vp(i,1, a)	1/0	Vo(i,1)
B(2)	Ves(i,2)	Vd(i,2)	Vr(i,2)	rx(i, 2, a)	Vp(i,2, a)	1/0	Vo(i,2)
...							
B(i-1)	Ves(i,i-1)	Vd(i,i-1)	Vr(i,i-1)	rx(i, i-1, a)	Vp(i, i-1, a)	1/0	Vo(i,i-1)
B(i+1)	Ves(i,i+1)	Vd(i,i+1)	Vr(i,i+1)	rx(i, i+1, a)	Vp(i,i+1,a)	1/0	Vo(i,i+1)
...							
B(n)	Ves(i,n)	Vd(i,n)	Vr(i,n)	rx(i, n, a)	Vp(i, n, a)	1/0	Vo(i,n)

Fig. 3. Trust evaluation matrix

Experience statistics: This is statistic data of prior experience accumulated during the communications with other nodes. The communication success through some node will increase the trust index of that node. The communication failure through that node will decrease the trust index attached to that node. Just like human being's communications, the trust we established on one person is generally based on the proportion of communication success and the level of satisfaction. By digitizing the value of the experience statistics, its value can be expressed as:

$V_{es}(i, j) = F_{ES}$ (proportion of successful communication between B(i) and B(j), level of satisfaction from B(i) to B(j)), where F represents a function.

Data value: This is the value of communication data. The higher value of data, the higher trust needed from other PTBs to transfer. The data value sent from B(i) to B(j) can be expressed as:

$V_d(i, j) = F_D$ (importance of data transferred between B(i) and B(j), security level of the system and B(i))

Intrusion black list: The black list of malicious nodes based on intrusion detection of the host PTB. The value of intrusion black list can be expressed as:

$V_{ibl}(i, j) = 1/0$. 1: B(j) is good node treated by B(i); 0: B(j) is malicious node treated by B(i).

Reference: Some reference, such as other bubbles' recommendation, reputation of the evaluated node, and other PTBs intrusion detection report, may also impact the

final evaluation result, especially when other information is lacked at the beginning of the network running. The value of reference is expressed as:

$V_r(i, j) = F_R$ (other PTBs' recommendation on B(j), reputation of B(j), other PTBs' intrusion detection report on B(j), ...)

Personal preference: The bubble owner's personal preference also affects the decision of trust as a subjective factor. The rate of the trust factor is one example.

$r_x(i, j, a) = F_r$(preferred rate of B(i) on x factor when evaluating trust on B(j) on action a), where x can be *es*, *d*, *r*, etc.
Actually, $r_x(i, j, a)$ is a set of values for different network actions, $r_x(i, j, a) = \{ r_x(i, j, a_k) \mid k = 1,\ldots,n) \}$.

PTB policy: Like the personal preference, the PTB's policy is also a subjective factor that affects the trust evaluation result. It is related to the whole network's security requirements and policy. It also affects the personal preferences. Most importantly, the trust threshold is also decided by the PTB's policy. In addition, the policy can also be tailored for different PTBs. The value of B(i)'s policy on some special action *a* for B(j) can be described as:

$V_p(i, j, a) = F_P$ (network's security policy, B(i)'s security policy on B(j), basic security requirements, ...)
The $V_p(i, j, a)$ is in practice a set of values for different network actions, $V_p(i, j, a) = \{ V_p(i, j, a_k) \mid k = 1,\ldots,n)\}$.

Other factors (e.g., frequency of routing request from a node, energy left, etc) can also be considered in the trust evaluation on particular action if needed. They can be involved into the evaluation based on the preferred evaluation algorithms.
$V_o(i, j) = F_O$ (frequency of routing request message from B(j), energy left on B(i), ...)

Herein, the authors suggest a linear function that can work for the simple trust evaluation. $TE_a(i, j)$ stands for trust evaluation result conducted by B(i) on B(j) for particular action *a*. It is calculated by considering the objective factors tailored by the subjective factors.
$TE_a(i, j) = [r_{es}(i, j, a)* V_{es}(i, j) + r_d(i, j, a)* V_d(i, j) + r_r(i, j, a)* V_r(i, j) + r_o(i, j, a)* V_o(i, j)] * V_{ibl}(i, j)$

In the above function, $r_x(i, j, a)$ (x = es, d, r, or o) is a factor rate that is decided by the personal preference. The total sum of $r_x(i, j, a)$ is 1 and the value of $r_x(i, j, a)$ may be different for different actions. The experience statistics, data value, reference and other factors can be digitized and applied a digital value. In addition, it is noted that the value of intrusion black list, $V_{ibl}(i, j)$, is either 1 or 0. Therefore, the result of the above function is a value. If the value exceeds the trust threshold $V_p(i, j, a)$ defined by the PTB policy on the particular action, the host PTB (B(i)) can trust the evaluated PTB (B(j)) on that action. If the value is below the trust threshold, the host PTB (B(i)) can avoid using the evaluated PTB (B(j)) or apply corresponding protection to the particular action. Other trust evaluation algorithms (such as those in [19-21]) could also be applied, but they may not be quite adaptive to the ad hoc networks.

5 Secure Routing Based on Trust Evaluation

In this part, a *source-initiated on-demand driven* routing is illustrated as an example to apply the above trust evaluation based security solution into secure ad hoc routing. Here, it is assumed that the ad hoc nodes can authenticate with each other correctly. In order to follow traditional routing description, node herein instead of PTB is used for easy understanding. The routing algorithm is described as follows.

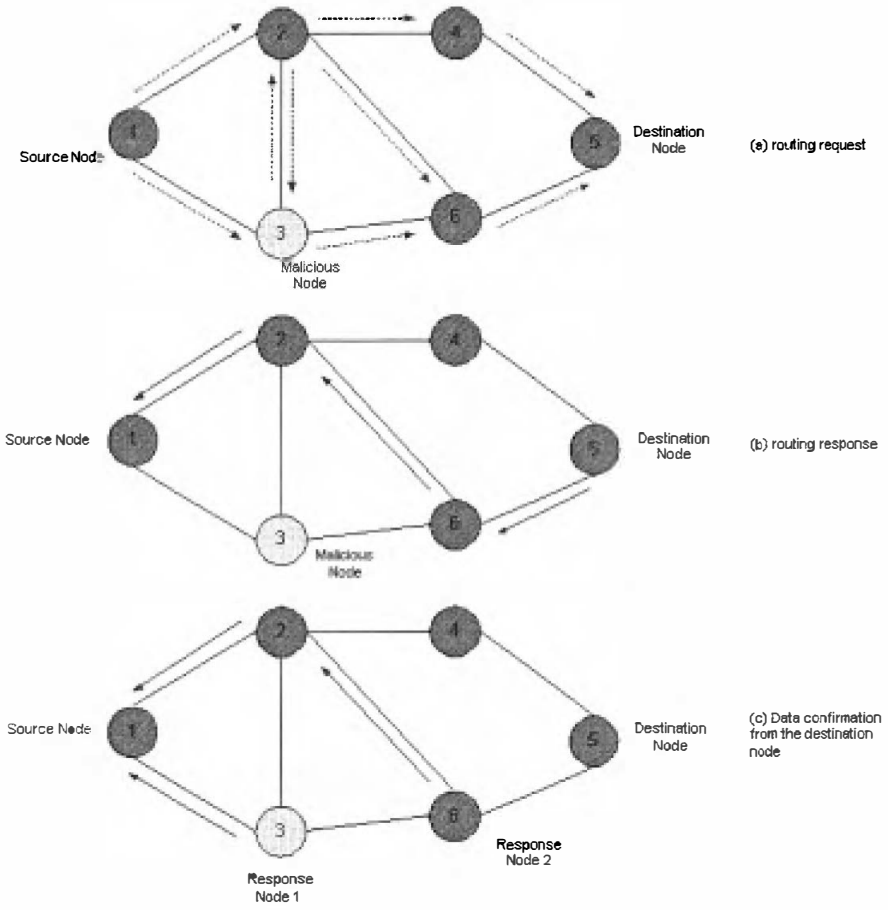


Fig. 4. Secure routing with trust evaluation

1. Source node broadcasts routing request message to its neighbors in order to find a route to destination node.
2. The neighbors of the source node forward the request to their neighbors if the trust evaluation on the source node pass its predefined threshold, and so on, until either the destination or an intermediate node with a "fresh enough" route to the

destination is reached. And that node would like to accept the data transfer based on its trust evaluation. (Figure 4 (a))

3. If some nodes respond that they have fresh enough route to the destination node and would like to reserve some time slot for serving data transfer, the source node checks the trust evaluation matrix and conducts the trust evaluation on the responded nodes. Based on the evaluation result and hops of the routes, the source node selects one preferred route, which it believes the best. (Figure 4 (b))
4. The source node sends (test) data packages to the destination using the selected route and set preferred time slot waiting for the destination node's confirmation and indicates that which package is required to respond confirmation of receival.
5. After receiving the data packages, the destination node applies the same method above to reply the confirmation message if the source node requests it. It is not mandatory to use the same route as the source for better security consideration. (Figure 4 (c))
6. If within the time slot, the destination's confirmation arrives and can be verified as valid, the source node will continue sending data packages via the underlying route. If the destination's confirmation cannot receive within the preferred time slot, the source node will update its trust evaluation matrix data on the routing nodes by reducing the trust value of experience statistics. If the source node makes sure the response node of underlying route is malicious, it will put the node into the intrusion black list, set that value to 0. The source node also propagates the malicious node over the networks. This information is used for updating the reference of other nodes' trust evaluation matrix and the update should also follow the trust evaluation on the source node. Then processing either jumps to step 1 for higher security or goes to step 7 for better performance.
7. The source node selects the second best route. Then go to step 4.

The proposed protocol can be implemented based on any on-demand ad hoc routing protocol with suitable modification and by adding knowledge accumulation and trust evaluation mechanism. Next, we further evaluate the security of our proposed routing protocol by analyzing it over several active ad hoc routing attacks described in [4].

Black hole attacks: In this attack, a malicious node uses the routing protocol to advertise itself as the shortest path to other nodes. The proposed routing protocol can defend this attack because it randomly requires the destination node's confirmation of the data package. If the source cannot receive the confirmation within the indicated time slot, it will change the route. In addition, the confirmation message may not be transferred via the same route as the source node selected. The route of the confirmation message is selected based on the destination's trust evaluation matrix. In addition, the confirmation response is requested randomly by the source node. Therefore, it will greatly reduce the risk that the confirmation message is intentionally transferred by the malicious node to the source node. What is more, if the malicious node is found by any node in the network, this attack can be avoided in our protocol based on the trust evaluation mechanism.

Denial of service: The DoS attack happens when the network bandwidth is hijacked by a malicious node. Any intrusion detection mechanism can be deployed and its result will contribute to the trust evaluation matrix, therefore affect any security-related decision. For instance, a malicious node might generate frequent route requests to make the network resources unavailable to other nodes. The proposed protocol fights against this attack in the following way. In step 2, the neighbor node processes the routing request according to the trust evaluation, in which the frequency of routing request message from a node is considered as one of main factors. If the frequency of request exceeds the threshold defined in the PTB's policy, the neighbor node will ignore the request. And at the same time, the neighbor node may broadcast the possibility of intrusion in the network. Any intrusion report broadcast in the network is recorded by every node and used for updating the value of reference in the trust evaluation matrix.

Routing table overflow attacks and energy consummation: In the first attack, the attacker attempts to create routes to nonexistent nodes. The goal is to have enough routes so that creation of new routes is prevented or the implementation of routing protocol is overwhelmed. In the second attack, an attacker can attempt to consume batteries by requesting routes or forwarding unnecessary packets to a node. In the proposed protocol, every node has right to ignore or reject route serving or data receiving according to the trust and ability evaluation. And the service time for other nodes can be set according to the evaluation result. In this way, it can be effectively against these kinds of attacks.

6 Further discussion

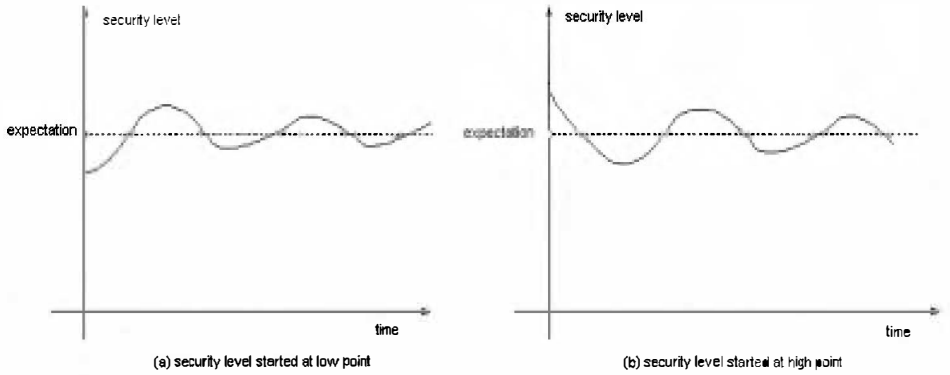


Fig. 5. Gradual security achievement

The security for the ad hoc networks is still in its infancy. Since the ad hoc networks are dynamic by nature, they require a dynamic security solution that fits this fundamental characteristic.

The proposed solution tries to simulate human being's social contact procedure on decision-making and introduces it into the ad hoc networks. The perfect security solution is hard to reach. But the average security level (for a node) can be achieved as expectation based on accumulated knowledge and as well as the trust relationship built and adjusted. With this way, it could greatly reduce security threats. As shown in Figure 5 (a), at the beginning, the security level reached may be quite low because the nodes (potentially malicious) do not have much knowledge about other nodes. With time elapsing, the nodes know each other more and more. So the trust evaluation on particular actions is more and more close to correctness. This causes the security level reaching its expectation. Figure 5 (b) shows another case where every node is a good node and full of capability at the beginning. So the security achieved at the start point is high. With the network running, some nodes are compromised. Some lack energy. Those cause security level to decrease. On the other hand, the trust evaluation is more and more correct. It directs any decision on security and pushes the security level reaching the expectation.

But the above trust evaluation result may not keep correct longer because of the dynamic characteristic of the network and its vulnerabilities. Further understanding is needed among the nodes. The trust evaluation will be close to correctness gradually since knowledge and experience accumulated by every node should not be updated frequently and totally. This is because the possibility of the whole network crash is low.

Even though there are some problems left, this method will help in avoiding further loss. The proposed mechanism is also flexible to resist new attacks by introducing new factors into the trust evaluation. Due to the dynamic characteristics of the networks, it is suggested that the trust evaluation should be conducted at real time if the security requirement is high. The authors call this solution as gradual-security approach.

7 Conclusions and future work

The new paradigm of the ad hoc networks presents new challenges on security due to its salient characteristics that are totally different from the conventional wired and wireless networks. In this paper, the authors studied the security issues in the ad hoc networks and analyzed the problems. The existing solutions cannot solve the security issues for the ad hoc networks well.

Based on the study, a trust evaluation based security solution was proposed by introducing human being's social contact procedure into any security-related decision-making. The authors believe that data protection approach, secure route selection, and any other decision related to security should be based on trust analysis and evaluation among network nodes. Based on this mechanism, the authors further applied the mechanism to a *source-initiated on-demand driven* routing protocol and analyze its security over several active attacks. The analysis showed that the proposed protocol against those attacks effectively. In addition, we further discussed the solution as a gradual-security solution, which can achieve average security level as expectation

based on knowledge and experience accumulation and inference. It is hard to achieve perfect security, but it is possible to greatly reduce the threats.

Immediate future work includes study of efficient and effective trust evaluation algorithm, simulation and proof of the proposed routing protocol. The authors are also working on how to establish basic trust identity to enforce the trust analysis and evaluation are conducted on the correct target nodes.

References

[1]S. Corson, J. Macker. Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations. IETF RFC2501, 1999.

[2]L. Zhou, Z. J. Haas. Securing Ad Hoc Networks. IEEE Network, 13(6): 24-30, Nov/Dec 1999.

[3]Yongguang Zhang, Wenke Lee. Intrusion Detection in Wireless Ad-Hoc Networks. Proceedings of MobiCom 2000, Sixth Annual International Conference on Mobile Computing and Networking, Boston, MA, USA, 6-11 Aug. 2000.

[4]HongMei Deng, Wei Li, Dharma P. Agrawal. Routing Security in Wireless Ad Hoc Networks. IEEE Communications Magazine, October 2002, p70-75.

[5]Diamadi, Z. Fischer, M.J. A simple game for the study of trust in distributed systems. International Software Engineering Symposium 2001 (ISES'01), Wuhan University Journal of Natural Sciences Conference. March 2001.

[6]Shillo, M.; Funk, P.; Rovatsos, M. Using trust for detecting deceitful agents in artificial societies. Applied Artificial Intelligence, vol.14, no.8, p.825-48 Sept. 2000.

[7]Warne, D., Holland, C.P. Exploring trust in flexible working using a new model. BT Technology Journal, vol.17, no.1, p.111-19. Jan 1999.

[8]Gerck Ed. Overview of Certification System: X.509, PKIX, CA, PGP & SKIP. In http://www.thebell.net/papers/certover.pdf

[9]Perlman, R. An overview of PKI trust models. IEEE Network, vol.13, no.6 p.38-43.

[10]Yao-Hua Tan. Thoen, W. Toward a generic model of trust for electronic commerce. International Journal of Electronic Commerce vol.5, no.2, p.61-74.

[11]Egger Florian N. Towards a Model of Trust for E-Commerce System Design. In Proc. Of the CHI2000 Workshop: Designing Interactive Systems for 1-to-1 E-commerce.

[12]Abdul-Rahman Alfarez, Halles Stephen. A Distributed Trust Model. In Proc. Of New Security Paradigms Workshop, ACM, New York, NY, USA, 1998.

[13]Daniel W. Manchala, Xerox Research and Technology. E-Commerce Trust Metrics and Models. IEEE Internet Computing, vol.4, no.2 p.36-44, 2000.

[14]Mui Lik, Mohtashemi Mojdeh, Halberstadt Ari. A Computational Model of Trust and Reputation. In Proc. Of the 35th Annual Hawaii International Conference on System sciences, 7-10 Jan. 2002, Big Island, HI, USA

[15]Seung Yi, Prasad Naldurg, Robin Kravet. Security-aware ad hoc routing for wireless networks. http://www.cs.uiuc.edu/Dienst/Repository/2.0/Body/ncstrl.uiuc_cs/UIUCDCS-R-2001-2241/pdf

[16]Jiejun-K, Petros-Z, Haiyun-Luo, Songwu-Lu, Lixia-Zhang. Providing robust and ubiquitous security support for mobile ad-hoc networks. Proceedings Ninth International Conference on Network Protocols. ICNP 2001, Riverside, CA, USA, 11-14 Nov. 2001

[17]McKnight, D. Harrison, Chervany Norman L. What is Trust? A Conceptual Analysis and An Interdisciplinary Model. In Proceedings of the 2000 Americas Conference on Information Systems (AMCI2000). AIS, Long Beach, CA, August 2000..

[18]Ramanujan-R, Ahamad-A, Bonney-J, Hagelstrom-R, Thurber-K. Techniques for intrusion-resistant ad hoc routing algorithms (TIARA). Proceedings of IEEE Military Communications Conference (MILCOM'00), vol.2, Los Angeles, CA, USA, 22-25 Oct. 2000.

[19]A.Jøsang. *An Algebra for Assessing Trust in Certification Chains*. In J.Kochmar, editor, Proceedings of the Network and Distributed Systems Security (NDSS'99) Symposium, The Internet Society, 1999.

[20]Daniel W. Manchala: Xerox Research and Technology. E-Commerce Trust Metrics and Models. IEEE Internet Computing, vol.4, no.2 p.36-44 (2000).

[21]Mui Lik, Mohtashemi Mojdeh, Halberstadt Ari: A Computational Model of Trust and Reputation. In Proc. Of the 35th Annual Hawaii International Conference on System sciences, 7-10 (Jan. 2002), Big Island, HI, USA.

Investigating Spyware on the Internet

Johan Wieslander
Dept. of Software Engineering and Computer Science
Blekinge Institute of Technology
SE-37125 Ronneby, Sweden
Email: jws@bth.se

Martin Boldt
Dept. of Software Engineering and Computer Science
Blekinge Institute of Technology
SE-37125 Ronneby, Sweden
Email: mbo@bth.se

Bengt Carlsson
Dept. of Software Engineering and Computer Science
Blekinge Institute of Technology
SE-37125 Ronneby, Sweden
Email: bca@bth.se

***Abstract*— Today, the distinction between a virus and an aggressive marketing tool on the Internet is sometimes insignificant. As an example, Peer-to-Peer (P2P) tools connected to the Internet often display ads (adware) or include activity monitoring or information collecting software that spies on the users (spyware). We present a method for examining P2P tool installations and present test results from a few of the most common P2P tools. The method itself will be evaluated and suggestions of refinements will be proposed. We will also discuss whether these tools, with their bundled software, make any privacy intrusions. An extended arms race is predicted in the future where a merged society of virus, spyware and adware is defeated by a combination of anti-virus and anti-spyware tools.**

***Index Terms*— Peer-to-Peer, adware, spyware.**

I. Introduction

When more and more people are getting connected to the Internet, the issue of privacy has become more important. Some techniques that were originally created to provide sensible functionality are today misused to monitor user activity. Examples of such techniques are web browser cookies, HTTP referrers and HTML image source tags. By using these techniques, and combining them, companies such as advertising agencies or media content providers can collect data for commercial use.

This kind of information gathering form the basis of two new techniques called adware and spyware. They are a kind of software that is generally bundled with (i.e. packaged with) other software, often free- or shareware such as games, audio/video players or P2P (Peer-to-Peer) tools. Although both of them are concerned with security and privacy issues, the main purpose of adware is to display ads on a web page or in a program GUI, rendering profit for the software copyright owners. Spyware, on the other hand, has the purpose of also gathering information about the user or the users' activities and to secretly send it to some predefined recipient. Martin et al.[10] reported such activities in a business-to-business browser holding bundled spyware.

Similar to web-based virus and anti-virus programs, there is an ongoing arms race[5][11] within the field of spyware. Unlike most viruses there is no immediate action being taking, i.e. as a user you cannot see the outcome of the battle. Instead of an all-or-nothing situation, spyware are gradually and indirectly affecting computer and web systems. An apparently well functioning anti-spyware, may lack capacity to detect a certain spyware. The difference, compared to a virus, is the response time within an arms race, making the upgrading delayed or even cancelled for the anti-spyware program.

A few good tools against adware and spyware exist, such as Ad-aware, but they typically accomplish a single or only a few tasks. There is a lack of tools when it comes to monitoring possible adware or spyware in runtime and logging their network activities.

We have chosen to examine P2P tools to find possible adware or spyware. All of the tools have the purpose of file sharing, except for ICQ, which is an instant messenger. The reason for choosing P2P tools is that several popular P2P tools are known to have contained spyware and adware. Furthermore, the use of P2P tools lies entirely in networking — they have no other purpose and are useless without a network connection. It is thus a fair assumption that a network connection is available, most likely to the Internet, whenever the P2P tool is running. Any bundled software is likely to succeed in attempts to transfer data, making it possible for both adware and spyware to work as intended and possibly also go undetected in file sharing P2P tools. Moreover, other common activities among Internet users are web surfing, e-mail transferring and instant messaging or chatting. Any spyware could have the opportunity to eavesdrop on that communication. Merely by monitoring instant messaging activities, not necessarily eavesdropping the content, sensitive information such as employee work habits[8] can be obtained.

The P2P tools that have been known to contain spyware have file sharing as their main purpose, in contrast to instant messaging services. Early P2P tools would share only mp3 music but the most popular tools today have the ability to share any kind of files. Sharing mp3 music, however, is still very popular. A typical mp3 song (3–5 MB) is considerably larger than a text file containing a couple of pages of text (say 20 KB). Transferring such an mp3 file would of course take much longer time than transferring the text file. With today's ADSL or ISDN networks, transferring a 20 KB file is a second's work. Also, a moderately fast home computer today could easily compress (at moderate compression) even a hundred pages of text without showing any noticeable load. Thus, it would be no problem compressing and transferring large amounts of data collected on a P2P tool user's computer. Keeping the activity hidden from the user would also not be

a problem.

Finally, we put forward the questions if the currently most common P2P tools contain spyware, if these activities intrude on user privacy and how a method for investigating them could be constructed. In section II we give further background to the P2P society considered, followed by the investigation in section III and an analysis in section IV. Section V discusses the results found and, finally, in section VI we conclude the work.

II. Background

Privacy intrusion[1][7][12] may involve one or more of the following actions:

- Information about the computer or the user(s) is sent without explicit permission. Such information could be hardware configuration, operating system and version, installed software, system configuration (computer name), user name, number of users or account settings (group).
- Explicit location information (other than traceroute equivalent results) is sent without explicit permission.
- Information about file names and file formats is collected and sent without explicit permission (both files on disk and files being searched for, or transferred, in the P2P tool).
- Information is collected from browser cache, history or cookies and sent.
- Information is collected from other files on the hard drive, or any other device that is connected to the computer, and sent.

It is important to understand that information gathering software is not considered spyware. Even software that gathers information about the user and then sends this information to some central processing unit is not considered spyware, as long as the user is notified about these activities in advance.

Protocols that use the peer-to-peer model[8] to communicate let all hosts (peers) communicate on equal conditions "In essence, peer-to-peer simply means equal communicating with equal"[8]. This means that any peer in a P2P network can connect to any other peer on that network. A comparison to the client/server model[14] would be that any peer is both a client and a server at the same time and therefore equal communication between these nodes is possible.

III. Investigation

The P2P tools selected were BearShare, iMesh, Kazaa, and Morpheus together with the instant messaging service ICQ. For a detailed description of these tools and all the details concerning the investigation, see [3]. These tools were selected because they are widely spread (between 18 and 200 million downloads[1]), have bundled software (with the exception of ICQ) and claim some anti-spyware policy (with the exception of Morpheus). The P2P tools of interest for this investigation all run on the Win32 platform. Though P2P file sharing itself is not restricted to that platform, the adware and spyware most likely is. Moreover, the majority of users most likely do not use P2P tools on any other platform. As we have only tested the investigation tools in Windows 2000 Professional, that operating system version was the one of choice. Our investigation method is based on state preservation. By preserving the state of a system together with complementing information (such as network traffic) it is later possible to retrieve a specific state for analysis. The main payback of this approach is that we avoid doing both data collection and data analysis simultaneously in real-time. Instead, we are able to collect data only once.

During both planning and execution of the investigation we had two main goals concerning the laboratory environment:

1) Preserving identical hardware and software configurations during all P2P tool investigations.
2) Using default software configurations whenever possible, including updates to device drivers and operating system.

To provide our investigation computers with the exact same software configuration we used a hard disk cloning system[2] developed in our local security lab. The use of this system gives our investigation a few advantages. As far as we can control it, the execution environment is the same for all installations. If necessary, it would therefore be possible to compare results from the investigations of the different tools. By creating reinstallable system images, the investigation results can be reproduced. Another advantage of the cloning system is that system images can be created at any time during the investigation, making it possible run any step any number of times.

We decided to split the investigation into a number of clearly separated steps for a better overview. These steps are:

1) Installation
2) 30-minute run
3) 100-minute run
4) Removal

These steps all contain a common set of tasks that create representations of the system state before and after every task. The result is a handful of files in different formats ready for analysis or storage. Every step includes this list of tasks:

1) Create pre-step file system list
2) Create pre-step registry export
3) Clear the firewall log
4) Start dumping packets
5) Perform the actual step
6) Stop dumping packets
7) Save the firewall log to file
8) Create post-step file system list
9) Create post-step registry export
10) Make Ad-aware search and save log to file

The file system lists are made using the long listing format (`ls -l`) and include properties like permissions and file

[1] These figures were retrieved from Download.com.

[2] We used the Seclab cloning system that is based on the FreeBSD operating system.

size. Registry exports include the entire registry and are of Win9x/NT4 Registration File format, a human-readable text format. Network packets are dumped using a tool with the ability to store a binary packet stream to file. These dump files are in tcpdump (binary) capture format and need to be parsed to be of any use.

Comparing the differences between the states before and after a step makes is possible to retrieve information like, for example, creation, modification or removal of files. It is also possible to find any registry changes and any information sent over the network. The data analysis is described in the next section.

IV. Analysis

The proposed analysis method was to start by compilation of Ad-aware logs and comparison of firewall logs. The result would be a list of known ad- and spyware components and an approximate measure of which ones had accessed the network. Next, these components could be verified in the file system lists and registry exports. These components could then be excluded in a more thorough search for any unknown components. The result could be a number of known components and maybe a few unknown ones. The next step would be to look at the network activities of these components.

The Ad-aware and firewall log files contain human readable text and are easy to filter and search through using any standard text editor. The network packet dump files, however, are more difficult to analyse. Because of their binary format, they must be analysed using a tool capable of parsing tcpdump format dump files. These dump files also contain a lot of information compared to e.g. the firewall logs. Analysing them is often a time-consuming task.

Component Name	Host Software	Adware	Spyware
B3D Projector	Kazaa	x	—[1]
CommonName	iMesh	x	x
Cydoor	iMesh	x	x
Cydoor	Kazaa	x	x
Cydoor	Morpheus	x	x
DownloadWare	Kazaa	x[2]	—
eZula	iMesh	x	—
FlashTrack	iMesh	x	x
Gator	Morpheus	x	x
Gator	iMesh	x	x
Hotbar	iMesh	x	x
NewDotNet	Kazaa	—	—
NewDotNet	iMesh	—	—
NetworkEssential	Kazaa	x	x
PromulGate	Kazaa	x	—
SaveNow	BearShare	x	x
SaveNow	Kazaa	x	x
WeatherBug	BearShare	—	—
Web3000	iMesh	x	x
WurldMedia	Morpheus	—	x

(1) Technically, not spyware. However, the component authors prepare to include the users in distributed computation projects without their permission.
(2) Technically, not adware. However, it downloads and installs adware.

TABLE I
Classification of identified components.

We gathered over 630 MB of data during our investigation (in about 11 hours of P2P tool execution). A typical list file includes about 100 000 entries in a single file and there was a total of 147 MB of data captured this way. In addition, we also gathered over 440 MB of registry data and 45 MB of network packet data. Due to this massive amount of data we had to restrict what to analyse, we simply could not analyse all of it. Therefore we chose to investigate all list files of iMesh and Morpheus. This choice was based on the fact that Ad-aware found most suspicious components in iMesh and Morpheus. For all tools we also chose to verify all files found by Ad-aware and to check what components survived removal of their host software for all tools. We restricted our registry data analysis by only looking at the parts handling what programs to start automatically at system startup.

Ad-aware found a number of suspicious components. A list of identified components is presented in table I. In addition to these, we managed to find two more components, Sentry.exe and Eac_rvndl, by analysing the data from our investigation.

The registry was checked after each investigation step for automatically starting components. By automatically starting each time the system rebooted, these components could constantly run in the background, performing their business. Table II lists these components together with their host software. Eac_rvndl is here of special interest since the component was not found by Ad-aware.

Component Name	Host Software
CMESys	Morpheus
DateManager	Morpheus
Eac_rvndl	iMesh
eZmmod	iMesh
GStartup	Morpheus
Hotbar	iMesh
MediaLoads Installer	Kazaa
Msbb	iMesh
New.net Startup	iMesh, Kazaa
PrecisionTime	Morpheus
PromulGate	Kazaa
SaveNow	BearShare, Kazaa
Sentry.exe	Morpheus
Trickler	iMesh
WheatherCast	BearShare
Zenet	iMesh

TABLE II
Automatically activated components in system startup.

Unfortunately, ZoneAlarm did not capture all outbound network connections. Also, we could not correlate all sessions captured in the packet dump file to the entries in the firewall log.

When analysing Morpheus we managed to identify one component that Ad-aware did not find. The component is called Sentry.exe and was installed during P2P tool installation. This component was located in the `C:\WINNT\` directory together with a file called Sentry.ini.

We did not manage to identify any new components for iMesh, other than the ones already found by Ad-aware. However, we found that the file msbb.exe (Web3000) was not

installed during the installation of iMesh. Instead this file was installed during the 100-minute execution. The file was located in the `C:\WINNT\System32\` directory together with a file called msbb.dll.

Apart from these findings, the list files also showed that several components managed to survive removal of their host software. Table III specifies these components.

Host	Component	Related file(s)
BearShare	SaveNow	save.exe
"	WeatherCast	weather.exe
iMesh	Cydoor	cd_clint.dll
"	"	cd_htm.dll
"	eZula	ezstub.exe
"	"	ezinstall.exe
"	Web3000	msbb.exe
"	"	msbb.dll
Kazaa	BrilliantDigital	bdeclean.exe
"	"	bdedetect1.dll
"	Cydoor	cd_clint.dll
"	SaveNow	savenow.exe
"	"	savenow.db
Morpheus	DateManager	DateManager.exe
"	Gator	CMESys.exe
"	"	GStartup.lnk
"	GMT	gmt.exe
"	PrecisionTime	PrecisionTime.exe
"	WurldMedia	mbho.dll

TABLE III
COMPONENTS THAT SURVIVED REMOVAL OF THEIR HOST SOFTWARE.

V. DISCUSSION

Though privacy is not at all trivial in the adware and spyware context, there are other important issues related to those technologies, for example information security. More than one of the bundled ad/spyware components have the ability to download and run software from servers in some way affiliated with the component author. Apart from consuming resources (CPU time, network bandwidth and disk space), this functionality poses a big threat to the P2P tool users. They can only hope that destructive viruses, trojans or worms are not spread in large networks like the Kazaa network. A recently reported Internet worm is called Slapper[2]. This worm includes a P2P protocol and is able to execute distributed denial-of-service (DDoS) attacks.

Because the described download-and-run functionality in essence means that the users' CPU time is sold by the downloading component's author, the users will have little or no control over what software is actually running in their systems.

Almost all of the P2P tools contain adware and spyware, see table I. Some of the found components are classified as both adware and spyware. One P2P tool, ICQ, contained no adware or spyware components. We found two suspicious components that were not detected by Ad-aware, Sentry.exe and Eac_Rvndl. Of these two components, at least Sentry.exe should be classified as spyware. From Cexx.org[3] we found that a company called IPinsight develops this software together with a description stating that Sentry.exe: "Provides Web sites with demographic and geographic information about you (the company brags that it can determine what city you live in to 90% accuracy), along with connection-speed and other data." It turns out that the Sentry.exe software eavesdrops on web forms submitted by a user. IPinsight acknowledge in their consumer policies[4] that they, apart from collecting information regarding users' age, gender, birthday and birth year, also collect "information about the computer's hardware configuration, such as the amount of free space on your hard drive, and information about the computer's software configuration, such as the name and version of the operating system".

The information collected by IPinsight is of private nature. IPinsight is probably using the information gathered by Sentry.exe to create very sophisticated user profiles that are sold to third parties. What adds an extra amount of excitement to this finding is that Ad-aware failed in locating this component.

When analysing the Windows registry data collected from iMesh we found a suspicious registry key called Eac_Rvndl. We found that `ANTIVI~1.EXE` might be a trojan backdoor called Troj/Canary. This information comes from a web page called Sophos virus analyses[5]. Sophos states that "Troj/Canary is a backdoor Trojan client program that can be used to download and run files from a remote server".

One conclusion we make from the two findings presented above is that Ad-aware does not find all ad/spyware circulating the Internet. We discovered two components, of which at least one was spyware, which Ad-aware did not find.

The method used in this investigation was based on the idea that as many relevant properties of the system as possible should be measured. Examples of such properties are file system contents and changes in content, network communication and Windows registry settings. Properties that would be less interesting to monitor are e.g. mean CPU load, overall memory usage and disk I/O. Monitoring these properties can merely imply activity but not provide any detailed information about installed ad/spyware and its activities.

When looking upon the two parts of the investigation, data collection and data analysis, we see that the method in general was simple and that it provided coverage of relevant system properties, although its sufficiency was not fully established. However, as it was constructed, the investigation method failed to provide means for associating processes to network data. A tool with the ability to perform that kind of association could greatly improve the extraction of interesting network data from the gathered. A well-performed investigation would mean several days of continuous packet dumping, creating extremely large amounts of data to analyse. Without good tool support such investigations would clearly not be feasible. Moreover, an open source, well-documented tool, could possibly help users and anti-spyware organisations keep up with spyware authors.

[3] `http://www.cexx.org`

[4] `http://www.ipinsight.com/consumer.asp`

[5] `http://www.sophos.com/virusinfo/analyses/`

Furthermore, the adware and spyware domain is rapidly changing. Anti-adware and anti-spyware tools and firewall blacklists make ad/spyware authors update or modify their software, and creating new ways of doing business. Software bundles may be added or removed between P2P tool versions, or existing bundles might be updated. It is therefore not certain that a P2P tool that used to contain adware or spyware still does. Maybe the principle of "innocent until proven otherwise" should be applied. It certainly should if any proactive measures are going to be possible because loose accusations against the software companies will not solve the spyware problem.

The hard part here is to collect the data because it has to be done on the Win32 platform (the Win32 API is much more complex than those of POSIX systems). Once log files exist it is a matter of filtering the data and inserting it into a database to make it searchable. The entire analysis could be performed perfectly well on any other platform, preferably in a Unix system because of the extensive tool support. We do acknowledge that a number of tools have been ported to Win32 (e.g. in the Cygwin environment), but do not see any purpose in substituting a real Unix environment.

VI. Conclusion and further work

We present a method for investigating spyware that is bundled with any type of software. Four out of five investigated P2P tools were bundled with adware and/or spyware. To assist us in identifying spyware we used one of the most popular anti-spyware tools, Ad-aware. During the investigation, two suspicious components overlooked by Ad-aware were found.

The investigation we performed allowed us to test the investigation method in practice. Despite the improvements done, there were a number of problems concerning the tools used in some of the more central steps of the investigation.

We propose that further efforts should be focused towards creating a new analysis tool. Its purpose would be to enable per-process tracking of network packet data. This is interesting because defining which processes to examine (i.e. finding process names) in a Win32 system is both easy and a good place to start an analysis. Further analysis of BearShare, iMesh, Morpheus and Kazaa is probably an interesting prolongation of our investigation.

There is an ongoing arms race in the area of web-empowered products[4]. Bundled software range from legal commercial adware to viruses executing distributed denial-of-service attacks. Spyware may perform the same kind of actions as a virus but are not classified as illegal by the anti-virus tools. The leading anti-spyware tool Ad-aware is only detecting a proportion of all spyware.

We predict an extended arms race in the future where a merged society of virus, spyware and some adware are defeated by a combination of anti-virus and anti-spyware tools. As long as collected information is encrypted and there are opportunities to install uncontrolled components we cannot distinguish the commercial purpose from illegal actions in spyware. By implementing the improvements described above, such an investigation should be able to find and analyse the network traffic sent between spyware and their central servers. If such network traffic could be linked to a certain process there could be no disagreements when classifying the components as spyware.

References

[1] R. Anderson, Security Engineering — A Guide to Building Dependable Distributed Systems. *John Wiley & Sons*, 2001.

[2] I. Arce, E. Levy, An Analysis of the Slapper Worm. *IEEE Security & Privacy*, Vol. 1 No. 1 2003.

[3] M. Boldt & J. Wieslander, Investigating Spyware in Peer-to-Peer Tools. *Blekinge Institute of Technology*, 2003.

[4] B. Carlsson, The Tragedy of the Commons — Arms Race within Peer-to-Peer Tools. *in eds. Omicini, A., Petta, P., and Tolksdorf, R., proceedings of the 2nd International Workshop Engineering Societies in the Agents' World, Lecture Notes in Artificial Intelligence 2203*, Springer-Verlag, 2001.

[5] R. Dawkins, The Selfish Gene 2nd Ed. *Oxford University Press*, 1989.

[6] Z. Demetrios, Exploiting the Security Weaknesses of the Gnutella Protocol. *University of California*, 2002.

[7] D. Gollmann, Computer Security. *John Wiley & Sons*, 1999.

[8] B. Leuf, Peer-to-Peer — Collaboration and Sharing over the Internet. *Addison Wesley*, 2002.

[9] M. Kirk McKusick, K. Bostic, M. J. Karels, J. S. Quarterman, The Design and Implementation of the 4.4BSD Operating System. *Addison Wesley*, 1998.

[10] D.M. Martin Jr, R.M. Smith, M. Brittain, I. Fetch, and H. Wu The Privacy Practices of Web Browser Extensions. *University of Denver*, 2000.

[11] J. Maynard Smith, Evolution and the theory of games. *Cambridge University Press*, 1982.

[12] B. Schneier, Secrets & Lies — Digital Security in a Networked World. *John Wiley & Sons*, 2000.

[13] A. Silberschatz, P. Baer Galvin, G. Gagne, Operating System Concepts, 6th Edition. *John Wiley & Sons*, 2002.

[14] A. S. Tanenbaum, Computer Networks, 3rd Edition. *Prentice Hall*, 1996.

Consolidation and Evaluation of IDS Taxonomies

Magnus Almgren Emilie Lundin Barse Erland Jonsson
Department of Computer Engineering, Chalmers University of Technology
SE-412 96 Göteborg, Sweden
{almgren,emilie,erland.jonsson}@ce.chalmers.se

Abstract

Accurate taxonomies are critical for the advancement of research fields. Taxonomies for intrusion detection systems (IDSs) are not fully agreed upon, and further lack convincing motivation of their categories. We survey and summarize previously made taxonomies for intrusion detection. Focusing on categories relevant for detection methods, we extract commonly used concepts and define three new attributes: the *reference model type*, the *reference model generation process*, and the *reference model updating strategy*. Using our framework, the range of previously used terms can easily be explained. We study the usefulness of these attributes with two empirical evaluations. Firstly, we use the taxonomy to create a survey of existing research IDSs, with a successful result, i.e. the IDSs are well scattered in the defined space. Secondly, we investigate whether we can reason about the detection capability based on detection method classes, as defined by our framework. We establish that different detection methods vary in their capability to detect specific attack types. The *reference model type* seems better suited than *reference model generation process* for such reasoning. However, our results are tentative and based on a relatively small number of attacks.

Keywords: intrusion detection, taxonomy, classification, detection methods

1 Introduction

It is a well-known fact that the research in a field greatly benefits from a good taxonomy and hence a good classification. A shared vocabulary enables efficient communication and a shared classification may direct future research into open areas, i.e. holes, in the classification. In some cases, a classification not only helps cluster the field but may also well reflect some intrinsic value, which in turn leads to a refinement of the models in the field.

There have been several defined taxonomies, classifications and subsequent surveys for intrusion detection. The goals of these efforts have also been quite diverse; some only try to survey the field and find it easier with labels on the systems, while others try to use the taxonomies for a deeper understanding or to guide future research efforts. Despite these previous efforts, intrusion detection still lacks a widely applicable and accepted taxonomy. This may in part be because of it being a young research field, part of it being fast-paced and maybe part of it owing to its inherent complexity.

We begin this paper with a survey of several taxonomies for attacks and intrusion detection systems (IDSs). In our view, many of the existing taxonomies have used different terms for the same or similar underlying concepts. We limit the scope of the paper by only considering terms related to the *detection method* in IDSs, thus by purpose excluding audit source location and other distinguishing features. After surveying previous taxonomies, we present three general attributes important for the detection method strategy. We conclude the first part of the paper with a discussion of how the terms in earlier surveys should be interpreted in our consolidated framework.

In the second part of the paper, we evaluate the surveyed taxonomies through our unified framework. Clearly, there are no strict rules defining a good taxonomy but mostly guidelines. For example, Howard [1] enumerates six attributes of a *satisfactory* taxonomy. We use two empirical tests to further investigate whether the attributes of the taxonomies are relevant for a researcher in intrusion detection. In the first test, we classify several well-known research IDS prototypes according to the unified taxonomy and evaluate the result. As many taxonomies are established in conjunction with a survey, we find the first empirical test represents typical use. One of the challenges is, of course, to ensure that also future (to be developed) systems can accurately be described by the taxonomy.

The second empirical test reflects the purpose of intrusion detection systems, i.e. their ability to detect attacks. We base this test on the assumption that different detection methods might be more or less suitable for detecting a specific attack. We start by building an argument why our assumption is valid and we then classify different detection methods to see whether their classification reflects their suitability to detect a certain attack.

The organization of this paper is as follows. In Section 2 we survey several existing classifications relevant for intrusion detection. In Section 3 we present a consolidated framework built upon the underlying concepts we found in the survey. The section is concluded by a discussion of how previous terms fit into the new framework. Section 4 contains the empirical evaluations of the taxonomies. Section 4.1 contains the first test, a classification of well-known IDSs and Section 4.2 describes the second test; how well the classification of a detection method reflects its suitability to detect a specific attack. We then discuss our results in Section 5 and conclude the paper with Section 6.

2 Classifications Relevant for Intrusion Detection

In this section, we present some of the previous taxonomies and classification efforts relevant for intrusion detection. This list is meant to summarize previous efforts, and should not be seen as an exhaustive enumeration of all previously made classifications. We have divided the classifications based on their main area of focus, but a few efforts span several areas and are thus reiterated in several sections below.

2.1 Intrusion Detection System Classification

Debar et al. [2] probably did the first taxonomic survey of intrusion detection system research. The main categories used in their taxonomy of IDSs are *detection method*, *behavior on detection*, *audit source location*, and *usage frequency*. These categories are further divided into two possible sub-categories. For example, the detection method is further refined as either behavior-based or knowledge-based detection.

Axelsson [3] also provides a survey and taxonomy of intrusion detection systems. His approach is more fine-grained than the one by Debar et al. The taxonomy treats detection principles and system characteristics separately. The detection principles are divided into *anomaly*, *signature*, and *signature-inspired*, with the sub-categories *self-learning* and *programmed*. Additional layers of sub-categories are used, depending on the specific class. Table 1 shows an overview of his detection principles. The system characteristics categories used in the taxonomy are *time of detection*, *granularity of data processing*, *source of audit data*, *response to detected intrusions*, *locus of data processing*, *locus of data collection*, *security*, and *degree of interoperability*.

The contribution of the two IDS surveys mentioned above is mainly to give a systematic overview of IDS research. In addition, Axelsson uses his survey to point out current trends in IDS research.

Table 1: Axelsson's [3] detection principles

Anomaly	Self-learning	Non time series
		Time series
	Programmed	Descriptive statistics
		Default deny
Signature	Programmed	State modeling
		Expert-system
		String-matching
		Simple rule-based
Signature-inspired	Self-learning	Automatic feature selection

A European Union funded project, entitled the MAFTIA project [4], attempts to form a detailed taxonomy over IDSs, that is then used for determining their detection scope and how different IDSs can be used to complement each other. The taxonomy aims at describing the capabilities of an IDS with respect to the analysis of activities (attacks and related events) and includes a large number of attributes for describing both the detector (analysis mechanism) and the sensors (information sources) of the IDS. It should be possible to match the IDS description to any activity description and determine whether the IDS generates an alarm for that activity. Unfortunately, there are no detailed examples of IDS descriptions in this report. As previously stated, we are focusing on the detection method in this paper and for that reason we limit our discussion to the detection component categories and attributes found in the MAFTIA project. Alessandri et al. [4] start by separating the attributes into *generic characteristics*, *data preprocessing*, and *instance analysis*. *Instance analysis*, which determines what kind of detection the IDS does, is then further divided into such sub-categories as *single-instance analysis* and *cross-instance analysis*. Subsequent division leads to the *analysis-level* category, which can take the possible values of *basic analysis*, *logic verification*, and *semantic verification*. They also consider the attribute *analysis technique*, which are divided into *timing analysis*, *information item analysis*, *data category analysis*, *sequence analysis*, and *statistical analysis*. The attributes for these categories are, e.g., string matching, advanced string matching, regular expression, and size verification.

Another classification is proposed by Halme and Bauer [5], where they use the classes of *anomaly detection* and *misuse detection* to describe IDSs.

Ko et al. [6] have not done a formal taxonomy, but divide IDSs into systems using *anomaly detection*, *misuse detection*, and *specification-based monitoring*.

There are two more significant surveys of the IDS field [7, 8], but both these efforts' main focus is to describe the existing IDSs to provide information for someone who wants to compare and deploy an IDS and not to create a taxonomy with a classification. Many of the characteristics they study are about usability and management of the IDSs and no attempt has been made to say anything about the detection coverage. For that reason, the surveys of Jackson [7] and Kvarnström [8] are not discussed in further detail.

2.2 Attack Classification

Going beyond the IDSs to the attacks that are the reason for their existence, we have several additional classifications. Kumar [9] includes a classification scheme for *intrusion signatures* with hierarchical classes, based on the complexity of matching the attack (signature). He divides the signature types into four categories: *existence patterns*, *sequence patterns*, *RE patterns*,[1] and *other patterns*.

The MAFTIA project [4], presents a fine-grained activity taxonomy to describe an attack in such detail that it can be matched against the detection capabilities of an IDS. For example, there are attributes for the *affected object*, such as *storage device*, *memory*, *operating system core*, *file system object*, or *process*. There are also attributes for the method *invocation type*, e.g. *object creation*, *object execution* etc.

Lindqvist [10] extends the classification scheme for *intrusion techniques* by Neumann and Parker [11]. The main categories in Lindqvist's classification are *bypassing intended controls*, *active misuse of resources*, and *passive misuse of resources*. Each main category has a number of sub-categories and the paper also includes a taxonomy of intrusion results.

Mell [12] tries to capture statistics of attacks based on special characteristics, such as *script goal*, *target type*, *attacker platform*, and *transmission method*. He also mentions that the database of classified attacks can be used for two different purposes: forensics, when creating a list of possible attacks that may have compromised a penetrated system, and as a search tool for specific attack scripts.

[1] RE stands for Regular Expressions, equivalent to regular grammars defining regular languages in the Chomsky hierarchy.

There are also a series of classifications focused on the intended result of the attacks. For example, the DARPA 1999 intrusion detection evaluation experiment used five main types of attacks: *denial of service*, *remote to user*, *user to superuser*, *surveillance/probing*, and *data attack*. More information is found in [13].

Finally, worth mentioning is also the seminal paper by Denning [14] with a list of attacks. It is not a classification, but a description of different types of attacks that is then used when designing the intrusion detection system described in the paper.

2.3 Other Classifications

There are also classifications concerning the *attackers* and the underlying *vulnerabilities* (or flaws) that are being exploited. For example, Landwehr et al. [15] and Krsul [16] have done vulnerability taxonomies and classifications. Their objective is to document flaws, hoping that knowledge of how systems fail would help build systems that can better resist failure. Howard [1] has classified the attackers of computer systems and their objectives. The attributes found in these efforts could be relevant for intrusion detection, but they are not within the scope of this article.

2.4 Detection Method Suitability for Attack Detection

In Section 4.2, we empirically test whether we can judge if a certain detection method class is suitable to detect a whole attack class based on the underlying taxonomies.

Few of the taxonomies we have surveyed discuss in any detail what their detection method division implies for the ability to detect different types of attacks. However, brief discussions can be found in some papers. For example, Ko [6] says that anomaly detection provides a method to detect penetrations without specific knowledge about the operating system or its security flaws and that it is the only viable technique to detect masqueraders. It may, however, be difficult to establish normal behavior and to set thresholds for what is considered anomalous. Misuse detection, on the other hand, is good at detecting known attacks but not at detecting novel attacks.

Axelsson [3] talks about the high-level categories of *well-known intrusions*, *generalizable intrusions*, and *unknown intrusions*. Well-known intrusions are easily detected by signature-based systems, and the signature-inspired systems are more sophisticated in that they consider both normal and attack behavior. More advanced pattern matching systems may detect generalizable intrusions. Finally, the anomaly detection systems are the only ones that may have a chance at detecting previously unknown intrusions.

3 IDS Terminology Revisited

In this section, we revisit the classifications described in the previous section, with the objective of providing a unified framework where all authors' work is set in relation to each other. Many of the surveyed taxonomies in Section 2 have used a similar terminology, but they differ when it comes to the details. Table 2 summarizes some of the terms related to the detection method that has previously been proposed.

Table 2: Detection Method Terminology used in Different Surveys

Author	*Terms used in Surveys*		
Halme and Bauer [5]	anomaly		misuse
Debar et al. [2]	behavior-based		knowledge-based
Ko [6]	anomaly	specification-based	misuse
Lindqvist [10]	dynamic knowledge		static knowledge
	default deny		default permit
Axelsson [3]	anomaly	signature-inspired	signature
	self-learned		programmed
MAFTIA [4]	*no corresponding mapping*		

3.1 Consolidation of IDS Terminology

Considering the terms used in previous taxonomies, we believe they can be summarized in three underlying concepts: the reference model type used in the detection process, the reference model generation process and finally the

reference model updating strategy. Let's have a closer look at these attributes.

- **Reference model type** The reference model in the detection system, i.e. the "knowledge base" that input data is compared to, can define the normal user model, the attacker model, or both. Detection is basically about classifying observations as belonging to the normal model or the attack model.
- **Reference model generation process** The second attribute is about how the reference model is created, i.e. basically how much of the information in the reference model that is knowledge encoded by experts and how much is learned directly from the data itself by an algorithm.
- **Reference model updating strategy** The third attribute is about how timely the information used in the reference model is. The better the updating strategy, the better chance we have of detecting new attacks. For our purposes, we do not make any distinction on *how* the information in the model is updated. For example, continuous updates (e.g. learning from data) are treated the same way as many (small) discrete updates (e.g. daily additions of rules to the knowledge base).

We consider these attributes to constitute three separate axes forming three continuous scales, as opposed to their previous presentation as attributes with binary values. For example, the axis reflecting the *reference model type* have the categorical value *normal information* at one end and *misuse information* at the other end. Clearly, these are not strict quantitative scales but more a guidance of the type of information found in the models.

We find that each of the existing taxonomies surveyed in Section 2 has referred to these underlying concepts with a wide variety of terms. In the next section we discuss how these previously used terms fit into our unified framework.

3.2 Resolving Prior Terminology

Let us now revisit the IDS taxonomies described in the earlier sections and show how they can be interpreted with the three attributes presented in Section 3.1. In this discussion, Table 2 is used as a visual reference and its columns indicate the implied similarities between the terms that we highlight below. Let us again stress that we are mostly concerned with the detection component of the IDS, and we do not consider attributes such as the source of events for the IDS (audit events collected at either the network or the host).

Halme and Bauer [5] refer to `anomaly` and `misuse` and this clearly describes the two extreme points on the scale *reference model type*. Misuse detection system specify (or model) the signs left by an attack. Anomaly detection systems model the normal behavior for the systems, and alerts when the observed behavior is far enough from the reference model.

Debar et al. [2] define knowledge-based detection as an encoding of knowledge of attacks and vulnerabilities, which in our view corresponds directly to one end of the scale of *reference model type*. Their behavior-based detection is found at the other extreme.

Ko [6] describes three relevant terms: `anomaly`, `misuse`, and `specification-based` detection. The first two are naturally treated as the equivalent terms above. The term specification-based is a bit harder to place in relation to the new attributes. Based on Ko's description and use of the term, we find that specification-based detection relates to both the *reference model type* and the *reference model generation process*. Basically, an expert writes a specification of the normal behavior of a system or part of a system (e.g. a protocol).

Lindqvist's classification [10] uses two classes of terms. The `dynamic` versus `static` knowledge would in our terminology be the two extremes of the *reference model updating strategy*. His `default deny` and `default permit` first seem not to fit our framework. However, let's consider these terms in relation to a system that might use such policies. A firewall with a default deny policy has an explicit list of all kinds of traffic it lets through, i.e. the firewall thus uses a normal reference model. A system with default permit, on the other hand, explicitly lists the services and protocols that are not allowed and will let all other types of traffic through. The firewall is then using a description of unwanted traffic, i.e. some sort of misuse. Based on the type of reference model the system employs, we can then automatically apply its policy, i.e. default permit or default deny.

Axelsson [3] defines `signature detection` as detection based on knowledge of a model of the intrusive process, and `anomaly detection` as detection based on a model of normal behavior. His `signature-inspired` detection class uses a mixed model of both intrusive and normal behavior. Only one system in his survey falls into this class, and it focuses on the intrusive model, and mainly uses the normal model to avoid choosing features prone to false alarms.

MAFTIA [4] includes two binary attributes called `behavior-based` and `knowledge-based` in the *generic attributes* class, but these are not central in the detection component taxonomy. The taxonomy in MAFTIA is very interesting, but it is not easily mapped to previous efforts and has no obvious high-level classes. As we want to study the importance of commonly-used high-level IDS attributes for detection, we do not further discuss this work.

4 Empirical Evaluation of Classifications

In the previous sections we have looked at taxonomies relevant to intrusion detection and made an effort to consolidate the main categories into a single framework. Previous taxonomies have worked well in describing ongoing research prototypes, but they have not managed to convey why their chosen categories are better than others. Here, we present an attempt to empirically evaluate the qualities of taxonomic attributes based on two tests. These can formally be described in the following way.

Test 1 *(IDS Classification Test).* Given a set of IDSs and a chosen taxonomy, we classify the IDSs. We consider the taxonomy to be well designed if the classification results in the IDSs being well separated into a number of clusters (where at least some clusters contain more than one object).

Test 2 *(IDS Detection Capability Test).* Given a set of IDS detection methods[2] and a set of attacks, we define a corresponding set of labels, l_i, as a value reflecting how well a certain IDS detection method, I_i, can detect an attack, A_i. If we can determine the value of l_i based on $T_{ids}(I_i)$ and $T_{attack}(A_i)$ $\forall i$ with T representing specific IDS and attack taxonomies, we consider these taxonomies successful in that their transformation preserves l.

One of the main uses of previous taxonomies has been to survey intrusion detection systems. Therefore, our first test is similar in nature and we classify some IDSs based on two of the attributes presented in Section 3.1. We do not consider the *reference model updating strategy* for two related reasons. Firstly, the updating strategy of the IDSs found in the literature is not well described because it more reflects an operational parameter than a system design parameter. Secondly, we needed to limit the scope of the paper and the *reference model updating strategy* being the least described, it was the one we omitted.

The ability to detect attacks is crucial for intrusion detection systems. It has previously been pointed out (see Section 2.4) that different detection methods may vary in their effectiveness for detecting different types of attacks. The *IDS Detection Capability Test* measures whether the classes defined by a taxonomy signify if specific IDSs within that class have the same capability to detect attacks. This is an important property, as it lets us reason about IDS system classes and their suitability to detect a new type of attack. Through the *IDS Detection Capability Test* we can then better understand statements such as the one by Ko [6] where he claims that anomaly detection is the only viable technique to detect masqueraders. As with the first test described above, we limit our discussion to only two of the three attributes described in Section 3.1.

Before going on with the tests, we would like to point out that the two tests capture different properties of a taxonomy. The first test shows the ability of the classification to abstract higher-level classes. The second test ensures that these classes have some relevance for practitioners in the field.

4.1 IDS Classification Test

For the first empirical test, we classified several well-known intrusion detection systems using the attributes *reference model type* and *reference model generation process*. The result is shown in Figure 1. Completely self-learning systems are on one end of the scale as opposed to programmed systems that are found on the other. In between, there are systems using some prior (programmed) knowledge in combination with learning. The other axis shows whether the systems model (or try to predict) intrusive or normal behavior. Our classification of the IDSs is based on how we believe they are used after studying the relevant research papers. The systems could be used differently with another knowledge base. Furthermore, the placement of each system is not exact, as we prioritized readability in the figure. Descriptions of the systems can be found in Appendix A.

We find that the classified IDSs are dispersed in Figure 1, making an empirical argument that the attributes *reference model type* and *reference model generation process* are relevant for survey classifications.

4.2 IDS Detection Capability Test

The *IDS Detection Capability Test* requires some IDS detection methods and some attacks, which we then use to determine how well suited a certain detection method is to detect different attacks. We describe the three detection methods and the two attacks we use in Section 4.2.1 and Section 4.2.2 respectively. Our premise is that different detection methods do differ in their detection capability, and we show that this seems to be the case by carefully considering each of the detection methods and each of the attacks in the following sections.

[2] A more general version would use all relevant attributes of IDSs and not be limited to only the detection methods. We present this version as the taxonomies we use are themselves limited to detection method attributes.

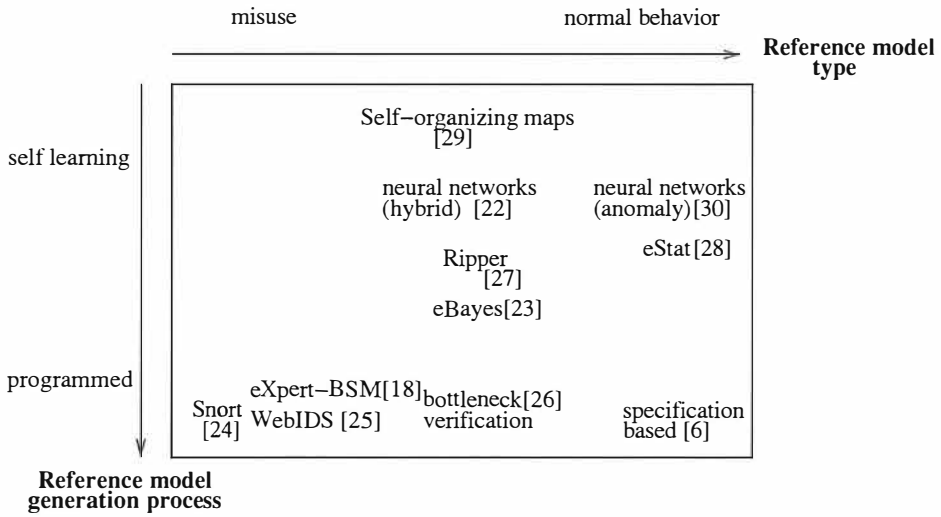


Figure 1: IDS detection method categorization

4.2.1 Detection Method Descriptions

Instead of discussing abstract detection methods, we have chosen to illustrate our points by using existing IDS systems that we believe reflect the detection method classes we have previously discussed. The three methods (systems) we use are chosen from different points in Figure 1 with respect to both axes. One should keep in mind that these existing systems often have capabilities that goes beyond a single detection method, but it should be clear from the context exactly what we mean.

Stateless Pattern Matching One of the most common methods used to detect attacks is built on some kind of (stateless) pattern matching, and Kumar [9] describes some relevant aspects of such detection. Such detection can be very efficient in terms of keeping up with high-speed event streams. Snort uses a form of stateless pattern matching for its signature rules, and we use it as an example in our discussion. We do recognize that Snort has many capabilities that we do not use for this study.

Expert Systems Forward-chaining rule-based systems have successfully been used for intrusion detection. They provide a powerful inference engine, and separate the knowledge base from the fact base. In our discussion, we are going to use rules from the experts used in the EMERALD framework [17, 18].

Neural Networks Artificial neural networks are a powerful computational tool and maybe the most well-known artificial intelligence system capable of inductive learning. There have been several systems using neural networks for intrusion detection, but there is no research paper that describes the experience from a general system used for detection of a wide range of attacks. For that reason, the discussion of the suitability of neural network detection for the attacks is kept at a general level. A pure anomaly-based neural network detector is useful for detecting unknown attacks, but as we are targeting specific attacks we assume a neural network that models both intrusive and normal behavior.

4.2.2 Attack Descriptions

This being the beginning of our project to evaluate taxonomies, we did not have any formal strategy for choosing suitable attacks. We used our expert knowledge and chose attacks that intuitively felt relevant and different. We hoped they would prove to require different detection methods to be successfully detected. The attacks are relatively old and should be well-known. The details are provided in the references.

echo/chargen attack [19] A DoS attack created by any UDP packet addressed to the destination port 7 (*echo*), and with the source port 19 (*chargen*), or the other way around.

syn flood [20] A DoS attack where the server receives *too many* SYN packets without accompanying ACKs, thus exhausting state in the server.

The *echo/chargen attack* is well-defined using only misuse information. It consists of a single event (or UDP packet) and this event clearly distinguishes it from normal behavior. The *syn flood attack* definition is also based on

Table 3: The attacks classified according to Kumar [9], Lindqvist [10], Mell [12], and DARPA [13]

Kumar	**echo/chargen attack** Existence pattern **syn flood attack** Sequence → Interval
Lindqvist	**echo/chargen attack** *Intrusion technique:* Active misuse of resources → Resource exhaustion *Intrusion result:* Denial of service → Unselective → Affects all users **syn flood attack** *Intrusion technique:* Active misuse of resources → Resource exhaustion *Intrusion result:* Denial of service → Unselective → Affects all users
Mell	**echo/chargen attack** *Script goal:* DoS remote → Crash/freeze host *Target type:* Hosts → UNIX *Attacker platform:* All *Transmission method:* IP → UDP *Attacker requirements:* Network access *Target of attack:* Application/daemon **syn flood attack** *Script goal:* DoS remote → Crash/freeze application (or host) *Target type:* Hosts, Router, Firewall *Attacker platform:* All *Transmission method:* IP → TCP *Attacker requirements:* Network access *Target of attack:* Network protocol stack
DARPA	**echo/chargen attack** Denial of service **syn flood attack** Denial of service

the misuse model, but we need normal data to define *too many* in relation to the normal traffic at a specific site. This is an example of an attack consisting of several events (SYN packets) and with no distinct decision boundary between normal and malicious behavior.

Table 3 shows classifications of the attacks according to Kumar [9], Lindqvist [10], Mell [12], and DARPA [13].

4.2.3 Attack Detection

Before we can evaluate the result of the *IDS Detection Capability Test*, we need to ensure that the chosen detection methods vary in their detection capability on the chosen attacks (i.e. we are trying the establish l_i as per definition in Section 4).

Detection of echo/chargen Snort, illustrating *stateless pattern matching*, already comes equipped with some rules to detect the exploits described in the previous section, as shown in Table 4. The rule used to detect the echo/chargen attack is listed first. By matching on the right protocol and with specific port numbers, we have an efficient detection signature. Clearly, stateless pattern matching works well for the echo/chargen attack. It is efficient in that pattern matching can be done fast and with a small memory footprint. We do not expect to see any false positives or negatives from the signature either, based on the knowledge that no normal traffic should have the characteristics of the attack. The administrator does not need to tune the rule and the alerts are distinct with good information, such as the type of attack and the computers targeted. We lack the actual attacker, as the source address is spoofed, but that can be captured at the network boundary.

As the basis for our discussion of *expert system* detection we use the material found in [17], even though Lindqvist et al. do not list an explicit rule for the echo/chargen attack. We believe the rule for detecting this attack would be trivial to construct, resulting in a single rule similar to the Snort rule described above. For the same reasons, the single echo/chargen rule should not create false positives or negatives.[3]

Neural networks use inductive learning, instead of specified rules, and are especially suited for tackling certain types of problems (e.g. described by Mitchell [21]). The key term when discussing neural networks is learning. Considering the echo/chargen attack, we would thus not design a network to detect that specific attack instance.

[3] As this attack is also easily detected with stateless pattern matching, one might in practice choose to avoid describing such a rule within the expert system for performance reasons.

Table 4: Snort signatures for the two attacks.

echo/chargen attack

```
alert udp any 19 <> any 7 (msg:"DOS UDP echo+chargen bomb"; reference:cve,CAN-1999-0635;
reference:cve,CVE-1999-0103; classtype:attempted-dos; sid:271; rev:3;)
```

syn flood attack

```
alert tcp $HOME_NET any <> $EXTERNAL_NET any (msg:"DDOS shaft synflood"; flags: S;
seq: 674711609; reference:arachnids,253; classtype:attempted-dos; sid:241; rev:2;)
```

Instead, we would look at the characteristics of that attack, i.e. the attack uses an anomalous combination of port numbers, and train a network with normal port combinations and possibly labeled instances of the echo/chargen attack. As it turns out, some other ports (*daytime* on port 13 and *time* on port 37) may also form a DoS attack. Our neural network might flag these combinations as anomalous, thus also detecting variants of the attack.

Both the pattern matching system and the expert system miss the "unknown" variants of this attack. However, we expect the generality of detection in the neural net also results in both false negatives and false positives. Furthermore, as the neural network works like a black box, the system administrator might find it difficult to interpret the alerts and then track down the false positives. It is difficult to evaluate a more complex neural network but there is clearly a trade-off between the accuracy of the detection method and its ability to detect variants of the attacks.

Detection of syn flood There is a signature in Snort to detect a *syn flood attack* (see Table 4). However, this signature is directed at a specific syn flood attack created by a Shaft agent where the sequence number is always the same. There is no simple way to write a *stateless pattern matching* signature for a general syn flood attack. Either you can match on no one, or all instances of syn packets. However, as syn packets are normal to network traffic (and there is a significant number of them), we cannot match on every syn packet the IDS sniffs. As a parenthesis, there are plugins[4] to Snort that extends its detection capabilities to also include syn floods. When it comes to the syn flood attack we have apparently reached the limits of stateless pattern matching. There is an example signature in Snort for a specific kind of syn flood, but there is no general signature and we claim no such signature can be created without extending the pattern matching language.

Contrary to the stateless pattern matching, we can create *expert system* rules for detecting syn flood attacks, because the expert system keeps state through the fact base. An example of detection rules is shown in Table 5. When it comes to the syn flood attack, the expert system excels over stateless pattern matching. By defining several constants that depend on the site in question, the expert system keeps track of ongoing connections and alerts when there are too many of them. However, the rules for the syn flood attack are relatively complex. Further, the thresholds must be manually adapted *per installed site*.[5] The system administrator must not only be competent enough to be able to specify the traffic profile at his site but also learn enough of the expert system to correctly set the necessary variables. The addition of new detection rules might be beyond the reach of a user of the IDS.

The syn flood attack seems to be a good match for a *neural network* as there is a range of acceptable behavior that changes per site. By using the network's ability to manually adapt to local traffic, there is no need for the administrator to tune detection rules. Cannady [22] claims a good detection of syn floods with a forward-feed neural network. A recurrent neural network providing short-term memory should also work well for such detection.

4.2.4 Results of the IDS Detection Capability Test

Summarizing the result of the attack detection discussion, we see that the three detection methods presented earlier vary in their capability to detect the two different attacks. Both the stateless pattern matching and the expert system detect the echo/chargen attack equally well, with possibly the only differences being the complexity of the algorithm and its performance, which are outside the scope of our discussion. The neural network can detect the attack but it is less specific and probably produces false positives and false negatives.

The stateless pattern matching fails to detect the syn flood. The expert system does detect the attack but it needs manual tuning, which may be difficult. The neural network can detect the attack, and needs no tuning. Possibly we again get false positives and false negatives from the neural network but as there is no clear decision boundary between the intrusive and the normal behavior, we expect that also the expert system might give us false alerts.

Having set up the preliminaries, it is now time to perform the *IDS Detection Capability Test*. Starting by classifying the three detection methods according to the concepts defined in Section 3.1, we find that the stateless pattern

[4] Snort plugins are programmable modules able to implement any detection algorithm, hence going beyond the simple pattern matching rules.

[5] In some rare cases, we expect it to be possible to define such thresholds once and then be applicable for most sites.

Table 5: Expert rules for detection of syn flood attacks (from [17])

```
1  rule[create_open_conn(*):
2    [+ev:conn_event| e_type == 0 ]
3  ==>
4    [+open_conn | seq_id=ev.seq_id,
5          sec = ev.sec,
6          expired = 0
7          client_ID = ev.client_ID]
8    [-|ev]
9  ]
```

```
rule[add_to_bad_cons(*):
  [+ts:time]
  [+oc:open_conn| expired == 0 ]
  [?|(ts.sec - oc.sec) > 'expire_time]
  [+bc:bad_conn|count < 'max_bad_conns]
==>
  [/bc| count += 1]
  [/oc| expired = 1]
]
```

```
10 rule[destroy_open_conn(*):
11   [+ev:conn_event| e_type == 1 ]
12   [+oc:open_conn | seq_id = (ev.seq_id - 1) ]
13 ==>
14   [-|oc]
15   [-|ev]
16 ]
```

```
rule[max_open_cons_cons(*):
  [+ts:time]
  [+oc:open_conn| expired == 0 ]
  [?|(ts.sec - oc.sec) > 'expire_time]
  [+bc:bad_conn|count == 'max_bad_conns]
==>
  [?|syn_alert("SYN Attack:  Last Host %s.
       SeqID = %d.  Time = %d:,
        oc.client_ID, oc.seq_id, oc.sec)]
  [/bc|count = 1]
  [/oc|expired = 1]
]
```

```
17 rule[ignore_spurious_acks(*):
18   [+ev:conn_event| e_type == 1 ]
19   [+oc:open_conn | seq_id = (ev.seq_id - 1) ]
20 ==>
21   [-|ev]
22 ]
```

```
23 rule[first_bad_conn(*):
24   [+ts:time]
25   [-bad_conn]
26   [+oc:open_conn|expired == 0]
27   [?|(ts.sec - oc.sec) > 'expire_time]
28 ==>
29   [+bad_conn|count = 1]
30   [/oc| expired = 1]
31 ]
```

```
rule[free_bad_open_cons(*):
  [+ts:time]
  [+bc:bad_conn]
  [+oc:open_conn|expired == 1]
  [?|(ts.sec - oc.sec) > 'bad_conn_life]
==>
  [-|oc]
  [/bc|count -= 1]
]
```

matching (Snort) and the expert system (eXpert-BSM) are positioned very close together while the (hybrid) neural network is in the middle of Figure 1. It seems likely that the position of the neural network, further from the other two detection methods, might reflect its problems concerning the echo/chargen attack.

However, by the same reasoning there should be a difference in placement of Snort and eXpert-BSM to reflect their varying detection success of the syn flood attack (where the former completely failed). This is not the case, probably meaning that some other property of the detection methods that are not used in the previous taxonomies are important to determine what kind of attacks the detection method can detect. The difference in placement between the neural network and eXpert-BSM, on the other hand, seems to explain why it is easier to use the neural network when it comes to syn flood detection.

Having only pointed out weaknesses in IDS taxonomies, it is time to turn to the attack taxonomies we have surveyed (see Table 3). In Lindqvist's and DARPA's classifications, the *echo/chargen* and the *syn flood* looks exactly the same. However, as we have seen there is a difference in how they are best detected. Mell's classification shows differences for the attacks but does not give any more hints for detection than Lindqvist's classes. The difference is that Mell's classification provides more information, e.g. about what data to collect and which platforms that are vulnerable. Kumar's classification reveals that the detection of the echo/chargen attack may be simpler than the detection of the syn flood attack. None of these classifications seem to map to any of the three detection method attributes that we are studying, i.e. *reference model type*, *reference model generation process*, and *reference model updating strategy*.

We have shown that detection methods vary in their ability to detect attacks. Unfortunately, neither IDS nor attack classifications consider enough vital attributes for us to take a more abstract system view and consider whole classes of detection methods and attacks.

5 Discussion

When building the unified framework presented in the first part of the paper, we made a few observations. Most of the surveyed taxonomies include a division based on the *reference model type*, and only a few mention the other attributes, *reference model generation process* and *reference model updating strategy*. Intuitively, we interpret this as endorsement from researchers in the field of the underlying concepts; many people find the *reference model type* relevant and most articles in intrusion detection also start with such references; fewer people have used the other terms implying that they are not as widely accepted or that these divisions are more recently proposed.

We then evaluated the concepts empirically. The result of the first test, the *IDS Classification Test*, was successful as the categorized systems showed a good spread with respect to the chosen attributes and one could distinguish certain clusters of systems in the figure. The success is not surprising, as most of the taxonomies we investigated were constructed with a survey in mind, and then presented in such a framework in the papers. We did note that with our addition of continuous scales for the parameters, even recent systems, i.e. built after the taxonomy papers were written, are classified successfully. For example, Valdes [23] calls his IDS *hybrid* as something between misuse and normal behavior.

The result of the second test, the *IDS Detection Capability Test*, was more uncertain. None of the taxonomies were designed with such a purpose in mind, and even though previous papers have made some allegations we find that no one has previously rigorously tested taxonomies in such a way. First, we could show that different detection methods (as exemplified by real systems) vary in their capability to detect attacks. Thus, we have established the assumptions leading up to the *IDS Detection Capability Test*. However, it was difficult to go further than that as the taxonomies for IDS and attacks are clearly not designed for such reasoning. Many of the attack classifications we investigated classified the attacks in an identical fashion, even though the attacks clearly differed in features relevant for detection.

Systems in the extreme ends of both the *reference model type* axis and the *reference model generation process* axis seem to be able to detect the echo/chargen attack well. Thus, we would tentatively like to suggest that a pre-programmed pure misuse system might be the most suitable for detecting this attack. We expected such as result as the misuse model is more important than the anomaly model with intrusive behavior that is clearly distinguished from normal behavior. Further, when there are no parameters that require tuning, the self-learning property is superfluous.

The syn flood attack, as presented here, requires state for successful detection. Its intrusive behavior is also close to the normal usage of the systems, implying that you need to model both for successful detection. There is a difference of detection ability between the stateless pattern matching (Snort) and the expert system. We believe the difference can be attributed to the statelessness of the pattern matching algorithm, and also to the fact that the rules in Snort concentrate on the misuse model only; the expert system implicitly model the normal behavior with its thresholds. However, we cannot make any stronger claims about the relevance of the *reference model type*. The expert system and the neural network are placed differently in relation to the *reference model generation process* axis, but both detect the syn flood attack. For that reason, this parameter does not seem to matter when it comes to the detection ability of this attack.

We would like to point out that the syn flood attack indicates that the self-learning capability of the neural network makes it much easier to deploy, indicating that the *reference model generation process* parameter does matter in other types of considerations. For all detection methods described above, there are clear trade-offs in terms of memory footprint, efficiency, and ease-of-use for the operator. However, in this case study we concentrated on evaluating how well detection methods classified with the two attributes *reference model type* and *reference model generation process* can detect the two attack examples.

To summarize, we have shown that different detection methods vary in their ability to detect attacks. The *reference model type*, widely used in the literature, is relevant both when it comes to surveys and when it comes to judging the detection method capabilities. The *reference model generation process* is relevant for surveys but seems less important for the *IDS Detection Capability Test*. Clearly, we need to extend the study with more attacks to be able to draw better conclusions. These attacks should be chosen to reflect the claims of previous papers described in Section 2.4, for us to validate their hypotheses. Finally, we implicitly assumed TCP/UDP packets as the audit source and then the ability of the algorithms to keep state mattered. We need to further investigate the property of stateful/stateless IDS.

6 Conclusions and Future Work

In this paper, we have surveyed previous classifications done for intrusion detection. After having summarized the different terms used, we consolidated the key attributes relevant for detection methods into three new attributes: the *reference model type* used by the detection system, the *reference model generation process*, and the *reference model updating strategy*. We consider each of these attributes to be of a continuous scale instead of previous presentations where the attributes have binary values. Many systems today use mixed reference models, which motivate the continuous axes. Based on our unified framework we then resolved the previously used terms in the field concerning detection methods.

In the second part of the paper, we evaluated the new taxonomy framework with two different tests. The result from the first test, where we classified 11 IDSs based on their detection method, was that most IDSs were well dispersed in the defined space, thus making an empirical argument that the attributes *reference model type* and the *reference model generation process* are relevant for surveys.

Before applying the second test, we established that different detection methods vary in their capability to detect

specific attack types. The second empirical test measures how well we can reason about the detection capability based on detection method classes. Both the IDS and the attack taxonomies seemed to be unsuitable for such a purpose and the small number of attacks in our study also limited the generality of our conclusions. We believe that the *reference model type* expresses a valid class for reasoning about detection capability, but the *reference model generation process* has less impact.

To extend this study, we need to identify and classify more attacks with a wide variety of characteristics relevant for detection purposes, thus allowing us to strengthen our conclusions based on the second test. We have only used two dimensions in this study, and the work should of course be extended to also incorporating other important aspects. For example, our tests indicate that stateful versus stateless detection is an important property.

References

[1] John D Howard. *An analysis of security incidents on the Internet 1989-1995*. PhD thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania, April 1997.

[2] Hervé Debar, Marc Dacier, and Andreas Wespi. Towards a taxonomy of intrusion-detection systems. *Computer Networks*, 31(8):805–822, April 1999.

[3] Stefan Axelsson. Intrusion detection systems: A taxomomy and survey. Technical Report No 00-4, Dept. of Computer Engineering, Chalmers Univerity of Technology, Sweden, March 2000.

[4] D. Alessandri, C. Cachin, M. Dacier, O. Deak, K. Julisch, B. Randell, J. Riordan, A. Tscharner, A. Wespi, and C. Wüest. Towards a taxonomy of intrusion detection systems and attacks. Technical Report RZ 3366, IBM Research, Zurich Research Laboratory, 8803 Rüschlikon, Switzerland, 2001. MAFTIA project, report D3.

[5] Lawrence R. Halme and R. Kenneth Bauer. AINT misbehaving - a taxonomy of anti-intrusion techniques. In *Proceedings of the 18th National Information Systems Security Conference*, pages 163–172, Baltimore, MD, USA, October 1995. National Institute of Standards and Technology/National Computer Security Center.

[6] C. Ko, M. Ruschitzka, and K. Levitt. Execution monitoring of security-critical programs in distributed systems: A specification-based approach. In *Proceedings of the 1997 IEEE Symposium on Security and Privacy*, Oakland, California, 1997.

[7] Kathleen A. Jackson. Intrusion detection system (IDS) product survey. Technical Report LA-UR-99-3883, Los Alamos National Lab, 1999.

[8] Håkan Kvarnström. A survey of commercial tools for intrusion detection. Technical Report TR99-8, Chalmers University of Technology, 1999.

[9] Sandeep Kumar. *Classification and detection of computer intrusions*. PhD thesis, Purdue University, West Lafayette, Indiana, August 1995.

[10] Ulf Lindqvist. *On the fundamentals of analysis and detection of computer misuse*. PhD thesis, Chalmers University of Technology, Göteborg, Sweden, 1999.

[11] Peter G. Neumann and Donn Parker. A summary of computer misuse techniques. In *Proceedings of the 12th National Computer Security Conference (NCSC*, pages 396–407, Baltimore MD, 10-13 October 1989.

[12] Peter Mell. Understanding the global attack toolkit: Using a database of dependent classifiers. In *2nd Workshop on Research with Security Vulnerability Databases*, January 21-22 1998.

[13] Richard Lippmann, Joshua W. Haines, David J. Fried, Jonathan Korba, and Kumar Das. The 1999 darpa off-line intrusion detection evaluation. *Computer Networks*, Volume 34(Issue 4):579–595, October 2000. Elsevier Science B.V.

[14] Dorothy E Denning. An intrusion-detection model. *IEEE Transactions on Software Engineering*, SE-13(2):222–232, February 1987.

[15] Carl E Landwehr, Alan R Bull, John P McDermott, and William S Choi. A taxonomy of computer program security flaws. *ACM Computing Surveys*, 26(3):211–254, September 1994.

[16] Ivan V Krsul. *Software vulnerability analysis*. PhD thesis, Purdue University, West Lafayette, Indiana, May 1998.

[17] Ulf Lindqvist and Phillip A Porras. Detecting computer and network misuse through the production-based expert system toolset (P-BEST). In *Proceedings of the 1999 IEEE Symposium on Security and Privacy*, pages 146–161, Oakland, California, May 9–12, 1999.

[18] Ulf Lindqvist and Phillip A Porras. eXpert-BSM: A host-based intrusion detection solution for Sun Solaris. In *Proceedings of the 17th Annual Computer Security Applications Conference (ACSAC 2001)*, pages 240–251, New Orleans, Louisiana, December 10–14, 2001.

[19] CERT. Cert advisory ca-1996-01 udp port denial-of-service attack. `http://www.cert.org/advisories/CA-1996-01.html`, February 1996.

[20] Christoph L Schuba, Ivan V Krsul, Markus G Kuhn, Eugene H Spafford, Aurobindo Sundaram, and Diego Zamboni. Analysis of a denial of service attack on TCP. In *Proceedings of the 1997 IEEE Symposium on Security and Privacy*, pages 208–223, Oakland, California, May 4–7, 1997.

[21] Tom M Mitchell. *Machine Learning*. McGraw, 1997. ISBN 0-07-042807-7.

[22] James Cannady. Artificial neural networks for misuse detection. In *Proceedings of the 1998 National Information Systems Security Conference (NISSC'98)*, pages 443–456, Arlington, VA, October 5-8 1998.

[23] Alfonso Valdes and Keith Skinner. Adaptive, model-based monitoring for cyber attack detection. In H. Debar, L. Me, and F. Wu, editors, *From Recent Advances in Intrusion Detection (RAID 2000)*, number 1907 in Lecture Notes in Computer Science, pages 80–92, Toulouse, France, October 2000. Springer-Verlag.

[24] Martin Roesch. SNORT - lightweight intrusion detection for networks. In *Proceedings of the 13th Systems Administration Conference - LISA '99*, Seattle, Washington, USA, November 7-12 1999. USENIX.

[25] M. Almgren, H. Debar, and M. Dacier. Lightweight tool for detecting web server attacks. In *Proceedings of the Network and Distributed System Security Symposium*, 2000.

[26] Robert K Cunningham, Richard P Lippman, and Seth E Webster. Detecting and displaying novel computer attacks with Macroscope. In *Proceedings of the 2000 IEEE Workshop on Information Assurance and Security*, United States Military Academy, West Point, NY, June 6–7 2000.

[27] Wenke Lee, Sal Stolfo, and Kui Mok. A data mining framework for building intrusion detection models. In *Proceedings of the 1999 IEEE Symposium on Security and Privacy*, Oakland, CA, May 1999.

[28] H.S. Javitz and A. Valdes. The NIDES statistical component description and justification. Technical report, Computer Science Laboratory, SRI International, Menlo Park, CA, March 1994.

[29] Kevin L. Fox, Ronda R. Henning, Jonathan H. Reed, and Richard P. Simonian. A neural network approach towards intrusion detection. In *Proceedings of the 13th National Computer Security Conference*, pages 125–133, Washington D.C., October 1990. National Institute of Standards and Technology/National Computer Security Center.

[30] Hervé Debar, Monique Becker, and Didier Siboni. A neural network component for an intrusion detection system. In *Proceedings of the IEEE Symposium on Research in Computer Security and Privacy*, 1992.

Appendix A

Here we have collected the descriptions of the classified intrusion detection systems. The descriptions refer to Figure 1 found in the main text.

Snort In the lower left corner of the map are systems like *Snort* [24]. These use rather simple, often stateless, string matching to detect known attacks and all knowledge is pre-programmed. This is the most common approach in commercial detection systems today.

WebIDS *WebIDS* [25] is found to the right of Snort. It is a pattern matching system but also uses thresholds and it is a little bit more complex than Snort. Setting the thresholds requires expert knowledge about the difference between normal behavior and attack behavior.

eXpert-BSM It is possible to make more general rules that may catch groups of intrusions instead of specific instances. This is done in *eXpert-BSM* [18], where an expert system[6] is used, which can handle more complex compound rules with several ordered or unordered events and negations. This is a pure programmed system, based on misuse knowledge, but it implicitly encodes knowledge about normal behavior in the general rules. This system is placed close to WebIDS in the map.

Bottleneck verification Even further to the right, is *Bottleneck verification* [26]. This system models the valid privilege transition paths and parses user commands to find out if high privilege operations are performed without going through the "bottleneck", i.e. login or the *su* command in a UNIX system, to get higher privilege. This means that they use programmed models of both normal and attack behavior.

Specification-based In the lower right corner of the map is the *specification-based* system by Ko et al. [6]. They manually specify valid sequences of execution events for important programs and detects deviations from these sequences. This is a pure programmed system using only normal behavior data as reference for detection.

eBayes In the middle of the map we find *eBayes* [23], which monitors TCP sessions. It uses Bayes nets to model different forms of normal and attack behavior. At least theoretically, these models can be either specified, learned or hybrid. The Bayes nets rely on conditional probability tables that can be learned either off-line or adaptively.

Ripper Above eBayes, also in the middle is *Ripper* [27]. Ripper needs labeled data containing both normal data and attacks. From a large number of features Ripper selects the most discriminating and generalizable features. It can be used to create both misuse and anomaly detection models, even though the best results are achieved for misuse models. The requirements of this approach is rather close to the neural network model. The advantages of Ripper is that it gives information about which features it uses and creates more readable "rules". However, the readability of the rules probably depend on the level of detail in the labeling of training data and should be better for misuse models than anomaly models. Higher level of detail in the labeling requires more knowledge, and thus brings Ripper more towards the programmed systems. If labels are only *attack* or *normal*, it is very close to the misuse neural network in [22].

eStat To the right of Ripper, we find *eStat*, which is a pure anomaly detection system based on NIDES statistical component [28]. It is close to (but below) the anomaly-based neural network, and it has been suggested that these kinds of systems can be interchanged. However, the statistics-based systems require slightly more encoded knowledge and programming, since suitable statistical measures must be defined and adjusted, which are done automatically in the neural network.

Neural networks In the upper part of the map, we find several types of *neural networks*. They can implement unsupervised learning using Kohonens *self-organizing maps* as in [29], or they can use supervised learning with labeled data as in [22]. The unsupervised learning strategy is completely self-learning, except for some initial choice of parameters to use. The supervised learning strategy is a little bit more towards the programmed systems, since it needs chosen training examples to do its job well. The supervised systems may be pure *anomaly* detection systems (e.g. [30]) using only normal behavior data for training, or *"misuse"* detection systems (e.g. [22]), which use a mix of (labeled) normal and attack data. Misuse detection using neural networks is not very common, however, since it is difficult to obtain labeled data with a high percentage of attacks. Cannady [22] uses about 30% simulated attacks mixed into normal data.

[6] More specifically, it is a forward-chaining production system.

A WAVELET-BASED APPROACH TO DETECT COMPUTER NETWORK ATTACKS

Mohamed Hamdi, Noureddine Boudriga

National Digital Certification Agency, Tunis, Tunisia

mmh@certification.tn, nab@supcom.rnu.tn

ABSTRACT. Anomaly-based intrusion detection is a crucial issue as it permits to identify attacks that do not necessary have known signatures. Approaches using anomalies often consume more resources than those based on misuse detection and have a higher false alarm rate. This paper presents an efficient anomaly analysis method that is shown to be more accurate and less complex than the existing techniques.

KEYWORDS. Network attacks, anomalies, wavelet transform, LLipschitzipschitz regularity.

1. INTRODUCTION

Computer network system security is a growing concern due to the rapid increase in enterprises' connectivity and accessibility that has resulted in more-frequent intrusions, misuses and security attacks. The problem of detecting intrusions, attacks and other forms of communication network abuses and misuses can be considered as finding security violations, or non permitted deviations, of the characteristic properties in the monitored network. This is because one can assume the fact that the suspicious activity might be different from the normal activity. However, in various cases, realize or detect such differences before any damage can be observed is a very complex task.

The ideal detection of a security abuse would be to discover the related activity before the attack can achieve its objectives. This would require recognition of the attack as soon as it takes place. Different approaches have been developed for this purpose, [1, 2, 3]. They mainly include two categories: pattern-based detectors and anomaly detection systems. While the former methods locate security problems by examining patterns of users activities within flows, logs, and usage files, the latter class looks for deviation in normal usage behavior of the network, information systems, and user profiles. Even though some of these methods are witnessing an increasing interest, no method can catch all types of intrusions.

The behavior-based techniques of detecting anomalies would involve monitoring the computer network state over a period of time while collecting information about the network normal behavior. Accordingly, some parameters of the computer network are identified as indicators of the anomalies. The decision is based on the availability of methods that are able to classify the anomaly by monitoring a sensors network for abnormal patterns of the system usage.

This paper describes the design of an anomaly-classifier system for intrusion and misuse detection and addresses the related mathematical issues. The classifier can monitor the activities of the computer network at multiple levels (from traffic engineering to user activity levels) and determines the correlation among the monitored parameters (or metrics). Our

method is based on the concept of period of observation and uses wavelet theory. Our method has three main features: first it does not require the storage of users profiles, data files on attacks, or statistical usage. Second, it offers a reduced complexity of the form $O(n)$, where n is the size of the observation periods. Last, it does not require an a priori information about the noise (or anomaly).

The remaining part of this paper is organized as follows: Section 2 describes the commonly used techniques to represent and detect computer network anomalies. Section 3 gives a formal representation of these anomalies. Section 4 explains how anomaly detection can be performed using wavelet theory and presents a case study where wavelet-based classification is demonstrated. Section 5 concludes this paper.

2. Anomalies in Network Security

The objective of intrusion detection is to monitor network traffic and generate alerts when the occurrence of an attack is detected. Two principal detection mechanisms are often considered: signature-based detection and anomaly based-detection. Signature based-detection is closely related to pattern recognition as the sniffed traffic is compared to a set of known attacks. The efficiency of this approach depends essentially on the number of attack signatures that the system knows. In fact, it does not generate an alert for an attack that has not a corresponding signature. In the second mechanism, detection results depend on the values of several measurable features, called *metrics*. This assumes that system's normal behavior can be described. Therefore, what deviates from this proper behavior is the source of an alert.

Two categories of techniques are commonly used in this context. The first, network profiling, consists in generating a periodically updated profile of the system's normal behavior by the use of historical data (frequency tables, means, entropies, etc.). Then, during each observation period, the monitored packet flow is classified as "normal" or "abnormal". The second class is called thresholding and is based on the comparison of certain attributes of the system to values that correspond to the boundary of the normal events. In practice, threshold detection is a small component of an IDS and is seldom used as an independent system because it is generally based on a limited set of metrics that do not express completely the state of the monitored network.

A crucial problem that needs to be addressed when discussing thresholding is the threshold setting mechanism. In fact, detection can give a high rate of false positives if the threshold value is too low or a great amount of false negatives if it is excessively high. In most cases, Bayesian decision theory is used to select the optimal threshold [1]. According to this theory, a threshold is said optimal if it minimizes the Bayes risk function that is computed using the false positive and the false negative probabilities. The major disadvantage of this approach is that one of its fundamental hypotheses can hardly be verified in our context. In fact, to compute the Bayesian risk, an a priori knowledge about the distribution of the network attacks is needed. As the occurrence of network attacks is mainly related to human factors, it is not easy to statistically model it even by the use of historical data.

Another shortcut of classic anomaly-detection techniques is that they are not built upon a strong relationship between anomalies and attacks. This can lead to a high false alarm rate as many abnormal values of a measured signal can correspond to a normal behavior of the system under analysis. For instance, consider a web server that was victim of a SYN/flooding attack (see Example 1). If the security analyst observes the number of packets per second that the server receives, he would notice a peak (abnormal value) at the moment of the attack.

However, and as illustrated in Figure 1, a peak does not always imply the occurrence of a malicious action. Indeed, while the first peak corresponds to the attack mentioned above, the second peak results from a quasi-simultaneous connection of big amount of clients. The latter event is normal as it occurs at 08:00 a.m. (beginning of office hours), a time when most of employees connect to the website from their offices. The conclusion from this example is that the detection of abrupt changes or abnormal values is not sufficient to characterize computer network attacks.

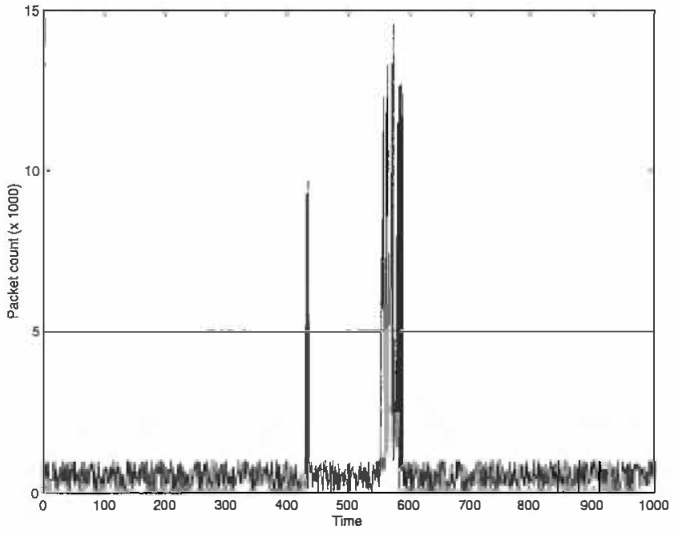


FIGURE 1. Evolution of packet count according to time for a public web server.

Therefore, we need an other detection mechanism that should be more adapted to network security in the sense that it takes into consideration the tight difference between anomalies and attacks. This mechanism should also have a low computational complexity because of the limited resources of the intrusion detection components.

To address this issue, we propose to model attacks as a noise that affects a signal, or a set of signals, that represent the normal behavior of the system. The benefit from this reasoning is that several techniques commonly used in signal processing to detect and cancel different kinds of noise can be adapted to our context. In the frame of our work, we focused on wavelet theory and Lipschitz regularity as will be detailed in Section 4. However, to apply these methods, a convenient modeling of network anomalies is needed.

3. ANOMALIES REPRESENTATION

Typically, an information system is constituted by a ***network***, a set of ***hosts*** and a set of ***users***. Then, a sound representation of anomalies should include these three fundamental components. In this section, we introduce an approach to discuss information system anomaly on the basis of the monitoring of users' actions, host behavior and network traffic.

3.1. Metrics representation. The first task in the anomaly representation is to define the metrics that have to be monitored. Consider three families of functions denoted by $(v_i(t))_{i\in\{1,..,\Upsilon\}}$, $(\eta_j(t))_{j\in\{1,..,\mathbf{H}\}}$ and $(\nu_k(t))_{k\in\{1,..,\mathbf{N}\}}$ that model the following activities respectively:

- User-level activities. These include attributes, such as most used commands, typing frequency, and login/logout period, that help develop the profiles of behavior patterns of users.
- Host-level activities. These include attributes (such as file structure, consumed CPU, and consumed memory) that provide indication of resource usage.
- Network-level activities. These use attributes (such as total packet count, packets with specific source or destination ports, and packets with specific source or destination addresses) to provide information that are gathered on network usage and engineering.

The above functions are the metrics that represent efficiently the information system state. They should be measured and monitored continuously. Information system administrators have to determine what attributes to record to ensure efficient representations of their systems. Moreover, for each of these metrics, a set of decision rules have to be defined in order to decide whether a value taken by a given function corresponds to a misuse or a legitimate use of the system. To this purpose, we introduce the following sets:

- $(N_{\upsilon_i})_{i\in\{1,..,\Upsilon\}}$ (resp. $(A_{\upsilon_i})_{i\in\{1,..,\Upsilon\}}$): the sets of normal (resp. abnormal) values that could be taken by the functions $(\upsilon_i(t))_{i\in\{1,..,\Upsilon\}}$,
- $\left(N_{\eta_j}\right)_{j\in\{1,..,\mathbf{H}\}}$ (resp. $\left(A_{\eta_j}\right)_{j\in\{1,..,\mathbf{H}\}}$): the sets of normal (resp. abnormal) values that could be taken by the functions $(\eta_j(t))_{j\in\{1,..,\mathbf{H}\}}$,
- $(N_{\nu_k})_{k\in\{1,..,\mathbf{N}\}}$ (resp. $(A_{\nu_k})_{k\in\{1,..,\mathbf{N}\}}$): the sets of normal (resp. abnormal) values that could be taken by the functions $(\nu_k(t))_{k\in\{1,..,\mathbf{N}\}}$.

An action at an instant t_0, denoted $\tau(t_0)$, can force the modification of the values of the aforementioned functions at t_0. In other terms:

$$\tau(t_0) = (\upsilon_1(t_0),..,\upsilon_\Upsilon(t_0),\eta_1(t_0),..,\eta_\mathbf{H}(t_0),\nu_1(t_0),..,\nu_\mathbf{N}(t_0))\,.$$

Definition: An action $\tau(t_0)$ is said to be ***anomalous*** if one of the following conditions holds:

(1) $\exists i \in \{1,..,\Upsilon\}$ such that $\upsilon_i(t_0) \in A_{\upsilon_i}$,
(2) $\exists j \in \{1,..,\mathbf{H}\}$ such that $\eta_j(t_0) \in A_{\eta_j}$,
(3) $\exists k \in \{1,..,\mathbf{N}\}$ such that $\nu_k(t_0) \in A_{\nu_k}$.

This means that if, during a action, a metric takes an abnormal value, then the whole transaction is abnormal. Therefore, the efficiency of this decision rule depends mainly on the efficiency of the elementary decision rules stating whether the value of a single metric is normal or not. The characterization of network anomalies is then easy as it consists simply in checking if the values of the monitored metrics belong, at a given instant, to several predefined sets. A more sensitive problem is to state wether an anomaly is related to an attack or not.

3.2. **The attack/singularity analogy.** In its broadest definition, a computer attack is any malicious action performed against a computer system or the services it provides. In [4], B. Schneier describes computer attacks as ***exceptions*** or ***events that take people by surprise***. This feature makes the detection of these attacks a difficult task. In fact, the detection system is often faced with the sensitive problem of stating if an event corresponds to a malicious action or not without having enough knowledge to do.

The aforementioned definition of computer attacks is particularly convenient in our context. In fact, most of these attacks can be assimilated as abrupt changes in some measurable signals. Thus, anomaly detection can be performed through the detection of singularities in a set of significant metrics. Obviously, the choice of these metrics is of primal importance as it has

a bing influence on the performance of the detection system. However, this issue is out of the scope of this paper. In the following, we give two examples illustrating the selection of efficient metrics depending on the nature of the attack.

Example 1. ***SYN flood (Neptune) attack***

This is a Denial of Service (DoS) attack to which most of TCP/IP implementations are vulnerable. It consists in opening enough half-open TCP connections to exceed the capacity of the victim machine who will be unable to accept more connections. This attack can be efficiently detected by monitoring the number of open connections on the protected machine.

Example 2. ***Mailbombing***

A mailbomb is an attack in which many messages are sent to a user registered on the mail server overflowing its queue and causing the failure of his account. It is rather harmful to particular users than to the whole server. It can be detecting through the measurement of the number of messages (or the amount of bytes) received by the server per time-unit.

In these two examples, anomalies can be easily identified by detecting the exceedance of the monitored metrics to a specific value. Nonetheless, to state if an anomaly is an attack or not, a more deep mathematical analysis of the anomaly is required. A detailed study of the first example is given in Section 4.

4. Wavelet-based Detection of Attacks

4.1. **Theoretical development.** In practical situations, the signal of interest $\sigma(t)$ (number of open connections, number of transmitted packets, etc.) is not known for all abscissa t but it is uniformly sampled. Then, what is really handled is a set of values $(\sigma(n))_{n\in\{1,..,N\}}$ where N is the number of samples. As $\sigma(t) \in \mathcal{L}^2(\mathbb{R})$ (it has a finite energy), its discrete wavelet transform can be computed. According to the results discussed in the previous section, the Lipschitz regularity of $\sigma(t)$ can be determined by studying the decay of $\left|W_\sigma(2^{-j}, k2^{-j})\right|$ across the scales j. $\left|W_\sigma(2^{-j}, k2^{-j})\right|$ is the wavelet transform of the function $\sigma(t)$ at the resolution j computed through a dilated convolution with a wavelet function denoted $\Psi(t)$. The following proposition is a straight-forward implication of theorem 3 (we assume that the wavelet $\Psi(t)$ has a compact support, n vanishing moments and is n times continuously differentiable).

Proposition 1. ***Characterizing computer attacks using the wavelet transform***

Let $\sigma(t) \in \mathcal{L}^2(\mathbb{R})$ be a function representing a monitored metric. If there exist $J \geq 2$ and k_0 such that:

1. *There exists $j_0 \in \{1, .., J\}$, such that $(2^{-j_0}, k_0 2^{-j_0})$ is a local maxima,*
2. *For all $j \in \{j_0, .., J-1\}$, $\left|W_\sigma(2^{-j-1}, k_0 2^{-j-1})\right|$ is a local maxima,*
3. *For all $j \in \{j_0, .., J-1\}$, $\left|W_\sigma(2^{-j-1}, k_0 2^{-j-1})\right| \geq \left|W_\sigma(2^{-j}, k_0 2^{-j})\right|$,*

Then, the point k_0 corresponds to a computer attack.

More simply, if a local maxima is detected at a point $(2^{-j_0}, k_0 2^{-j_0})$ and if k_0 corresponds to local maxima with increasing modulus at scales finer than 2^{-j_0}, then k_0 corresponds to a computer attack.

The main advantages of this method are the good spatial localization and the low numerical cost. The first feature is due to the wavelet spatial properties while the second one comes from

the fact the wavelet transform of a signal of size N can be computed in $O(N)$ steps according to Mallat's pyramidal algorithm [7].

4.2. **Case study.** In this section, we illustrate the result of proposition 1 in a concrete case. We discuss a representative example to demonstrate the benefit of the use of wavelet theory in computer network intrusion detection. We use data corresponding to a real attack provided by the Information System Technology Group of MIT (Massachusetts Institute of Technology)[1]. That is a distributed denial of service attack using the ***mstream*** software and corresponding to the first scenario (Lincoln Laboratory Scenario DDoS 1.0) for year 2000. It was performed over multiple steps (probing, breaking in, installing trojan, launching DDoS) against a US government which IP address was 131.84.1.31.

The measured metric in our case is the packet count $c(t)$ received by the victim machine over observation periods that were fixed to one minute. Figure 2 represents the evolution of this function over time (several little fluctuations corresponding to normal traffic can not be seen on the plot because of the great amount of packets received at the moment of the attack).

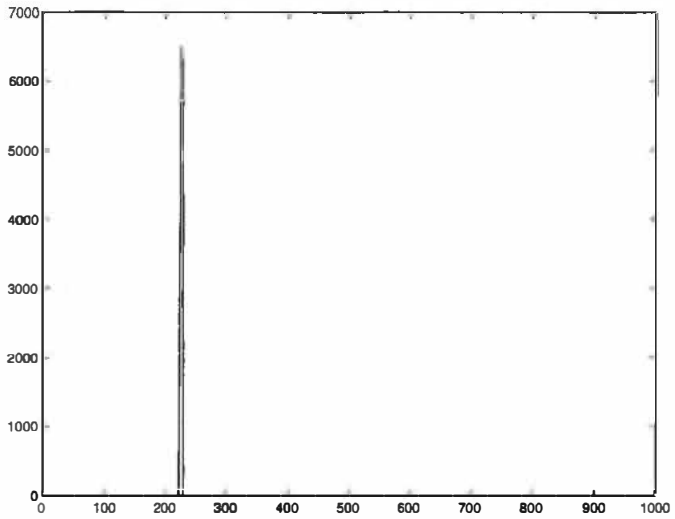


FIGURE 2. Time-evolution of the packet count.

The reader would notice the presence of an important peak that corresponds to the DDoS attack. Several little fluctuations, corresponding to normal traffic, can not be seen because of the huge difference between their amplitude and the number of packets received by the victim machine at the moment of the attack. In a real case, the administrator could not determine if this peak corresponds to normal traffic or to an attack.

In mathematics, the local regularity of functions is often evaluated by the use of Lipschitz exponents. In [5], wavelets have been proved to be particularly efficient to evaluate the local regularity of functions unlike the Fourier transform which is adapted only to global regularity. Indeed, it offers the possibility to analyze the pointwise regularity of a function. This is due to the good localization in the scale-space domain of the wavelet basis. In fact, through the measurement of the decay of $|W_f(s,t)|$, the wavelet transform of a function f at a scale s and a point t (see Appendix 1 for more details about the wavelet transform), in a two-dimensional

[1]**http://www.ll.mit.edu/IST/ideval/index.html**

neighborhood of t_0 in the scale-space (s,t), one can estimate the Lipschitz exponent, thus stating about the singularity, at the point t_0.

As this method may have, in some cases, a heavy computational load, it is rarely applied in practice. Often, in real world applications, only the decay of $|W_f(s,t_0)|$ is measured (a one-dimensional neighborhood is handled instead of a two-dimensional one). Nonetheless, it was shown in [6], through a counter-example, that this practical approach does not always yield reliable results. Thus, a method that allies mathematical tractability and numerical ease was proposed in [5] (theorem 3).

To address this problem, we perform the wavelet decomposition of the packet count $\mathbf{c}(t)$ to the sixth resolution using the well-known Daubechies wavelet (with four vanishing moments) and we study the evolution of the wavelet maxima across scales. Figure 3 represents the spatial localization of wavelet maxima at each scale. From Figure 4, it can be deducted that maxima wavelet coefficients corresponding to the abscissa $t_1 = 219$ are increasing over scales thus implying that the singularity of interest is an attack. In Figure 5, the modulus of another local maxima, situated at the abscissa $t_2 = 683$, is plotted for each decomposition scale. As this modulus decreases, it can be concluded, according to proposition 1 that this abscissa does not correspond to an anomaly, which is effectively true.

5. Conclusion

In this paper, a method for detecting computer network attacks through the use of wavelet theory and Lipschitz exponents is proposed. The theoretical fundamentals of the method are established and experiments are carried out on traffic containing real attacks to check the correctness of our reasoning. It is shown that the proposed technique allows an efficient detection of computer attacks modulo a good choice of the measured metrics. The merit of our approach is that it exploits the intrinsic properties of the wavelet transform (essentially the low computational cost and the good spatial localization).

References

[1] P. Helman, G. Liepins, "Statistical Foundations of Audit Trail Analysis for the Detection of Computer Misuse," IEEE Transactions on Software Engineering, Vol. 19, No. 9, September 1993.

[2] D. Dasgupta and F.A. Gonzalez,"An Intelligent Decision Support System for Intrusion Detection", Proc. Int. Workshop on mathematical methods, models and architectures for computer network security (MMM-ACNS), LNCS, May 21-23, St Petersburg, Russia.

[3] M.L. Gassata, "A genetic algorithm as an alternative tool for security audit trail analysis", in Proc. of the first Int. workshop on recent advances in intrusion detection, 1998, Louvain, Belgium.

[4] B. Schneier, "*Secrets and Lies: Digital Security in a Networked World,*" John Wiley & Sons,ISBN: 0471253111, 2001.

[5] S. Jaffard, "Exposants de Holder en des Points Donnés et Coefficients d'Ondelettes," Notes au Compte-Rendu de l'Académie des Sciences, France, Vol. 308, Série I, pp. 79-81, 1989.

[6] S. Mallat, W.L. Hwang, "Singularity Detection and Processing with Wavelets," IEEE Transactions on Information Theory, Vol. 38, n° 2, pp. 617-643, March 1992.

[7] S.G. Mallat, "A Theory of Multiresolution Signal Decomposition: the Wavelet Representation," IEEE Transactions PAMI, Vol. 2, n° 7, pp. 674-693, 1989.

Appendix A. Wavelet Theory

A.1. Continuous Wavelet Transform (CWT). A function $\Psi(t)$ belonging to $\mathcal{L}^2(\mathbb{R})$ (Holder space of functions with finite energy) and centered around zero ($\int_{-\infty}^{+\infty} \Psi(t)dt = 0$) is said to

be a a wavelet if, and only if, its Fourier transform $\widehat{\Psi}(\omega)$ satisfies the following condition:

$$\int_{-\infty}^{+\infty} \frac{\left|\widehat{\Psi}(\omega)\right|^2}{\omega} d\omega < +\infty. \tag{1}$$

$\Psi(t)$ is also called the **"mother wavelet"**.

The wavelet transform of a function $f(t) \in \mathcal{L}^2(\mathbb{R})$ is defined by:

$$W_f(s,t) = f * \Psi_s(t) = \int_{-\infty}^{+\infty} f(u)\Psi_s(t-u)du, \tag{2}$$

where $s \in \mathbb{R}_+^*$ is the scale factor, $*$ is the convolution operator and $\Psi_s(t) = \frac{1}{\sqrt{s}}\Psi(\frac{t}{s})$ is the dilation of the wavelet $\Psi(t)$ by s.

It offers an alternative to the windowed Fourier transform (or Short-Time Fourier Transform, STFT) for non-stationary signal analysis. The principal shortcut of the STFT is that is uses a constant-length window that does not permit a spatial analysis of the signal. On the other hand, in the wavelet transform, wide windows are applied for low frequencies and short windows for high frequencies. This provides an idea about the global and the local properties of the signal $f(t)$.

The function $W_f(s,t)$ has two main properties. The first results from the fact that the Fourier transform of the convolution of two functions is the product of their Fourier transforms. In other terms:

$$\widehat{W_f}(s,\omega) = \widehat{f}(\omega)\widehat{\Psi}(\omega). \tag{3}$$

The second feature is that the wavelet transform is invertible as $f(t)$ can be reconstructed as follows:

$$f(t) = \int_0^{+\infty}\int_{-\infty}^{+\infty} W_f(s,u)\Psi_s(t-u)du\frac{ds}{s}. \tag{4}$$

A.2. **Discrete Wavelet Transform (DWT).** Often in practice, $W_f(s,t)$ are computed for discrete values of s and t. The main constraint when choosing these values is to ensure that the transform is not redundant. According to Equation 4, this requirement can be satisfied if the functions $(\Psi_{s,u}(t))_{s\in\mathbb{R}_+^*,u\in\mathbb{R}}$ given by:

$$\forall s \in \mathbb{R}_+^*,\ \forall u \in \mathbb{R},\ \Psi_{s,u}(t) = \Psi_s(t-u) = \frac{1}{\sqrt{s}}\Psi(\frac{t-u}{s}), \tag{5}$$

are a basis of $\mathcal{L}^2(\mathbb{R})$.

The DWT assumes the computation of wavelet coefficients for discrete scale factor $s = 2^{-j}$ and translation $u = k2^{-j}$ for $j, k \in \mathbb{Z}$. In fact, these values of the wavelet transform parameters define an orthogonal basis $\left(\Psi_{j,k}(t) = 2^{-\frac{j}{2}}\Psi(2^{-j}t - k)\right)_{j,k\in\mathbb{Z}}$ called **the wavelet basis.**

To represent a function $f(t)$ at different resolutions through the use of the "mother wavelet", we have to introduce the scaling function defined by:

$$\Phi(t) = \sum_{k=-1}^{k=N-2} (-1)^k c_{k+1}\Psi(2t+k), \tag{6}$$

where c_k are called the wavelet coefficients verifying $\sum_{k=0}^{k=N-1} c_k = 2$ and $\sum_{k=0}^{k=N-1} c_k c_u = 2\delta_{u,0}$ (δ is the delta function).

In [?], Mallat proposed a description of a discrete orthonormal wavelet transform that is based on the concept of multiresolution analysis.

Definition 1. ***Multiresolution analysis***

A multiresolution analysis of $\mathcal{L}^2(\mathbb{R})$ is defined as a set of closed subspaces V_j of $\mathcal{L}^2(\mathbb{R})$, $j \in \mathbb{Z}$, verifying the following properties:

(1) $V_j \subset V_{j+1}$
(2) $v(t) \in V_j \Rightarrow v(2t) \in V_{j+1}$
(3) $v(t) \in V_0 \Rightarrow v(t+1) \in V_0$
(4) $\bigcup_{j\in\mathbb{Z}} V_j = \mathcal{L}^2(\mathbb{R})$ and $\bigcap_{j\in\mathbb{Z}} V_j = \emptyset$
(5) $\exists g(t) \in V_0$ such that $(g(t-k))_{k\in\mathbb{Z}}$ is a Riesz base of V_0 (i.e. $\exists A > 0$, $B < +\infty$ s.t. $A \leq \sum_{n\in\mathbb{Z}} |\widehat{g}(\omega + 2\pi n)|^2 \leq B$).

If we assume the approximation of a function f at a resolution j is its projection on V_j, then these properties would have the following significations:

- The first condition means that the information contained in the approximation of a function at a resolution j is necessarily included in the approximation at the resolution $j+1$.
- The second condition is the one that implies scale and dilation invariance. From a function $v(t) \in V_j$ (contains no details or fluctuations at scales smaller than 2^{-j}), the function $v(2t)$ can be obtained by squeezing $v(t)$ by a factor of 2. Thus, this function does not contain details at scales smaller than 2^{-j-1}.
- The third condition corresponds to shift invariance of the spaces V_j. If the approximation of a function at the j^{th} resolution $v_j(t)$ belongs to V_j, so do its translates by integers $(v_j(t-k))_{k\in\mathbb{Z}}$.

Mallat showed that every function $f(t) \in \mathcal{L}^2(\mathbb{R})$ can be represented at the r^{th} resolution by the expression

$$f(t) = \sum_{k\in\mathbb{Z}} \lambda_{-r,k}\varphi_{-r,k}(t) + \sum_{j=-r}^{j=-1} \sum_{k\in\mathbb{Z}} \gamma_{j,k}\psi_{j,k}(t), \tag{7}$$

where $\varphi_{j,k}(t) = \sqrt{2^j}\Phi(2^j t - k)$ and $\psi_{j,k}(t) = \sqrt{2^j}\Psi(2^j t - k)$, $\forall j \in \{-r, .., -1\}$.

$(\lambda_{-r,k})_{k\in\mathbb{Z}}$ can be seen as the coefficients of the projection of f on V_{-r} and $(\gamma_{j,k})_{j\in\{-r,..,-1\},k\in\mathbb{Z}}$ as those of the projection of f on O_j that verifies the following equation:

$$V_{j+1} = V_j \oplus O_j, \forall j \in \{-r, .., -1\}\,. \tag{8}$$

This means that $(\varphi_{j,k}(t))_{k\in\mathbb{Z}}$ (resp. $(\psi_{j,k}(t))_{k\in\mathbb{Z}}$) is a basis of V_j (resp. O_j) for every $j \in \{-r, .., -1\}$.

As an extension to this reasoning, the function $f_{-r}(t) = \sum_{k\in\mathbb{Z}} \lambda_{-r,k}\varphi_{-r,k}(t)$ can be written as the sum of its projections on V_{-r-1} and O_{-r-1}:

$$f_{-r}(t) = \sum_{k\in\mathbb{Z}} \lambda_{-r-1,k}\varphi_{-r-1,k}(t) + \sum_{k\in\mathbb{Z}} \gamma_{-r-1,k}\psi_{-r-1,k}(t). \tag{9}$$

By combining equations 7 and 9, we obtain the following expression of $f(t)$:

$$f(t) = \sum_{k\in\mathbb{Z}} \lambda_{-r+1,k}\varphi_{-r+1,k}(t) + \sum_{j=-r+1}^{j=-1} \sum_{k\in\mathbb{Z}} \gamma_{j,k}\psi_{j,k}(t). \tag{10}$$

This brings to evidence the main feature of the wavelet transform: the representation of a function at a given resolution (or scale) can be obtained from its representation at a coarser resolution. The transform is then said to be multiscale.

Appendix B. Singularities and Lipschitz regularity

In mathematics, the local regularity of functions is often evaluated by the use of Lipschitz exponents.

Definition 2. ***Local Lipschitz regularity, uniform Lipschitz regularity***

A function $f(t)$ is said to be Lipschitz-α, for $0 \leq \alpha \leq 1$, at a point t_0, if, and only if, there exists a constant A such that for all points t in a neighborhood of t_0

$$|f(t) - f(t_0)| \leq A\,|t - t_0|^{\alpha}. \tag{11}$$

$f(t)$ is uniformly Lipschitz-α over the interval $]a, b[$ if there exists a constant A such that Equation 11 holds for every $(t_0, t) \in]a, b[^2$.

Furthermore, $f(t)$ is said to be singular in t_0 if it is not Lipschitz-1 in t_0.

It is noteworthy that this definition can be extended to $\alpha > 1$.

Theorem 1. ***Lipschitz regularity characterization with the Fourier transform***

A function $f(t)$ is bounded and uniformly Lipschitz-α over $\mathbb{R}$ if

$$\int_{-\infty}^{+\infty} \left|\widehat{f}(\omega)\right| (1 + |\omega|^{\alpha})\, d\omega \;<\; +\infty. \tag{12}$$

In [1], Jaffard proposed a theorem giving a necessary condition and a sufficient condition for characterizing local Lipschitz regularity by using the wavelet transform.

Theorem 2. ***Lipschitz regularity characterization with the wavelet transform***

Let $\Psi(t) \in \mathcal{L}^2(\mathbb{R})$ an n times continuously differentiable wavelet having n vanishing moments ($\int_{-\infty}^{+\infty} t^k \Psi(t) dt = 0$ for all $0 \leq k \leq n+1$) and a compact support. If $f(t) \in \mathcal{L}^2(\mathbb{R})$ is Lipschitz-α at a point t_0, $0 \leq \alpha \leq n$, then there exists a constant A such that for all points t in a neighborhood of t_0 and any scale s

$$|W_f(s,t)| \leq A\,(s^{\alpha} + |t - t_0|^{\alpha}). \tag{13}$$

Conversely, $f(t)$ is Lipschitz-α at t_0, $0 \leq \alpha \leq n$, if the two following conditions are verified:

1. There exists $\varepsilon > 0$ and a constant A such that for all points t in a neighborhood of t_0 and any scale s

$$|W_f(s,t)| \leq A s^{\varepsilon}. \tag{14}$$

2. There exists a constant B such that for all points t in a neighborhood of t_0 and any scale s

$$|W_f(s,t)| \leq B\left(s^{\alpha} + \frac{|t - t_0|}{|\log|t - t_0||}\right). \tag{15}$$

Theorem 3. ***Lipschitz exponents estimation using wavelet maxima***

Let $\Psi(t)$ be a wavelet with compact support, n vanishing moments and n times continuously differentiable. Let $f(t) \in \mathcal{L}^2(\mathbb{R})$. If there exists a scale $s_0 > 0$ such that for all scales $s < s_0$ and $t \in]a, b[$, $|W_f(s,t)|$ has no local maxima, then for any $\varepsilon > 0$, $f(t)$ is uniformly Lipschitz-n on $]a + \varepsilon, b - \varepsilon[$.

This theorem indicates the presence of a maximum in the modulus of the wavelet transform $|W_f(s,t)|$ at the finer scales where a singularity occurs. Discontinuities in a function f can be assimilated to the fact that $|W_f(s,t)|$ remains constant over a large range of scales in a spatial neighborhood of t_0.

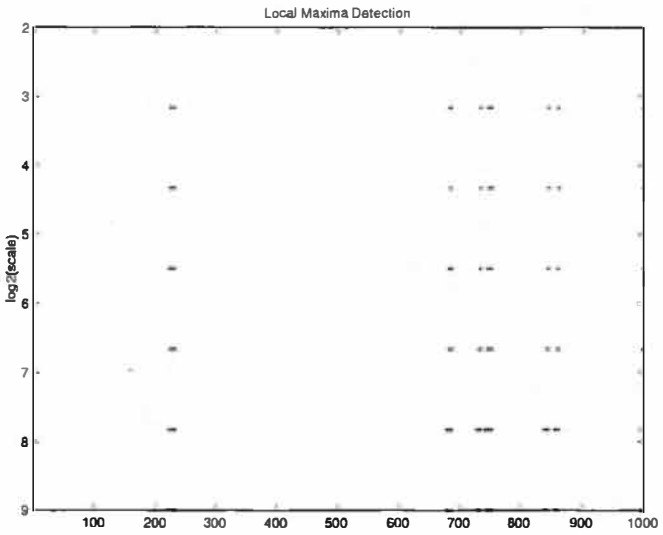


FIGURE 3. Detection of wavelet maxima.

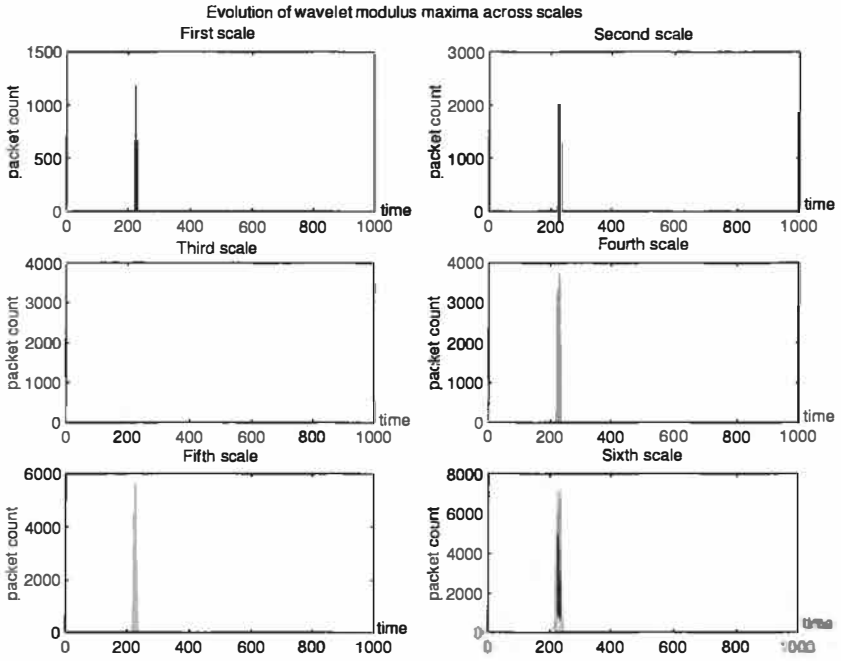


FIGURE 4. Evolution of wavelet maxima across scales for $t_1 = 219$.

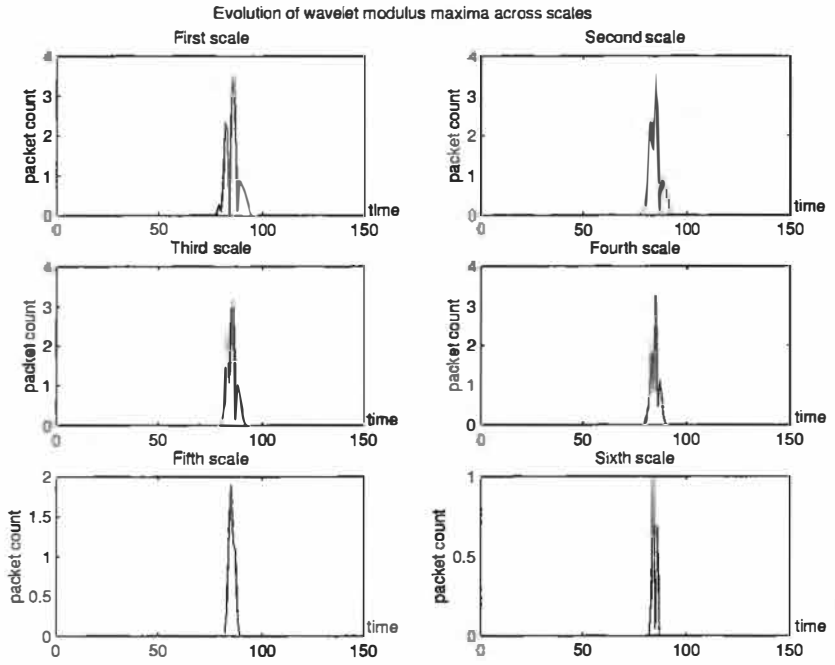


FIGURE 5. Evolution of wavelet maxima across scales for $t_2 = 683$.

On Visualising Intrusions

Henrik Almegren *Ola Söderström* *Erland Jonsson*
Department of Computer Engineering, Chalmers University of Technology
SE-412 96 Göteborg, Sweden
henrik@almegren.org,{uu,jonsson}@ce.chalmers.se

Abstract

In this paper we argue that *information visualisation* techniques can be used to analyse network and system events for the purpose of discovering attempts to compromise the security of computer systems. The idea is to use the human mind and its skills in pattern recognition and image interpretation as a basis for the non-trivial decision-making involved in this process.

Our method does not use any advanced pre-processing of data. We leave the analysis task totally to the human by providing him/her with a visualisation of the information. We believe that computer security in general and intrusion detection in particular could benefit from such an approach.

We work with input data created for evaluation of intrusion detection systems and analyse system logs and network traffic using a commercial visualisation tool. We have done no other pre-processing than converting the data to formats known by the visualisation tool.

We show that it is possible to visualise intrusions and attacks by using simple visualisation methods, and we believe that information visualisation could prove to be a very useful tool for future intrusion detection systems.

1 Introduction

Several methods have been developed to avoid intrusions in computer systems and withstand denial-of-service attacks. Very few of those are based on the visual paradigm, which takes advantage of the human mind and its ability to draw conclusions and capacity for image interpretation.

In intrusion detection, it is a difficult, if not impossible, task to draw a line between allowed and forbidden and between normal and abnormal. The result of automatic intrusion detection is often a loss of information or too many false alarms. It can be argued that automated intrusion detection systems (IDSs) will never be able to detect all attacks on a computer system or network — at least not while preserving an acceptable false alarm rate. The idea presented in this paper is to leave it to the human to make the distinction between normal and abnormal by using a graphical representation of data as the basis for the decision.

The distinction is made between information visualisation and scientific visualisation in that the latter is visualising something that has a geometry in itself, physically based data, e.g. a picture of the airflow around a car or chemical molecules. Information visualisation is visualising discrete objects or abstract data, e.g. documents, databases of any kind or statistics. What we do is information visualisation and our data is system logs and records of network traffic [1].

The primary purpose of this work is to investigate the possibilities of visualising intrusions and attacks on computer systems. Here, we first take a brief look at previous work, and then present four visualisations of attacks based on data from operating system logs and network traffic logs. We conclude our paper with a discussion of our results and conclusions.

2 Background

A small number of publications describe attempts to use visualisation for intrusion detection or similar tasks. A selection of these is given below. We also briefly discuss the more general field of information visualisation and talk about our data sources.

2.1 Related work

In [2] Girardin and Brodbeck describe an experiment in which they try to visualise firewall logs for monitoring purposes. To cope with the multidimensional data and classify log events automatically, two different pre-processing algorithms are used: spring layout and self organising maps. The results of these processes are then visualised with a few different basic two dimensional maps in which one can identify clusters of log events, recognise patterns and identify outliers. While these experiments detected no real intrusions in the firewall logs that were used, some scanning activity and abuse of services were detected. Girardin's work continues in [3] with analysis of network traffic but in this case he uses network traffic gathered using `tcpdump`.

The work of Erbacher and Frincke uses visualisation together with the intrusion detection system Hummer [4, 5]. The Hummer system can handle large distributed systems and provides data gathering and filtering. The system described in [4] visualises nodes in a network by focusing on one node and displaying network traffic between this node and other nodes in the network. This is also further developed in [6] and [7].

Another project in this field is Elisha, which is a visual-based anomaly detection system developed in a project at the University of California, Davis. The demo version of this system described in [8], addresses the MOAS conflict in BGP, Boarder Gateway Protocol.

2.2 Information visualisation

Information visualisation is the art of translating abstract information into a picture. This picture should serve to make the viewer understand the information and to make it possible to extrapolate and draw conclusions. Visualisation helps us to understand, to see, to think, to make decisions and to remember. A good visualisation can be very valuable; it helps the mind to be much more effective. The invention of visual artifacts, from writing in mathematics to diagrams and visual computing, can indicate the progress of civilisation [1, 9]. It is important that the visualisation provides interactivity. The user should be able to adapt the visualisation so that the important properties of the information are exposed. It is best if this can be done in real time, so called direct manipulation [10].

A visualisation always has an inherent limitation in that it can only display a certain number of dimensions. A scatter plot, for example, can only show two, perhaps three dimensions. This problem can be solved in different ways. One is to process the data with an algorithm that reduces the number of dimensions. The data can then be visualised in a simple way, but information is also lost in the process [2, 3]. Another is to add dimensions to the visualisation in different ways. In a scatter plot, this could be to apply a different size, shape or colour to each marker [10]. We have used the latter in our work.

2.3 Data sources

Exposure of malicious activity is possible only if the input data contain the activity in the first place. Two common data sources of intrusion detection systems are network traffic and host data. IDSs that make analyses of network data are called network-based IDSs, while IDSs that analyse data generated by one or more hosts are called host-based IDSs.[1]

Although network data are the most dominant form of input data processed by intrusion detection systems, we believe that host-based monitoring provides an important complement [11, 12], in particular if the host-based IDS takes data generated by the operating system kernel as input. As both network-based and host-based IDSs are limited by the content of the data they take as input, attack coverage is likely to increase if both data sources are analysed. This hypothesis is supported by the experimental results reported in this paper.

3 Visualisation of attacks

The following sections show the results of visualising data from the operating system kernel as well as network traffic data. In total, we show four visualisations, three of attacks exposed in system events and one attack exposed in network data. Exposed here means creating a visualisation in which the anomalous pattern generated by the attack is clearly outlined.

We look at one denial-of-service attack, two buffer overflow attacks, where the user assumes root privileges, and one probe attack where a so-called port scanning is done. Further information on the meaning of these attack categories may be found in [13].

3.1 Input data

The data we analyse in this paper originate from the DARPA sponsored intrusion detection evaluation made at MIT Lincoln Laboratory in 1998, hereafter referred to as the IDEVAL data. The data contain attacks embedded in normal background traffic and normal operating system events.

Network traffic was recorded by the `tcpdump` software, and system audit records were generated by the Basic Security Module (BSM) present on the Unix Solaris operating system.

3.1.1 BSM

BSM is a kernel-based auditing mechanism providing TCSEC C2 capabilities [14]. Kernel-based auditing means that support for auditing is compiled into the kernel itself, which allows for the interception of system calls and direct access to user credentials and other relevant information for security auditing. BSM also allows user level events such as su(1M), login(1), mount(1M) etc, to be monitored. The target system on which BSM auditing was enabled was running Solaris 2.5.1.

The audit records are stored in a binary format but may be converted to ASCII by praudit(1M). We built a tool that converted the output from praudit into a format suitable for our visualisation tool.

3.1.2 tcpdump

`Tcpdump` is an open source packet capture program for systems running UNIX or Linux. It was used in the IDEVAL for the recording of network traffic offered to IDSs taking part in the experiment. A large number of different output formats are available; choosing the hexadecimal notation makes `tcpdump` include headers. We built a tool that parsed the whole IP packet, including TCP, UDP and ICMP headers, and converted the hexadecimal notation to a format accessible by our visualisation tool.

[1] By network-based IDSs we mean IDSs that analyse network traffic data, wherever the monitor is located. By host-based IDSs we mean IDSs that analyse data from applications or the operating system kernel.

3.1.3 Visualisation tool

The visualisation experiments were done using *Spotfire® DecisionSite*. This is commercial software that originated from Christopher Ahlberg's research on *dynamic queries* [10]. Spotfire offers a graphical interface in which the user is allowed to make choices as to what parts of the data should be visualised. It then composes the corresponding queries and updates the visualisation in real time. Combined with dynamic queries, Spotfire provides common and basic visualisation techniques such as two dimensional scatter plots where the size, shape and colour of markers may be adjusted as desired. Even though the tool offers many extra features, such as parallel coordinates, 3-D plots and certain statistical measures of the data being visualised, we have found that simple 2-D scatter plots suffice for our purposes.

3.2 Visualising operating system events

Three attacks are visualised in the following subsections, Mailbomb, eject, and ps [13]. Mailbomb is a denial-of-service attack, while the other two are successful buffer overflow attacks. In the first two visualisations the attacks as such are exposed, while after-the-attack activity is displayed in the attack on ps(1). We start by discussing some of the ways system activity may be visualised for the purpose of detecting malicious behaviour.

3.2.1 Data used for visualisation

Unauthorised transitions between different user IDs are most probably the result of malicious activity. Due to the existence of set-user-ID and set-group-ID program files, there is more to this than keeping track of changes in effective user IDs alone. Recall that if the set-user-ID bit is activated on a program file, the effective user ID is set to the owner of the file upon execution. As the effective user ID is used for determining file access permissions, and the owner of set-user-ID program files is typically root, these files are often targeted by intruders. The set-group-ID case works in a similar fashion.

To make it easier to follow these transitions, an additional user ID, called audit ID, is provided by the BSM system. As this ID is given to users at login and is inherited by all child processes started by the user's initial process, it serves to identify users regardless of what roles they assume during their login sessions, including successful calls to su(1M). This audit ID should make it easy to track transitions between different user IDs. In the end, it is a question of creating a visualisation that allows the operator to distinguish illicit from permissible root activity.

Information on user IDs and additional interesting information is recorded by the BSM audit system as a part of execution events, i.e. as a part of calls to `execve(2)`. A vast amount of security relevant information is included, should these events occur. This includes information on user IDs discussed above, the access path of the program file being executed, the name of the program executed, file permission settings on the file, user and group owner, the argument vector of `execve()`, the process ID of the calling process, return value and, of course, time of occurrence. Further documentation may be found in [15].

To expose the attacks on the host running BSM, we need only a fragment of all the information recorded by the BSM audit system. In the visualisations of the Mailbomb, eject and ps attacks we have used the following information:

- Audit ID
- Effective user ID
- Real user ID
- Process ID
- File access permissions
- Audit record size
- Invocations of su(1M)
- Invocations of execve(2)
- Time of audit record creation

The following sections show how these pieces of information are used to expose malicious activity by means of visualisation.

3.2.2 Visualisation of the Mailbomb attack

By plotting process ID versus time it is possible to detect certain abnormal system activity. Here, we chose this visualisation setting in order to expose the Mailbomb attack. Mailbomb is a denial-of-service attack in which the attacker sends a large number of messages to a mail server, overflowing the server's mail queue [13]. This attack is shown in Figure 1, a visualisation that contains 638,069 records in total. We plotted process ID on the vertical axis and time on the horizontal axis.

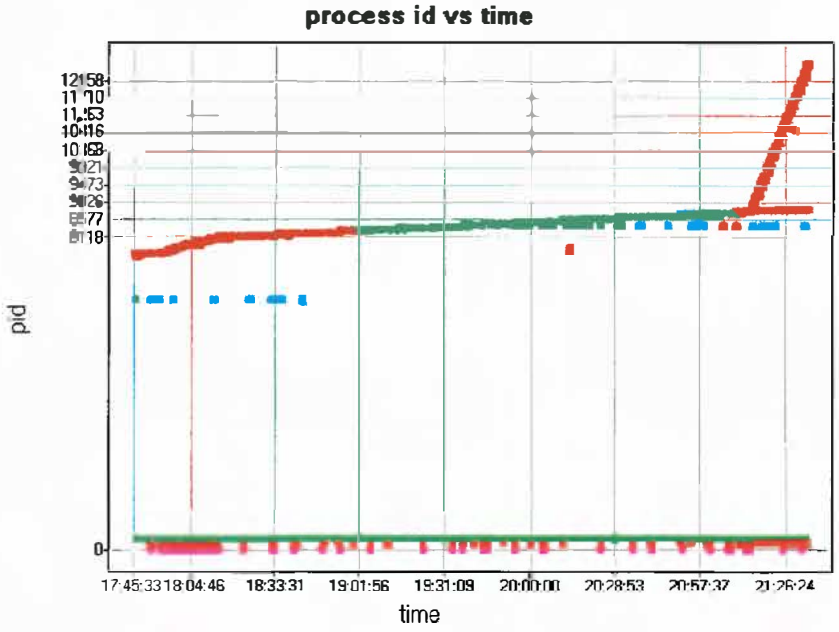


Figure 1: Visualisation of Mailbomb attack.

The steep stair in red in the upper right corner indicates that new processes are created at a very high rate. Regardless of what is causing such an activity, it must be investigated. In this case it was the trace of a denial-of-service attack, but plotting process ID versus time could reveal abnormal system activity in general.

We have coloured on `audit_id`, where network daemons have been given the colour red. It is possible to make this colouring since all network daemons share the same audit ID. Hence, the operator is informed that the excessive creation of new processes is the result of network activity, as opposed to a normal user. In the Mailbomb attack, it is Sendmail that tries to manage the 10,000 one-kilobyte messages sent to one single user. This illustrates the way colour is used in most visualisations; it adds an extra dimension and thereby provides an extra layer of information. It is the pattern, however, that uncovers the malicious activity.

3.2.3 Visualisation of the attack on eject(1)

The attack method that comes to mind when discussing abnormal arguments passed to `execve(2)` is *buffer overflow* attacks. By definition, the size of the arguments given in these attacks should deviate from what is normally the case. As the argument vector of `execve()` is present in the BSM audit records, and since we have information on the size of audit records, one approach to detect buffer overflow attacks would be to look at the size of audit records. This approach is taken in the visualisation shown in Figure 2 on the following page.

We plotted audit record size on the vertical axis and audit ID on the horizontal axis. We coloured in real user ID, using red for user root. In the visualisation of the eject attack, the largest record is 893 bytes, more than 270 bytes larger than the second largest record. Such a deviation in such a large record is abnormal and calls for attention.

When we looked at the marker at 893 bytes we discovered that it actually contained two records; both were successful buffer overflow attacks on `eject(1)`. Shown below are the arguments given to `execve()`, recorded in the BSM `exec_args` token, truncated for brevity:

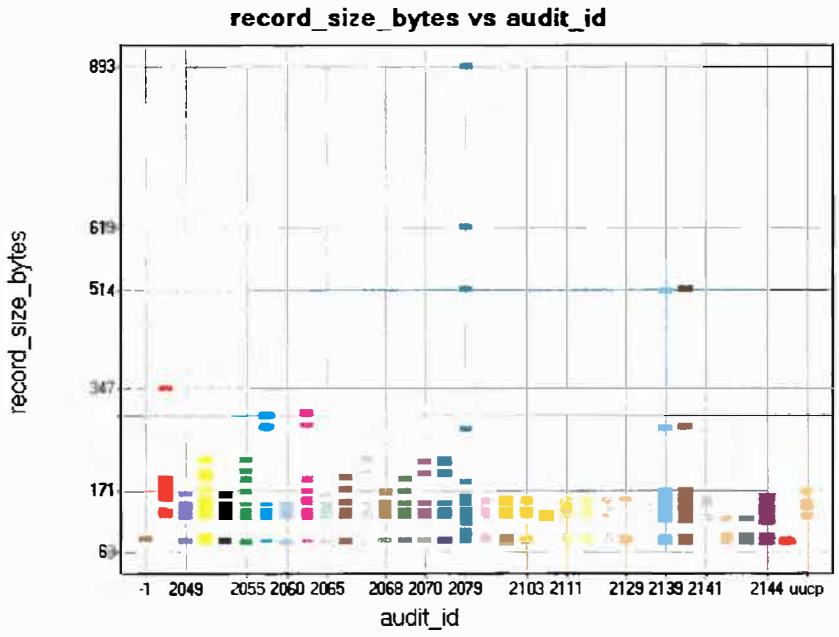


Figure 2: Visualisation of buffer overflow attack on eject(1).

eject, `^\@^S&^\@^S&^\@^S&^\@^S&^\@^S&^\@^S&...`

The presence of control characters is explained by the fact that the argument is originally given in hexadecimal notation and then translated to ASCII. Note that the figure also reveals root activity that is not caused by the execution of a set-user-ID program file. For instance, on the right end of the horizontal axis, we see the markers of 2144 and it is possible on the vertical axis to read the size of the audit records generated. Disregarding the record size, we see that some parts of the markers are red and others are blue. Since we have coloured in real user ID and user root has been assigned red, we are informed that user 2144 sometimes assumes real user ID root. As it turns out, this is explained by legitimate calls to `su`. In the same way as in the previous visualisation, it is the *pattern* that exposes the attack, and the colour is a means of additional information.

3.2.4 Visualisation of the attack on ps(1)

The buffer overflow attack on `ps` is exposed in the visualisation shown in Figure 3 on the next page. Our previous attempt to expose this type of attack by looking at audit record size proved to be successful. However, that visualisation has a rather limited attack coverage; it seems that buffer overflow attacks are the only type of attacks that are likely to appear in such a visualisation. We would like a visualisation in which general illicit root activity is displayed, be it by means of a buffer overflow or some other method. This is achieved in Figure 3. The following rules were used to create this picture:

1. Limit system events in the visualisation to `execve()` and `su`. In BSM parlance, `execve()` and `su` are the only event IDs present in the visualisation.
2. Allow only records where effective user ID is set to root. The purpose of the visualisation is to expose illegal transitions to effective user ID root.
3. Colour set-user-ID, using blue colour for records for which the set-user-ID or set-group-ID bit is activated and red for the complement of this set. Since only effective user ID root is present in the visualisation, this means that the red marker should not exist without there having been a preceding call to `su`. Blue markers simply tell us that the user executed a set-user-ID or set-group-ID program owned by root, such as `su`.
4. Rotate the markers by set-user-ID. Records in which the set-user-ID or set-group-ID bit is activated are shown orthogonally to records in which none of these bits is activated. This adds redundancy

to the visualisation as colour is used for the same purpose. The intention is to facilitate an understanding of the visualisation. This measure makes all red markers appear orthogonally to the blue ones.

5. Size the markers by event ID. Small triangles indicate that `execve()` has been called, while large triangles indicate that someone has executed the `su` program. This help us to distinguish `su` from other set-user-ID programs.
6. Use lines to connect markers sharing the same real user ID. A horizontal line between two or more markers means that the user shares his/hers real user ID with no one. A vertical line connecting two users means that these users share the same real user ID.
7. Filter out mail.local(1M) activity. This local mail delivery agent runs with a real user ID of a normal user and does not have the set-user-ID bit activated, but runs with effective user ID root. In addition, it is owned by user bin.

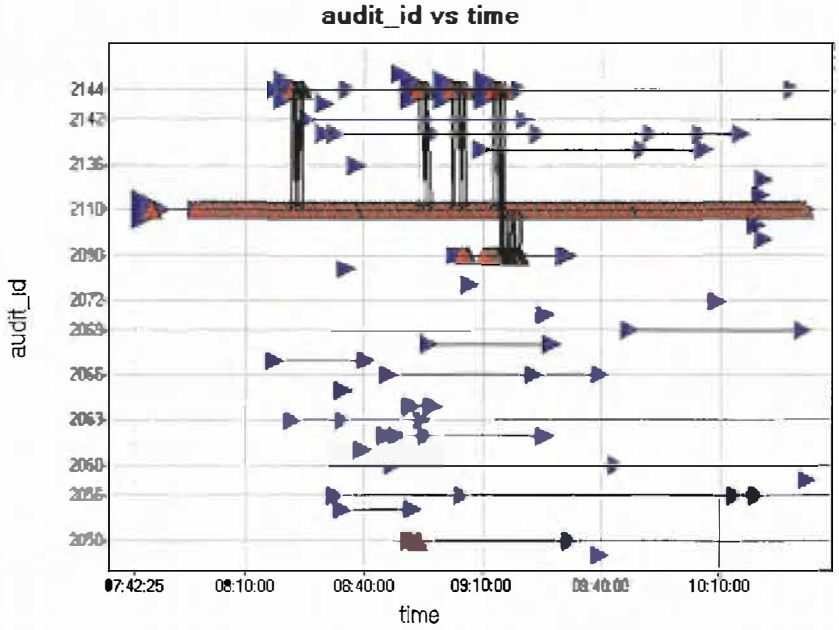


Figure 3: Visualisation of buffer overflow attacks on ps(1). Colour settings: **Blue**=set-user-ID or set-group-ID programs, **Large blue**=the su program, **Red**=non set-user-ID or set-group-ID programs. In addition, only effective user-ID root activity is present.

In the end we are left with Figure 3. Perhaps the first thing that catches the eye is the horizontal highway of red markers belonging to user 2110. This is a trace of the system administrator, whose activity begins by the execution of the su program at approximately 07:50. The density of red markers that follows indicates great activity. This is explained by the execution of the `ps_monitor` script, in which `date`, `ps -elf`, and `sleep 60` are executed in a loop. The red colour is explained by the fact that neither the set-user-ID nor the set-group-ID bit is activated on these program files.

Legal activity is also shown in the trace of 2144. This user is responsible for four more successful calls to `su`, shown in the top of the figure. That this user has succeeded in changing identity to root is revealed by the appearance of red markers and the fact that only effective user ID root activity is present in the visualisation. It is also revealed by the vertical lines that connect user 2144 with 2110. These lines indicate the these users share the same real user ID.

The malicious activity of user 2050 is exposed almost at the bottom of Figure 3, starting at approximately 08:50. The red marker shows that a non set-user-ID program is executed. As this event is not proceeded by a large blue marker, i.e. by a call to su, and since only effective user ID root activity is present in the visualisation, we conclude that user 2050 is launching a successful attack on the system. Table 1 on the following page shows the details.

The execution of non set-user-ID programs as effective ID root without a preceding call to `'su - root'` calls for attention. In Table 1, we see that the file permissions on `pwd(1)`, `id(1)` and

time	audit event	ruid	euid	file mode	exec_args
08:53:42	execve(2)	2050	root	4555	`ps,-z,-u,^p^p^p...`
08:53:42	execve(2)	2050	root	0555	`/bin/pwd`
08:53:42	execve(2)	2050	root	0555	`id`
08:53:44	execve(2)	2050	root	0555	`more,/etc/shadow`
...	...	...	...	...	...

Table 1: Information found in the audit records of user 2050 as the attack is launched. Real user ID is abbreviated as ruid and effective user ID as euid.

`more(1)` are all set to 0555, i.e. `-r-xr-xr-x`. Yet they are executed with effective user ID root. Only `ps` has the set-user-ID bit activated, which of course is a prerequisite for the attack to be successful.

Another successful buffer overflow attack is launched by user 2090 at approximately 09:00. This is revealed by the appearance of red markers without a proceeding large blue marker, thus indicating that the su program has not been executed. The density of markers tells us that this intruder is far more active than our buffer overflow friend 2050 discussed above. The vertical lines connecting user 2090 with user 2110 shows that these users share the same real user ID. As it turns out, user 2090 installs a rootkit and real user ID root is taken as part of this process.

Aside from the illicit transition to real user ID root, detection could quite easily be avoided if this visualisation method is used. Prior to the attack, the malicious user/intruder may copy all the program files to `/tmp` and change file permissions so that they become set-user-ID programs (not without certain implications, however). The red markers would then be replaced with benign blue ones. In order to detect this we could visualise on file owner instead of file permissions. Instead of colour in set-user-ID activity, we look at programs *not owned* by root but executed with effective user ID root. For instance, if root owns all set-user-ID program files and bin (or the intruder) owns `pwd`, `id` and `more`, we could give all files owned by root a unique colour. This would make it easy to distinguish illicit from benign root activity.

What might seem as yet another way for the intruder to avoid detection is to invoke `su` in close succession to the attack. This is made possible as the present visualisation does not reveal whether the user failed or succeeded in the call to `su`. However, a real world security site officer unaware the identity of the users who are allowed to assume the role of root is probably misplaced to begin with.

We have presumed that root is the almighty user whose precious identity the intruder seeks to take. All our reasoning is of course equally valid for whatever user ID the operator wishes to protect, e.g. apache, sendmail, ftp etc.

3.3 Visualising network events

From the IDEVAL we had tcpdump data from a sniffer placed on the outside interface of the router in the simulated network. These are the data that we have used to visualise the network traffic, including the attacks.

3.3.1 Approach

Our idea was that simple two-dimensional plots should be useful in the case of visualising network traffic. Consider normal network traffic and the attributes of each packet, which constitute a series of numbers and digits that form a specific pattern. These attributes are port numbers representing services, high source port numbers incrementing by one for each new session, IP addresses that are number series and have large parts in common if there are two computers in the same IP net, well-known port numbers for the most commonly used services and so on. Now consider a probe attack or a denial-of-service attack where the network traffic generated clearly breaks the normal pattern. A probe sweeps over many ports of IP addresses, a SYN flood generates packets with the whole range of source ports and in other denial-of-service attacks the attacker sends unusually large packets. The scatter plot below was created on the basis of this idea.

3.3.2 Visualising an Nmap FIN scan

This visualisation is a scatter plot of TCP packets. Time was assigned to the horizontal axis and destination port number to the vertical axis. Each packet is represented by a circular marker, and we chose to colour each marker by what TCP flags the packet carried. The reason for this is the idea that one should be able to see and follow a TCP session along the time line. Further, we have zoomed in on areas with a high density of packets. For example, the range of port numbers from 1 to 200 naturally has a high density of packets as the most common services listen to ports in this range.

Figure 4 shows what kind of traffic passes through the network and what kind of traffic that is dominant. Port 80, HTTP, has a well-filled blue line, meaning that we have a great deal of HTTP traffic on this network. Port 25, SMTP, and port 23, `telnet`, also have well-filled lines as a result of these services being heavily used on this network. We can also see some `ftp` traffic (port 20 and 21), some use of the `ident` service (113) and some use of `finger` (79).

What really breaks the normal pattern in this view is the presence of the malplaced FIN packets, which are coloured pink. On almost every port number between 1 and 200 there is a group of three TCP packets for which the FIN-flag is set. These packets are part of an Nmap FIN scan done against a computer on the internal network.

Many other attacks would also be visible in this visualisation. A SYN port scan done in a short time frame would generate a straight dark blue horizontal line. A denial-of-service attack called Neptune SYN flood would create slanting solid lines, also breaking the normal traffic pattern.

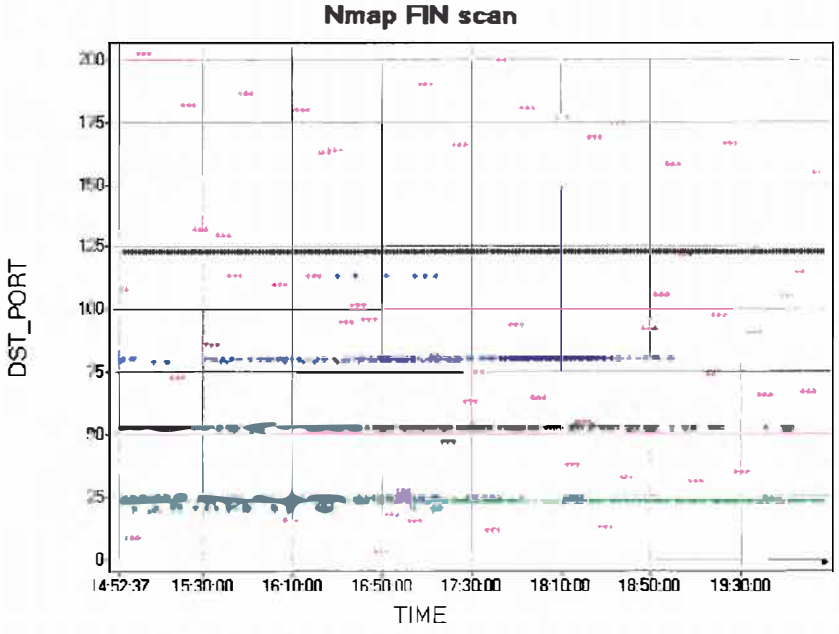


Figure 4: SYN = dark blue, SYN ACK = light blue, ACK = green, RST = red, FIN = pink, UDP packets = black

4 Discussion

The four examples here are visualisations of four different kinds of information. The first example visualises system activity in a way that lets the user see the rate at which new processes are created. Rapid growth often indicates some kind of denial-of-service attack, and this can easily be seen in the visualisation. The second visualises specific system calls and the size of these records. When someone gives a program an abnormally large argument in an attempt to overflow a buffer, this will break the normal pattern of records and their size. The third visualisation exposes illegal transitions between different kinds of user IDs. This visualisation truly gives the user a chance to spot intrusions. In this example the possibility to apply different shapes, colours and alignments to the markers on the basis of different record properties is very useful. We even use some visual attributes for the same thing, as it often

improves the quality of a visualisation to have some redundancy [1]. The fourth and last visualisation described in this paper is a simple but effective plot to visualise network traffic. It shows what kind of traffic passes through the network and visualises traffic that goes to services not regularly used, which makes them easy to see. There are of course many other ways to visualise network traffic, but that is left to future work.

All visualisations we use are two dimensional scatter plots. In one example we use the filtering functions provided by the visualisation tool in order to display only the data that is interesting in that case. Since this is done manually, advanced pre-processing is not done in any of the cases presented. What we do can be compared with anomaly detection. We create visualisations of data, and it is as anomalies in these visualisations that the intrusions become visible. This means that it will probably also be possible to visualise many other intrusions in a similar manner.

5 Future work and open issues

At present, the visualisation tool used in our experiments allows only for postmortem or off-line analysis. A real-time visual presentation of the large data sets given as input would require a huge amount of processing power. However, we do believe it would be possible to update the visualisation in intervals of a few minutes, possibly less than a minute. Such an updating frequency would probably suffice for continuous monitoring, and we would have a near real-time intrusion detection system.

The step from creating visualisations that expose malicious activity to a full-grown intrusion detection system based on visualisation techniques should not be underestimated. However, allowing the end-user to interact with the data and manipulate the visualisations dynamically seems a promising approach. Since the pre-processing is reduced to simple conversions of data from one format to another, this means that the classification of data is also left to the end-user. In this process the end-user is equipped with a tool that allows the creation of different views of the same data and manipulation of the visualisations as desired. In the end, we believe our approach addresses the complexity of intrusion detection in a way that increases attack coverage.

6 Conclusions

The paper gives a few examples that show that it is possible to visualise intrusions and attacks by using simple visualisation methods that make use of system event logs and captured network traffic. No advanced pre-processing seems to be necessary, although dynamic filtering can probably be used to widen the range of attacks that can be exposed. Thus, our experiments indicate that information visualisation has the potential of being a powerful tool for intrusion detection.

References

[1] Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman. *Readings in information visualization Using vision to think*. Morgan Kaufmann Publishers, Inc. San Francisco, California, USA, 1999.

[2] Luc Girardin and Dominique Brodbeck. A visual approach for monitoring logs. Technical report, UBS, Uiblab, 1998.

[3] Luc Girardin. An eye on network intruder-administrator shootouts. In *Proceedings of the Workshop on Intrusion Detection and Network Monitoring*, Santa Clara, California, USA, April 1999. USENIX.

[4] Robert F. Erbacher and Deborah Frincke. Visualization in detection of intrusions and misuse in large scale networks. In *Proccedings of the International Conference on Information Visualization*, pages 294–299, London, January 2000.

[5] Dr. Deborah Frincke. Hummer project, June 2001. *http://www.csds.uidaho.edu/ hummer/*.

[6] Robert F. Erbacher, Zhouxuan Teng, and Siddharth Pandit. Multi-node monitoring and intrusion detection. In *Proceedings of the IASTED International Conference on Visualization, Imaging, and Image Processing*, pages 720–725, Malaga, Spain, September 2002. IASTED.

[7] Robert F. Erbacher and Zhouxuan Teng. Analysis and application of node layout algorithms for intrusion detection. In *Proceedings of the SPIE '2003 Conference on Visualization and Data Analysis*, Santa Clara, California, USA, January 2003.

[8] Soon-Tee Teoh, Kwan-Liu Ma, and S. Felix Wu. Elisha: A visual-based anomaly detection system. In *Recent Advances in Intrusion Detection, Proceedings*, Zurich, Oktober 2002. RAID.

[9] Donald A. Norman. *Things that make us smart--Defending human attributes in the age of the machine*. AddisonWesley, 1993.

[10] Christopher Ahlberg. *Dynamic Queries*. PhD thesis, Chalmers Tekniska Högskola, 1996.

[11] Magnus Almgren and Ulf Lindqvist. Application-integrated data collection for security monitoring. In Wenke Lee, Ludovic Mé, and Andreas Wespi, editors, *Recent Advances in Intrusion Detection (RAID 2001)*, volume 2212 of *LNCS*, pages 22–36, Davis, California, October 10–12, 2001.

[12] Ulf Lindqvist and Philip A. Porras. eXpert-BSM: A Host-based intrusion detection solution for Sun Solaris. In *Proceedings of the 17th Annual Computer Security Applications Conference*, pages 240–251, New Orleans, Louisiana, USA, Dec 2001. IEEE Computer Society.

[13] Kristopher Kendall. A database of computer attack for the evaluation of intrusion detection systems. B.s. paper in computer science and engineering and m.s.e. paper in electrical engineering and computer science, Massachusetts Institute of Technology, June 1999.

[14] U.S. Department of Defense. *Trusted Computer System Evaluation Criteria*, December 1985. 5200.28-STD.

[15] Sun Microsystems, Inc, 901 San Antonio Road, Palo Alto, CA 94303, USA. *SunSHIELD Basic Security Module Guide*, February 2000. *http://docs.sun.com/db/doc/806-1789*.

Extending EMV to support Murabaha transactions

Mansour A. Al-Meaither* and Chris J. Mitchell

Information Security Group, Royal Holloway, University of London,
Egham, Surrey, TW20 0EX, United Kingdom
{M.Al-Meaither, C.Mitchell}@rhul.ac.uk

Abstract. Conventional credit card transactions are not consistent with Islamic principles, as exemplified by the Islamic banking system and the 'Murabaha sale'. On the other hand, EMV-compliant IC (Integrated Circuit) cards have been developed to secure traditional Point of Sale (POS) transactions. Thus, if Islamic principles are to be applied to card payments, a new and secure card payment process is required. In this paper, we propose a method for applying Islamic principles to card payments, where EMV IC cards are used to conduct card Murabaha transactions. After introducing the notion of Murabaha sale within the Islamic banking framework, we outline the EMV payment transaction process. Security requirements are then identified for a secure card Murabaha transaction. We then present a possible modification that allows an EMV card to conduct a Murabaha sale transaction. Finally, we analyse how the proposed scheme matches the identified security requirements.

Keywords:

Murabaha, EMV cards, electronic commerce, payment systems, security.

1 Introduction

A key concept in the Islamic economic system is the prohibition of payment and receipt of interest on deposits and loans. Instead, the sale of goods and the sharing of profits and losses among parties to any business transaction are encouraged. Islamic law puts many restrictions on contracts to attain maximal justice in a financial transaction, minimise the potential for legal disputes, and build a healthy and stable financial and economic system [1].

Modern banking systems were introduced into the Muslim countries in the late 19th century. Many Muslims confined their involvement with these banks to transaction activities such as current accounts and money transfers. Borrowing from banks was strictly avoided in order to avoid dealing in interest, which is prohibited in Islam.

* This author's work is supported by the Saudi Arabian Government

As a result, Islamic banks began to offer financial instruments consistent with Islamic religious beliefs.

Although it is difficult to obtain exact figures on the size of the Islamic financial sector, it is nevertheless experiencing strong growth. According to [7], the Islamic banks assets grew from $5 billion in 1985 to over $100 billion in the late nineties. While conventional banks guarantee the capital and rate of return, Islamic banks, working on the principle of profit and loss sharing, cannot, by definition, guarantee any fixed rate of return on deposits.

Meanwhile, the growing volume of fraud in credit/debit card transactions at the Point of Sale (POS) has led the card associations MasterCard and Visa to support their members in providing card-holders with a chip-based card. The advantages include combating fraud, validating both card and cardholder, and managing cards remotely. In order to allow a single merchant terminal to be used with cards issued by different card issuing banks and different brands, the EMV [2–5, 9] specifications have been developed, which define the physical and electrical characteristics of the IC card, the IC terminal specifications, and how the IC terminal communicates with the IC card. The security services supported by the EMV specifications include the following:

- Card authentication, where a terminal can be certain that a card is genuine,
- Risk management, where the card and the terminal independently decide which transactions need to be referred back to the card issuer at the time of the transaction,
- PIN verification on the card itself, and
- Transaction authorization by which a card issuer can be certain that a transaction has come from a specific and authentic card, as well as the card ensuring that the approval/decline response has been sent by the authentic issuer.

However, conventional credit card transactions are not consistent with Islamic principles, since they involve dealing in interest. Therefore, if Islamic principles are to be applied to card payments, a new and secure card payment process that is consistent with Islamic principles is required. Meeting this need by extending the EMV specifications to enable cards to conduct Murabaha transactions is potentially attractive for a number of reasons.

1. There will potentially be a significant reduction in Murabaha sale transaction expenses.
2. There will undoubtedly be a significant increase in Murabaha transactions, which will result in additional revenue stream for both merchants and issuers.
3. EMV is a good basis for inter-operability and global coverage.
4. Cardholders already have a trust relationship with their issuer.
5. Exploiting the existing EMV infrastructure provides a cost-effective solution.

In this paper we present a method of using EMV cards for secure card-based Murabaha transactions. After introducing the notion of Murabaha sale within the Islamic banking framework, we outline the existing EMV payment process.

Security requirements are then identified for a secure card-based Murabaha transaction. We then present a possible modification to EMV that allows the conduct of a Murabaha sale transaction, Finally, we analyse how the proposed protocol matches the identified security requirements.

2 Murabaha Sale

Murabaha sale is one of the most commonly used forms of financing provided by Islamic banks. Hasanin [6] notes that Murabaha is the mode of contract most frequently used in Islamic banking, in some cases accounting for 90% of all financing.

Murabaha is an Arabic term that means obtaining profit, and is a type of trust trading. Financially, it means cost plus profit sale, but in Islamic law it is a term that refers to a particular kind of sale [6].

A customer wishing to purchase goods requests the Islamic bank to purchase these items on his behalf and then sell them to him with a certain amount of profit agreed upon added to the initial cost. In the period up to the resale the bank has title to the goods, and hence a legal responsibility. The basic component of Murabaha is that the seller discloses the actual cost he has incurred in acquiring the goods, and then adds some profit thereon.

2.1 Rules Governing a Murabaha Sale

The validity of a Murabaha transaction depends on certain conditions, which should be properly observed to make the transaction acceptable in Islamic law. The rules that govern this principle, as stated in [6], are as follows.

- The two sale transactions making up a Murabaha payment, one through which the financial institution acquires the commodity and the other through which it sells it to the customer, should be separate and real transactions.
- The financial institution must own the commodity before it is sold to the customer.
- It is essential to the validity of the Murabaha that the customer is aware of the original price, including the costs necessary to obtain the commodity, and the profit. This is because Murabaha is a sale with a mark-up, and if the customer did not know the basic price then a violation of the Murabaha sale conditions has taken place.
- Both parties, i.e. the financial institution and the customer, have to agree on the profit for the financial institution from the sale, where the sum of the cost and profit is equal to the selling price charged by the financial institution.
- Murabaha is valid only where the exact cost of a commodity can be ascertained. If the exact cost cannot be ascertained, the commodity cannot be sold on a Murabaha basis.
- It is also necessary for the validity of Murabaha that the commodity is purchased from a third party. The purchase of the commodity from the

customer on a "buy back" agreement is not allowed in Islamic law. Murabaha based on a "buy back" agreement would be nothing more than an interest-based transaction.
- Cash is not permitted to be withdrawn on a Murabaha basis.

Unless these conditions are fully observed, a Murabaha transaction becomes invalid under Islamic law.

3 The EMV Transaction process

In this section, we give an overview of the EMV transaction flow, with a focus on the security mechanisms. A more detailed description of these mechanisms can be found in [3].

An EMV card payment transaction involves interactions between four parties: the cardholder, the merchant, the acquirer, and the issuer, with roles as follows.

- **Issuer**: A financial institution that issues a payment card to the cardholder.
- **Cardholder**: An authorised holder of an card issued by the issuer. The card stores the cardholder's payment data and is capable of generating authentication data and verifying a cardholder's PIN. During a transaction, a cardholder has a connection only to the merchant, which passes authorisation messages to the issuer (via the acquirer).
- **Merchant**: This is the business that accepts the card payment for the purchased goods. It uses a terminal to interact with the card. The terminal also interacts with the issuer (via the acquirer) to receive authorization for payment transactions.
- **Acquirer**: This is a financial institution that processes card payment authorizations and payments for the merchant. The acquirer and the issuer communicate through a secure financial network.

3.1 EMV transaction security

EMV transaction security is accomplished in two phases:

1. **Authentication.** Card authentication to the terminal is achieved using digital signatures. A chain of trust is established from the card scheme, which acts as the top-level certification authority (CA). Each terminal has a trusted copy of the CA's public key; the CA signs a certificate for the issuer public key and this certificate is stored on the card. The CA's public key and the issuer certificate are used to verify the authenticity of data stored in the card and messages sent by the card to the terminal during a transaction.
   Cardholder authentication is performed by PIN entry at the terminal. The PIN can be verified offline by the card, or online by the issuer. If supported, PIN encryption for offline PIN verification is performed by the terminal using an asymmetric encipherment mechanism [3]. The card may have a separate

key pair for PIN encryption or it may use the signature key pair. The card's public key is then used by the PIN pad to encrypt the PIN, and the private key is used by the card to decrypt and then verify the encrypted PIN.

2. **Transaction authorisation.** Transactions can be approved either offline by the card or online by the issuer. In both cases, symmetric cryptographic mechanisms are used to generate and verify Application Cryptograms (AC). The ACs exchanged by the issuer and the card are cryptographically secured using MACs (Message Authentication Codes). These are computed using a session key derived from a long term secret key shared between the card and the issuer.

The EMV Specifications allow both phases to be completed offline, without communicating with the issuer. However, the card or the terminal may force the transaction online, in which case an authorisation message is sent to the issuer for verification.

3.2 EMV transaction flow

The EMV transaction flow begins when the buyer card is inserted into the merchant terminal. The terminal reads data from the card for use in its risk management and to establish the card authenticity.

There are two types of card authentication, Static and Dynamic Data Authentication (SDA and DDA), where not all cards support DDA. For the card to support DDA it must have its own signature key pair and the means to generate signatures. In both cases the terminal uses a stored copy of the card brand public key to verify the issuer public key certificate; in DDA, the terminal also verifies an issuer-signed certificate for the card public key. In SDA, the terminal verifies the issuer's signature on critical card resident data so that unauthorised alteration of issuer data after personalisation is detected. In DDA, the terminal uses a public key based challenge response protocol to authenticate the card and verify the integrity of card resident data [3].

Next the Cardholder verification method is invoked to ensure that the person presenting the card is the one to whom the application in the card was issued. For this purpose EMV uses a secret PIN, where this PIN can be verified either offline by the card or online by the issuer. Upon successful cardholder verification, the terminal then decides whether the transaction should be approved offline, declined offline, or an online authorisation is necessary. Providing it does not reject the payment at this stage, the terminal passes the payment request to the card in the form of a GENERATE AC command. In response, the card performs 'action analysis'. Depending on the card risk management policy the card's action analysis can return one of three results [4].

1. A Transaction Certificate (TC), when the payment is approved offline.
2. An Authorisation ReQuest Cryptogram (ARQC), when either the card or the terminal want to go online so that the issuer can authorise or reject the transaction. The issuer then responds with an Authorisation ResPonse

Cryptogram (ARPC) which the card verifies and acts on. The terminal then issues a second GENERATE AC command that includes the issuer response and possibly a command script that the issuer may send. If the transaction is approved by the issuer, the card computes a transaction certificate (TC).
3. An Application Authentication Cryptogram (AAC), when the request is declined.

Finally, by returning either a TC or an AAC to either the first or second GENERATE AC command issued by the terminal, the card indicates its willingness to complete transaction processing. If the terminal decides to go online, completion shall be achieved when the second GENERATE AC command is issued.

4 Using EMV cards for a Murabaha transaction

The method proposed here for using EMV to support a Murabaha transaction involves the same participants and roles as the standard EMV payment process. However, using EMV for a Murabaha transaction requires extensions to the current security model and message flows. A key feature of a Murabaha transaction is that it is composed of two transactions, one between the merchant and the issuer, and the other between the cardholder and the issuer. Therefore, using an EMV card for a Murabaha transaction requires online communication with the issuer for every EMV Murabaha transaction that takes place. Moreover, the following assumptions are made.

1. The issuer has an agreement with the cardholder to sell him goods on a Murabaha basis. The issuer undertakes to purchase commodities as specified by a buyer, and then resell them on a Murabaha basis to the cardholder for the cost price plus a margin of profit agreed upon previously by the two parties. The issuer does not make a purchase unless the buyer requests it and makes a prior promise to purchase.
2. The EMV card is DDA capable.
3. Every acquirer participating in the scheme has their own signature key pair.
4. Every acquirer participating in the scheme has obtained a certificate for their public key from the brand CA, and this certificate is loaded into every merchant terminal supported by this acquirer.
5. Every issuer is equipped with a copy of the public key of the brand CA, as necessary to verify acquirer certificates.
6. Every terminal is equipped with its own public key certificate signed by its acquirer. [Alternatively the acquirer could append this certificate to every signed message sent from a merchant terminal to an issuer, as it passes through the acquirer network]. Additionally, it has the means to compute signatures, as well as a privacy-protected location in which to store its private key.
7. In a standard EMV transaction, terminal risk management is performed to protect the system against fraud. It provides positive issuer authorisation for

high-value transactions and ensures that transactions initiated from the card go online periodically to protect against threats that might be undetectable in an offline environment [4]. However, since this function is related to offline transactions, and the proposed extension requires the terminal to go online for every transaction, terminal risk management is not performed in the proposed extension.

4.1 Security requirements

No participant in any transaction want to suffer any loss. Therefore, we need to define precisely the security requirements to meet the needs of the transaction participants. We therefore next identify what security services are required for a secure card-based Murabaha transaction. The security services can be divided into four categories: authentication, confidentiality, integrity, and non-repudiation.

Authentication

Entity authentication provides assurance to one party regarding the identity of a second party involved in a protocol, and that the second has actually participated [8]. In the proposed extension to EMV, this security service can be sub-divided into the following:

1. Verification by the issuer that the merchant is as claimed.
2. The merchant needs to be sure that the payment card is genuine.
3. The merchant needs evidence that the cardholder is the legitimate owner of the payment card.
4. The issuer needs to be sure that the source of the payment instruction is a legitimate card.
5. No attacker can authorise a false EMV card-based Murabaha transaction on behalf of a cardholder.

Confidentiality

Confidentiality for information exchanged between the transaction participants is needed. The main reason for confidentiality is to prevent misuse of transaction data by unauthorised parties [8]. This security service can be sub-divided into the following:

1. The cardholder PIN must be kept secret from non-authorised parties.
2. The cardholder may require privacy of his order information.

Integrity

Integrity ensures that data is not altered in an unauthorised manner since it was created, transmitted, or stored by an authorised participant [8]. This security service can be sub-divided into the following.

1. The cardholder must be aware of the original price of the goods being purchased and the amount of profit the issuer is charging him before buying the goods. This is important for the transaction to be compatible with Murabaha sale conditions.
2. The buyer requires assurance that the issuer owns the goods being offered.
3. The cardholder payment authorisation must be protected against alteration, or any alteration must be detectable.

Non-repudiation

Non-repudiation prevents a participant from denying an action he has performed [8]. This security service can be sub-divided into the following.

1. The merchant must have evidence that the issuer has bought the goods and authorises him to sell the goods on his behalf to the cardholder.
2. The issuer must possess evidence that the cardholder has authorised payment for the goods on a Murabaha basis.

4.2 Interaction

We now describe the processes necessary to complete an EMV-based Murabaha transaction. In the description, $X||Y$ denotes the concatenation of data items X and Y and $S_X(M)$ is the signature of entity X on message M using the private signature key of entity X.

Figure 1 illustrates a transaction in which a cardholder uses his EMV card to purchase goods on a Murabaha basis from a merchant. It begins when the card is inserted into the merchant terminal (step 1). To authenticate the card, DDA is performed.

The terminal first issues the READ RECORD command (step 2) which returns the Primary Account Number (PAN), the CA identifier, the issuer public key certificate Cert_I, and the card public key certificates Cert_{IC}. In order to authenticate the card's public key in Cert_{IC}, the terminal verifies the issuer public key certificate Cert_I using its copy of the CA public key. The issuer signature on Cert_{IC} is then verified.

After successful verification of the card certificate Cert_{IC}, the terminal constructs the Purchase Information (PI), which contains a description of the goods, price and the date. Moreover, the terminal generates authentication data ($Data$) which contains a random number generated by the terminal, the current date and time, and the card PAN. The terminal then sends a challenge to the card using an INTERNAL AUTHENTICATE command containing $Data||PI$ (step 4). The EMV specification allows the INTERNAL AUTHENTICATE command to carry data of size up to 252 bytes [4], which is enough for the authentication data and the additional PI. Upon receipt of the message in step 4, the card computes the signature $S_{IC}(Data||PI)$ and sends it to the terminal (step 5). The card response (step 5) acts as a promise from the cardholder to buy the goods from the issuer on a Murabaha basis. Using the card certificate Cert_{IC}

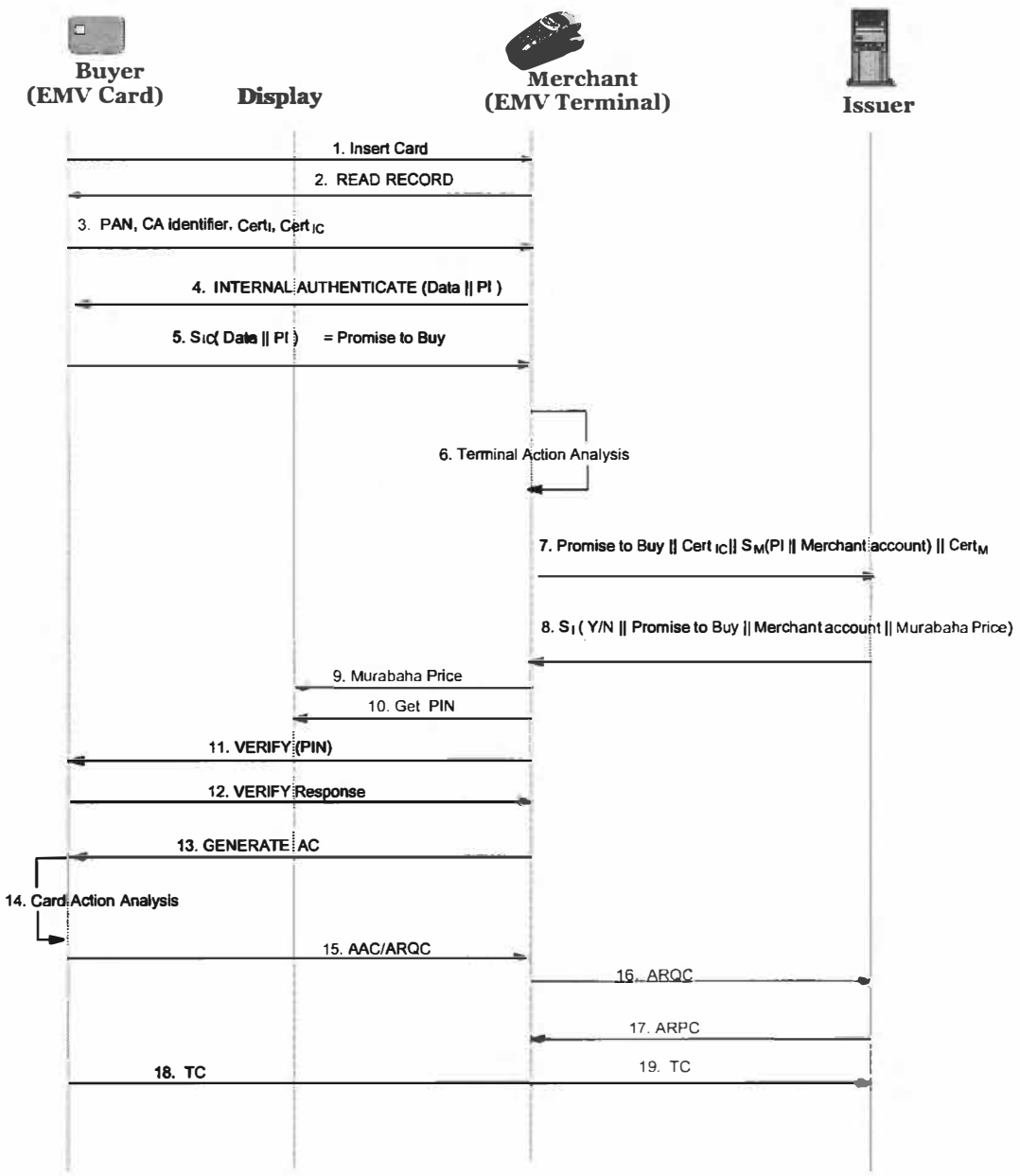


Fig. 1. EMV card-based Murabaha transaction

retrieved in step 3, the terminal verifies the signature $S_{IC}(Data||PI)$. Then, it checks that the same random number sent in step 5 is present in the signed data.

After successful verification of the signature received in step 5, the terminal action analysis is performed (step 6), where the decision is made as to whether the transaction should be declined offline, or continue online. If the decision is to reject the transaction, the terminal will issue a GENERATE AC (step 13) asking for an Application Authentication Cryptogram (AAC) from the card. If the outcome of the decision is to go online, the terminal constructs a message that contains the cardholder promise to buy the goods made in step 5 and the merchant signature over its bank details ('Merchant account') and PI. The terminal then sends this message (step 7) to the issuer along with its public key certificate Cert_M and the card public key certificate Cert_{IC}.

The information sent in step 7 notifies the issuer that a cardholder wishes to buy the goods, with the description given in PI, on a Murabaha basis. The issuer checks the cardholder promise to buy. If the issuer decides to proceed with the transaction, he first buys the goods from the merchant by crediting the goods price to the ('Merchant account') received in step 7. Then, the issuer constructs and sends a signed authorisation message to the terminal (step 8). This message contains the issuer decision as to whether to proceed or decline the transaction, and the price at which the goods will be sold to the cardholder ('Murabaha price'). In addition, this message authorises the merchant to sell and deliver the goods on behalf of the issuer to the cardholder. The signature in step 8 can be verified by the merchant terminal using the issuer's public key obtained during step 2.

After successful verification of the signature received in step 8, the terminal displays the 'Murabaha price' to the cardholder (step 9) and requests the buyer to enter his PIN (step 10). EMV allows the PIN to be verified offline by the card or online by the issuer. In the proposed scheme, both options remain valid. If the decision is to perform offline verification, the terminal sends a VERIFY command to the card (step 11). On receipt of the VERIFY command, the card returns a VERIFY response message (step 12), which indicates success or failure. In addition to ensuring that the person presenting the card is the person to whom it was issued, correct PIN entry by the cardholder is regarded as agreement by the cardholder to purchase the goods from the issuer on a Murabaha basis at the specified price (the 'Murabaha price').

After successful cardholder verification, the terminal sends a GENERATE AC command to the card (step 13). Next, the card action analysis process (step 14) begins, where a card performs its own risk management to protect against fraud or excessive credit risk [4]. Details of card risk management algorithms within the card are specific to the issuer. A card may decide to complete the transaction online or reject the transaction offline. If the outcome of the decision is to reject the transaction offline, an AAC is returned by the card to the terminal (step 15) and the transaction ends. If the outcome of the decision is to complete the transaction online, an ARQC is generated and sent by the card to the terminal (step 15) and then forwarded to the issuer (step 16). The issuer

responds to the ARQC with an ARPC (step 17). If the transaction is accepted, the card generates a TC (step 18) and sends it to the terminal which forwards it to the issuer (step 19) and the transaction ends.

In the proposed extension, most of the transaction procedures are similar to those in the standard EMV payment process. However, some messages have been modified and additional messages have been added to satisfy the Murabaha sale rules. For example, the INTERNAL AUTHENTICATE command (step 4) includes the Purchase Information (PI) in addition to the authentication data required by the EMV specifications. Therefore, the response signature (step 5) computed by the card must be computed on the PI in addition to the authentication data. Moreover, completely new messages (steps 7&8) have been added between the merchant and the issuer. These messages are necessary to complete the first transaction as specified in section 2.1, which is not part of the EMV standard.

Extending the EMV standard payment process requires that the terminal firmware be upgraded to allow the storage of a terminal-specific signature key pair. Additionally, the acquirer and the issuer transaction processing software must be modified to carry the new EMV messages.

An advantage of the proposed extension is that the existing EMV card needs not be changed to perform the extended transaction. The proposed changes affect only the merchant, the acquirer, and the issuer.

5 Security analysis

In this section, we examine to what extent the generic security requirements outlined in section 4.1 are met by the extended EMV transaction.

5.1 Authentication

1. *Verification by the issuer that the merchant is as claimed.* The "standard" EMV transaction does not provide mechanisms to authenticate the merchant terminal to the cardholder and the issuer [10]. However, in our extension, the merchant terminal has its own public key certificate Cert_M, which is sent along with the terminal signature in step 7 to the issuer. The issuer verifies the merchant certificate and that the merchant signature is valid; if this verification process fails then the transaction is not completed. Nevertheless, it is still possible for the cardholder to interact with a different merchant than intended.
2. *The merchant needs to be sure that the payment card is genuine.* This is performed using DDA. The merchant terminal verifies that the card certificate Cert_{IC} retrieved in step 3 is valid. In addition, the terminal verifies the validity of the card response (step 5).
3. *The merchant needs evidence that the cardholder is the legitimate owner of the payment card.* This is accomplished using PIN entry which can be verified either offline by the card or online by the issuer. An authentic cardholder will

enter the correct PIN. Moreover, the EMV Specifications limit the number of unsuccessful PIN entries [4].

4. *The issuer needs to be sure that the source of the payment instruction is a legitimate card.* The issuer can use the *Promise-To-Buy* and Cert_{IC} received in step **7** to verify the legitimacy of the card. Moreover, the ARQC (step 16) and TC (step 19) are generated using a key shared between the card and the issuer.
5. *No attacker can authorise a false EMV card-based Murabaha transaction on behalf of a cardholder.* The cardholder PIN is assumed to be a secret known only to the cardholder. Therefore, nobody but the cardholder can authorise an EMV card-based Murabaha transaction. This is based on the assumption that the merchant terminal displays the correct transaction data to the cardholder. This is a standard assumption for merchant terminals, where the cardholder is required to trust the reputation of the merchant when using a card in the merchant premises. In addition a fraudulent merchant is easy to track and prosecute.

5.2 Confidentiality

1. *The cardholder PIN must be kept secret from non-authorised parties.* If PIN verification is done using the card (offline), then the terminal can encrypt the cardholder PIN when sent from the PIN pad to the card. The card might use a separate key pair for PIN encryption or use its signature key pair. The card public key is used by the terminal (PIN pad) to encrypt the PIN, and the private key is used by the card to decrypt the encrypted PIN [3] in order to verify it.
2. *The cardholder may require privacy of his order information.* Order information is not encrypted and can be read by the terminal, the acquirer, and the issuer. Therefore, this requirement is not satisfied.

5.3 Integrity

1. *The cardholder must be aware of the original price of the goods being purchased and the amount of profit the issuer is charging him before buying the goods.* Since the cardholder chose the goods that are to be bought on a Murabaha basis from the merchant, we assume that he/she is aware of the original price of the goods. Moreover, in step 8 of the interaction, the terminal receives confirmation of the issuer willingness to sell the goods on a Murabaha basis with the 'Murabaha price' being the price at which the goods should be sold to the cardholder. This is followed by the terminal asking the cardholder for PIN entry. Entry of the correct PIN is taken as confirmation of the willingness of the cardholder to continue the transaction.
2. *The buyer requires assurance that the issuer owns the goods being offered.* By sending the message in step 8 to the terminal, the issuer provides an undeniable assurance to the terminal that it has purchased the goods. The buyer has to trust the merchant terminal to display the 'Murabaha price'

only if the message in step 8 is verified correctly and indicates that the issuer has credited the 'Merchant account'.

3. *The cardholder payment authorisation must be protected against alteration, or any alteration must be detectable.* This requirement is met, because MACs are used to protect the integrity of the AC generated by the card. The card and the issuer can verify MACs generated by each other using a shared key.

5.4 Non-repudiation

1. *The merchant must have evidence that the issuer has bought the goods.* The merchant can verify the message received in step 8 from the issuer. If it verifies successfully, then it provides a payment guarantee from the issuer, because it contains the issuer agreement to buy the goods, the goods description, and the merchant bank details ('Merchant account').
2. *The issuer must possess evidence that the cardholder has authorised payment for the goods on a Murabaha basis.* Entry of the correct PIN by the cardholder upon the display of the 'Murabaha price' will trigger the generation of an ARQC and a TC by the card. The TC sent to the issuer in step 19 can be regarded as evidence of cardholder authorisation; however, the TC is generated using a shared key with the issuer. Therefore it is of no value in providing non-repudiation unless combined with evidence from audit trails, e.g. held by the acquirer.

6 Conclusion

In this paper, we have proposed an extension to the EMV specifications to enable cards to conduct Murabaha transactions at POS terminals. We described the extension in detail, and explained how it meets the identified security requirements. In the proposed extension, most of the transaction procedures are similar to those in the standard EMV payment process. However, additional messages have been included to satisfy the Murabaha sale rules. The proposed extension can be seen as a step towards adapting electronic payment schemes to the Islamic economic system.

Finally, a future possible research area is to see how to conduct a Murabaha transaction using a mobile phone.

7 Acknowledgement

The authors thank Sami Al-Suwailem from the Center for Research and Development of Al-Rajhi Banking and Investment Corporation, for discussions that helped shape some of the ideas in this paper.

References

1. Mahmoud Amin El-Gamal. *A Basic Guide to Contemporary Islamic Banking and Finance.* Islamic Society of North America, Plainfield, IN, USA, 2000.
2. EMV. *EMV2000 Integrated Circuit Card Specification for Payment Systems Version 4.0 — Book 1: Application Independent IC Card to Terminal Interface Requirements.* EMVCo, 2000.
3. EMV. *EMV2000 Integrated Circuit Card Specification for Payment Systems Version 4.0 — Book 2: Security and Key Management.* EMVCo, 2000.
4. EMV. *EMV2000 Integrated Circuit Card Specification for Payment Systems Version 4.0 — Book 3: Application Specification.* EMVCo, 2000.
5. EMV. *EMV2000 Integrated Circuit Card Specification for Payment Systems Version 4.0 — Book 4: Cardholder, Attendant, and Acquirer Interface Requirements.* EMVCo, 2000.
6. Fayad Hasanin. *Murabaha Sale in Islamic Banks.* The International Institute of Islamic Thought, Herndon, VA, USA, 1996.
7. Zamir Iqbal and Abbas Mirakhor. Progress and challenges of Islamic banking. *Thunderbird International Business Review*, 41(4–5):381–405, 1999.
8. A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of applied cryptography.* CRC Press, Boca Raton, FL, USA, 1997.
9. Donal O'Mahony, Micheal Peirce, and Hitesh Tewari. *Electronic Payment Systems for E-Commerce.* Artech House, Norwood, MA, USA, 2001.
10. Mostafa Hashem Sherif. *Protocols for Secure Electronic Commerce.* CRC Press, Boca Raton, FL, USA, 2000.

Implementing elliptic curve cryptosystems using Hesse curves over prime fields

Terje Gjøsæter, Kjetil Haslum, and Trond Stølen Gustavsen

Agder University College
Department of Information and Communication Technology
Grooseveien 36, N-4876 Grimstad, Norway
{tgjosate,khaslum}@siving.hia.no,trond.gustavsen@hia.no

Abstract. This paper presents results of experiments comparing elliptic curve cryptosystems using Hesse curves with systems using traditional Weierstrass curves. Elliptic curves on Hesse form are believed to give better performance than the forms presently used in implementations. To perform these experiments, we have implemented point operations on curves in C++. The results of the experiments show that use of Hesse curves leads to 20-30% faster implementations than Weierstrass curves. The Hesse form makes elliptic curve cryptography even more interesting as an alternative to RSA in practical applications, and is well suited for constrained computing devices such as smart cards.

Keywords: Cryptography, public key cryptography, elliptic curve cryptography, implementation.

1 Introduction

Since Koblitz and Miller independently suggested use of elliptic curves in public key cryptography in 1985, increasingly effective implementations of elliptic curve cryptographic systems (ECC) has been developed. Today, these systems are as fast as systems based on integer factoring with same key length, see [1].

Because elliptic curve cryptosystems has the same security as RSA, but with shorter keys, they are more effective, and can often replace RSA. ECC is particularly suited for use in smart cards, cellular phones and other constrained computing devices. This has lead to an increased interest in ECC, and many important public key protocols has ECC counterparts.

Points on an elliptic curve constitute an additive abelian group, with addition as group operation.

Multiplication of a point on an elliptic curve with an integer, that is to add the point with itself several times, is the foundation for use of elliptic curves in cryptography. We use the notation

$$[n]\mathbf{P} = \underbrace{\mathbf{P} + \mathbf{P} + \cdots + \mathbf{P}}_{n}$$

where $\mathbf{P}$ is a point on an elliptic curve, and n is an integer. One simple and efficient algorithm for calculating $[n]\mathbf{P}$ uses a combination of point doublings

and point additions. The number of point doublings will always be equal to or higher than the number of point additions. By also using point subtractions, the number of point additions can be reduced. It is therefore most important that point doubling is fast.

All elliptic curves can be written in *long Weierstrass form*:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{1}$$

Under some conditions the equation of the elliptic curve can be simplified to *short Weierstrass form*:

$$E_{a,b} : y^2 = x^3 + ax + b \tag{2}$$

Under some other conditions they can be written in *Hesse form*:

$$E_D : x^3 + y^3 + 1 = Dxy \tag{3}$$

The choice of form influences the performance of the operations in cryptographic systems, because the different representations has different formulas for point multiplication.

Recent research indicates that curves in Hesse form has a number of properties that make them well suited for use in cryptography, not only because of increased performance, but also because it can be used to protect against some types of side channel attacks. We will focus on performance experiments comparing short Weierstrass form and Hesse form.

There are two main types of ECC implementations, over $\mathbb{F}_{2^r}$, and over $\mathbb{F}_p$.

In this paper we will describe the experiments we have performed with elliptic curves on Hesse form over finite fields $\mathbb{K} = \mathbb{F}_p$. A description of ECC using the Hesse form over $\mathbb{F}_{2^r}$, can be found in [2].

The rest of this paper is organized as follows. In Section 2 we review the group law for points on elliptic curves. Section 3 explains how Hesse form can be used to protect against side channel attacks. In Section 4 we describe the experiments we have performed, and present the results of these experiments.

2 The Group law

Points on an elliptic curve make an abelian group under point addition.

If $\mathbf{P}$ and $\mathbf{Q}$ are two points on an elliptic curve E, then there is a group law such that $\mathbf{P} + \mathbf{Q}$ is a well defined point on E.

2.1 Projective coordinates

In projective coordinates, two points (x_1, y_1, z_1) and (x_2, y_2, z_2) (where we may assume that $z_1, z_2 \neq 0$) are considered to be equivalent if there exists $\lambda \neq 0$ such that $\lambda(x_1, y_1, z_1) = (x_2, y_2, z_2)$. If (x, y, z) represents a point in projective coordinates, the cooresponding affine representation is obtained as $(x/z, y/z)$.

By using projective coordinates, it is possible to perform addition and doubling of points without division. Division over finite fields is usually much slower than multiplication, so it is usually advantageous to use projective coordinates, even if more additions and multiplications are needed than with affine[1] coordinates.

There are several kinds of projective coordinates. It is common to use Jacobian projective coordinates, see [3], for Weierstrass form, and normal projective coordinates for Hesse form.

2.2 Group law for points on Hesse curves

Implementation based on Weierstrass curves is well documented and tested, and is most commonly used in implementations of ECC today, see [4]. Recommended standard curves are Weierstrass curves. There are good guidelines and methods for choosing cryptographically strong curves. We have chosen not to go into details about use of Weierstrass form, since many good descriptions of this already exist, for example [3], [1] and [5].

Let $\mathbf{P} = (x_1, y_1, z_1)$ and $\mathbf{Q} = (x_2, y_2, z_2)$ be two different points in normal projective coordinates on an elliptic curve E_D, $\mathbf{P} + \mathbf{Q}$, $-\mathbf{P}$ and $[2]\mathbf{P}$ can then be expressed as follows:

$$(\mathbf{P} + \mathbf{Q}) = (y_1^2 x_2 z_2 - y_2^2 x_1 z_1, x_1^2 y_2 z_2 - x_2^2 y_1 z_1, z_1^2 x_2 y_2 - z_2^2 x_1 y_1) \quad (4)$$

$$-\mathbf{P} = (y_1, x_1, z_1) \quad (5)$$

$$[2]\mathbf{P} = (y_1(x_1^3 - z_1^3), x_1(z_1^3 - y_1^3), z_1(y_1^3 - x_1^3)) \quad (6)$$

A big advantage by using Hesse form is that the point operations can be performed concurrently, see [2].

Addition of two points $\mathbf{P}+\mathbf{Q} = \mathbf{R}$ where $\mathbf{P} = (x_1, y_1, z_1), \mathbf{Q} = (x_2, y_2, z_2), \mathbf{R} = (x_3, y_3, z_3)$ and $\mathbf{P}, \mathbf{Q}, \mathbf{R} \in E_D(\mathbb{K})$ can be executed concurrently in the following way:

$$\begin{array}{lll} \lambda_1 = y_1 x_2 & \lambda_2 = x_1 y_2 & \lambda_3 = x_1 z_2 \\ \lambda_4 = z_1 x_2 & \lambda_5 = z_1 y_2 & \lambda_6 = z_2 y_1 \\ s_1 = \lambda_1 \lambda_6 & s_2 = \lambda_2 \lambda_3 & s_3 = \lambda_5 \lambda_4 \\ t_1 = \lambda_2 \lambda_5 & t_2 = \lambda_1 \lambda_4 & t_3 = \lambda_6 \lambda_3 \\ x_3 = s_1 - t_1 & y_3 = s_2 - t_2 & z_3 = s_3 - t_3 \end{array} \quad (7)$$

The doubling formulas can be executed concurrently in a similar way, see [2].

Table 2 contains an overview of the number of basic operations needed to perform addition and doubling of points in Weierstrass form and Hesse form.

3 Side channel attacks

All point operations in Hesse form can be computed with only the formula for addition. This can to a certain degree protect against side channel attacks where

[1] By the affine plane we mean the set of all tuples $(x, y) \in \mathbb{K}^2$

Table 1. Timings for operations on field elements with lengths 160 bit and 240 bit.

Operation	Time(p_{160})	Time(p_{240})	Abbreviation
add(c,a,b)	0.46μs	0.51μs	A
subtract(c,a,b)	0.44μs	0.51μs	SU
negate(c,a)	0.52μs	0.57μs	
a.multiply_by_2()	0.47μs	0.52μs	M2
a.divide_by_2()	23.76μs	28.34μs	I2
multiply(c,a,b)	2.57μs	5.50μs	M
divide(c,a,b)	39.72μs	63.08μs	
invert(c,a)	33.39μs	53.64μs	I
square(c,a)	2.33μs	4.84μs	SQ
power(c,a,3)	11.46μs	16.80μs	-

the system is cracked with the help of side channel information (such as power, time, etc.) that can be used to calculate the number of additions and doublings that are performed in a multiplication. In some situations this can be used to find the secret integer that the point has been multiplied with.

In [6] is described how point doubling and point subtraction can be performed by swapping coordinates, and then use the formula for point addition. This technique is claimed by [6] to be at least 33% faster than other existing methods for protection against side channel attacks.

Let the point $\mathbf{P} = (x, y, z)$ be a point on a Hesse curve, then $[2]\mathbf{P}$ can be calculated by adding the points (z, x, y) and (y, z, x) with equation 4, see [6] page 6.

We have not used this method in our tests.

4 Implementation and experiments

We have implemented point operations on elliptic curves using C++, compiled with GCC, see [7]. We have also used the libraries Gnu MP, see [8], and LiDIA, see [9]. All results in this section are averages from 100 tests (with randomly chosen points).

We have timed the execution of some operations on elements of finite fields from LiDIA, the results from these tests are shown in Table 1. This is the foundation for considerations about time consumption of the formulas for addition and doubling.

The test of basic field operations are performed over the same finite field as the point operations, so the results from the different tests are comparable.

The tests are performed over two different finite fields $\mathbb{F}_p$ where p is a prime p_{160} of length 160 bit, and another prime p_{240} of length 240 bit, these are realistic sizes for use in cryptography.

Table 2. Basic operations in addition and doubling on an elliptic curve over a finite field. (Abbreviations as in Table 1.)

Op.		Basic operations
Add.	LiDIA affine	$2 \cdot M + SQ + I + 6 \cdot SU$
	LiDIA proj.	$12 \cdot M + 4 \cdot SQ + A + 7 \cdot SU + 2 \cdot M2 + I2$
	Weierstrass	$12 \cdot M + 4 \cdot SQ + 2 \cdot A + 5 \cdot SU + 6 \cdot M2$
	Hesse	$12 \cdot M + 3 \cdot SU$
Add. mix.	LiDIA proj.	$8 \cdot M + 3 \cdot SQ + A + 7 \cdot SU + M2 + I2$
	Weierstrass	$8 \cdot M + 3 \cdot SQ + A + 5 \cdot SU + 7 \cdot M2$
	Hesse	$10 \cdot M + 3 \cdot SU$
Dbl.	LiDIA affine	$3 \cdot M + 2 \cdot SQ + I + A + 3 \cdot SU + 2 \cdot M2$
	LiDIA proj.	$4 \cdot M + 6 \cdot SQ + 4 \cdot A + 3 \cdot SU + 6 \cdot M2$
	Weierstrass	$4 \cdot M + 6 \cdot SQ + 4 \cdot A + 3 \cdot SU + 6 \cdot M2$
	Hesse	$6 \cdot M + 3 \cdot SQ + 3 \cdot SU$

We have found two curves which we use in the tests. The first curve is specified by the following data:

$$\begin{aligned} p_{160} &= 1224753567915253525600877180059052116597297173971 \\ D &= 1550842421627942258257328785351007532033094402 42 \\ a &= 1808901272343108614406190635530977964674453038 76 \\ b &= 6387231065610304706782316703719324216503513898 55 \end{aligned}$$

The Hesse form is given by the specified D, and the corresponding[2] Weierstrass curve is given by a and b. The second curve is given similarly by:

$$\begin{aligned} p_{240} &= 169207162111028669914134189641167009619598713171362450223626077518140610 3 \\ D &= 702497238573896875692799960114136297227310413820769850347558251120978749 \\ a &= 431643474101790531809507705073497143389255228180223876860393494532849250 \\ b &= 993890749750054797374570702618347228585971779775823602477561598691887183. \end{aligned}$$

The order (number of points) of the curves are

$$\#E(\mathbb{K}) = 3 \cdot 40825118930508450853362583953995751896695610107 1$$

and

$$\#E(\mathbb{K}) = 3 \cdot 56402387370342889971378063213722336581827064017500807068379783198811271 1$$

respectively. For both curves the order $\#E(\mathbb{K})$ is a prime number multiplied by three. Note that the order of a Hesse curve is always divisible by three.

Table 2 shows the number of basic field operation used in point operations. *LiDIA affine* and *LiDIA proj.* refers to the implementation provided by LiDIA. *Weierstrass* and *Hesse* refers to our implementation.

LiDIA proj. and *Weierstrass* are both implementations of point operations using Jacobian projective coordinates, but we have done some improvements in

[2] The Hesse curve and the Weierstrass curve are birationally equivalent.

Table 3. Timings for addition and doubling on the elliptic curve over the finite fields with characteristics p_{160} and p_{240}

Op.		160 bit		240 bit	
		Measured	Estimated	Measured	Estimated
Add.	LiDIA affine	83.91μs	43.5μs	114.37μs	72.54μs
	LiDIA proj.	91.88μs	68.4μs	143.02μs	118.2μs
	Weierstrass	52.32μs	46.1μs	98.71μs	92.05μs
	Hesse	36.31μs	32.16μs	72.27μs	67.53μs
Add. mix.	LiDIA proj.	79.07μs	55.32μs	116.24μs	91.46μs
	Weierstrass	38.55μs	33.50μs	70.78μs	64.18μs
	Hesse	31.58μs	27.02μs	61.58μs	56.53μs
Dbl.	LiDIA affine	73.36μs	48.48μs	105.31μs	82.90μs
	LiDIA proj.	55.26μs	30.24μs	83.34μs	57.73μs
	Weierstrass	37.18μs	30.24μs	65.08μs	57.73μs
	Hesse	26.1μs	23.73μs	51.75μs	47.97μs

our implementation, one of the changes is to substitute one `divide_by_2()` by 5 `multiply_by_2()`, this has lead to increased performance.

If the z-coordinate of one of the points equals 1, the formula for addition can be simplified, this is called mixed coordinates. If the point to be multiplied is known in advance, all the doublings can be precalculated, and the points resulting from these precalculations can be normalized with $z = 1$. We have implemented this, and refer to it as *add. mix.* in the tables 2 and 3.

Table 3 shows the results from the experiments with point operations.

The estimated timings shown in Table 3 are based on the timings of the basic operations in Table 1. They can be considered as a lower bound, because we have only taken into account the field operations in the point operations, since they are the most time-consuming operations.

Table 3 shows a bigger difference between measured and estimated timings for LiDIA's point operations. We think there are three main reasons for this; LiDIA's operations contain local variable declarations in the point operation functions, more tests are performed, and they use function pointers to make the functions more general.

4.1 Point multiplication

Point multiplication can be performed by repeated doublings and additions, one doubling is needed for every bit in the representation of the number the point is to be multiplied with. If one uses subtraction in addition to doubling and addition, multiplication can be speeded up by using so called SD2 (Signed Digit base 2) representation. In SD2 representation, the number to be multiplied is represented by { -1,0,1} instead of only 0 and 1, such that the number of zeros is minimized. This leads to a reduced number of additions. SD2 is the algorithm used in our implementation of point multiplication.

Table 4. Timings for point multiplication on the elliptic curve over the finite field with characteristic p_{160}, and p_{240}

	160 bit		240 bit	
Multiplication	Measured	Estimated	Measured	Estimated
LiDIA affine	$16.35ms$	$16.14ms$	$34.81ms$	$34.32ms$
LiDIA Jacobian	$14.58ms$	$13.69ms$	$32.14ms$	$31.44ms$
Weierstrass Jacobian	$8.91ms$	$8.70ms$	$23.72ms$	$23.40ms$
Hesse	$6.24ms$	$6.09ms$	$18.39ms$	$18.17ms$

The important operation during encryption and decryption is point multiplication, $[k]\mathbf{P}$. In this calculation, k is often a random value of the same size as the order of the curve. In our tests, we have chosen k to be a random value between 0 and the number of points on the curve. The results are given in Table 4.

Our implementations of the two forms show that Hesse form is 30% faster than short Weierstrass form for 160 bit and 20% better for 240 bit. We belive that our results clearly indicate that cryptosystems using Hesse curves perform better than systems using Weierstrass curves.

The main reason for the decrease in the difference in execution time for 240 bit compared to 160 bit, is that the execution time for multiplication of elements in a finite field increases much faster than the other basic operations. Therefore the difference in number of multiplications becomes most important for the result.

Note that we in all our estimates count all basic field operations, and not only multiplications (and squarings) as commonly seen in the litterature. For 160 bit, an estimate based only on a count of multiplications (and squarings) would (see Table 2) in fact predict a smaller preformance gain (about 15%), compared to our results and estimates (about 30%).

5 Conclusion

Our experiments indicate that cryptosystems using Hesse curves perform better than systems using Weierstrass curves. Moreover, the performance gain is bigger than one would expect from a count of field multiplications as commonly seen in the literature. This is because the formulas for point operations for Hesse form are simpler than those for short Weierstrass form.

Based on this, we conclude that elliptic curve cryptosystems using the Hesse form should be considered as an interesting alternative, especially in constrained computing devices such as smart cards.

Finally we believe that our results show that the Hesse form can make elliptic curve cryptography even more interesting as an alternative to RSA in practical applications.

References

1. M. Rosing, *Implementing Elliptic Curve Cryptography.* Manning, 1999.
2. N. P. Smart, "The Hessian form of an elliptic curve," in *CHES 2001*, Koc, Naccache, and Paar, Eds. Springer-Verlag LNCS 2162, May 2001, pp. 118–125.
3. Blake, Seroussi, and Smart, *Elliptic Curves in Cryptography.* Cambridge university press, 1999.
4. *IEEE 1363-2000: Standard Specifications for Public Key Cryptography*, IEEE Std. [Online]. Available: http://grouper.ieee.org/groups/1363/
5. A. Miyaji, T. Ono, and H. Cohen, "Efficient elliptic curve exponentiaion," in *Advances in Cryptology-Proceedings of ICICS'97.* Springer-Verlag LNCS 1334, 1997, pp. 282–290.
6. M. Joye and J.-J. Quisquater, "Hessian elliptic curves and side-channel attacks," in *CHES 2001.* Springer-Verlag LNCS 2162, 2001, pp. 402–410.
7. GCC home page. [Online]. Available: http://gcc.gnu.org/
8. The Gnu MP home page. [Online]. Available: http://www.swox.com/gmp/
9. LiDIA-Group. (2001) LiDIA - A library for computational number theory. [Online]. Available: http://www.informatik.tu-darmstadt.de/TI/LiDIA/

Secure Storage for Mobile Terminals

Jani Suomalainen[1], Aarne Rantala[1], Markku Kylänpää[1],
Jarkko Tolvanen[2] and Janne Mäntylä[2]

[1]*VTT Technical Research Centre of Finland, P.O. Box 1203, FIN-02044 VTT, Finland*
[2]*Nokia Research Center, P.O. Box 407, FIN-00045 NOKIA GROUP, Finland*
E-mail: {jani.suomalainen,aarne.rantala,markku.kylanpaa}@vtt.fi
{jarkko.tolvanen,janne.mantyla}@nokia.com

Abstract. *Mobile terminals accompanied by open software platforms are increasingly used in legally binding applications, such as payments, votes and content playing, and, hence, should be secured against thefts, network attacks and malicious programs to achieve public acceptance for their use. Secure storage - a memory area accessible only for authorized entities - is needed to protect users', manufacturers', operators' and content providers' critical data. This paper provides a discussion on requirements and open challenges for secure storage solutions from mobile terminals' point of view as well as presents a classification of different techniques for securing mobile data. Furthermore, the paper presents a short survey of existing cryptographic, access control and tamper protected hardware solutions for secure storage.*

Keywords. secure storage, personal trusted device, mobile terminal

1 Introduction

Mobile terminals, such as smart phones and personal digital assistants (PDAs), open for external programs are becoming more and more common. Also, use of these devices in electronic commerce, voting, content playing and other applications handling critical information is gaining popularity. To ensure trustworthiness of these devices, mobile platforms should provide protection against thieves, crackers and malicious programs, attacking from various directions.

Secure storage, which is a memory area accessible only for authorized entities, can be used to protect data. Secure storage solutions can be roughly divided to cryptography-based mechanisms, such as per-file and per-filesystem encryption, and to access control mechanisms, based on operating systems, middleware platforms or tamper protected hardware elements.

However, when applying different securing mechanisms to mobile terminals, various restrictions are faced. These devices are portable, which limits their memory, processing and user interaction capabilities. Furthermore, they do not have fixed network connection, have limited power supply and, finally, should be low-priced.

This survey aims for gaining an understanding of different data securing strategies' applicability for mobile terminals. The paper presents requirements and surveys existing secure storage solutions from mobile terminals' point of view. First, Section 2 presents some use cases illustrating needs for securing mobile users' data. Section 3 lists different attacks and vulnerabilities threatening secure storage. Then, Section 4 describes requirements, characteristics and building blocks of secure storage and proposes alternative techniques for different usage and implementation strategies. Finally, Section 5 presents various academic and industrial solutions that could be used to implement secure storage and to defend against different threats and attacks.

2 Needs for Securing Mobile Data

2.1 Protecting Users' Interests

Mobile terminals have various needs for securing data. Use of personal trusted devices for managing personal information, for confidential communication, as well as for performing legally binding transactions requires users' critical data and credentials to be protected.

To achieve public acceptance and legitimacy of applications like payments, votes and electronic keys, users' digital credentials should be trusted. Essentially, to protect users from identity thefts and to prevent repudiations of transactions, credentials - data indicating identity and authority level - must be kept secret. For example, keys used to sign transaction documents or to identify house residents should be concealed in a way that they are kept secret and available only for the user.

In many cases users want to secure *confidentiality* of their data, when stored in user's own device, when backed up or when transmitted to trusted partners. Motivation may span from concealing business secrets to just protecting one's privacy. As in the case of identity, confidentiality is typically based on the use of private or secret keys. For instance, security protocols may use private keys to decrypt traffic. The user may well have several devices from which he wants to access essentially the same data. For instance, the user may want to access his calendar with smart phone, laptop or office workstation. Access to calendar and other private information might be required outside network coverage, too.

Additionally, users require *authenticity* and *integrity*, i.e. the ability to assure that the counterpart, e.g. a sender of a message, is not some other entity than what the user is led to believe and that no third-party has modified data. As in the case of confidentiality, public keys can be used to ensure the credentials of different entities. Therefore, even though these keys are not secret, they should be guarded or monitored to prevent them from being modified or destroyed or, at least, tampered without leaving a trace. Additionally, to ensure that a device functions as expected, the platform should be able to store integrity verification data, such as hashes of program codes and audit logs, in a trusted place.

2.2 Protecting Others' Interests

In certain cases, users themselves can be seen as adversaries who should be prevented from violating the rights of content, service and device providers. For example, files whose copyrights are owned by content providers impose some restrictions to users. users should be given rights to use or delete content files such as e-books and music, but not to make any unauthorized and uncontrolled copies. Also, some programs should be protected so that they are executable only during a limited evaluation period or after a customer has received a license key. A more detailed scenario could be the controlled sharing of files, where a library (e.g. an enhancement module to an existing program) can be downloaded and installed freely but is accessible only to particular programs. In conclusion, telecom operators and content providers require that their rights are guarded before they will start to trust and to utilise these devices.

2.3 An Example Application Requiring Data Security

Ticketing [1] is a typical application, which is specific for mobile environment and requires security solutions to protect both users and content providers interests. In ticketing, users are given access to services or goods provided by an issuer of tickets. Implementations of tickets may differ. However, typically a device must be able to store securely either a ticket - an electronic data object - itself or information used to prove users identity. Different kinds of tickets have differ-

ent requirements concerning applicable operations like modifying or copying, level of protection and handling throughput. Major subtypes can be characterized in the following way:

- *Event ticket* is used exactly once, e.g. when entering a theatre. Its value is significant and it is not modified after purchase but it is preferable that it can be transferred to another person.
- *Plane ticket* is also used exactly once. However, it is sometimes necessary to modify it when either a passenger changes plans or an airline company has to change its schedule. These kinds of tickets need not to be transferable and their value is very high.
- *Public transport ticket* often requires complicated processing because it may be time-limited, value-limited or use-count limited. Its value is usually relatively low but the volumes are often very high. In mass transit systems, the ticket verification speed is a critical requirement.
- *Coupon ticket* is often distributed because of promotional purposes, to make a new product or service known or get a customer to revisit, and so its value is low.

3 Attacks Against Mobile Data

Reasons for attacking against sensitive data in mobile devices are various. In criminal attacks, the goal is to acquire significant financial gain, e.g., by impersonating somebody, by stealing secrets, or by reselling pirated contents. Personal grudges could motivate attacks against target's professional or social reputation and, if successful, cause serious damage like loss of job, spouse or personal freedom. Some people might attack devices just for fun whereas others might be motivated by publicity.

In addition to deliberate attacks, there are also several kinds of unintentional incidents that may cause significant damage without appropriate precautions. Devices may get lost due to negligence, accidents or even natural disasters. Data within a device being repaired is usually accessible to servicing personnel. It can be presumed that even people without any criminal intentions could take advantage of such an opportunity to gain publicity and financial gain.

Attacks can be classified to three categories according to a method an attacker utilises to gain access to a device:

- *Remote attacks* are attacks through network interfaces against programs which are executed on a terminal. Although, mobile devices do not typically have many servers there may be peer-to-peer applications containing vulnerabilities. Since mobile devices are typically used in open networks and are out of external firewall protection, attack frequency may be high.
- *Local software attacks* are caused by malicious programs, which are executed on a terminal. The attacker has to implement attacking software and to find a way to get the user to install it. Currently, many mobile terminals are open for external applications and, therefore, downloaded third-party programs or e-mail attachments, containing Trojan horses or worms, cause a significant threat for device trustworthiness.
- *Physical attacks* are caused by a user possessing a terminal. A user may be a thief or the device owner trying to break e.g. copyrights. Since mobile terminals are small and portable they are more easily stolen, broken, or lost than desktop workstations and, thus, are more vulnerable for physical threats.

After an attacker has access to the device, some software or hardware vulnerability must be found to read protected data. The main attack types are described in the following list and their relation to user, access control mechanism and secured data is presented in Figure 1.

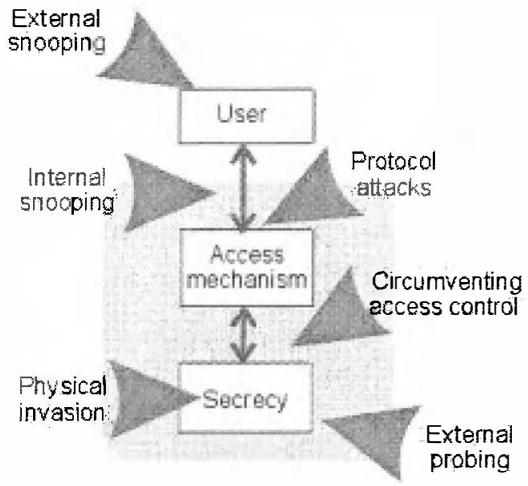


Figure 1. Attack types

- *External snooping* where operations needed to access secure storage are observed without invading the device, e.g. in its simplest form simply by looking over-the-shoulder or by scanning electromagnetic emissions from the device.
- *Internal snooping* where a piece of hostile executable code is inserted into the device that, e.g., monitors user actions (keystrokes, etc.) that are needed to access secure storage.
- *Protocol attacks* where design weakness of an access mechanism is mislead to carry on operations or to provide information required for access. Example of these attacks is dictionary attack i.e. trying heuristically different passwords to access secure storage.
- *Circumvention* where attacker accesses secure storage by bypassing protection mechanisms. For example, implementation weaknesses such as buffer overflows can be utilized to access secure storage.
- *External probing* where e.g. power consumption during secure storage accesses are monitored in order to speed up guessing the correct password.
- *Physical invasion* where the device is opened and its structure and operation are studied with external reader, microscope, logic analyser, etc.

4 Building Blocks of Secure Storage

Secure storage for mobile terminal can be composed of various technologies and mechanisms. Four main alternatives for secure storage media include local memory, tamper-protected hardware, removable media cards as well as remote network servers. These alternatives as well as main protection strategies - cryptography and access control - and techniques, to which the trust to protection is based on, are described in the following subsections. Figure 2 illustrates how protection strategies relate to storage media in a case where the trust relies on a fact that the device is untampered. The ovals indicate the mandatory mechanisms for protecting the different media. However, depending of the threat model also other combinations of mechanisms and media are possible.

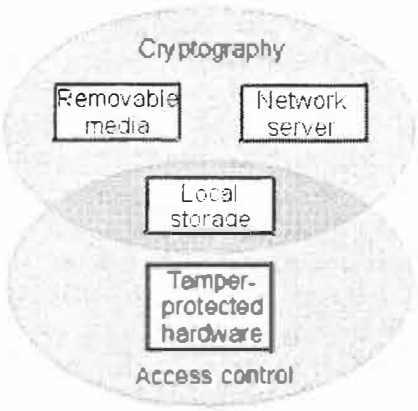


Figure 2. The relation of protection strategies to storage media

4.1 Storage Media

Tamper-protected hardware can be the most secured solution against situations where an attacker has access to a device. However, solutions where are all secure data is stored in special hardware may be expensive.

Local storage space provides more scalability to the solution. Because the device is usually in user's control, it is quite safe from many attacks requiring physical connection. Also, local memory used in mobile terminals is, typically, more resilient for physical attacks than disks in desktop workstations, since it is integrated to actual device. However, if the device falls under an attack, there are less protection mechanisms than in secure hardware solutions.

Removable media has basically the same characteristics as local storage. However, when removed from the device, it is out of access control protection of operating system and vulnerable. Also, small memory cards can more easily be stolen or lost.

Network storage provides large memory space that could be accessed with different terminals. A drawback of the use of network is that, as it is not in user's physical control, it can be easily attacked from different directions. Also, storage might be unavailable in some situations.

In practice, location where data is stored should depend of how critical it is. For instance, secret keys, which form the basis of data confidentiality of all data, could be stored in tamper-protected hardware. These keys can then be used to create large encrypted files, which can be stored in remote locations.

4.2 Access Control

Critical data should be stored in a way that information is accessible only for authenticated and authorized applications or users. Confidentiality can be between a device owner and a device or between a third-party and a device. In consequence, a device should support user and application recognition as well as discretionary and mandatory access control. A platform implementation should enforce that secure storage is available only through mechanisms, which are preventing unauthorized users and applications from accessing secured information.

Mobile terminals are typically single user devices and, hence, user identification is mainly needed to protect user against physical threats. A problem with user identification, based e.g. on use of personal identification numbers (PINs), passwords or biometrics, is that it may effectively defeat itself by being too tedious and awkward to use. Requesting passwords too often makes it easier to snoop them visually and makes people choose short passwords, and enforcing different PINs for every use just makes user to write them down. If passwords are weak i.e.

short or poorly chosen, they may be determined easily with these brute-force tactics. However, since input capabilities in current mobile devices are limited but good usability is highly important, it is problematic to demand the use of strong passwords.

Program code based authentication is needed to prevent malicious programs from accessing critical resources such as secure storage. For instance, programs trustworthiness can be concluded on a base of their origin and trusted programs can be given more privileges than programs downloaded from Internet. Program code based authentication enables also mandatory access control so that terminal providers and content providers are able to control access to certain information. For instance, a content file might be available only to a certain media player program.

Enforcing access control can be based on the use of access control lists (ACLs) i.e. listing entities, which are allowed to access particular resource, or capabilities i.e. listing resources, which a particular entity is allowed to access.

4.3 Tamper-Protected Hardware

An attacker possessing a device might be able to circulate software access control with physical means. Therefore, device packaging and interfaces should be such that it is not easy to monitor or disturb device operation and read out or modify contents of device memory.

Tamper-protected hardware solutions can be used to diminish attackers ways to circumvent protection mechanisms by limiting capabilities to interact with secured memory area. Although any hardware cannot be guaranteed to be completely tamperproof, compromising usually requires experience and time and, therefore, increases the costs of attacks. Furthermore, attacks involving physical tampering cannot be easily repeated and distributed. Also, if the protected secrecy is individual and unrelated to keys in other devices, compromised secret cannot be utilized against other devices.

When considering responses to attacks hardware can be categorized to following classes:

- *Tamper-resistant* solutions try to make the unauthorized access to hardware difficult, e.g. by employing exceptionally sturdy packaging methods.
- *Tamper-evident* solutions aim to leave a trace (e.g. damage in packaging) if somebody has tried to tamper with the hardware. Tamper-resistant packages are usually tamper-evident, too, but it is also possible to produce light and weak packages that are not really tamper-resistant, but are tamper-evident by being practically irreparable.
- *Tamper-responding* solutions try to render intrusions fruitless by detecting tampering attempts and destroying memory contents (and possibly other circuits, too) before the attacker is able to read or inspect it.

Hardware can be built-in to a device or it may work as a separate, often removable module communicating with the rest of the device. A removable module may be trusted by different entities than the terminal itself, e.g. a content owner may trust a removable module but not a terminal.

4.4 Cryptography

Protection, which is enforced with access control mechanisms verifying read and write requests, can be used to conceal data when attacker cannot access secure storage through any other route. However, it does not provide security to situations where the attacker can circumvent access control, for example, by accessing a memory card or a disk with another device. Therefore, when confidential information is kept in a non-access controlled area or when additional security is required, data should be encrypted.

Using cryptography to secure data is more processing intensive than protecting data with access control systems. Hence, more battery power and time will be consumed. Due to processing and battery limitations, large-scale encryption might be problematic in the mobile environment.

Safeguarding secret decryption keys for the entire lifetime of stored data is a major challenge for secure storage implementation. If the key is exposed, also the data is compromised. If the key is lost, all data ever encrypted with it becomes inaccessible. There are four main alternative strategies for securing the secret key:

- *User originated keys* (e.g. passwords) can be used as encryption keys or to secure encryption keys.
- *Software-based mechanism* can provide protection for certain applications. Software obfuscation (see e.g. [2] for an overview), i.e. hiding secret keys to software implementations and making these implementations hard to understand when reverse-engineered, can be used to make software attacks more expensive. Alternatively, keys protected with software, e.g. with access control, can be used to conceal secret keys, which are protecting data in removable or network media.
- *Tamper-protected hardware* can be used to physically secure secret keys.
- *Network-based techniques* such as those for key escrow, i.e. using one or several trusted third-party to safe keep secret keys [3], and function sharing, i.e. performing cryptographic operations in cooperation with other devices (see e.g. [4]), can be used in secure storage solutions when network connection is available.

4.5 Trusted Path

A trusted path from the user or from an application to secure storage is needed to ensure that critical data cannot be attacked when processed, input or displayed. In essence, all system components handling secure data should be kept trusted. Additionally, communication between these components should be trusted.

Different software mechanisms, both proactive (e.g. access control enforcement) and reactive (e.g. using cryptography for integrity checks), can be used to guard system integrity, which is required for trusted path. However, complete trust cannot be achieved with software-only solutions. One particular problem is the integrity of system startup process. Access enforcement is not in place before an operating system has been loaded and, therefore, if tamper-protection can be circumvented an attacker might be able to replace system components with own arbitrary code.

In open environments, protection mechanisms must be flexible and ubiquitous to enable execution of different types of applications but simultaneously protecting sensible system services. This typically leads to complex implementations where various components must be guarded in order to prevent attackers from circumventing protection. To minimize the complexity and potential vulnerabilities, only essential components should be kept trusted and allowed to access critical data.

4.6 Information Backup, Recovery and Access with Multiple Devices

Secured data should not be always tied to a single mobile terminal but available with other devices also. For instance, there should be recovery mechanisms in case some critical data is lost due to device failure. A related issue is a change of devices during which all data, including private keys, need to be migrated to the new device. Furthermore, the user might have several devices, which all should be capable to access same secured information.

A simple way to protect against the loss of encrypted information is to store backups of keys and data in a separate secure location. However, this is a potential security weakness, as it cre-

ates an exception to the rule that some data, like private keys, are never copied outside the device. Typically, encryption keys can be given for a trusted third-party for safekeeping.

5 Secure Storage Implementation Examples

Secure storage solutions can be classified according to a mechanism a protection is based on. In the following subsections, examples for cryptography, software platform and tamper-protected hardware-based secure storage solutions are presented. Cryptographic solutions are further divided to explicit data encryption as well as to automatic solutions, i.e. cryptographic file systems, and hardware solutions are divided to smart cards, tokens and high capacity systems.

5.1 Explicit Data Encryption

Encrypting information explicitly on requests can be viewed as the most primitive form of secure storage. This type encryption enables applications and users to control load encryption causes to a device, both in a form of processor load and power consumption.

Pretty good privacy (PGP) [5] can be used to protect local files as well as files sent over communications channels. Contents of the files to be protected are encrypted with some symmetric cryptosystem and a session key, which is encrypted with user's private key stored in user's private key file. The contents could also be protected against unauthorized modifications by just signing it. The encrypted files themselves are relatively secure if the PGP application has been used properly (long enough keys, etc.). The situation changes drastically if the attacker is able to acquire the private key file. The private key is protected with a passphrase but in practice it is usually much easier to find the passphrase (e.g. by a dictionary attack) than break the contents encryption or derive the private key from the public key. Therefore, PGP encrypted files residing in some removable storage module or copied over network are usually secure, but losing the device itself, with private keys, constitutes a much more serious security breach.

Trusted Computing Group (TCG) [6] promotes a more comprehensive solution for creating cryptography based confidential and integrity aware storage, which is based to hardware supported and integrity protected platform. They goal of the standardisation body is to provide definition and specification of a trusted computing platform, suitable for various hardware environments. TCPA secure storage is realized by using a hardware module as a portal to confidential data. All secured information is kept in encrypted files and can be stored in any storage media. The hardware module stores the secret key and performs cryptographic operations.

5.2 Cryptographic File Systems

Cryptographic file systems provide transparent secure storage access for applications and are usually significantly user-friendlier than the simple file encryption tools described in the previous subsection. They are also more likely to be utilized to protect all sensitive files as they protect everything by default - and without any additional work - whereas with simple file encryption the user must select every file to be protected. This means a very significant security boost, too, because forgetting to protect a sensitive file is a very common cause of security breaches.

Cryptographic File System (CFS) [7, 8] provides a location transparent way to implement secure storage. CFS can use any available file system (like remote file servers using Network File System mounts) as its underlying storage. No modifications to existing file systems are needed and also backups can be made in a normal way without sharing any key information. For the user CFS works transparently and encrypted files are just ordinary files for user's applications. *Smart Card Secured Cryptographic File System* (SC-CFS) [9] is an extension to CFS adding

support for smart cards to minimize user chosen password threat. CFS is using a single password to encrypt all files in secure directory. SC-CFS is using a separate key for every file.

Transparent Cryptographic File System (TCFS) [10, 11] is a cryptographic file system similar to CFS. TCFS has some advantages over CFS. TCFS allows groups of users to share encrypted data. It is also possible to have directories, which contain both encrypted and non-encrypted files. Because TCFS does not require users to explicitly attach directories as CFS, it is also more transparent.

5.3 Software Platforms

Secure operating systems and middleware platforms restricting users and applications access to system resources can be used to protect confidentiality and integrity of secure storage. For instance, most modern operating systems support discretionary access control, where users can set limitations to applications to use resources. Additionally, advanced security features supporting mandatory access control, where users itself are restricted to access certain resources, have been implemented.

Multi-level security (MLS), proposed, for instance, in Bell-LaPadula model [12, 13] provides mandatory access control. Mandatory access rules are typically set by administrators and used to limit information flow between users with different clearance levels. Multi-level security systems have traditionally been targeted for military and large organizations. For single user mobile terminals, limiting information flow between users is typically not a requirement.

Java Micro Edition (J2ME) is a middleware platform targeted for third-party application developers implementing software for mobile terminals. Basically J2ME provides security model where application's privileges are based on the application code identity. Therefore, it is suitable defence against malicious programs, such as viruses and Trojan horses. Additionally, if system integrity can be maintained, mandatory access control can be used to protect interest of other parties by giving special privileges to certain applications. J2ME Mobile Information Devices Profile (MIDP) [14] can be utilised to create secure storage against malicious MIDlets in order to protect users' interests. However, a problem with the J2ME security is that, if the underlying platform does not provide additional support making data in persistent storage available only for J2ME, an attacker can circumvent J2ME altogether and read protected data.

5.4 Smart Cards

Smart cards are very small and portable storage devices of relatively low capacity and some tamper resistance, like Subscriber Identity Modules (SIMs), Wireless Identity Modules (WIMs) or Java Cards. One motivation for them is that it is relatively easy to move them from one device to another, thus transporting e.g. tickets or user's credentials.

Packaging as well as physical, electrical and logical interfaces of smart cards have been standardized [15]. The simplest kinds of smart cards consist of memory only but the more developed ones have on-board processors. The interface to the outside world may utilize a contact pad with eight contacts, through which power is fed to the card and data is transferred using a serial interface.

Main advantages of a smart card with a processor, when compared to a 'passive' encrypted file, are that it can implement active defensive measures like counting unsuccessful attempts and locking up if a set limit is reached. Other advantages are that critical data like private keys can be stored in it, and used to protect hashes and session keys without transporting it outside the card, and that the host system has no way of destroying some data (like root public keys or root certificates) that the card is instructed to keep intact.

Although breaking smart card security from the outside is at least one degree of magnitude more difficult than breaching an ordinary operating system they are not impervious, either. For instance, by measuring its power consumption during cryptographic operations it may be possible to deduce the keys used. This kind of an attack requires physical possession of the targeted card, and it is possible to make such attacks more difficult with 'obfuscation' circuits and code, which add 'noise' to the above mentioned measurements [16]. The part of the application that resides on the card itself is quite secure but it depends on the host system integrity when interacting with the user, and this can be the weak link.

5.5 Security Tokens

Security tokens like USB hardware tokens and iButtons can be used as easily portable secure storage. Tokens are separate elements, which can be easily to connect to an actual device through a wireless link or a plug-in socket. They are more robust than smart cards and it is easier to use them with several devices.

USB hardware tokens, which are about a size of a housekey and can be easily carried in a keyring or similar, are today sometimes used as a secure storage for private keys etc. Examples of USB devices are Aladdin Knowledge System's eToken [17] R1 and Rainbow Technologies' iKey [18] 1000 and 2000. Security in these devices is elementary [19]: no obvious attempts at tamper-proofing is evident and it is possible to open the physical housing and gain access to the printed circuit board without any signs of forced entry. A determined attacker can thus retrieve the secret data by mechanical and electrical attack.

Dallas Semiconductor iButton [20] is one of a new breed of devices that can be used as a hardware tokens in the PKI arena. It is used by touching a reader with it for a short time. The iButton is housed in a water-proof, stainless steel metal housing, so it can withstand rough handling and extreme environmental conditions. Its physical size also means it is wearable unlike credit card-sized smart cards. For example it can be crafted into a ring. Java-powered cryptographic iButtons have a number of tamper-proofing features, which prevent the device from being physically attacked with invasive methods and prevent intruders from accessing critical data and obtaining private keys.

5.6 High Capacity Cryptographic Hardware

Secure MultiMediaCard (MMC) [21] is an example of a larger, higher capacity, and performance storage device that has security features similar to those of smart cards. Correspondingly, the weak point may be that the trusted path between the memory device and the user depends on the host machine in the same way as with smart cards.

The first implementations have been intended mainly to copyrighted contents delivery according to the 'Keitaide-Music' framework [22]. The principle of operation is such that the contents files themselves can be delivered freely because they are encrypted. In order to decrypt them an encrypted license and a genuine player that has a player key are needed. The player is usually a separate hardware module, which contains a public/private key pair. The license, containing a content decryption key, is generated by a license server when content is purchased, and it is encrypted with the public key of purchaser's MMC module (so it is not transferable). After that the Secure MMC will decrypt the license and then the contents with the content key, encrypt them with the player public key and give them to the player. The player then decrypts them with its private key and plays the contents. This last scheme is to prevent copying cleartext content within the device, between player and MMC. The license (that is tied to the memory module) will thereafter be stored in the module, and if it contains e.g. playback count limitation the contents will be decrypted only until the limit is reached.

Security coprocessors such as IBM 4758 [23] are used to perform the most security-critical subtasks like signing, checking signatures, generating, encrypting and decrypting session keys. It is a tamper-responding package providing hardware accelerators for speeding-up cryptographic operations; protected local storage for private keys, certificates and public keys as well as tamper-detection circuitry, which initiates local storage overwrite in a case of an attack. The hardware detects attempts to physically penetrate the processor package, temperature extremes, voltage variation and radiation, and in such cases zeroes internal memory without any software intervention.

6 Conclusions

In the paper, requirements and an overview of existing secure storage solutions were presented from mobile terminals point of view. For mobile phone, which is closed from external software, hardware based solutions provide sufficient way to secure data against most attacks. However, with current advanced mobile devices, which are open for third-party applications, a layered solution is required. Firstly, operating system level support is required for high-granularity access control for various applications. Secondly, cryptography is required for scalability and for protecting external data. Finally, hardware support is required to provide physical protection for secure storage.

References

1. MeT Ticketing Framework. Mobile Electronic Transactions initiative. *www.mobiletransaction.org*. February 2001.
2. C.S. Collberg, C. Thomborson. Watermarking, tamper-proofing, and obfuscation - tools for software protection. *IEEE Transactions on Software Engineering*. August 2002.
3. D.E. Denning, D.K., Branstad. Taxonomy for Key Escrow Encryption Systems. *Communications of the ACM*. Vol. 39. No. 3. March 1996
4. P. MacKenzie, M. K. Reiter. Networked Cryptographic Devices Resilient to Capture. *Proceedings of the IEEE Symposium on Security and Privacy*. 2001.
5. The International PGP Home Page. The PGPi project. *www.pgpi.org*.
6. Trusted Computing Group. *www.trustedcomputinggroup.org*.
7. Matt Blaze. A Cryptographic File System for Unix. *Proceedings of the ACM Conference on Communications and Computing Security*. November 1993.
8. Matt Blaze. Key Management in an Encrypting File System. *Proceedings of the USENIX Summer 1994 Technical Conference*. June 1994.
9. Naomaru Itoi. SC-CFS: Smartcard Secured Cryptographic File System. *CITI Technical Report 01-6*. January 2001.
10. Giuseppe Cattaneo, Luigi Catuogno, Aniello Del Sorbo, Pino Persiano. The Design and Implementation of a Transparent Cryptographic Filesystem for UNIX. *Proceedings of the USENIX Annual Technical Conference 2001*. June 2001.
11. Giuseppe Cattaneo, Luigi Catuogno, Aniello Del Sorbo, Pino Persiano. Transparent Cryptographic File System. *Linux EXPO 2001*. June 2001
12. D. E. Bell, L. J. LaPadula. Secure Computer Systems: Mathematical Foundations. *Technical report 2547*. Volume i. MITRE Corporation. 1973.
13. D. E. Bell, L. J. LaPadula. Secure Computer Systems: A Mathematical Model. *Technical report 2547*. Volume ii. MITRE Corporation. 1973.
14. Mobile Information Device Profile – Specification, version 2.0. Java Community. JSR-118. *jcp.org/aboutJava/communityprocess/review/jsr118/*. November 2002.

15. ISO/IEC 7816. Information technology -- Identification cards -- Integrated circuit(s). *International Organization for Standardization.*
16. Paul Kocher, Joshua Jaffe, Benjamin Jun. Differential Power Analysis: Leaking Secrets. *Proceedings of the Advances in Cryptology - Crypto'99.* Lecture Notes in Computer Science, vol. 1666, Springer-Verlag. 1999.
17. eToken Family of Products. Aladdin Knowledge Systems. *www.ealaddin.com/etoken/*
18. iKey Authentication Tokens. Rainbow Technologies. *www.rainbow.com/ikey/.*
19. Kingpin. Attacks on and Countermeasures for USB Hardware Token Devices. *Proceedings of the Fifth Nordic Workshop on Secure IT Systems Encouraging Co-operation.* October 2000
20. iButton -Java-Powered Cryptographic iButton. Dallas Semiconductor. *www.ibutton.com/ibuttons/java.html.*
21. PIN Secure MultiMediaCard with a User-authentication Function. *Hitachi Review Special Issue: Hitachi Technology 2002-2003.* Vol. 51. Pp. 38. August 2002.
22. Keitaide-Music Consortium. *www.keitaide-music.org.*
23. Joan G. Dyer, Et al.: Building the IBM 4758 Secure Coprocessor. *IEEE Computer Magazine.* October 2001.

SRBAC: A Spatial Role-Based Access Control Model for Mobile Systems

Frode Hansen and Vladimir Oleshchuk

Agder University College,
Department of Information and Communication Technology,
Grooseveien 36, 4876 Grimstad, Norway
{frode.hansen,vladimir.oleshchuk}@hia.no

Abstract. Role-based access control models are receiving increasing attention as a recent generalized approach to access control. In mobile computing environments (that offers location based services), availability of roles and permissions may depend on users location. To cope with the spatial requirements, we extend the existing RBAC model and propose a Spatial Role-based Access Control (SRBAC) model that utilize location information in security policy definitions.

Keywords: Role-Based Access Control, security policy, location information, mobile systems

1 Introduction

Role-Based Access Control (RBAC) models [1, 2] are receiving increasing attention as a recent generalized approach to access control. It differs from traditional identity based access control in that it takes advantage of the concept of role relations. In such models, the user's rights to access computer resources (objects) are determined by the user's membership to roles and by these roles' permissions to perform operations on objects. Thus, a role is a collection of permissions (or operations on a set of objects) determined by the system, based on the users organizational activities and responsibilities, as well as policies for an organization. Therefore, whenever a user has been properly authenticated by the system, this user may activate a subset of roles assigned to the user in order to accomplish his/hers tasks.

The advantages of the concept of roles are several. Firstly, it simplifies authorization administration because a security administrator needs only to revoke and assign the new appropriate role memberships if a user changes its job function. Furthermore, RBAC has shown to be policy neutral [3] and supports security policy objectives as least privilege and static and dynamic separation of duty constraints [2]. Moreover, RBAC offers flexibility with respect to different security policies and in fact [4] shows that RBAC can be configured to enforce mandatory and discretionary access control policies. Recent models [3, 5] extend

the RBAC model by specifying temporal constraints on roles that is associated with a user.

Because of the mentioned above, RBAC has been widely investigated. However, even though this great interest for RBAC as way of constraining users access to computer systems and the maturity of models, there are still issues not addressed by the existing RBAC models. One such requirement is related to that the system should be able to base its access decisions depending on the spatial dimension in which the user is situated. The reason for this is that mobile computing devices and wireless networks are increasingly being utilized by organizations. This enables users gaining access to networked computer resources, anywhere and anytime, through their mobile terminal. In organizations where access to resources are limited to a specific location, location-based services require means for obtaining the position of the requesting user in order to mediate the authorization request. Consider, for instance, the case of a doctor that has permission to access a patient's electronic patient record (EPR). However, due to the sensitive information that this EPR contains, the doctor is only authorized to access the EPR in designated areas. Thus, if the doctor request to access a particular patient's EPR from less trustworthy locations such as a hospital cafeteria or reception where there can be a considerable accumulation of people (doctors, nurses, patients, visitors, etc.); the doctor's access request is denied (more information on the application of location based RBAC in healthcare environments can be found in [6, 7]).

In order to cope with the spatial requirements, we propose a Spatial Role-based Access Control (SRBAC) model, an extension of the existing RBAC model proposed in [2], to be able to specify spatial constraints on enabling and disabling of roles. SRBAC support can be used to constrain the set of permissions available to roles that a user may activate at a given location.

Several solutions related to our work have been discussed in the literature. Spatial security policy for mobile agents and mechanisms to provide such policies are discussed in [8]. However, authors do not discuss how it can be used to extend RBAC. In [9, 10], authors extend the RBAC model by introducing the notion of environmental roles in order to control permission sets by (de)activation of roles based on spatial information. The main difference with our work is that in our solution the availability of permission sets depend on spatial information within the same active role. It reduces a number of roles we need to specify within the system and therefore simplify security administration.

The remainder of this paper is organized as follows. In Section 2, we describe the Role-Based Access Control model on which we based our model on together with the formalism used to present location information of a mobile terminal used in the authorization procedure. In Section 3 we present the formal model of SRBAC and finally, Section 4 concludes our work.

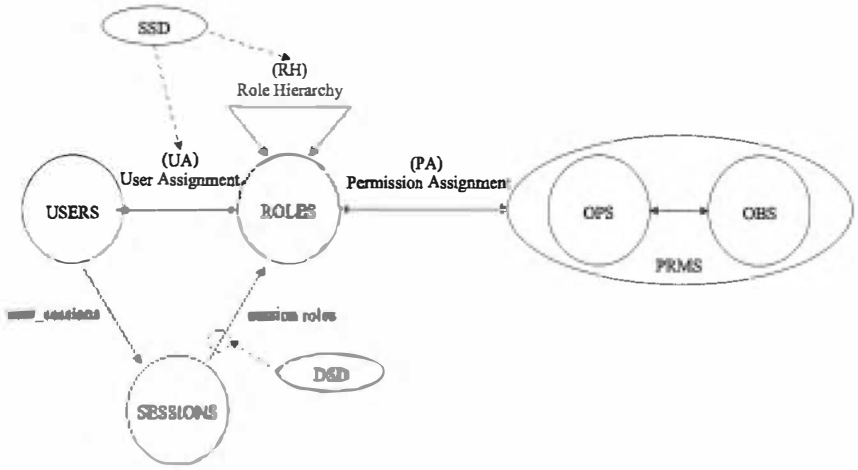


Fig. 1: NIST RBAC Model

2 Preliminaries

This section provides a short description of the standard for Role-Based Access Control (RBAC) proposed by National Institute of Standards and Technology (NIST) [2]. Our model, proposed in Section 3, extend this model by adding the spatial domain component such that authorization decisions can be made also with regards to location. To be able to accomplish this we provide a description of the formalism used to present spatial information about a mobile terminal, which can be used in our Spatial RBAC model.

2.1 RBAC Model

The NIST model, depicted in Figure 1, is defined through four different model components: Core RBAC, Hierarchical RBAC and Static- and Dynamic Separation of Duty Relations. Core RBAC is the base model (minimum requirement) for any RBAC system. Hierarchical RBAC adds the concept of role hierarchies where roles inherit permission from other roles. The two final model components add constraints to the model. The RBAC reference model element sets and relations are further explained below in this section.

Core RBAC Core RBAC encompasses the most essential aspects of Role-Based Access Control and consists of five basic data element sets: $USERS$, $ROLES$, $OBJECTS$ (OBS), $OPERATIONS$ (OPS) and $PERMISSIONS$ (PRMS). Here a user is defined as human beings, machines, networks and autonomous agents. A role is a job function within an organization and permissions are approvals to execute operations on one or more RBAC objects. In addition, the model include sets of $SESSIONS$ where each session is a mapping between a user ($user\_sessions$) and an activated subset of roles ($session\_roles$) that are assigned to the user.

In RBAC, several functions are defined that can be executed on the data element sets. The two relations, the User Assignment (UA) relation and the Permission Assignment (PA) relation, model the assignment of users to roles and the assignment of permissions to roles. Here a user can be assigned to many roles and a role can be assigned to several users. Similarly, a role may be granted several permissions and a permission may be assigned to many roles. Furthermore, the function *user_session* associates a session with a single user, and each user is associated with one or more sessions, where a session is a mapping of one user to one or more roles.

Hierarchical RBAC Hierarchies are natural means for structuring roles to reflect the organization's lines of authority and responsibility. In RBAC, role hierarchies define an inheritance relation between roles, denoted by $\succeq$. Such that if $r_i \succeq r_j, r_i, r_j \in ROLES$ then r_i inherits the permissions from role r_j. It is also possible to limit the scope of inheritance by forcing restrictions on the role hierarchy such that inheritance is limited to a single immediate descendant role, denoted $\succ\!\!\succ$. Thus if $r_i \succ\!\!\succ r_j$, the role r_i is a single immediate descendant of r_j, if $r_i \succeq r_j$, but no role in the role hierarchy is situated between r_i and r_j.

Constrained RBAC The RBAC model adds the notion of *separation of duty* [2]. A *Separation of Duty* relation is enforced on a set of operations that are mutually exclusive, i.e. no single user may execute all the operations within the set and no single user may be assigned to roles that are conflicting. Constraining the user's actions through the establishment and definition of roles, role hierarchies and role relations may contend this. RBAC accomplish this by enforcing *static separation of duties (SSD)* and *dynamic separation of duties (DSD)*. Through SSD constraints are placed on the assignment of users to roles and especially their ability to form User Assignment associations such that a user may not be authorized for two roles that are mutually exclusive. In addition, it is possible to apply SSD relations in the presence of a role hierarchy. Here, the SSD limitations are inherited such that if a role r_i inherits role r_j and role r_j has an SSD relation with role r_k, this would imply that role r_i also has an SSD relation with role r_k, where $r_i, r_j, r_k \in ROLES$.

Furthermore, introducing DSD relations constrain the permission sets that are available to a user. Contrary to SSD, where constraints are placed on a user's entire permission space, DSD relations restrict the availability of the roles that can be performed simultaneously within a user's session. Therefore DSD allows a user to be authorized to mutually exclusive roles, but these may not be active simultaneously (i.e. DSD relations define constraints on roles that can be activated within a user's session).

2.2 Location Model

For the system to be able to make authorization decisions based on the spatial dimension in which the user is situated, the mediator must be able to obtain

the location of the mobile terminal in which the access request was made from. There exist several location-sensing techniques that vary in granularity for both indoor and outdoor position estimation of mobile terminals. The GPS (Global Positioning System) system is a well-known technique for location sensing and can be used to estimate the location of mobile terminals. GPS emits coded radio signals that can be processed in a mobile terminal (with a GPS receiver) to determine its position, time and velocity. The GPS provides high position accuracy and the GPS radio signals can be utilized to compute positions in three-dimensional space. However, since this technique requires line-of-sight of the mobile terminal, it works properly only for outdoor determination of the position of a mobile terminal [11].

For indoor location tracking of mobile terminals one may use location sensing systems such as the Active Badge [12] location system. This system was the first indoor location system developed for use in an office computing environment, and use infrared (IR) technology to keep track of active badges worn by employees. Another solution is to use the 3D-iD system from Pinpoint that use radio frequency (RF) signals and an array of antennas placed at known positions to be able to track a mobile terminal [13]. The Cricket location-support system [14] makes use of both ultra sound and RF signals for the mobile terminals to "*learn their physical location by using listeners that hear and analyze information from beacons spread throughout the building*" [14].

For wireless networks one would have to incorporate more than one of these techniques for location estimation of mobile terminals. The type of location estimation technique used depends on the requirement of accuracy to the mobile terminal' position, which is required by the system in the authorization process. For example, for a user requesting access to a secure service limited to a specific room in a building, may require fine granularity in order to ensure that the user does not try to access the service from the room next door. Therefore, a location system must be able to cope with several location spaces [15]: radio field or infrared cells, access point addresses and geographical coordinates. In addition, due to the diversity in location-spaces, the location information must be represented in a universal and flexible way, such that it can yield for a *ubiquitous computing environment* [15].

In addition to obtaining the location information of the mobile terminal, the system must also be able identify the authenticity of the spatial information obtained. The service that provides the location information used in the authorization process must be able to provide secure and trusted data. This is particularly important for a service which requires precise accuracy of the mobile terminal in order to prevent disclosure of classified information. However, this is beyond the scope of this paper, we assume that the system can identify and verify location of any legitimate user based on a *trusted* underlying network architecture.

In our access control model, in order to ensure this viability, locations are represented by means of symbolic formalism that defines locations as location expressions which describes location areas identifiable by the system.

3 SRBAC Model

As explained earlier, in traditional RBAC, users are assigned to roles and permissions are associated with roles, such that a user may activate permissions dependent on their role assignments. Incorporating traditional RBAC in a mobile environment, one would have to define roles for each location in an organizational domain. Therefore, in organizations where the location domains are many, due to location specific services, roles defined in the system becomes considerable. Moreover, roles defined for these locations may have a lot of the same permissions assigned to them. Thus, in a mobile setting, we can achieve more flexibility defining the security policy when permissions are assigned dynamically to a role limited by the location in which a user is situated. Therefore, a role is dynamic in the sense that it may have different permissions assigned to it for two distinct locations. For example, in a bank a customer is assigned to the role *customer_role*, and has permission to open his/hers safety-deposit box (and of course other permissions related to the nature of such a role). However, the permission to open the safety-deposit box is restricted by the position of the customer, i.e. the system should only grant permission to open the box only when the customer is nearby the safety-deposit box. Thus, the customer may activate his/hers role of a customer when entering the bank, but may not activate this particular permission until entering the strongroom, where the safety-deposit-box is located. If the Figure 2 shows the wireless environment of the bank subdivided

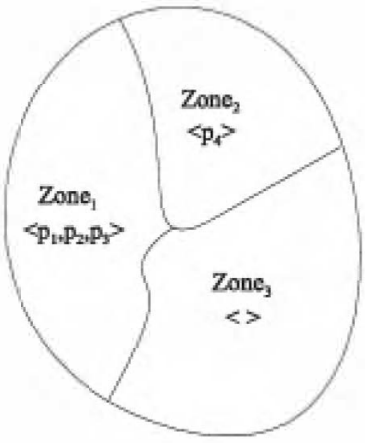


Fig. 2: Logical location domains with available permissions.

into three different zones and $\mathtt{Zone}_2$ is the strongroom location.

For the *customer_role*, permissions associated with it, varies with location such that a user assigned to it has permissions $\mathtt{p}_1, \mathtt{p}_2, \mathtt{p}_3$ in $\mathtt{Zone}_1$, only permission $\mathtt{p}_4$ in $\mathtt{Zone}_2$ and no permissions in $\mathtt{Zone}_3$ (denoted $\varnothing$). For example, permission p_4 can indicate that a user assigned to the *customer_role* may open his/hers

Table 1: Location Permission Assignment List (LPAL) where the *customer_role* has different permissions for distinct locations

ROLES	LOC	PERMISSIONS
customer_role	$Zone_1$	p_1, p_2, p_3
customer_role	$Zone_2$	p_4
customer_role	$Zone_3$	$\varnothing$

safety-deposit box only when located in $\mathbf{Zone_2}$. The $\varnothing$ implies that there are no permissions associated with *customer_role* in $Zone_3$, indicating that users associated with the *customer_role* may not access any of the services offered in this particular zone. For a system, the permissions assigned to *customer_role* with regards to locations, can be listed in a *Location Permission Assignment List* (LPAL) as shown in Table 1.

In the remainder of this section we introduce the formal model components of the Spatial Role-Based Access Control (SRBAC) model.

3.1 Core SRBAC

We extend the existing RBAC model [2] to be able to utilize location information in security policy definitions. The SRBAC model consists of the following five basic components: sets $Users$, $Roles$, $Permissions$ ($PRMS$), $Sessions$ and $Locations$ (LOC), representing the set of users, roles, permissions, sessions, and spatial locations respectively. $Users$ are considered to be mobile units that can establish (wireless) communication with system resources to perform some activities. *Roles* are described as a set of permissions to access system resources (objects). *Permissions* are approvals to execute some operation on one or more RBAC objects, and depend on the role and role owner location. *Locations* are represented by means of symbolic expressions called location expressions that describe location domains identifiable by the systems. We assume that wireless network can identify and verify location of any legitimate user based on underlying network architecture (as discussed in Subsection 2.2).

We assume that areas defined in LOC cover the whole responsibility domain Z of SRBAC. The domain Z is divided on the physical layer into subareas, called *primary location cells* denoted as π_i, $i = 1, .., k$, which reflect the ability of the underlying architecture to uniquely map user location into cells. We assume that underlying infrastructure is unable to distinguish different locations inside π_i for any $i = 1, .., k$. However, using primary location cells in SRBAC can be unpractical because primary location cells represent infrastructure of location detection system but we need the structure location domains reflecting organizational infrastructure. Therefore we introduce *logical location domains* that reflect organizational location infrastructure and organizational security policy. For example, within a University we can define logical location domains representing locations such as departments, laboratories and even individual offices. They can be defined as composition of primary cells.

For example, allocation of ICT department can be described by location expression `ICT_dom=`$[\pi_1, \pi_3]$ as area covered by primary location cells π_1 and π_3. Similarly, `LIB_dom=`$[\pi_2, \pi_4, \pi_5]$ defines library location area. Assuming that `CS_dom`, `EE_dom` and `IS_dom` are logical location domains for departments of Computer Science, Electrical Engineering and Information Science, respectively, we can define domain for School of Computing `CSchool_dom` as composition of all its departments in the form of location expression, i.e., `CSchool_dom=ICT_dom +CS_dom+EE_dom+IS_dom`. The example demonstrates the idea of using location expressions to define new domains. Since logical location domains can be seen as sets we define new location domains by using domain operations that are similar to operations used in set theory, i.e., union (denoted as '+'), intersection (denoted as '×'), difference (denoted as '−') and complementation (denoted as '¬' or '*outside*'), etc.

Generally, the same position can belong to different logical location domains. In order to simplify definitions and implementations, it is desirable to identify a least set of locations that can be used in location expressions to define all meaningful location domains in SRBAC.

A location l from LOC is called *homogeneous* with respect to role r from *Roles* if r has the same permissions available in any position inside l. Location l from LOC is called *homogeneous* (with respect to *Roles*), if it is homogeneous with respect all r from *Roles*.

Definition 1. *Set of locations* $L = \{l_1, l_2, ..., l_k\}$ *from* LOC *are called* normalized *with respect to set of roles* R *from Roles if it is*

- *a partition of* LOC, *that is,* $LOC = \bigcup_{i=1}^{k} l_i$ *and* $l_i \cap l_j = \varnothing$ *for* $i \neq j$, *and*
- *any location* l_i *from* LOC *is homogeneous with respect to* R.

It is easy to see that any meaningful location expression can be presented as a subset of normalized LOC. From now we assume that LOC is a normalized set of locations (with respect all roles from *Roles*) that is a partition of the entire domain area controlled by SRBAC.

On the sets $Users$, *Roles*, *Permissions* ($PRMS$), *Sessions* and *Locations* (LOC) several functions are defined. The user assignment relation UA, represents the assignment of a user from $Users$ to roles from *Roles*. The permission assignment relation PA, represents the assignment of permissions to roles based on location. We model user assignments to sessions by function $user\_sessions$ where users can be associated a single session.

Definition 2. *SRBAC model consists of the following components.*

- $USERS$, $ROLES$, $PRMS$, $SESSIONS$ *and* LOC, *represent the finite set of users, roles, permissions, sessions and locations respectively;*
- $UA \subseteq USERS \times ROLES$, *the relation that associates users with roles;*
- $assigned\_users(r : ROLES) \rightarrow 2^{USERS}$, *the mapping of a role onto a set of users. Formally:* $assigned\_users(r) = \{u \in USERS \mid (u, r) \in UA\}$;

- $PA \subseteq ROLES \times LOC \times PRMS$, *the relation that assigns a permission to a role available in location;*
- $assigned\_permissions(r : ROLES, l : LOC) \rightarrow 2^{PRMS}$, *the mapping of a role* r *onto a set of permissions based on location. Formally:* $assigned\_permissions(r, l) = \{p \in PRMS | (r, l, p) \in PA\}$;
- $user\_sessions(u : USERS) \rightarrow 2^{SESSIONS}$, *assigns a user onto a set of sessions;*
- $session\_roles(s : SESSIONS) \rightarrow 2^{ROLES}$, *the mapping of each session to a set of roles;*
- $avail\_session\_permissions(s : SESSIONS, l : LOC) \rightarrow 2^{PRMS}$, *the permissions available in a session for a location,* $\bigcup_{r \in session\_roles(s)} assigned\_permissions(r, l)$.

3.2 Hierarchical SRBAC

As explained earlier, hierarchies in RBAC define an inheritance relationship between roles, such that a role r_i inherits the permissions from role r_j if all permissions of r_j are also permissions of r_i. Since we present a model where the permissions assigned to the roles varies with location, the permission inheritance relationship among roles in presence of a role hierarchy must also depend on the location. That is, a role r_i would inherit the permissions of role r_j in locations L if all the permissions of r_j in locations L are also permissions of r_i in locations L and if $L \subseteq LOC$.

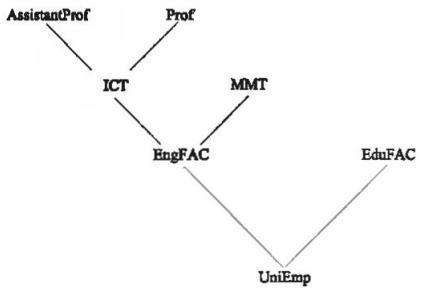


Fig. 3: Role hierarchy example where the role `Prof` would inherit permissions from the roles; `ICT`, `EngFAC` and `UniEmp` in locations specified by L.

This would mean that the role `Prof` in Figure 3 can activate all the permissions inherited from the roles `ICT`, `EngFAC` and `UniEmp` dependent on location together with the permissions assigned to the `Prof` directly.

The location dependent role hierarchy can be formally defined as follows.

Definition 3. *Role Hierarchies in SRBAC*

- $RH \subseteq ROLES \times ROLES \times LOC$ *is a partial order on roles with respect to locations, called dominance relation, written as* $\underset{(L)}{\succeq}$, *where* $r_i \underset{(L)}{\succeq} r_j$, *means*

that role $r_i \in ROLES$ *inherits all permissions that role* $r_j \in ROLES$ *has in locations* $L \subseteq LOC$*, and all the users of* r_i *are also users of* r_j*. If* L *is omitted then role* r_i *inherits all the permissions of* r_j *with respect to locations where* r_j *is defined;*

- $auth\_permissions(r : ROLES, l : LOC) \rightarrow 2^{PRMS}$ *is the mapping of a role* r *onto a set of permissions based of location* l *in presence of a role hierarchy (the permission set assigned directly to the role for that location together with permissions assigned to its junior roles in that location). Formally:* $auth\_permissions(r, l) =$

$$assigned\_permissions(r, l) \cup \left\{ \bigcup_{\forall r' : r \underset{(l)}{\succeq} r'} auth\_permissions(r', l) \right\};$$

- $auth\_usr(r : ROLES) \rightarrow 2^{USERS}$*, the mapping of a role* r *onto a set of users in presence of a role hierarchy. Formally:* $auth\_usr(r) = \{u \in USERS | r' \succeq r, (u, r') \in UA\}$.
- *Generally: Let* L *be a set of locations* $\{l_1, l_2, ...\}$ *normalized with respect to roles* $r_i, r_j \in ROLES$ *and* $l_i \in LOC$*. Then* $r_i \underset{(L)}{\succeq} r_j$ *means* $\bigvee_{l \in L} \left(r_i \underset{(l)}{\succeq} r_j \right)$.

From the above definition follows that if $r_i \underset{(l)}{\succeq} r_j$, then $auth\_permissons(r_j, l) \subseteq auth\_permissons(r_i, l)$ and $auth\_usr(r_i) \subseteq auth\_usr(r_j)$.

3.3 Constrained SRBAC

The proposed RBAC model [2] defines separation of duties to be enforced on a set of roles that may not be executed simultaneously by a user. Our model, extend the concept of separation of duties to allow users to be authorized to mutually exclusive roles if they cannot be executed in the same location. It is similar to SSD and DSD in that intends to limit the permissions available to a user. It differs from SSD and DSD in that the roles are mutually exclusive reliant on the location in which a user is situated. That is, two roles with assigned permissions may be mutually exclusive for a given location, however, for another location a user may be authorized to activate these two roles, since the set of permissions assigned to the roles may be different for distinct locations.

We define in our model both *Spatial Static Separation of Duty* and *Spatial Dynamic Separation of Duty* relations and are further elaborated and defined in the next two subsections.

Spatial Static Separation of Duty Relations. *Spatial Static Separation of Duty relations (SSSD)* enforce constraints on the assignment of users to roles with regards to location. This implies that if a user is assigned to a role in one location, the user cannot be assigned to another role in this location if these to roles are conflicting. Thus, a user may never activate to two roles that share a SSSD relation for a specified location. This is the stronger separation of duty

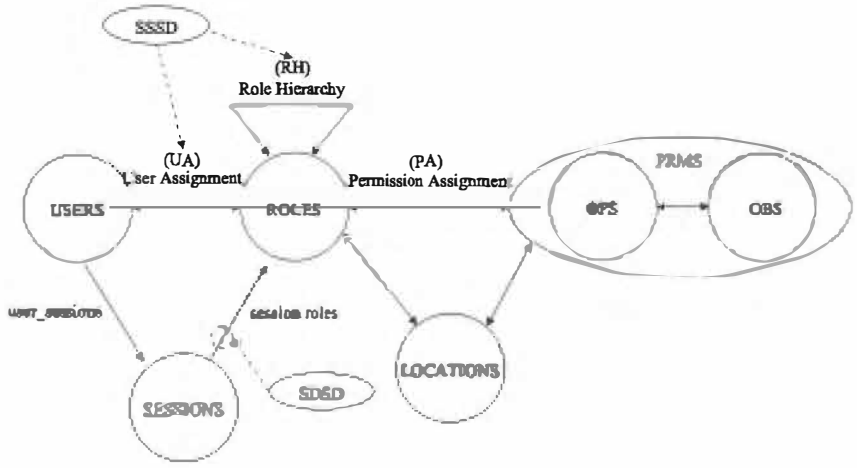


Fig. 4: Spatial Separation of Duty relation

relation, and our model would be similar to the standard RBAC model if the SSSD relation is valid for the entire location space.

Let us illustrate this with an example shown in Figure 5. This example environment contains two roles $\mathtt{R}_1$ and $\mathtt{R}_2$ that a user may activate for all locations except location $\mathtt{Zone}_3$, that is since we assume that $(\langle \mathtt{R}_1, \mathtt{R}_2 \rangle ; \mathtt{Zone}_3) \in SSSD$. In classical SSD, enforcing constraints on $\mathtt{R}_1$ and $\mathtt{R}_2$ would result in that no one user can be assigned to both roles for all the locations, that correspond to restriction $(\langle \mathtt{R}_1, \mathtt{R}_2 \rangle ; \mathtt{Zone}_1, \mathtt{Zone}_2, \mathtt{Zone}_3, \mathtt{Zone}_4) \in SSSD$ that is same as $(\langle \mathtt{R}_1, \mathtt{R}_2 \rangle) \in SSD$.

$\mathtt{Zone}_1$ $\mathtt{R}_1, \mathtt{R}_2$	$\mathtt{Zone}_2$ $\mathtt{R}_1, \mathtt{R}_2$
$\mathtt{Zone}_3$ $\mathtt{R}_1, \mathtt{R}_2$	$\mathtt{Zone}_4$ $\mathtt{R}_1, \mathtt{R}_2$

Fig. 5: Example on SSSD constraint where no one user is allowed to be assigned to both $\mathtt{R}_1$ and $\mathtt{R}_2$ in $\mathtt{Zone}_3$ and an example on Violation of a SDSD relation.

The formal definition of Static *Spatial* Separation of Duty is given below.

Definition 4. *Spatial Static Separation of Duty (SSSD).*

- $SSSD \subseteq \left(2^{ROLES} \times 2^{LOC} \times N\right)$ *is a collection of triples* (rs, ls, n) *where each* rs *is a role set,* ls *is normalized location set, and* n *is a natural number,* $n \geq 2$*, with the property that no user can be assigned to* n *or more roles from the set* rs *in any normalized location* l *from* ls*. Formally:* $\forall\, (rs, ls, n) \in SSSD$, $\forall l \in ls$, $\forall t \subseteq rs : |t| \geq n \Rightarrow \bigcap_{r \in t} authorized\_usr(r, l) = \varnothing$.

Spatial Dynamic Separation of Duty Relations. *Spatial Dynamic Separation of Duty relations (SDSD)* are enforced on the permissions assigned to roles that are activated in a user's session (see Figure 4). SDSD relations allow users to be assigned to two or more roles that are not conflicting when activated in separate sessions for specified locations, however, it would generate policy concerns when activated simultaneously in a user's session for other specified locations. This offers a great advantage compared with classical DSD, due to the fact that one can limit the validity of the constraint to yield in specific locations. A classical DSD constraint enforce restrictions on roles on the entire organization, i.e., in our case, the whole location space, while SDSD, limit the constraint only to be valid dependent on location such that a user may activate conflicting roles within a session for a location, other than the location where the SDSD constraint is specified. Let illustrate with an example.

In Figure 5, we see an example of a wireless environment where a user takes up two roles, $\mathtt{R}_1$ and $\mathtt{R}_2 \in ROLES$. These two roles may are authorized to be activated dependent of location, on various resources offered in this wireless environment. In addition, no one user is allowed to activate both $\mathtt{R}_1$ and $\mathtt{R}_2$ in location $\mathtt{Zone}_3$ in a single session, $(\langle \mathtt{R}_1, \mathtt{R}_2 \rangle ; \mathtt{Zone}_3) \in SDSD$. No SDSD constraint on $\mathtt{R}_1$ and $\mathtt{R}_2$ are specified for locations $\mathtt{Zone}_1$, $\mathtt{Zone}_2$, $\mathtt{Zone}_4$, thus a user may activate these two roles in a single session for the three locations. However, in $\mathtt{Zone}_3$, the user would violate the spatial separation of duty constraint defined in the security policy if the user was to activate $\mathtt{R}_1$ and $\mathtt{R}_2$ in one session (marked by an ellipse in Figure 5). Therefore, the user is not capable of activating both these roles in the same session in location $\mathtt{Zone}_3$. In classical DSD, the constraint on $\mathtt{R}_1$ and $\mathtt{R}_2$ would not only apply to $\mathtt{Zone}_3$, but the entire location space, consisting of $\mathtt{Zone}_1$, $\mathtt{Zone}_2$, $\mathtt{Zone}_3$ and $\mathtt{Zone}_4$.

The formal definition of Dynamic Spatial Separation of Duty is given below.

Definition 5. *Spatial Dynamic Separation of Duty (SDSD).*

- $SDSD \subseteq \left(2^{ROLES} \times 2^{LOC} \times N\right)$ *is a collection of triples* (rs, ls, n) *where each* rs *is a role set,* ls *is normalized location set, and* n *is a natural number,* $n \geq 2$*, with the property that no user may activate* n *or more roles from the set* rs *in any normalized location* l *from* ls*. Formally:* $\forall\, (rs, ls, n) \in SDSD$, $\forall l \in ls$, $\forall s \in SESSIONS$,
  $\forall t \subseteq session\_roles(s) \cap rs : |t| \geq n \Rightarrow \bigcap_{r \in t} authorized\_usr(r, l) = \varnothing$.

4 Conclusions

In this paper we have presented Spatial RBAC (SRBAC), a novel model that extends RBAC to incorporate location information associated with roles in order to permit location-based definition of security policy. In the SRBAC model, permissions are dynamically assigned to the role dependent on location, thus a user assigned to a role may have different permissions reliant on the location. Incorporating spatial information in RBAC as proposed in this paper would enable RBAC to implemented in future mobile computing environments.

References

1. Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role-based access control models. IEEE Computer **29** (1996) 38–47
2. Ferraiolo, D.F., Sandhu, R., Gavrila, S., Kuhn, D.R., Chandramouli, R.: Proposed NIST standard for role-based access control. ACM Transactions on Information and System Security (TISSEC) **4** (2001) 224–274
3. Bertino, E., Bonatti, P.A., Ferrari, E.: TRBAC: A temporal role-based access control model. ACM Transactions on Information and System Security **4** (2001) 191–223
4. Osborn, S., Sandhu, R., Munawer, Q.: Configuring role-based access control to enforce mandatory and discretionary access control policies. ACM Transactions on Information and System Security (TISSEC) **3** (2000) 85–106
5. Joshi, J.B.D., Bertino, E., Latif, U., Ghafoor, A.: Generalized temporal role based access control model (GTRBAC) (part I)– specification and modeling. Technical report, CERIAS TR 2001-47, Purdue University, USA (2001)
6. Hansen, F., Oleshchuk, V.: Spatial role-based access control model for wireless networks. In: IEEE Vehicular Technology Conference VTC2003. (2003)
7. Hansen, F., Oleshchuk, V.: Application of role-based access control in wireless healthcare information systems. In: Proc. For Scandinavian Conference in Health Informatics. (2003) 30–33
8. Scott, D., Beresford, A., Mycroft, A.: Spatial security policies for mobile agents in a sentient computing environment. In: Lecture Notes in Computer Science, Springer-Verlag Heidelberg (2003) 102–117
9. Covington, M.J., Long, W., Srinivasan, S., Dev, A.K., Ahamad, M., Abowd, G.D.: Securing context-aware applications using environment roles. In: Proceedings of the sixth ACM symposium on Access control models and technologies, ACM Press (2001) 10–20
10. Mantoro, T., Johnson, C.: Location history in a low-cost context awareness environment. In: Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003, Australian Computer Society, Inc. (2003) 153–158
11. Kaplan, E.: Understanding GPS: Principles and Applications. Boston: Artech house Publishers (1996)
12. Want, R., Hopper, A., Falcão, V., Gibbons, J.: The active badge location system. ACM Transactions on Information Systems (TOIS) **10** (1992) 91–102
13. Werb, J., Lanzl, C.: A positioning system for finding things indoors. IEEE Spectrum **35** (1998) 71–78
14. Priyantha, N.B., Chakraborty, A., Balakrishnan, H.: The cricket location-support system. In: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, ACM Press (2000) 32–43
15. Leonhardt, U., Magee, J.: Towards a general location service for mobile environments. In: Proceedings of the 3rd IEEE Workshop on Services in Distributed and Networked Environments. (1996) 43–50

An Access Control System for Business Processes for Web Services

Hristo Koshutanski Fabio Massacci
Dip. di Informatica e Telecomunicazioni - Univ. di Trento
via Sommarive 14 - 38050 Povo di Trento (ITALY)
{hristo,massacci}@dit.unitn.it

Abstract— **Web Services and Business Processes for Web Services are the new paradigms for the lightweight integration of business from different enterprises.**

Security and access control policies for Web Services protocols and distributed systems are well studied and almost standardized, but there is not yet a comprehensive proposal for an access control architecture for business processes. The major difference is that business processes describe complex services that cross organizational boundaries and are provided by entities that sees each other as just partners and nothing else.

This calls for a number of differences with traditional aspects of access control architectures such as: credential vs. classical user-based access control; interactive and partner-based vs. one-server-gathers-all requests of credentials from clients; controlled disclosure of information vs. all-or-nothing access control decisions; abducing missing credentials for fulfilling requests vs. deducing entailment of valid requests from credentials in formal models.

Looking at the access control field we find good approximation of most components but not their synthesis into one access control architecture for business processes for web services, which is the contribution of this paper.

I. Introduction

Middleware has been the enterprise integration buzzword at the end of the past millennium. Nowadays a new paradigm is starting to take hold: Web Services (WS for short). Setting hype aside, the major difference between middleware solutions (CORBA, COM+, EJB, etc.) and WS is the idea of lightweight integration of business processes from different enterprises.

Basic WS are well studied and standardized, for what concerns access control and security. There are also many approaches [1], [2], [3] for controlling access to services in distributed systems, and an advanced standardization process (see for instance the OASIS XACML [4] proposal). With the notable exception of provisional access control [5] and trust negotiation [6], access control models rest on the idea that the server picks the evidence you sent on who you are (credentials), and what you want (request), checks its evidence on what you deserve (policies) and makes a decision.

Moving up in the WS hierarchy from single services to *orchestration and choreography of WS and business processes* the picture changes. Business processes describe complex services that cross organizational boundaries and are provided by partners.

This work is partially funded by the IST programme of the EU Commission, FET under the IST-2001-37004 WASP project and by the FIRB programme of MIUR under the RBNE0195K5 ASTRO Project.

The paradigmatic example in the WS standards is a travel agent WS that must orchestrate a combination of plane and train tickets, car rental, hotel booking and insurance, each service offered by different partner which may or may not be involved according to the actual unrolling of the workflow.

For example consider the problem of going to a nice "Shakespearian Tour" in Italy: you might decide to go to the city of Shylock, and from there rent a car and travel to Romeo and Juliet's last resort, to jump then on a train and visit the Senate's seat where Pompeous spoke after Caesar's death. However, you might as well decide to travel instead to Germany first and then the train to Verona from there. In the first case you might need to use a car rental company. The second path may require to contact a German train company for the schedule, which is not needed if you land directly in Italy.

Let us now consider the problem of "lightweight" credentials such as the German train discount card or the car rental gold member card. Should the user provide them anyway at the beginning? Obviously not. Should the server orchestrating the process require each partner to publish its policy on discounts? Obviously not. Such problems are not simply problems of practicality, but have major security implications:

1) Credential vs. identity based access control – A WS is something you publish on the Web for everybody to use it, so the system has to be close to trust management systems [3];
2) Orchestrating vs. combining – *partners* have different security policies and *are just partners* and not part of the same enterprise. They may not wish to disclose their policies to the server orchestrating the request. So, we cannot simply combine the policies, we need to orchestrate the request grant/deny/process of many different policies/partners.
3) Interactive vs. one-off access control – if partners have different policies they might as well require different credentials to a client. Privacy considerations make gathering all potentially needed credentials from clients difficult. Furthermore, this may simply be impossible. An airline may want to ask confidential information directly to its frequent fliers (e.g., confirmation of religious preferences for the food) and not to the Web travel agent orchestrator of the process. This calls for an interactive process in which the client may be asked on the fly for additional credentials and may grant or deny such

requests[1].

4) Abducing vs. deducing credentials – in most classical formal models we deduce that a request is valid because it is entailed by the combination of the policy and the set of available credentials. Here, a partner must be able to infer the causes of some failed request to ask the missing credentials to the client. The corresponding logical process is no longer deduction but it is abduction. So we must have co-existence of deduction (for deciding access and release of information) and abduction (for explaining failed accesses).
5) Data vs. source level communication – the choice of format for messages is always rather complicated, as it calls for the implementation of software that is able to interpret its meaning. In a Business Process scenario we no longer need messages, but just "mobile" processes. A client will receive a business process so that he can simply execute the source to obtain and send the missing credential. An authorization server can download a business process from a policy orchestrator and obtain the desired authorization.

Looking at the access control field we find a good approximation of most components: we have proposals for combining policies at the logical level [7], [8] and at the architectural level [4]. We have proposals for calculi for controlling release of information [9], and procedures for trust negotiations and communication of credentials [6], architecture for distributed access control [4], [2], [1].

What is missing is a way to synthesize *all* these aspects into one access control architecture for business processes of WS, which is the contribution of this paper.

In the next section we introduce some notion about WS and Business Processes for WS. Then we present our architecture and discuss how the entire message passing scheme can be implemented as "mobile" processes. Section V explains how we can use logical deduction and logical abduction to build a firm foundation for the interactive process of inferring disclosable credentials from access control policies and from release policies. Next we discuss how everything can be implemented using Business Process themselves. A brief discussion of related works concludes the paper.

II. A Primer on WS and Business Processes

A Web Service as defined by the standard[2] is "an interface that describes a collection of operations that are network-accessible through standardized XML messaging. A Web service is described using a standard, formal XML notion, called its *service description*. It covers all the details necessary to interact with the service, including message formats (that detail the operations), transport protocols and location."

[1]Note that the workflow may even take completely different paths based on the results of interaction. For example a rent-a-car operator may require a signed credit card number plus a physical address. The client may deny such requirement and thus another operator may be chosen that only asks for a credit card number.

[2]Web Services Conceptual Architecture (WSCA), http://www- 3.ibm.com/software/solutions/webservices/pdf/WSCA.pdf

The idea behind Web services is to encapsulate and make available enterprise resources in a new heterogeneous and distributed way.

Web Service Technology Stack		Access Control Issues
Layer	Standards	AC Granularity
Workflow	BPEL4WS	Workflow-level AC
Discovery	UDDI	Description-level AC
Service Description	WSDL	Service-level (End Point) AC
Messaging	SOAP/XML Protocol	Universal way to convey AC info
Transport Protocols	HTTP,HTTPS,FTP,SMTP	–

Fig. 1. Web Services Technology Stack & Access Control Issues

The WS architecture, as defined by W3C[3], is divided into five layers grouped into three main components - Wire, Description, and Discovery (Fig. 1). The *Wire* component comprises the messaging and transport layers with the SOAP protocol and the XML message format. *Discovery* offers users a unified and systematic way to find, discover, and inspect service providers over the Internet. There are two standards proposed at this level - Universal Description, Discovery and Integration (UDDI) and Web Service Inspection Language (WSIL).

Moving upward we found the *Service Description* layer and the *Business Process Orchestration* layer. The service description layer is responsible for describing the basic format of offered services (protocols and encodings, where a service resides, and how to invoke it). The standard for describing the communication details at this layer is Web Service Description Language (WSDL).

The Business Process Orchestration layer is an extension of the service model defined at the description layer. This layer is responsible for describing the behavior of complex business and workflow processes. Intuitively, business processes are graphs where each node represents a business activity and primitive nodes are in WSDL. The recently released standard at this layer is the Business Process Execution Language for WS (BPEL4WS)[4].

The BPEL4WS primitive activities are the following:

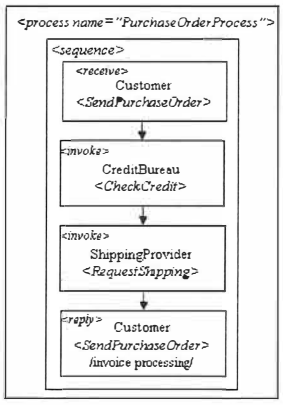
`<invoke>` - invoking an operation on some Web service;
`<receive>` - waiting for an operation to be invoked by someone externally;
`<reply>` - generating the response of an input/output operation;
`<assign>` - copying data from one place to another.

More complex activities can be constructed by composition:

`<sequence>` allows a developer to define an ordered sequence of steps;

[3]W3C. Web Services Architecture. http://www.w3.org/TR/ws-arch.

[4]BPEL4WS specification – http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/




```
<process>
   <sequence>
      <receive partner="Customer"
               portType="purchaseOrderPT"
               operation="SendPurchaseOrder"
               container="PO">
      </receive>
      <invoke  partner="CreditBureau"
               portType="CheckCreditPT"
               operation="CheckCredit">
      </invoke>
      <invoke partner="shippingProvider"
              portType="shippingPT"
              operation="RequestShipping"
              inputContainer="shipingRequest"
              outputContainer="shippingInfo">
                <source linkName="ship-to-invoice"/>
      </invoke>
      <reply partner="Customer"
             portType="purchaseOrderPT"
             operation="SendPurchaseOrder"
             container="Invoice"/>
   </sequence>
</process>
```

Fig. 2. Example of BPEL4WS Process

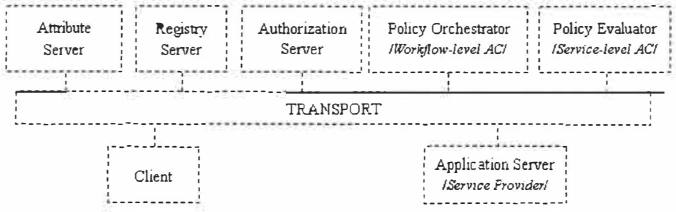


Fig. 3. Cross-section view of the architecture

`<switch>` - allows a developer to have branching;
`<while>` - allows a developer to define a loop;
`<flow>` - allows a developer to define that a collection of steps has to be executed in parallel.

An example of compositions of services is shown in Figure 2: a buyer service is ordering goods from a seller service, i.e. the buyer service invokes the order method on the seller service, whose interface is defined using WSDL. The seller service invokes a credit validation service to ensure that the buyer can pay for the goods and after that continue by shipping the goods to the buyer. The credit validation service can take place at a credit bureau site in a separate security domain. Notice that a number of partners participate in the process that therefore crosses administrative boundaries.

The XML code shown in Figure 2 is a very brief example of the scenario described above in the notations of BPEL4WS primitives. The structure of the processing section is defined by the `<sequence>` element, which states that the elements contained inside are executed in this order. The node contents is self explanatory.

III. Architecture

Combining the traditional proposals for distributed access control and the essential components used for Web services we propose here a security architecture for orchestrating authorization of Web Services Processes. Figure 3 shows a cross-section view of the architecture, whereas Figure 4 shows a horizontal view of it. A brief description of the servers shown in the figure is given below.

AttributeServer is responsible for providing group/role membership information as in [1], for instance in the form of membership and non-membership certificates.

RegistryServer is responsible for maintaining relations between services and service providers implementing a particular service. When a **Client** requests the **RegistryServer** for a specific service, the latter responds with a list of **ApplicationServer**s implementing the requested service.

AuthorizationServer decouples the authorization logic from the application logic. It is responsible for *locating*, *executing*, and *managing* all needed **PolicyEvaluator**s, and returning an appropriate result to the **ApplicationServer**. Also it is responsible for managing all the *interactions* with the **Client**.

PolicyEvaluator terminology borrowed from Beznosov et al [2], is an entity responsible for achieving endpoint decisions on access control (see Figure 3). All partners involved in a business process are likely to be as different entities, each of them represented by a **PolicyEvaluator**.

PolicyOrchestrator from the authorization point of view is an entity responsible for the workflow level access and release control. It decides which are the partners that are involved in the requested service (Web service workflow) and on the base of some orchestration security policies to combine the corresponding **PolicyEvaluator**s in a form of a Web process (*Policy Composition Process*) that is suitable for execution by the **AuthorizationServer**.

To secure the entire architecture we must make some assumptions on the security properties of the lower levels. Obviously we assume authentication, confidentiality, and message integrity at the transport and message levels. So, we assume that we have already in place the proposed standards.

At transport level we assume the adoption of the WS-Security specification[5] that describes enhancements to SOAP messaging to provide message integrity, confidentiality, and authentication. For the message level one can use the W3C and IETF specification for XML-Signature[6] and W3C XML-Encryption[7], or the recently release specifications by IBM and Microsoft for WS secure conversations[8].

Assuming security at lower level, the second key component is the languages and format of communications. We propose here a major innovation: the typical exchange of messages in an access control system is at "data" level (credentials, policies, requests, objects, etc.) that are interpreted by the recipients. This choice makes the actual implementation of proposed access control infrastructure difficult and often not easily portable. Here we propose to exchange messages at "source code" level and in particular at the level of business process description. It means that instead of sending just messages that have to be interpreted by entities, we truly have

[5]WS-Security – http://www-106.ibm.com/developerworks/webservices/library/ws-secure

[6]XML-Signature – http://www.w3.org/TR/xmldsig-core

[7]XML-Encryption – http://www.w3.org/TR/xmlenc-core

[8]WS-SecureConversation – http://www.ibm.com/developerworks/library/ws-secon

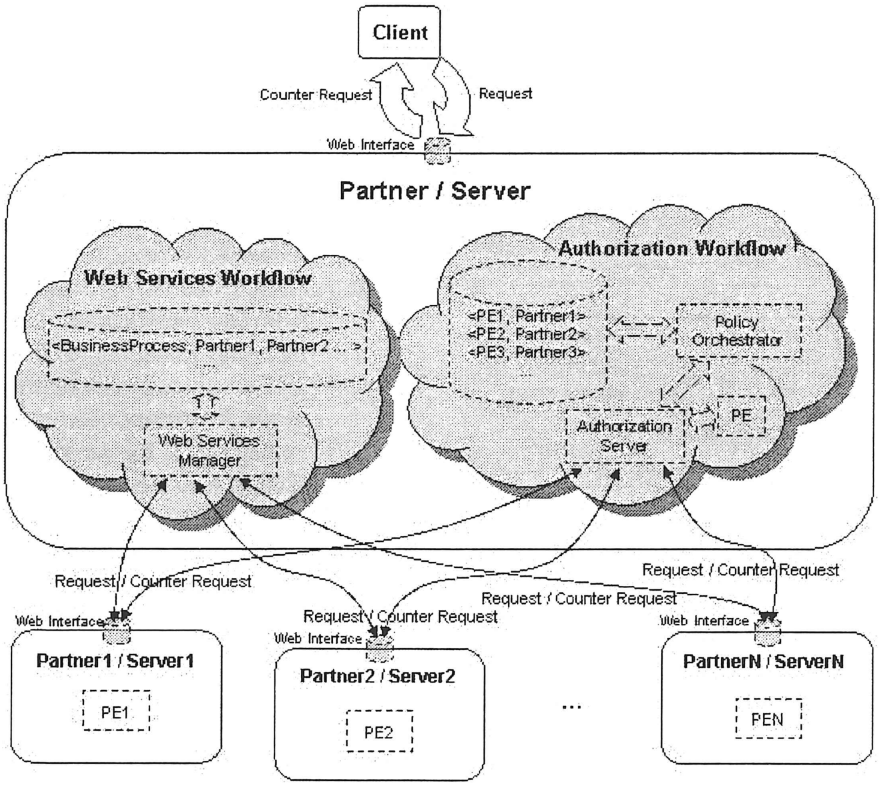


Fig. 4. Horizontal view of the architecture

mobile processes passing from one entity to another indicating themselves what the recipient has to do.

The mobility of authorization processes has a number of advantages. First of all a server simply needs an off-the-shelf interpreter for business processes for a quick implementation. Second we have more flexibility for describing the process leading to an access control decision. Some PolicyEvaluators may decide to disclose it XACML policies and therefore send a mobile processes, which just describe the evaluation of the policies along some XACML rules. Other PolicyEvaluators may instead decide to offer an external interface, so that they just specify a container for requests and an output container for its decision. All intermediate choices are possible so that one can accommodate also provisional access control or the interactive version that we advocate here.

Leading this approach at an extreme the AuthorizationServer can simply receive a business process from the orchestrator and execute it. The process may still be computationally intensive as an AuthorizationServer may have to process thousands or millions of authorization workflows, but it could be logically very simple thus reducing the TCB to the simple execution of certified processes from certified sources[9].

The role of the PolicyEvaluator is to encapsulate the connected with it partner's specific access control model, authorization policy, and requirements with their internal representation, interpretation, and mechanisms for computing an access decision and presenting it as a service using standardized Web service interface (e.g., WSDL).

[9]Recall that we assume that authentication, integrity, and confidentiality are assured at message and transport level.

The entity burdened with constructing the authorization workflow (Figure 4) is the PolicyOrchestrator. The PolicyOrchestrator functionality can be considered as having two main tasks: first one, called *Policy Composition Service*, is to select which are the partners involved in the requested process and to combine the corresponding PolicyEvaluators in a policy composition process, and return it back to the AuthorizationServer. After the AuthorizationServer having finished the execution of the policy composition process it asks[10] the PolicyOrchestrator for applying the workflow level release policies over the results from the execution – the second main task. The process of applying release control polices, called *Release Policy Service*, captures how the final authorization decision should be released to the Client.

IV. Interactive Communications as "Mobile" Processes

We have decided to use the term *mobile process* because it well expresses the idea of using mobile code together with the functionality of Web processes. The main advantages of using mobile processes in our authorization framework are *flexibility* and *simplicity* of entities. Flexibility because of recipient of mobile process is not limited to the functions and computational algorithms that the recipient's logic predefines. Migrations of actors in the system from one server to another is easier with mobile processes and the system as a whole is more flexible. Entities in the framework becomes simpler,

[10]This is the case if it is specified in the policy composition process, i.e. depends on the security policies being applied in constructing the policy composition process.

having little functionality pre-engineered into them, as we will see in section VI.

The next important step in advocating mobile processes is to specify a language that is needed for coding them. We have identified it as a *language for communicating interactive requests back to a* Client. This is even in the case when a Client is an AuthorizationServer waiting for a response either from a PolicyOrchestrator or from a PolicyEvaluator. This language can be designed with a black box view of the PolicyEvaluator, but must be easily interpretable from the Client side. Thus we propose to use BPEL4WS itself as a language in which requests are coded. The PolicyEvaluator/PolicyOrchestrator must represent its request as a WS business process that can then be interpreted and executed by the Client. If the PolicyEvaluator wants part of the request to be only visible to the Client it can use the available XML-crypto features to protect the relevant part.

Loosely speaking we may say that the Client starts by executing a simple `<invoke>R</invoke>` and obtain in return either its result or a more complicated process to execute. For example a BPEL4WS interactive request may specify a `<input container>` where to put a digitally signed copy of the travel contract sealed with the public key of the rent-a-car company (a process that can be specified as a `<sequence>` of events).

The idea is intuitive and appealing but there is an essential detail that must be taken care of. Notably, the AuthorizationServer will receive a number of interactive requests while controlling its workflow and the combination of these requests and the service workflow specification is essential. The simplest solution is to ignore such interaction: all interactive requests are compiled into a `<flow>` and the result is sent back to the Client. Such solution is hardly satisfactory from the point of view of the Client: we often want to know "why" some additional information is needed. See the example of Figure 2: at some stage somebody may ask for a digitally signed declaration about our address. We may consider this request fair enough from the shipping agent, but not from the credit checking bureau. So, each BPEL4WS interactive request must be supplemented with a special tag [root/context]:

- root requests will be compiled with a `<flow>` construct and returned together with the overall result of the computation for contextual requests;
- contextual requests the PolicyOrchestrator will make a copy of the WS process (*not* the authorization process) and replace each step S for which an additional request I has been called with the request and a context indicating the WS (partner and all) that required the additional credential. The PolicyOrchestrator will then prune the WS process removing all nodes that were not on a path from the root to the newly modified nodes and sends the result to the Client.

The last step is necessary to protect the overall workflow from unnecessary disclosure.

This combination is sufficiently adequate for most uses, but still it offers the PolicyOrchestrator just the choice of compiling individual requests rather than combining them. Here we have identified an important point in the PolicyOrchestrator where we need to introduce a new language - a *language for combination of policies and interactive requests at workflow level*. So far we have not found a proposal that is entirely satisfactory, part because there are not enough case studies of WS Business Processes to guide the selection of policies at workflow level.

The proposal by Bertino et al. [10], is fairly expressive but only focuses on implementing snapshot constraints on a workflow level (i.e. safety properties). So it is not possible to express properties such as "if Y is repeatedly true then eventually X should happen".

The usage of algebraic constructs based on dynamic logic proposed by Wijesekera and Jajodia [8] seems more promising. Indeed `<invoke>` operation would be mapped into single action, `<sequence>` into sequential compounder, `<switch>` into non deterministic choice (each case represented by a test) and `<flow>` by intersection. This does not mean that we would use dynamic logic for actual implementation[11], but rather that the logical language may offer a formal foundation to policy written in BPEL4WS.

V. The Abduction of Missing Credentials

For the deployment of the architecture, the PolicyEvaluator must be able to determine the set of additional credential that are necessary to obtain a service in case of failure. This problem may of course be shifted on the implementors of PolicyEvaluators, as the architecture only needs that the outcome of this derivation is mapped into some BPEL4WS process that is then sent to the client.

However, there is no algorithm in either the formal or the practical models of access control and trust negotiations to derive such credentials from the access control policy. The works on trust negotiations [11], [6] focus on communication and infrastructure and assume that requests and counter requests can be somehow calculated from the access policy. The formal models on credential-based access control and policy combination [10], [7], [8] don't treat the problem of inferring missing credentials from failed requests, as they are within the frame of mind of inferring successful requests from present credentials. Also standardization efforts like the XACML proposals [4] gives rules for deriving what is right (evaluating policies) and not rule for understanding what is wrong.

Here, we present an approach based on logic that allows for a clean solution of these problems. For sake of simplicity (and popularity), assume that the policy is expressed using Datalog rules or logic programs with the stable model semantics (if we need negation to implement some constraints like separation of duties). What we need is a logical implementation of the following process:

1) the PolicyEvaluator receives the credentials and evaluates the request against the policy augmented with the credentials, i.e. whether the request is a logical consequence of the policy and the credentials;

[11] This is less critical than prejudice may suggest. The ML implementation of Peter Patel-Schneider at Bell-Labs can actually crack significant dynamic logic theorems in milliseconds.

2) if the request is granted nothing needs to be done;
3) if the request fails we evaluate the given credential against a release policy of the PolicyEvaluator to infer which are the credentials whose need can be disclosed on the basis of the credentials already received;
4) abduce the actually needed credentials by re-evaluating the request against the policy and considering the potentially disclosable credentials determined at the previous step; only the needed credential are communicated to the client.

In a nutshell, what we need for the implementation of PolicyEvaluator is to implement two main inference capabilities: *deduction* and *abduction* [12]. We need to use deduction to infer whether a request can be granted on the basis of the present credentials as in [9], [10], [7], we use abduction to explain which minimum set of credentials would be necessary to grant a failed request. Obviously it is not necessary to use logic, what we claim is that the underlying logical constructs that we need for our access decisions are these two conceptually different operations.

Due to lack of space, here we just give the basic hint of the formalization.

Definition 1 (Access Control): Let P be a datalog program (or stratified logic program) representing an access control policy, let r be an atom representing a request, let C be a set of atoms representing a set of given credentials, the *request is granted* if and only if $P \cup C \models r$.

Definition 2 (Release Control): Let P be a datalog program (or stratified logic program) representing a release control policy, let d be an atom representing a credential, let C be a set of atoms representing a set of given credentials, the *credential d is disclosable* if and only if $P \cup C \models d$.

Definition 3 (Access Control Explanation): Let P be a datalog program (or stratified logic program) representing an access control policy, let r be an atom representing a request, let C be a set of atoms representing a set of given credentials, let $D_P \supseteq C$ be a set of atoms representing disclosable credentials, an *explanation of missing credentials* $C_M \subseteq D_P$ such that

1) $P \cup C \not\models r$
2) $P \cup C \cup C_M \models r$
3) $P \cup C \cup C_M$ is consistent

The first conditions says that the missing credentials are indeed needed. The second condition says that they are sufficient and the last condition says that they are actually meaningful. In presence of positive Datalog program such as for Bonatti and Samarati's logic [9] and Li's Delegation Logic 1 [7], the consistency condition is satisfied by default. In presence of constraints on the execution or negation as failure, as in Bertino et al. Datalog programs for workflow policies [10] — which can be easily augmented with credentials — the consistency condition is essential to guarantee that the abduced set of atoms makes sense. Indeed, constraints could make $P \cup C \cup C_M$ inconsistent and therefore it would not make much sense to say that the request r should be granted from a system.

In Figure 5 is shown a logic program showing a university online library access and release rules. The notations for declarations, credentials, and services are borrowed from Bonatti and Samarati [9]. Here `decl` means that it is a statement (e.g., identity, address) declared by the client, while `cred` is a statement declared and signed by a key corresponding to some trusted authority. Consider rule 4 that says "to have access to service `reading` the client should have access to library (presenting Id and some library card) and a loan library card". Rule 10 says "to reveal the need for a loan library credential there should be a declaration of the library's Id and some library credential".

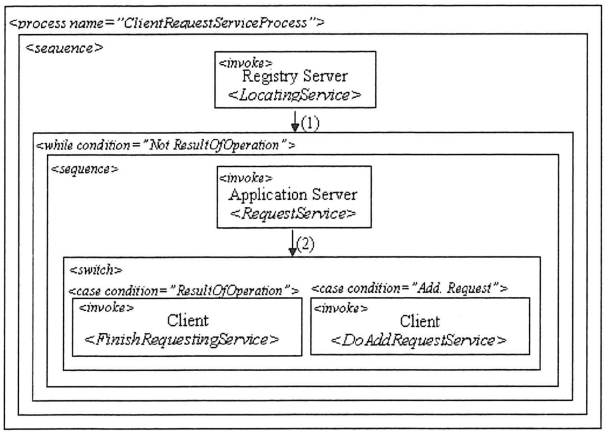


Fig. 6. Client Application Process Diagram

If the PolicyEvaluator is given the declaration $\mathrm{decl}(id1568)$ and the credential $\mathrm{cred}(card(user, john, id1568), bibK)$, together with the request for reading the journal articles on-line. The query $\mathrm{serv}(reading)$ does not follow from the policy and the given declarations and credentials. So, we apply the release policy and infer that the following credentials are disclosable:

$$\mathrm{decl}(john, cs), \mathrm{decl}(id1568),$$
$$\mathrm{cred}(researcher(id1568, cs), csK),$$
$$\mathrm{cred}(card(user, john, id1568), bibK),$$
$$\mathrm{cred}(member(john, cs), csK),$$
$$\mathrm{cred}(card(loan, john, id1568), bibK).$$

The abduction algorithm derive two possible answers for the credentials:

$$C_{M1} = \{\mathrm{decl}(john, cs), \mathrm{cred}(member(john, cs), csK)\}$$
$$C_{M2} = \{\mathrm{cred}(card(loan, john, id1568), bibK)\}$$

Both sets are minimal with respect to the subset inclusion ordering and only C_{M2} is minimal with respect to a set cardinality ordering. In case the first set is chosen the PolicyEvaluator will compile a `<flow>` node for sending the requests back to the client.

VI. Component Algorithms as Business Processes

This section shows how we can describe entities in our architecture and how they can communicate each other using BPEL4WS specification.

The Client process is shown in Figure 6. In the figure, after the Client has requested the ApplicationServer for getting a service R, presenting its credentials, there are two cases:

Access Policy:

$$\begin{aligned}
\text{serv}(query()) &\leftarrow \text{decl}(Id), \text{cred}(card(Type, Name, Id), biblioK) && (1)\\
\text{serv}(query(citations)) &\leftarrow \text{serv}(access), \text{cred}(member(Name, Dept), K_D), assoc(Dept, K_D) && (2)\\
\text{serv}(booking) &\leftarrow \text{decl}(Name, Dept), \text{cred}(card(loan, Name, Id), biblioK) && (3)\\
\text{serv}(reading) &\leftarrow \text{serv}(access), \text{cred}(card(loan, Name, Id), biblioK) && (4)\\
\text{serv}(reading) &\leftarrow \text{cred}(academic(Name, UnivId), K_U), assoc(university, K_U) && (5)\\
\text{serv}(reading) &\leftarrow \text{serv}(query(citations)), \text{cred}(researcher(Name, Dept), K_D), assoc(Dept, K_D) && (6)
\end{aligned}$$

Release Policy:

$$\begin{aligned}
\text{decl}(Name, Dept) &\leftarrow \text{decl}(Id) && (7)\\
\text{cred}(researcher(Name, Dept), K_D) &\leftarrow \text{decl}(Name, Dept), \text{cred}(card(Type, Name, Id), bibK) && (8)\\
\text{cred}(member(Name, Dept), K_D) &\leftarrow \text{decl}(Name, Dept) && (9)\\
\text{cred}(card(loan, Name, Id), bibK) &\leftarrow \text{decl}(Id), \text{cred}(card(Type, Name, Id), bibK) && (10)\\
\text{cred}(academic(Name, UnivId), K_U) &\leftarrow \text{decl}(UnivId), \text{decl}(Name, Dept) && (11)
\end{aligned}$$

Fig. 5. University Library WS Access and Release Policies

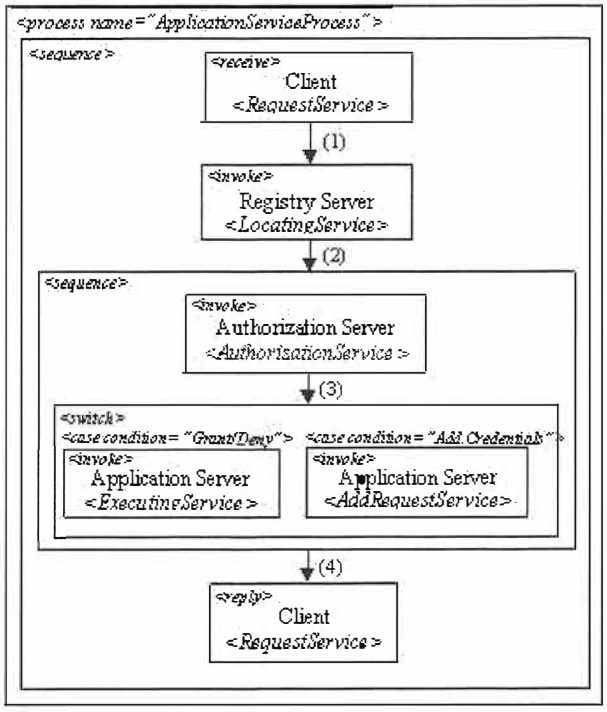


Fig. 7. Application Server Process Diagram

Additional Request - in this case is returned a counter request (a process), indicating what should be done by the Client. After that locally is invoked a service *DoAddRequestService* for executing the required process. Because of the while loop again is requested the service R with the result of the process; ResultOfOperation - in this case is returned the result of the requested service R and the Client's process finishes. The ApplicationServer, after the Client's request for accessing the service R, asks the RegistryServer (step 1 in Figure 7) for locating its *AuthorizationService*. After that the *AuthorizationService* is invoked along with Client's credentials and the requested service R for taking the authorization decision (step 2 in Figure 7). Then we can switch between explicit Grant/Deny response returned from the AuthorizationServer in the case of which is executed or not the requested service R and the results are returned back to the Client (step 4 in Figure 7), or in the case of additional credentials is executed the *AddRequestService*, which either executes some counter-requirements that have to be presented to the Client or redirects the entire request to the Client (step 4 in Figure 7).

The AuthorizationServer process, shown in Figure 8, is the following: after the *AuthorizationService* has been invoked by the ApplicationServer the *PolicyCompositionService* located in the PolicyOrchestrator is invoked. The result of the service invocation (step 1 in Figure 8) is a policy composition process (e.g., BPEL4WS) indicating what should be done by the AuthorizationServer in order to be taken the final authorization decision. After obtaining the process (step 2 in Figure 8), the AuthorizationServer starts executing it, requesting all needed PolicyEvaluators with respect to that process, i.e. some of them in parallel, others in a sequence etc. Here the policy composition process consists of a sequence indicating that first the AuthorizationServer has to execute all PolicyEvaluators relevant to the requested service R orchestrated in a specific way (where the most intuitive structure is a `<flow>` one indicating execution in parallel, as shown in Figure 8), and after that executing the *ReleasePolicyService* responsible for taking the final access decision. After finishing the policy composition process, the AuthorizationServer returns the final access decision to the ApplicationServer (step 4 in Figure 8).

VII. Conclusions and Related Work

As we have already discussed, a number of access control models have been proposed for workflows [10], role based access control on the web [13], entire XML documents [14], [15], tasks [16], and DRM [17], possibly coupled by sophisticated policy combination algorithms . However, they have mostly remained within the classical framework where servers know their clients pretty well: they might not know their names

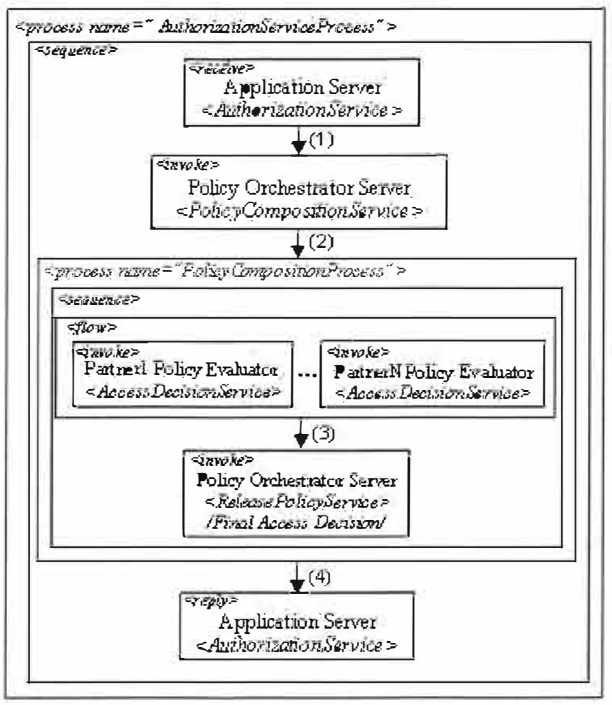


Fig. 8. Authorization Server Process Diagram

but they know everything about what, when, and how can be used by these clients.

In most proposals, the possibility that servers may get back to the calling Clients with some counter requests is not considered. This even in the case where the Client is actually an AuthorizationServer querying different PolicyEvaluator servers.

In one of the earliest work on distributed access control by Woo and Lam [1] the ApplicationServer offloads its authorization policy to an AuthorizationServer. After evaluating the policy the AuthorizationServer hands out authorization certificate to the Client, which the Client has to present along with its request.

An architecture close to ours has been proposed by Beznosov et al. [2]. Authorizations are managed by an Authorization Service, and its Access Decision Object (ADO). The ADO obtains references to all PolicyEvaluators related to the Client's request, asks a decision combinator for combining decisions according to a combination policy, and returns the decision back to the Client.

In this paper we have proposed a solution to address the challenges of WS processes: a possible architecture for the authorization of business processes for Web services. We have identified an interactive access control model as a way for protecting security interests wrt disclosure of information and access control of both servers and clients. Logical abduction is the solid semantical foundation upon which interaction can be build.

In the model a Client interacts (contracts) with the servent in order to finalize the necessary set of credentials needed to satisfy all partners' requirements related to the process. We propose to use "mobile" processes as messages exchanged in the architecture, and specified how entities in the architecture can be implemented using WS processes themselves.

References

[1] T. Y. C. Woo and S. Lam, "Designing a distributed authorization service," in *Proceedings of Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. INFOCOM*, vol. 2. IEEE Press, 1998, pp. 419–429.

[2] K. Beznosov, Y. Deng, B. Blakley, C. Burt, and J. Barkley, "A resource access decision service for CORBA-based distributed systems," in *Proceedings of 15th IEEE Annual Computer Security Applications Conference. (ACSAC '99)*. IEEE Press, 1999, pp. 310–319.

[3] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. D. Keromytis, "The role of trust management in distributed systems security," *Secure Internet programming: security issues for mobile and distributed objects*, pp. 185–210, 1999.

[4] S. Godik and T. Moses, *eXtensible Access Control Markup Language (XACML)*, OASIS, February 2003, www.oasis-open.org/committees/xacml/.

[5] M. Kudo and S. Hada, "XML document security based on provisional authorization," in *Proceedings of the 7th ACM conference on Computer and Communications Security*. ACM Press, 2000, pp. 87–96.

[6] T. Yu, M. Winslett, and K. E. Seamons, "Supporting structured credentials and sensitive policies through interoperable strategies for automated trust negotiation," *ACM Transactions on Information and System Security (TISSEC)*, vol. 6, no. 1, pp. 1–42, 2003.

[7] N. Li, B. N. Grosof, and J. Feigenbaum, "Delegation logic: A logic-based approach to distributed authorization," *ACM Transactions on Information and System Security (TISSEC)*, vol. 6, no. 1, pp. 128–171, 2003.

[8] D. Wijesekera and S. Jajodia, "Policy algebras for access control the predicate case," in *Proceedings of the 9th ACM conference on Computer and Communications Security*. ACM Press, 2002, pp. 171–180.

[9] P. Bonatti and P. Samarati, "A unified framework for regulating access and information release on the web," *Journal of Computer Security*, vol. 10, no. 3, pp. 241–272, 2002.

[10] E. Bertino, E. Ferrari, and V. Atluri, "The specification and enforcement of authorization constraints in workflow management systems." *ACM Transactions on Information and System Security (TISSEC)*, vol. 2. no. 1, pp. 65–104, 1999.

[11] M. Roscheisen and T. Winograd, "A communication agreement framework for access/action control," in *Proceedings of the Symposium on Security and Privacy*. IEEE Press, 1996, pp. 154–163.

[12] M. Shanahan, "Prediction is deduction but explanation is abduction," in *Proceedings of IJCAI '89*. Morgan Kaufmann, 1989, pp. 1055–1060.

[13] L. Giuri, "Role-based access control on the web," *ACM Transactions on Information and System Security (TISSEC)*, vol. 4, no. 1, pp. 37–71, 2001.

[14] E. Bertino, S. Castano, and E. Ferrari, "On specifying security policies for Web documents with an XML-based language," in *Proceedings of the Sixth ACM Symposium on Access control models and technologies*. ACM Press, 2001, pp. 57–65.

[15] E. Damiani, S. D. C. di Vimercati, S. Paraboschi, and P. Samarati, "A fine-grained access control system for XML documents," *ACM Transactions on Information and System Security (TISSEC)*, vol. 5. no. 2, pp. 169–202, 2002.

[16] J. B. D. Joshi, W. G. Aref, A. Ghafoor, and E. H. Spafford, "Security models for web-based applications," *Communications of the ACM*, vol. 44, no. 2, pp. 38–44, 2001.

[17] J. Park and R. Sandhu, "Towards usage control models: beyond traditional access control," in *Seventh ACM Symposium on Access Control Models and Technologies*. ACM Press, 2002, pp. 57–64.

Decentralized Credentials

Marius Gjerde* Stig Frode Mjølsnes† Aslak Bakke Buan‡

September 25, 2003

Abstract

A high security architecture employing a mobile electronic wallet was first developed in the seminal European research project CAFE. This paper deals with a generalized model based on this architecture that was recently proposed in [6], where a localized (residential) credential keeper maintains most of the content of the user's wallet, still the wallet is able to perform transactions. However, new threats against the security are introduced by this new model, hence the protocols to be used need careful construction and analysis to maintain strong multiparty security. We consider this problem here, by analyzing the new architecture of decentralized credentials, by proposing a new and efficient identification protocol, and by giving arguments for the claimed security properties hold under this new model.

Keywords: Mobile commerce, e-wallet architecture, privacy, payment protocols, digital credentials.

1 Introduction

The seminal European research project CAFE [2] developed technology and a working pilot of the concept of an electronic wallet. The project's main concern and target were payment transactions that could replace traditional cash by digital coins [5]. The electronic wallet contained currency and credentials, and provided users the ability to perform

*Diploma thesis [3], work carried out spring 2003 at the Department of Mathematical Sciences, NTNU.
†Department of Telematics, NTNU
‡Department of Mathematical Sciences, NTNU

payment transactions *offline*, ie. without contacting the bank during the payment transactions. Essential security requirements for the CAFE architecture were privacy for the user, and integrity for the service organizations. During payments, a user should be totally anonymous, and it should be impossible to link a user to a specific transaction, even if all other parties collaborated. Given several views of protocol executions, it should even be impossible to link the same (anonymous) user to two different transactions. This property is called *unlinkability*.

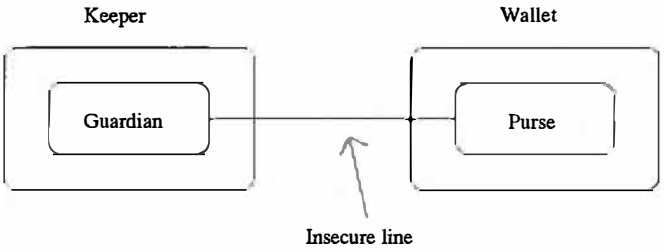


Figure 1: Segregation of guardian and wallet

Motivation This paper covers some results from diploma thesis work "Decentralized Credentials" [3], where the outset was the article "Online e-wallet system with decentralized credential keepers" [6] that generalizes the CAFE architecture to include *online* wallet transactions. A main idea was to separate the credentials from the electronic wallet within a fully decentralized architecture. This decentralizing requires the system to be online, as opposed to the offline system of CAFE. A major scalability challenge for most wallet schemes is the problem of multi-issuer. In practice, every service provider and credential issuer provide various types of physical tamper resistant tokens, such as smart cards, that contains and secures secret information on behalf of the service provider. It follows that users either have to juggle the cards in and out of the electronic wallet, or accommodate an ever growing number of smart cards by introducing new card slots. The introduction of a decentralized credential keeper provides a possible solution to this problem, as the keeper easily can provide a large number of card slots within its physical constraints.

The system of decentralized credentials, however, introduced some major threats against security in the architecture. In CAFE, the communication channel between the wallet and a service (ie. bank or shop) would be insecure, while the new system introduced another insecure communication channel between the wallet and the decentralized credential keeper. In turn, this introduced challenges concerning identification of the wallet to the keeper, and information leaking, ie. subliminal channels [8, 9], between the keeper and some service.

The possible applications of the digital wallet are not limited to payment transactions [2]. It could also be used for identification, digital signatures and key agreement protocols. Some obvious applications are accessing doors and signing receipts. Other possibilities are

starting cars and logging into computers.

2 The model

Mjølsnes and Rong [6] gave a conceptual introduction to the new architecture of decentralized credentials. Before introducing a new identification protocol, we have to give a detailed description of the system's composition.

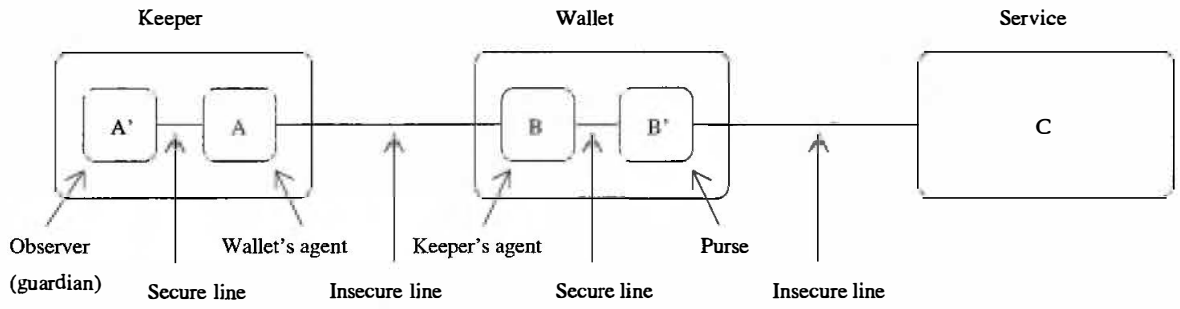


Figure 2: Decentralized credentials.

As stated, we assume the communication channels between the credential keeper and the wallet, and between the wallet and some service, to be insecure. On the other hand, the inner communication channels in the wallet and the credential keeper are assumed secure.

Keeper The credential keeper contains and protects the credential hardware and software at a fixed location. It will consist of two parts, where one part contains the actual credentials, and is called the *observer*. The second part of the keeper is needed to identify the wallet, and will contain some secret key needed to accomplish this. The device is assumed to be either tamper-proof or placed in a secured area in order to protect the secret key. This device is called the *wallet's agent*. There are two scenarios to identify the wallet to the keeper:

1. **Public key techniques** The keeper and the wallet each holds a public/private key pair, and identifies each other through the use of these.
2. **Shared secret** The keeper and the wallet share some symmetric secret, which can be used to identify each other.

Here, we only consider the scenario of a *shared secret*. By this, we get a one-to-one relation between the wallet and the wallet's agent, and thus exclude the ability of the wallet to

contact arbitrary credential keepers to perform some action.

Wallet The wallet will also have to hide the secret shared with the wallet's agent, and therefore consists of a tamper-proof device as well. This device is called the *keeper's agent.* The wallet also consists of a user-controlled device, used to download protocol instructions from the different credential suppliers. This device is called the *purse.* It could be possible to eliminate the tamper-proof keeper's agent from the wallet, and generate the secret key by input from a user. This, however, will make the user give a very long pass-phrase as input to generate enough entropy for a secret key, and would probably be unpractical if similar security level is required.

An overview of the system is sketched in Figure 2. In the figure, the different parts of the system are given short names, which we will continue using to make printing easier. Thus, the observer is from now on A', the wallet's agent is A, the keeper's agent is B, the purse is B' and the service is C.

Trust The observer A' is issued by, or at least trusted by the service C. The user trusts the issuer of the wallet's agent A and keeper's agent B, and therefore trusts these parts of the system, as well as the purse B', controlled by himself.

Subliminal channels A very important requirement for the protocol construction is to preclude information leakage between A' and C. With the introduction of *subliminal* (covert) channels [8, 9], Simmons showed that such information leakage could be present without A/B/B' being able to notice it. By this, the new protocol is constructed to preclude subliminal channels, which eliminates the possibility of information leakage from C to A' (*inflow*) and from A' to C (*outflow*).

3 The identification protocol

We now start describing the identification protocol for the new scheme of decentralized credentials[1]. The new protocol is an extension of the Schnorr identification protocol [7], and hence security is based on the complexity of calculating discrete logarithms in multiplicative cyclic groups [4].

[1] In the diploma thesis work [3], several other protocols are constructed. These include signature generation, key agreement and payment in the architecture.

Setup In the setup of the protocol, we first select two large primes p and q, such that q divides $p-1$. Let G be the cyclic group $\mathbb{Z}_q^*$ and fix a generator g for G. The observer A', or more precisely, one of the smart cards in the observer A', will hold a private key parameter $\mathcal{S}$, with a corresponding public key parameter $\mathcal{P} = g^{\mathcal{S}} \bmod p$. The parameter $\mathcal{P}$ then represents the claimed identity (name) of A', where $\mathcal{S}$ is the trap-door information for A' to validate this claim. The public parameters $(p, q, g, \mathcal{P})$ are assumed known to all parties in the protocol.

We next assume the wallet's agent A and the keeper's agent B share a common secret k. The size of k will correspond to the size of p and q. In addition, we define a message authentication code $\mathcal{H}_k : \{0,1\}^* \rightarrow \{0,1\}^q$, with the common secret k as input. We also define a collision-resistant hash function [4] $\mathcal{H} : \{0,1\}^* \dashrightarrow \{0,1\}^l$, where $l = 2^t$.

The protocol In the protocol, the wallet B/B' will identify itself to a service C with the help of the keeper A/A'.

The protocol starts by the observer A' generating a witness w, which it sends to the user trusted device A. The witness is then randomized to preclude outflow in the protocol, and sent to the service C through the wallet. Together with the randomized witness $\bar{w}$, A sends a commitment to a variable a_2, and by this starts a coin-flipping protocol with C. The coin-flipping protocol will randomize the challenge given by C in such a way that neither A nor C will control the output. This precludes inflow to A' through the challenge.

On the receival of the (not yet randomized) challenge c from C, the wallet is still not identified by A. Hence, B uses its secret common key k to generate a message authentication code (MAC) $\mathcal{H}_k(\bar{w})$ on the randomized witness it received from A. The randomized witness here serves as a challenge to B in its identification to A. After A has received the challenge and the MAC from B, it uses its secret common key k to check that B is valid, before randomizing the challenge c to $\bar{c}$. A then sends the randomized challenge to A', which uses its private key $\mathcal{S}$ to generate a response on the challenge $\bar{c}$. On receival of the response r, A randomizes it to $\bar{r}$, according to the randomizing of the witness w earlier in the protocol. A then sends the randomized response $\bar{r}$ to C. At the same time, A finishes the coin-flipping protocol with C by sending a_2 to C.

The protocol is shown in Table 1. In the table, the tamper-proof device B and the user-controlled part B' of the wallet are put together, as they trust each other and work as one (B will contain the secret key k, which is unknown to B'). For simplicity, we omit mod-endings from the table. By doing most of the calculation in the keeper, the protocol gains efficiency, as the wallet holds limited computing resources.

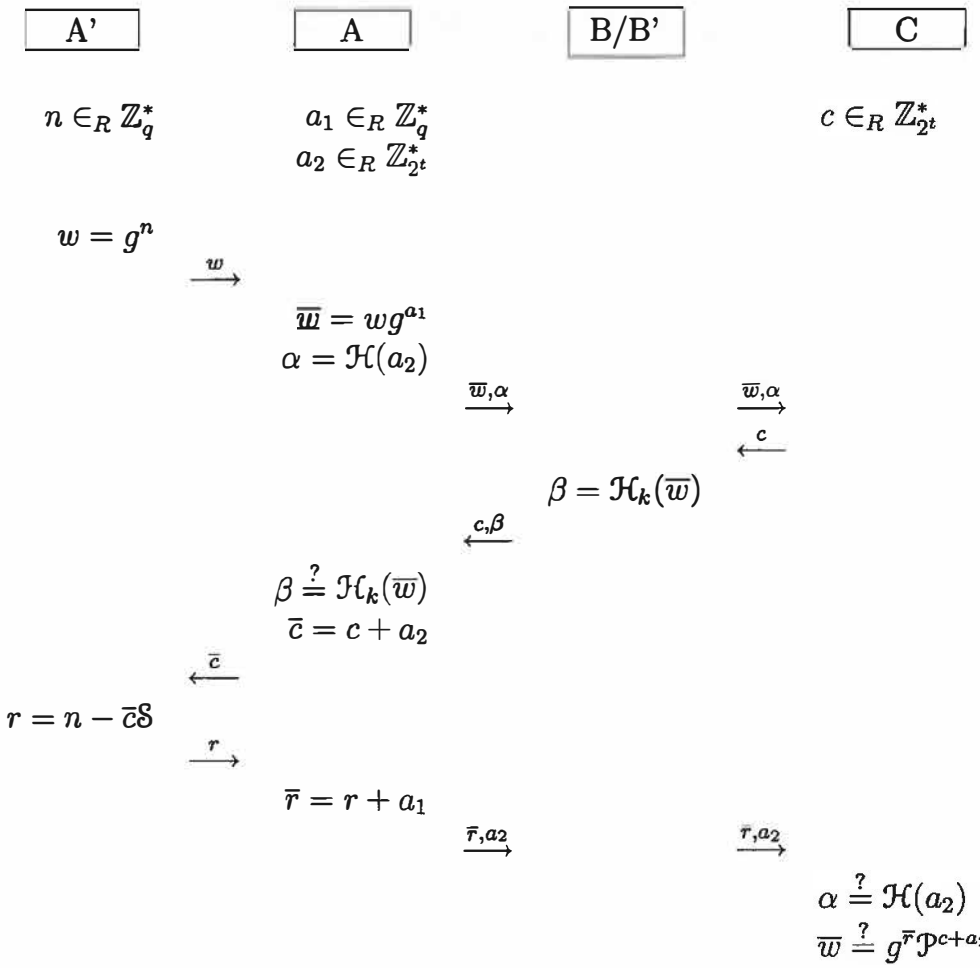


Table 1: Identification protocol for decentralized credentials. The identity (name) is represented by the public parameter $\mathcal{P}$.

4 Security

We now discuss the security of the new identification scheme. Our protocol is an adapted version of Schnorr identification scheme. We introduce the following modifications:

1. Randomization of the witness w.
2. Randomization of the challenge c.
3. The introduction of the wallet B/B' as an identifying 'man-in-the-middle'.

We now analyze the security aspects of introducing these modifications.

Randomization of the witness w. In the protocol, A gets a witness w from A', which A randomizes to $\bar{w}$ by using a random variable a_1.

First, we note that A needs to know the variable a_1 for C to validate in the protocol. In the final step, A randomizes the response r from A' by using a_1. If A does not know a_1, he will have to get $\bar{r}$ from the equation $\bar{w} \equiv g^{\bar{r}}\mathcal{P}^{\bar{c}} \bmod p$, which means that A has to solve a discrete logarithm problem.

Since A can choose a_1 *after* receiving w, A can mount an attack resulting in the private key $\mathcal{S}$ of A'. The attack can be based on the following property: If A gets hold of two protocol views with the same randomized witness $\bar{w}$, but with different challenge-response pairs $(\bar{c}_1, \bar{r}_1)$ and $(\bar{c}_2, \bar{r}_2)$, A gets the following equation:

$$\bar{w} \equiv g^{\bar{r}_1}\mathcal{P}^{c_1} \equiv g^{r_1}(g^{\mathcal{S}})^{c_1} \equiv g^{r_2}\mathcal{P}^{c_2} \equiv g^{r_2}(g^{\mathcal{S}})^{c_2} \bmod p,$$

from where A can calculate $\mathcal{S}$ by:

$$\mathcal{S} = (r_1 - r_2)(c_2 - c_1)^{-1} \bmod q.$$

To mount the attack, A has to store prior protocol executions with $(\bar{w}_i, \bar{c}_i, \bar{r}_i)$ in a big matrix. Here, $i \in \{1, \ldots, n\}$, where n are the number of prior executions of the protocol. At the receival of a witness w from A', A can try to find a $\tilde{w} = \bar{w}_i$, $i \in \{1, \ldots, n\}$, by iterating $\tilde{w} = wg^{a_1} \bmod p$ with different values of a_1. The probability of finding such a $\tilde{w}$ is $n/|\langle g \rangle|$, where $|\langle g \rangle| = 2^q$. By this, we see that the attack is unpractical.

Randomization of the challenge c. In the protocol, this randomization is performed by a coin-flipping protocol between A and C. If the hash-function $\mathcal{H}$ is one-way (the hiding property), then C would get no information about a_2 through α. Subsequently, if the hash-function $\mathcal{H}$ is collision-resistant (the binding property), then A will not be able to generate two different variables a_2, giving the same α. If we assume $\mathcal{H}$ is computationally hiding and binding, and that A and C does not hold unlimited computing power, the randomization of the challenge c gives no security flaws to the new protocol.

The introduction of the wallet B/B' as an identifying 'man-in-the-middle'. To preclude unauthorized wallets being identified as A' by the service C, the wallet B/B' will have to be identified by A. In the protocol, this is done by using the ideas of the protocol SKID 2 from ISO/IEC 9798-4 [1]. The parameters used by SKID2, the shared key k and the randomized witness $\bar{w}$, will in practical solutions be made large enough to preclude attacks against this protocol.

5 Conclusion and Future Work

We have presented a new identification protocol, adapted to the architecture of *decentralized credentials*, and argued for its security. The major advantage of the protocol is that the amount of computation done by the electronic wallet is kept to a minimum. This makes the protocol efficient, in the case where the electronic wallet has limited computing resources. The protocol also precludes inflow and outflow between the observer and the service, which gives a user the privacy of contacting whichever service he wants, without the observer getting any information about the service during protocol execution.

The security of the new identification scheme remains to be formally proven.

References

[1] ISO/IEC 9798-4. Information technology - security techniques - entity authentication - part 4: Mechanisms using a cryptographic check function. *International Organization for Standardization, Geneva, Switzerland*, 1995.

[2] Jean-Paul Boly, Antoon Bosselaers, Ronald Cramer, Rolf Michelsen, Stig Fr. Mjølsnes, Frank Müller, Torben P. Pedersen, Birgit Pfitzmann, Peter de Rooij, Berry Schoenmakers, Matthias Schunter, Luc Vallee, and Michael Waidner. The ESPRIT project CAFE - high security digital payment systems. In *ESORICS*, pages 217–230, 1994.

[3] Marius Gjerde. Decentralized credentials. Master's thesis, NTNU, Dept. of Mathematical Sciences, june 2003.

[4] A. J. Menezes, P. C. van Oorshot, and S. A. Vanstone. *Handbook of applied cryptography.* CRC Press, 1997.

[5] Stig F. Mjølsnes. Digital currency by cryptographic protocols. In R. Conradi, editor, *Norsk Informatikk Konferanse - NIK'89*, pages 139–157. Tapir Academic Publisher, 1989.

[6] Stig Frode Mjølsnes and Chunming Rong. On-line e-wallet system with decentralized credential keepers. *Mobile Networks and Applications*, 8(1):87–99, 2003.

[7] C. P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, pages 161–174, 1991.

[8] Gustavus Simmons. The prisoners problem and the subliminal channel. In *CRYPTO '83, Santa Barbara, CA*, pages 51–67, 1984.

[9] Gustavus Simmons. The subliminal channel and digital signatures. In *Advances in Cryptology - EUROCRYPT '84*, pages 364–378, 1985.

Privacy-Preserving Spatially Aware Authentication Protocols: Analysis and Solutions

Geir M. Køien and Vladimir A. Oleshchuk
Department of Information and Communication Technology
Agder University College
N-4876 Grimstad, Norway

Abstract

Location independence is an inherent property of the subscribers in wireless/cellular networks. In this paper we examine an authentication protocol which takes into account the spatial coordinates of the subscribers. Such protocols can be useful security policy tools, but they are problematic with respect to the spatial privacy rights of the principals. We investigate the use of secure multi-party computational geometry algorithms/protocols for the purpose of privacy preserving of spatial data in the context of authentication protocols in a mobile environment.

Key words: Location privacy, secure multi-party problem, mobile systems, authentication.

1 Introduction

In this paper we will investigate a spatial extension to the typical Authentication and Key Agreement (AKA) protocols found in mobile cellular networks. We therefore assume a general model with the following principal entities:

- a Home Domain (HD) controlling the subscribers
- a Serving Network (SN) with min. one associated Access Network (AN)
- the subscriber; identified by the User Equipment (UE)

The AKA extension will focus on enhancing the policy control aspects of the AKA protocol by placing spatial restriction on the execution of the protocol. There are various ways of providing a spatial control dimension to an AKA protocol. The obvious solution, which simply distributes location information to the controlling entities, suffers from inadequate location privacy. For publicly operated mobile networks it is realistic to expect regulatory requirements to maintain a reasonable degree of location privacy for the subscribers. Therefore, a Spatial AKA (SAKA) mechanism must simultaneously be able to achieve the seemingly contradictory goals of enforcing spatial control while still preserving the location privacy of the subscribers.

2 The Need for Spatial Verification

2.1 Entity Authentication

The purpose of peer entity authentication is to provide corroboration that the claimed identity of the peer is a true identity of the peer. Authentication is normally not an end to itself. It is usually better understood as a security context set-up procedure for subsequent secure communication between the involved entities. While the cryptographic basis for the authentication algorithms may be based on either symmetric or asymmetric (public-key) algorithms, the subsequent secure communication is most likely protected by symmetric crypto-algorithms. The resulting key material is intended to exist only for the duration of the ensuing secured session. Sessions may be long lived and the principals may periodically re-authenticate themselves to renew the security context.

2.2 Authentication in Cellular Systems

Authentication in a cellular network like UMTS is based on a model where the UE and the AN[1] have a defined trust relationship with the HD. What one wants to achieve in the scenario presented in fig 1 is to have mutual entity authentication between the UE and the AN/HD and to distribute/generate session keys (UE - AN). In cellular networks like the UMTS system, the trust relationships are not completely symmetrical. The UE is a subscriber to services at a cellular operator (HD). User identity and the user security credentials are normally unilaterally decided by the HD. The relationship between HD and AN/SN is regulated through a roaming agreement contract. In the context of the SAKA mechanism we shall assume that the UE does not trust the HD with respect to privacy of location data.

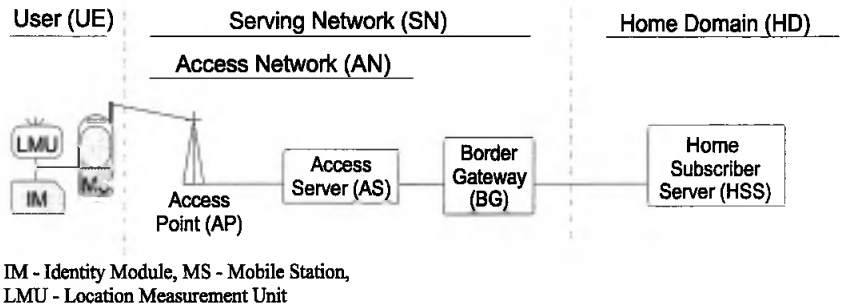


Figure 1: A generic mobile system (cellular/wireless) architecture

2.3 Access Security

Authentication can take place on different levels and for different purposes. The UMTS AKA mechanism and associated encryption/integrity protection are specifically designed to provide access security implemented at the link layer [1]. In general we have two distinct cases for access security:

[1] We use the term *Access Network* to denote both AN and SN interchangably.

- end-to-end security between the UE and the HD (transparent to AN)
- the HD provides an *authentication centre* function; the *challenge-response* protocol is executed between the UE and AN; subsequent encryption/integrity protection is established for protection of the connection between UE and AN

The last case is reminiscent of the model used in the 3GPP-WLAN interworking specifications (Fig. 2) [2, 3]. We shall focus our attention at this last model, but remain agnostic with regard to the question of where the encryption/integrity protection is terminated in the network.

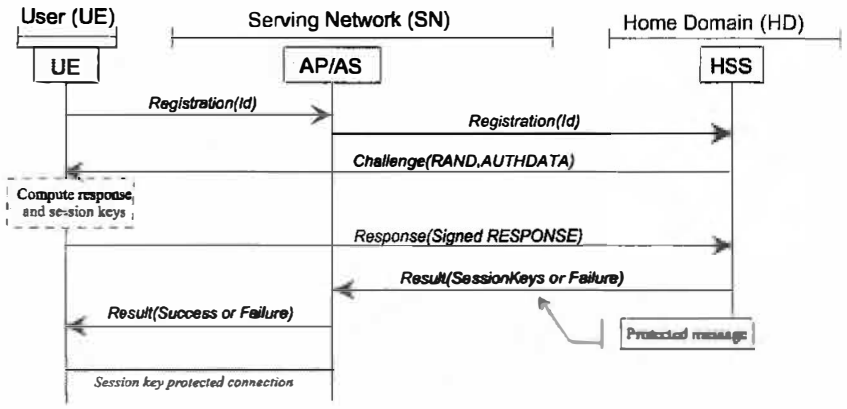


Figure 2: Simplified 3GPP-WLAN AKA architecture

2.4 The User Equipment

In Fig. 1 the User Equipment (UE) is depicted as consisting of a Mobile Station (MS) and an Identity Module (IM). Some mobile systems keeps the subscription data and the security credentials on a separate Identity Module implemented on a smartcard. The IM is an entity owned by the HD and is for practical purposes assumed to be an entity trusted by both the subscriber and the HD. We shall assume that the subscriber trusts the UE, including the Location Measurement Unit (LMU). A UE may become compromised, but for the purpose of our analysis we shall assume that the subscriber have reason to trust the UE. We further assume that the UEs always has an LMU. The UEs will thus be capable of independently determining its own position. We shall remain agnostic with respect to the inner workings of the LMU, but shall insist that it can establish the position independently of the mobile system infrastructure.

2.5 Exposure Control Dimensions for Mobile Systems

The AKA procedure establishes or renews a security context by authenticating the entities and providing fresh key material. The key material will from then on be exposed through usage and through being stored at the principals. Exposure control in typical systems focus on usage exposure control and temporal exposure control. For instance, in the IPsec protocol suite the *lifetime* parameter of

the Security Associations [4] can be defined both in terms of usage, measured in Kilobytes protected with the key, and temporal exposure, measured in seconds since the negotiation. The validity of the authentication and the associated key material is then by design tied to the *lifetime* intervals. For a mobile system one can add a spatial dimension to the existing exposure control dimensions. We then have three primary dimensions to exposure control:

- Usage exposure (KBytes)
- Temporal exposure (seconds)
- Spatial exposure (area)

So what should be the rationale for adding a spatial exposure dimension to mobile authentication protocols? Immediately we observe that the characteristics of the physical environment will differ depending on geographical location. The physical radio environment will obviously fluctuate, but other aspects will also vary with position. One aspect that will vary is the risks and threats from the environment. While it is obvious that the exposure of key material stored at the principals is related to the passing of time, it is not obvious that this is a linear function independent of the changing risk exposure from the environment. Some environments are clearly more hostile than others and the risks and threats present to the principal are relative to the location. A similar argument can be made for the usage exposure. In some environments the chances for active and/or passive attacks against the protected communication are almost negligible, in other environments the risks are substantial and cannot be ignored. So it is clear that location is a factor when deciding on the exposure risks one run. One important aspect of the exposure control is how to measure the exposure. We observe that:

- *Usage exposure:* a UE can easily, independently and inexpensively measure and establish the exact number of bytes/packets transmitted or received.
- *Temporal exposure:* a UE can easily, independently and inexpensively measure the passing of time.
- *Spatial exposure:* a UE cannot establish its own position without assistance either from the mobile system or from some out-of-band system like for instance the Global Positioning System (GPS). The measurements require additional equipment and will have limited resolution regarding both spatial accuracy and update frequency.

3 Considerations for Spatially Dependent Authentication Protocols

3.1 Spatial Access Control at Which Layer?

3.1.1 Link layer and Network Layer

One can execute the spatial access control in conjunction with the AKA protocol before the network layer has been established and presented to the principal.

The service would then be pervasive and apply to all user-plane communication between the principal and the access network. The link layer will in many cases be an appropriate choice for a SAKA mechanism, but implementing the mechanism at the link layer comes at the expense of binding the mechanism to a particular access technology. Although one have major success stories for homogeneous access technologies (like GSM), it seems undesirable to limit oneself to a particular link layer technology. There are also good reasons to implement a SAKA mechanism at the network layer. The general technology trend seems to be towards generic core networks with multiple access networks attached. The lowest layer common denominator for such network constellations will be the network layer.

Users have come to expect almost instant session establishment, although this expectation is mainly held for circuits switched telephony services (speech). In modern mobile systems one often have small cell structures and one must expect the users to be constantly on the move. Therefore it is extremely important that the connection is (re-)established quickly, otherwise the services will be perceived to be of low quality. So a SAKA procedure cannot be allowed to take too long time to complete or be executed too frequently if it is to be implemented at the link- or network layer.

3.1.2 Concerning the Transport Layer and the Application Layer

In this paper we will only investigate spatial control in conjunction with the AKA access security mechanism. However, SAKA protocols may also be useful for entities at higher layers. Here the requirements need not be as stringent with respect to set-up times as those found on the link/network layer. With substantially different design requirements and priorities, it's not obvious that a SAKA mechanism is the most appropriate at the higher layers. Depending upon the requirements one may find that a Spatial Role Based Access Control (SRBAC) mechanism is better suited for this purpose [5].

3.2 Spatial Exposure Control and Resource Usage

3.2.1 Required number of signaling round-trips

The number of signaling round-trips that must be completed before a connection is established is a very significant source of delays. This is true for local round-trips in the access (UE-AP/AS) and even more pronounced for round-trips involving the home domain (UE-HSS). The signaling delays are mainly composed of buffering/processing delays and of transmission delays. The transmission delays are a function of the signal propagation rate of the physical medium and there is no room for improvement here unless one can make the traveled distance shorter. That would require smaller cells and a tighter and more hierarchical infrastructure, but the infrastructure costs increases rapidly with deployment of a dense infrastructure. Still, even for a dense infrastructure one should still focus on reducing the number of signaling round-trips for the connection set-up to an absolute minium.

3.2.2 Bandwidth usage

All access networks have bandwidth limitations. This is particularly apparent for radio access networks, which may have acute restrictions on bandwidth during the access signaling procedure. System signaling additionally requires an error-free data channel even in the presence of adverse radio conditions. For the set-up procedure or handover events one cannot tolerate extensive delays, and thus error correction by retransmission is not generally applicable. Forward error correction (FEC) methods must therefore be used. Such methods typically employ a combination of block coding and convolutional coding. The data expansion is typically at 2-3 times the original data[2]. For future mobile radiosystems the bandwidth limitation may be less problematic. It would then be a useful optimization strategy to run parts of the access signaling in parallel. Such optimization would be most efficient if it can save complete signaling round-trips. At the other end of the spectrum (pun intended), one may have to serialize parts of the access procedures due to link layer frame[3] size limitations etc. For systems like GSM/GPRS the MTU size restriction imposes real problems.

3.2.3 Computational complexity

Authentication invariably involves cryptographic operations. Some of these operations are costly in terms of the required processing power and processing delays. We shall be most concerned with the delays, but we observe that higher processing demands will require more powerful processors and high energy consumption. The privacy preserving algorithms tend to use a number of public-key cryptographic operations. This puts restrictions on the usage of the privacy preserving algorithms. We ideally want to execute these algorithms on the IM, but in the short term that may not feasible. However, Moore's "law" of doubling of computational power every 18 months seems still to hold, and we are therefore optimistic with respect to the possibility of running the privacy preserving crypto-algorithms on the IM in the near future.

3.3 Limitations on Spatial Control

3.3.1 Spatial Resolution

A spatial exposure mechanism must take into account the dynamic behavior of the principal, cell size and cell cluster size, the capabilities of the UE/AN for determining the location (resolution and update frequency), computational limitations, the complexity of the geometric shapes and obviously also the intended usage of the spatial control function. One does not want to require a too high measurement frequency or to impose artificial upper bounds on the traveling speed of the subscribers. The SAKA protocol may potentially be executed for every cell change event, but high capacity systems tend to have very small cell sizes and thus this may be impractical. Realistically, re-authentication due to position change events will only occur when one leaves or enters a location area. The UE may therefore be able to move a considerable distance after the spatial

[2]For GSM the expansion is from 184 bits to 456 bits (approx. 2.5 x increase)
[3]Often called the Message Transfer Unit (MTU) size

verification until the next spatial verification event. In practice one must therefore allow the spatial verification to be valid for an area and/or a period of time. The UE may periodically check that the current position does not diverge with more than a defined distance from the SAKA point. For sake of simplicity in the analysis we shall assume that the *validity area* (VA) has circle shape. If a SAKA event occurred at position z, then the area VA is defined by the radius r from z. Given this, what would be the maximum practical resolution we can have? The question can be translated into what would be the smallest VA size that can be enforced. To start with we assume that the UE position is updated with a frequency of one measurement per second. We also assume a maximum subscriber speed of 360 km/h. This gives a maximum change in position of at most 100 meter between two consecutive measurements. We must also consider the measurement accuracy of the position. For the GPS system we will have a measurement error of approximately ± 10 meters if we assume that differential GPS (dGPS) is not available. This means that a VA cannot be smaller than a circle with a radius r of 120 meters. For cases where the mobility is more pedestrian one can attain better precision. Given a maximum UE velocity of 36 km/h, the UE will only move 10 meters between measurements. Given the same measurement error, the VA would be as small as 30 meters. For a case with maximum velocity of 36 km/h, use of differential GPS (± 1 meter) and a measurement rate of 4 times as second, one get a VA radius r of less then 5 meters. We note that one in general cannot assume that subscribers are slow moving.

3.3.2 Availability of Spatial Information

Location information may not be instantly available. For instance, location data from the GPS system requires a clear view of the sky. So we may have situations where the UE is not able to verify/obtain position data for a prolonged period of time. The most problematic situation is when location information is not available at the time of SAKA execution. If the position data has expired one cannot execute the spatial verification. It will be a matter of policy whether the access can then be granted. Somewhat less problematic is the case where the periodic verification with respect to inclusion in the VA cannot be run due to expired or lacking spatial information. A grace period may be called for. Either way, a decision procedure for these cases must be defined.

4 Location Inclusion and Privacy-Preserving

4.1 The Point Inclusion Problem

The problem we are going to study in this section is known as the *point inclusion problem* [6, 7]. It can be defined as following: Alice has a point z and Bob has a polygon P. We need to find an algorithm to determine whether z is located within the polygon P such that Alice does not have to disclose any information about the point z to Bob and that Bob does not have to disclose information on the polygon P to Alice. The only fact to be revealed is the solution to the problem. The problem is a case of the a secure multi-party computation problem and can be solved using a circuit evaluation protocol [9]. However this solution is infeasible because of high communication complexity. Specialized solutions to

the problem has been proposed in the literature [6–8]. They are more efficient but still impractical to be used in the framework described in this paper with respect to consumed computational resources, the number of signaling rounds trips required, and the volume of data exchanged.

4.2 Secure Two-Party Location Inclusion Protocol based on homomorphic public-key cryptosystems

In this section we propose a secure two-party privacy-preserving protocol that has lower communication complexity than described in literature [6–8]. As mentioned in [8], efficient solutions for 2-party model are hard to find. We want to investigate whether practical solutions for the models described above can be achieved.

Let us select a public-key cryptosystem with homomorphic property where encryption and decryption are denoted as $E(\bullet)$ and $D(\bullet)$ respectively. That is, there is an operation on encrypted data, denoted $\oplus$, that can be used to perform addition of the data without decryption: $E(x) \oplus E(y) = E(x+y)$. Many such systems have been proposed in the literature [11–13]. Further, since $E(x) \oplus E(y) = E(x+y)$ then $E(2x) = \underbrace{E(x) \oplus E(x)}_{2}$ and $E(xy) = \underbrace{E(x) \oplus \cdots \oplus E(x)}_{y}$. So we can multiply encrypted data if one of the multipliers is known. This attribute is desirable and it will be used in S2PLIP protocol proposed later in this section. To simplify further notation and make our protocol independent of particular selected homomorphic public-key cryptosystem we assume that operations $\oplus$ and $\otimes$ on encrypted data are defined as following:

$$\begin{aligned} E(x) \oplus E(y) &= E(x+y) \\ E(x) \otimes E(y) &= \underbrace{E(xy) = E(x) \oplus \cdots \oplus E(x)}_{y} \end{aligned}$$

Note that in the second equation we need one of multipliers in decrypted form.

Let polygon P be presented as a set of functions $\{f_i(x,y) \,|\, i = 1, 2, ..., n\}$ where $f_i(x,y) = 0$ represents the equation of the line boundary of the polygon P. We can also assume that functions $\{f_i(x,y) \,|\, i = 1, 2, ..., m\}$ represent the edges of the lower part of the boundary and $\{f_i(x,y) \,|\, i = m+1, ..., n\}$ represent the edges of the upper part of the boundary. Therefore a location $z = (\alpha, \beta)$ is inside the polygon P if $f_i(\alpha,\beta) > 0$, $i = 1, 2, ..., m$ and $f_i(\alpha,\beta) < 0$, $i = m+1, ..., n$. Further in the paper we assume that the polygon $P = \{g_i(x,y) \,|\, i = 1, ..., n\}$, where

$$g_i(x,y) = \begin{cases} f_i(x,y) & \text{for } i = 1, 2, ..., m \\ -f_i(x,y) & \text{for } i = m+1, ..., n \end{cases}$$

Thus location $z = (\alpha, \beta)$ is inside P if and only if $g_i(\alpha,\beta) > 0$ for all $i = 1, 2, ..., n$. It is easy to see that $g_i(x,y) = a_i x + b_i y = (a_i, b_i) \cdot (x,y)$, that is $g_i(\alpha,\beta)$ can be calculated as a scalar product of (a_i, b_i) and (α,β). Therefore $g_i(x,y)$ can be defined by (a_i, b_i).

Given that Alice (UE) knows her location z and Bob (HD) knows polygon P, we want to solve the *point inclusion problem* such that Bob will know the answer at the end of the procedure.

Protocol S2PLIP (*Secure Two-Party Location Inclusion Protocol*)
Inputs: Alice has a location $z = (\alpha, \beta)$, and Bob has a polygon $P = \{(a_i, b_i) \,|i = 1, 2, ..., n\}$; Alice and Bob use the same homomorphic public-key cryptosystem (E, D).
Outputs: Bob gets information whether z is inside P without knowing more about z and without disclosing P to Alice.

1. Bob generates a key pair for a homomorphic public-key cryptosystem and sends the public key to Alice.
2. Bob sends encrypted polygon $E(P) = \{(E(a_i), E(b_i)) \,|i = 1, ..., n\}$ and $E(\bullet)$ to Alice.
3. Alice calculates encrypted values of $E(g_i(\alpha, \beta))$ without decrypting g_i and encrypting z as following:

$$\begin{aligned} E(g_i(\alpha,\beta)) &= (E(a_i), E_k(b_i)) \cdot (E(\alpha), E_k(\beta)) \\ &= E(a_i) \otimes E(\alpha) \oplus E(b_i) \otimes E(\beta) \\ &= E(a_i\alpha) \oplus E(b_i\beta) \\ &= E(a_i\alpha + b_i\beta) = r_i \end{aligned}$$

   Note that α, β are known to Alice but not to Bob. The result is $\{r_1, r_2, ..., r_n\}$.
4. Alice generates a random value v, calculates $\widehat{v} = (r_{i_1} \oplus E(v))$, and asks Bob to decrypt $\widehat{v} = E(g_{i_1}(\alpha, \beta)) \oplus E(v) = E(g_{i_1}(\alpha, \beta) + v)$
5. Bob returns $D(\widehat{v}) = D(E(g_{i_1}(\alpha, \beta) + v)) = g_{i_1}(\alpha, \beta) + v$ to Alice
6. Alice calculates $D(r_{i_1}) = g_{i_1}(\alpha, \beta) = D(E(g_{i_1}(\alpha, \beta) + v)) - v$
7. Alice permutes $r_1, r_2, ..., r_n$ into $r_{i_1}, r_{i_2}, ..., r_{i_n}$, finds $D(r_{i_1})$ and calculates

$$\begin{aligned} e_1 &= r_{i_1} + r_{i_2} + ... + r_{i_n} \\ &= E(g_{i_1}(\alpha,\beta)) + E(g_{i_2}(\alpha,\beta)) + ... + E(g_{i_n}(\alpha,\beta)) \\ &= E(g_{i_1}(\alpha,\beta) + g_{i_2}(\alpha,\beta) + ... + g_{i_n}(\alpha,\beta)) \end{aligned}$$

   and

$$\begin{aligned} e_j &= D(r_{i_1}) r_{i_j} = D(E(g_{i_1}(\alpha,\beta))) E(g_{i_j}(\alpha,\beta)) \\ &= g_{i_1}(\alpha,\beta) E(g_{i_j}(\alpha,\beta)) \\ &= E(g_{i_1}(\alpha,\beta) g_{i_j}(\alpha,\beta)), \text{ for } j = 2, ..., n \end{aligned}$$

8. Alice permutes $e_1, e_2, ..., e_n$ into $e_{i_1}, e_{i_2}, ..., e_{i_n}$, and sends its to Bob.
9. Bob decrypts $e_{i_1}, e_{i_2}, ..., e_{i_n}$ and concludes that Alice is inside the polygon P if all decrypted values are positive, that is $D(e_{i_j}) > 0$, for all $j = 1, .., n$.
10. Security and Complexity Analysis

4.2.1 Security and Communication Complexity

In the above protocol, Bob sends the encrypted polygon P to Alice. Alice will never disclose her position z to Bob. Let us evaluate communication cost of this protocol. We assume that polygon P has n angles and selected cryptosystem has l bits keys. We assume, for simplicity, that each coefficient can be presented as l bits number. The following protocol steps will affect the communication complexity:

1. Bob sends encrypted P and public key to Alice. ($2nl$ bits are sent and communication complexity is $2n$)
2. Alice asks Bob to help to decrypt $g_{i_1}(\alpha, \beta)$ ($2l$ bits are sent and communication complexity is 1)
3. Alice sends result $e_{i_1}, e_{i_2}, ..., e_{i_n}$ to Bob ($2nl$ bits are sent and communication complexity is $2n$)

Thus, communication complexity is $4n + 1$ or $O(n)$, and no more then $2l(2n + 1)$ bits need to be sent between Alice and Bob. We should note that the best algorithm for point location within polygon is $O(\log n)$. However it doesn't take into account privacy concerns. We can compare our solution with other proposed in the literature [6]. Secure two-party point-inclusion protocol proposed in [6] has computational complexity $O(n)$. However in the setting considered here the communicational complexity is a bottle-neck. Analyzing the communicational complexity of that protocol we can see that protocol utilizes Secure Two-Party Scalar Product Protocol (S2PSPP) and Secure Two-Party Vector Dominance Protocol (S2PVDP) (see [6] for more details).

The more efficient S2PSPP for smaller n has communication complexity of $4nm$ where m is a security parameter such that n^m is large enough. It is clear that our protocol is more efficient then S2PSPP. But we should remember that in addition to S2PSPP we must use also S2PVDP. S2PVDP involves amongst others the Yao's Millionaire Comparison Protocol [14] which has communication complexity that is exponential in the number of bits of the involved numbers. By involving untrusted third party (which may misbehave) the communication complexity can be improved to $O(l)$ where l is the number of bits of each input number.

The communication inefficiency of the previously proposed solution has been acknowledged in the literature [7], and new modified solution with improved performance has been proposed [8]. The idea is that users can accept weaker security for the sake of better performance. The proposed secure scalar product protocol based on commodity-server model [10] has communication cost $4n$, and S2PSPP has communication cost only $2n$ (with significant increasing of computational cost). However to be practical the communication cost of S2PVDP must still be significantly improved.

4.2.2 Analysis of Delay Factors

As has already been discussed the temporal cost of public-key operations will decrease with improved processing power in the UE. The signal propagation delay will on the other hand remain constant unless the travelled signal distance can be made shorter. The only viable optimization strategy at the protocol level

is to reduce the number of round-trips (thus effectively reducing the accumulated travelled signal distance) and making sure that frame (MTU) fragmentation does not occur. The latter problem imposes message size restrictions while the former imposes restrictions on the number of signalling passes.

The S2PLIP protocol, as described in Section 4.2, requires 2,5 round trips (including *acceptance*). The plain vanilla AKA protocol is implemented as a single pass protocol for the successful case (with *acceptance* being implied with the continued signalling sequence). The combined SAKA protocol can similarly be realized with two passes for the successful case (again assuming implicit *acceptance* signalling). For a simple polygon (square or hexagonal) and with key size of for instance 1024 bit, the message size can be kept reasonably low (square yields appox. 1 KByte msg size) and fragmentation should not be a big problem[4].

5 Scenario for Public Mobile Networks

In this section we present a scenario that is fairly typical for a public cellular network. Based on signal propagation delay the AN will be able to determine the approximate distance from the AP to the UE. Advanced radio techniques provide fairly precise direction information and the AN may be able to triangulate the position using multiple APs. It is therefore clear that the UE cannot prevent the AN from determining its approximate position. Thus, a location privacy scheme should focus on preserving UE location privacy with respect to the HD and not the AN. Given this, we shall focus on a model where only the UE and the HD takes part.

Spatial Verification between the UE and the HD:

1. HD initiates the SAKA protocol. (AKA *challenge* and transfer of public part of public-key etc.)
2. UE and HD executes the S2PLIP protocol. UE also computes AKA *response* and transfers it to HD.
3. HD concludes on S2PLIP success only when z is within P. SAKA success depends on both AKA and S2PLIP success.

5.1 Discussion of the Scenario

In the above scenario we only sketch the SAKA protocol. Further work needs to be done in order to bind the AKA and S2PLIP protocols together. A scheme based on the use of the AKA sequence number seems promising. The unique sequence number[5] in the AKA *challenge* together with message authentication can be used to provide the necessary binding between the AKA and S2PLIP protocols to create an integrated SAKA protocol.

We note that the UE and HD need not trust each other with respect to exchange of spatial information. However, there is a requirement that the UE and HD must be honest when executing the S2PLIP protocol. That is, we

[4] Note that 1 KByte will exceed the MTU size of for instance GSM/GPRS

[5] The sequence number is identified by the SEQ information element, which is part of the AUTN parameter in the Authentictaion Vector (AV).

assume that neither the UE nor the HD will attempt to lie or deceive when executing the S2PLIP protocol.

The described scenario will have little operational impact on the architecture except for the required support at the UE and the HSS (at the HD). We observe that more control can be had if the AN also participates in the spatial verification, but then the complexity and system impact will increase correspondingly. A related spatial authentication solution with significantly less system impact has been investigated in [15], but this solution does not posses true privacy-preserving properties.

6 Summary and Conclusion

The S2PLIP protocol developed in this paper has made spatial control and location privacy attainable goals for a spatially dependent authentication protocol. We have demonstrated that reasonably efficient spatial control is possible without compromising location privacy for the subscribers. There are some practical limitations to spatial control and it may not yet be suitable for full scale deployment. Currently the SAKA protocol may be most appropriate when used for specific areas, customers etc. and perhaps only for limited time periods. Finally, we note that more work is required to fully develop the SAKA protocol, both in terms of secure binding of protocol message exchange as well as on specifying a suitable public-key algorithm.

References

[1] G.M. Køien, An introduction to access security in UMTS, To appear in *IEEE Wireless Communications magazine*, February 2004

[2] G.M. Køien and T. Haslestad, Security aspects of 3G-WLAN interworking, To appear in *IEEE Communications magazine*, November 2003

[3] 3GPP, TS 33.234 3G Security; Wireless Local Area Network (WLAN) Interworking Security; (Release 6), *(draft)*, 3GPP, Sophia Antipolis, Valbonne, France, 2003

[4] D. Harkins and D. Carrel, The Internet Key Exchange (IKE), RFC 2409, IETF, November 1998

[5] F.Ø. Hansen and V.A. Oleshchuk, Spatial Role-Based Access Control Model for Wireless Networks. *IEEE Vehicular Technology Conference VTC2003-Fall*, Orlando, USA Oct. 2003.

[6] M.J. Atallah and W. Du, Secure Multy-Party Computational Geometry. In *WADS2001: 7th International Workshop on Algorithms and Data Structures*, pp.165-179, USA, August 8-10, 2001.

[7] W. Du and M.J. Atallah. Secure Multy-Party Computation Problems and Their Applications: A Review and Open Problems. NSPW'01, September 10-13, 2002, pp. 13–21.

[8] W.Du and Z. Zhan, A Practical Approach to Solve Secure Multi-Party Computational Problems. In *Proceedings of New Security Paradigms Workshop*, September 23-26, 2002.

[9] O.Goldreich, S.Micali and A.Wigderson. How to Play Any Mental Game. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing*, pp. 218–229, 1998.

[10] D. Beaver, Commodity-Based Cryptography. In Proceedings of the 29th Annual ACM Symposium on the Theory of Computing. 1997.

[11] D. Naccache and J. Stern. A New Cryptosystem Based on Higher Residues. In *Proceedings of the 5th ACM Conference on Computer and Communication Security*, pp.59–66, 1998.

[12] T. Okamoto and S. Uchiyama. An Efficient Public-Key Cryptosystem as Secure as Factoring. In *Advanced in Cryptography - EUROCRYPT'98*, LNCS 1403, pp. 308–318, 1998.

[13] P. Paillier. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In *EUROCRYPT'99*, LNCS 1592, pp. 223–238, 1999.

[14] A.C. Yao. Protocols for Secure Computations, In *Proceedings of the 23th Annual IEEE Symposium on Foundations of Computer Science*, 1982.

[15] G.M. Køien and V.A. Oleshchuk, Spatio-Temporal Exposure Control; An investigation of spatial home control and location privacy issues, In *Proceedings of the 14th Annual IEEE Symposium on Personal Indoor Mobile Radio Communications (PIMRC)*, pp.2760-2764, September 2003

TECP – Tutorial Environment for Cryptographic Protocols

Jelena Zaitseva, Jan Willemson, Jaanus Pöial
Department of Computer Science, University of Tartu, Estonia
e-mail: jellen@ut.ee, jan@ut.ee, jaanus@ut.ee

Abstract

The availability of educational cryptography software is insufficient nowadays, especially for public key cryptography. We have developed a new software tool for teaching public key cryptography based on modular arithmetic. This tool is a tutorial environment allowing stepwise construction of public key cryptography protocols and demonstration of their work. It enables visualization of protocols (with values of secret parameters and intermediate results), allows adding/removing communicating parties, allows adding/editing/removing of arbitrary parameters (with dynamic recalculation of dependent parameters), handles number-theoretic and cryptographic primitives (modular arithmetic, prime numbers, generators, hash functions, etc.), allows working with gigantic integers, saves/loads constructed protocols. By means of these features the user is provided with flexible and configurable tutorial environment.

Keywords: tutorial environment, public key cryptography, modular arithmetic, visualization

1 Introduction

Since the importance of cryptography has lately greatly increased, it is very important to offer corresponding teaching courses for specialists in this area.

There is almost no educational software that would help to study this discipline. The only resource the authors managed to find was [6]. It includes only nine educational programs concerning symmetric Caesar cipher, block ciphers DES, Blowfish, IDEA and ElGamal public key encryption algorithm. It is also disappointing that there is much less educational software available explaining public key

cryptography than the software explaining symmetric algorithms. This specific part of cryptography however should be available with all its nuances. Experience has shown that some students initially cannot understand how it is possible to code some information by one key and decode it by another one.

It is easy to conclude the necessity of public key cryptography teaching software. Such a software must explain the idea of public key cryptography, discover nuances and assist in understanding public key cryptography techniques. It will hopefully accelerate learning of this discipline.

The rest of the paper is organized as follows: section 2 describes requirements to the tutorial environment, section 3 gives a brief acquaintance with possibilities of TECP and describes the representation of cryptographic protocols, section 4 describes the possibilities of using TECP, section 5 gives a short description of implementation of TECP, section 6 – Conclusions and Further Work.

2 Problem Formulation

First, we analyzed primitives needed for PKC [16]. We had in mind the following protocols: Diffie-Hellman key exchange algorithm [2], RSA signatures and encryption schemes [12], Rabin public-key encryption scheme [13], ElGamal signature and encryption schemes [3, 4], DSA [10], Chaum's blind signature scheme [1].

The analysis of the cryptographic schemes shows that the tutorial environment must be able to perform the following mathematical operations:

- enable visualization of protocols, including values of secret parameters and intermediate results (all values can be arbitrary large),
- allow adding/removing communicating parties,
- allow adding/editing/sending/removing arbitrary parameters,
- handle the next number-theoretic and cryptographic primitives:
  - calculation of a mod b, $a - b$, $a + b$, $a \cdot b$, a/b, a^b,
  - calculation of a^b mod n (-1 can also be a value of b),

- calculation of $\gcd(a, b)$,
- calculation of hash value of m,
- generation (and verification) of the prime numbers,
- generation (and verification) of a number from $\mathbb{Z}_n$ ($\mathbb{Z}_n^*$),
- generation (and verification) of a generator of $\mathbb{Z}_n^*$, provided n is a *safe prime number*, i.e. $n = 2 \cdot a + 1$, where a is a prime number. We require such a condition to be able to generate and verify efficiently a generator of $\mathbb{Z}_n^*$,
- generation (and verification) of a number congruent to a mod b,

where a, b, n and m are some positive integers.

3 Overview of Tutorial Environment

The tutorial program – TECP – is a visual environment for creation and manipulation of cryptographic protocols based on modular arithmetic (a good overview of such protocols can be found in [9, 14]). Its main part is the workfield where protocols can be created and visualized.

Protocols are viewed as sequence diagrams. By means of such diagrams the communicating parties that commit protocol steps are represented. The possible steps are generation of keys, calculations of auxiliary parameters and data transmission.

The environment has a set of tools for operating on different components of protocols, i.e. on communicating parties and protocol steps (protocol variables and data transmissions) [16]. At any time all protocol parameters can be changed by the user on the fly.

3.1 Representation of Protocol Components

Figure 1 presents a screenshot of the tutorial environment.

Parties. There are three types of parties: a regular party, an eavesdropper and a public authority.

- *Regular party* can create and calculate different protocol variables, and transmit data to any other party.

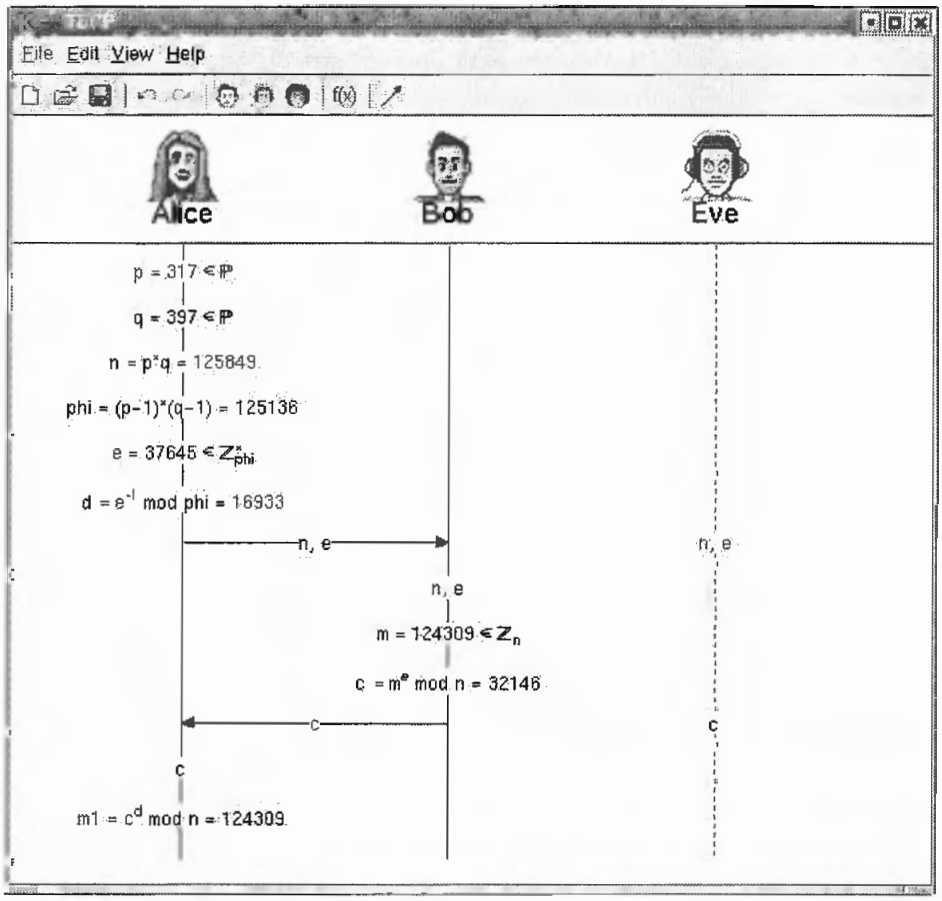


Figure 1: A Screenshot of the Tutorial Environment

- *Eavesdropper* is a regular party with a possibility to know all data transferred from one party to another, even if the eavesdropper is not a recipient. It can create and calculate different protocol variables, and transmit data to any other party.
- *Public authority* is a regular party able to share all received data with all other existing parties. It can create and calculate different protocol variables, and transmit data to any other party as well.

Protocol Variables. A created variable appears under a creating party as the next step of the protocol. If the value of a variable is more than 15 symbols long, only the first 5 and the last 5 symbols of

the value of protocol variable will be displayed with dots in between. In this case, value of a variable can be seen by its fly-by hint or in the special window ('All Values'), which contains information about the values of all protocol variables and all the parties containing them.

Data Transmissions. Each data transmission is represented by an arrow from a sending party to a recipient with transmitted protocol variables on/above it.

4 Usage of Tutorial Environment

In this section, considerations on how the software can be used during the studying process are presented.

Getting to understand communication protocols. The first step students can take for understanding the idea of a protocol is to construct one using the tutorial environment.

Before constructing a protocol student often does not understand basic principles by which the protocol works, in spite of being familiar with the theoretical side of this theme. During the modeling process of the protocol comes understanding of its structure and working principles.

Visualization of numerical values of all parameters. The software enables visualization of numerical values of all parameters (protocol variables). It is considered to be significant for understanding the mechanism of the work of a protocol.

Usually, when discussing protocols at lectures using blackboard-and-chalk presentations, actual values of protocol parameters including hash values are not displayed to students due to their lengthy nature.

The tutorial environment gives convenient tools to illustrate operations with large integers including hash operations. Students can see the value of hash function themselves (Figure 2), and perform different operations with it.

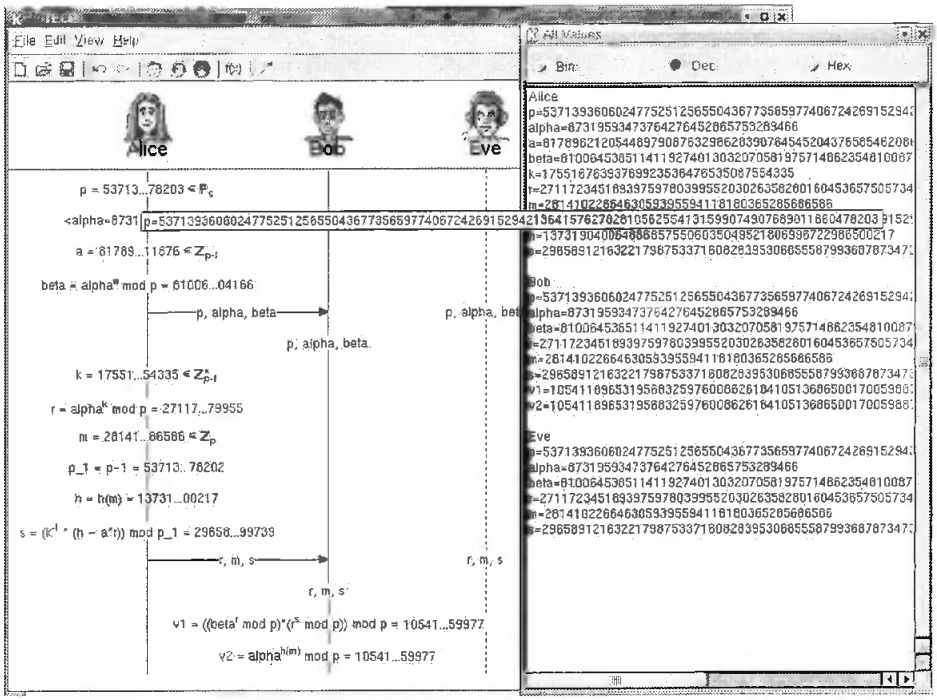


Figure 2: ElGamal Signature Scheme

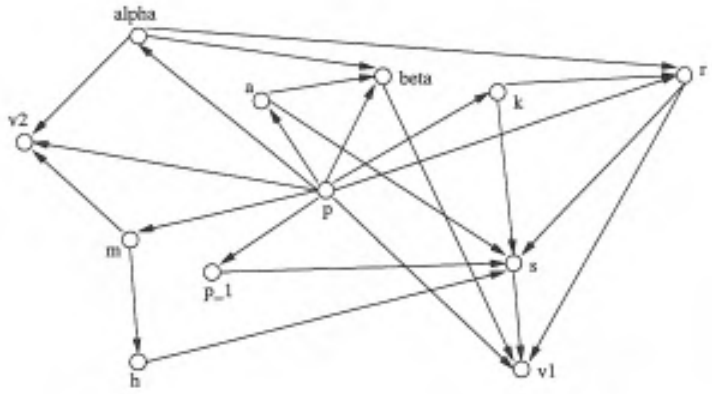


Figure 3: Dependency Graph of Protocol Variables

Possibility to change values of protocol variables. The other thing which definitely can help understanding the protocol, is the possibility to change the value of any protocol variable.

All protocol variables dependent on the edited one will be re-

calculated/regenerated on the fly. It becomes possible due to the dependency graph of protocol variables. In the case of ElGamal signature scheme (Figure 2) this graph is presented in Figure 3.

The change of the value of p causes the regeneration of *alpha*, a, k and m, and recalculation of *beta*, r, p_1, h, s, $v1$ and $v2$. Change of k, however, influences the values of r, s and $v1$. Change of the value of a protocol variable should convince students in the proper work of the protocol – values $v1$ and $v2$ remain equal.

Problem generation For tutors, the environment provides efficient means for generating different problem instances based on the same protocol.

E.g. a standard problem on RSA states that it is possible to recover the encrypted message if it encrypted using the public exponent $e = 3$ with three different moduli.

With TECP, an instance of this problem can be generated simply by regenerating six prime numbers.

Experimenting with protocols. Possibility of changing the protocol (addition/removal of a party, addition/change/removal of transmitted data, addition/change/removal of a protocol variable) gives the ability to see how it influences the security of a protocol.

In the simplest example of RSA encryption scheme (Figure 1), suppose, Alice sends not only n and e, but n, *phi* and e. In this case the user can ask Eve to calculate Alice's private key, and hence, to get to know what message Bob has sent (Figure 4).

Man-in-the-middle attacks can also be constructed using TECP.

5 Implementation

The tutorial environment was written using Borland® Kylix™ 3 Open Edition and Borland® Delphi™ 6 Personal Edition, and is available under GPL. It can be used on Linux and/or Windows machines.

Freeware package FGInt [11] and TParser 10.1 [5] were used in program implementation.

The tutorial environment can be downloaded from [15].

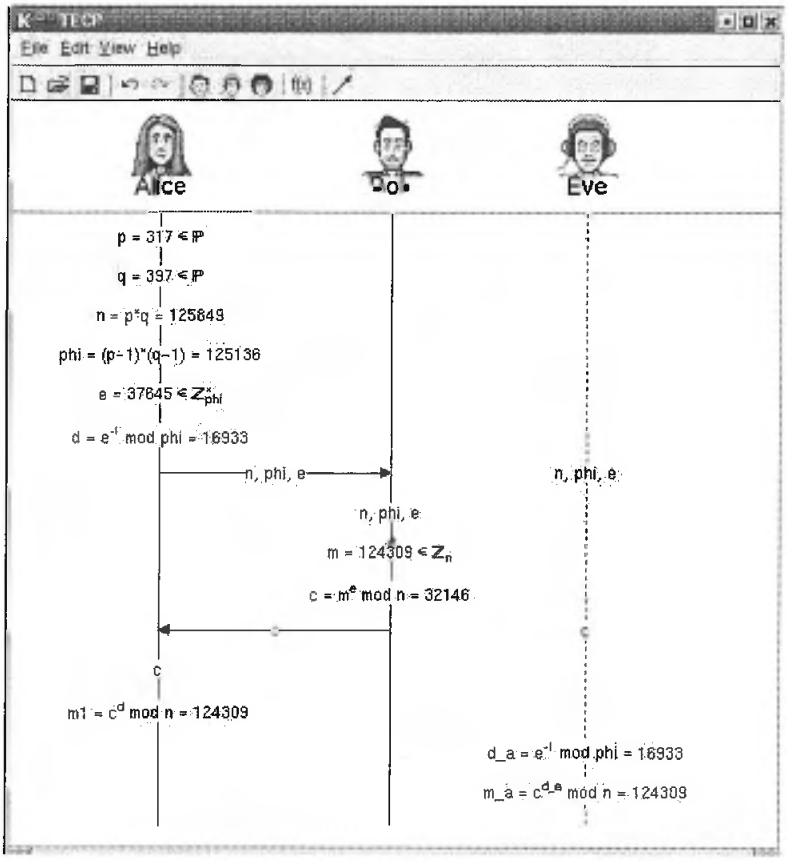


Figure 4: Successful Attack on RSA Encryption

6 Conclusion and Further Work

We have developed a new multi-platform (Linux/Windows) software tool for teaching public key cryptography based on modular arithmetic. This is an attempt to fill the gap in public key cryptography educational software.

The tool developed is a tutorial environment allowing stepwise construction of public key cryptography protocols and demonstration of their work.

TECP was used at the course 'Introduction to Cryptology' conducted of the University of Tartu in the fall term 2002. The course examination has shown that in contrast to past years examinations the amount of students who has failed it, is decreased. Although

we do not have enough statistical data to judge this result, TECP proved to be a comfortable tool for a lecturer to teach and a student to experiment with protocols. It will undoubtedly be used during future courses as well. Further development of the tutorial environment can involve addition of some mathematical operations (concatenation, for example) and addition of new modules enabling constructing cryptosystems based on sparse polynomials [7] and elliptic curves [8].

Acknowledgments

The authors thank Estonian IT Foundation for support.

References

[1] Chaum, D. Blind signatures for untraceable payments. *Advances in Cryptology - Proceedengs of Crypto 82*, pages 199–203, 1983.

[2] Diffie, W. and Hellman, M.E. Multiuser Cryptographic Techniques. *Proceedings of AFIPS National Computer Conference*, pages 109–112, 1976.

[3] ElGamal, T. A Public-Key Cryptosystem and Signature Scheme Based on Discrete Logarithms. *Advances in Cryptology: Proceedings of CRYPTO 84, Springer-Verlag*, pages 10–18, 1985.

[4] ElGamal, T. A Public-Key Cryptosystem and Signature Scheme Based on Discrete Logarithms. *IEEE Transactions on Information Theory*, IT-31(4):469–472, 1985.

[5] Hoffmeister, S., Flaider, A., and Schaaf, R. TParser 10.1 for Borland Delphi - a component for parsing and evaluating mathematical expressions specified at runtime. http://www.datalog.ro/delphi/parser.html, May 2003.

[6] Black Wolf's homepage. Cryptography. http://home.od.ua/~ blackw/Crypt/crypt.html, May 2003.

[7] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A Public Key Cryptosystem, August 1999.

[8] Koblitz, Neal. *Algebraic Aspects of Cryptography.* Springer Verlag, January 1998.

[9] Menezes A.J., P.C. van Oorschot, and Vanstone, S.A. *Handbook of Applied Cryptography.* CRC Press, October 1996.

[10] National Institute of Standards and Technology. Digital Signature Standard (DSS). *FIPS 186-2*, January 2000.

[11] Othman, Walied. Fast gigantic integers package. http://triade.studentenweb.org/GInt/gint.html, May 2003.

[12] R. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.

[13] Rabin, M.O. Digitalized signatures and public-key functions as intractable as factorization. *MIT Laboratory for Computer Science, Technical Report, MIT/LCS/TR-212*, 1979.

[14] Arto Salomaa. *Public-Key Cryptography.* Springer-Verlag, 2nd edition, 1996.

[15] TECP homepage. http://www.math.ut.ee/~ jellen/TECP, June 2003.

[16] Jelena Zaitseva. *TECP - Tutorial Environment for Cryptographic Protocols.* MSc Thesis, 2003.

Improving the Gnutella protocol against poisoning

Meelis Roos, Jan Willemson, Peeter Laud*
Cybernetica, Estonia
{mroos,jan,peeter}@cyber.ee

September 25, 2003

Abstract

Gnutella is a decentralized peer-to-peer file-sharing network on the Internet. One of many problems with this network is poisoning — flooding the network with false data. One kind of poisoning is serving junk instead of a piece of a file — when the downloader puts all the pieces together to get the file, the checksum does not match. We propose an addition to the protocol to protect against this kind of poisoning. We use cryptographic hash functions to prove that a certain piece of file really appears in the whole file in the specific position. The downloader can verify this against a previously distributed file checksum and discard exactly these pieces that fail the validation.

Keywords: Gnutella, file sharing, cryptographic checksums

1 Introduction

Like it or not, peer-to-peer file-sharing is becoming an important part in Internet. There are several file-sharing networks around and the amount of available peer-to-peer networking software is rapidly growing. The networks meet different problems as they grow. The goal of this paper is to solve one of these problems. The social background and legal aspects of file-sharing and peer-to-peer networking are beyond the scope of this paper.

*Supported by Estonian Science Foundation grant #5568

The first file-sharing networks were centralized (with Napster being the most famous system of centralized era). The weakness of centralized network showed itself with time and current file-sharing networks are mostly decentralized. KaZaA, eDonkey and Gnutella are perhaps the most widely known decentralized networks to date but there are also many others. Mojo Nation and BitTorrent are a couple of non-traditional and border-stretching technologies in peer-to-peer networking. It is also worth mentioning that there are usually several different software solutions to access one common file-sharing network. We are looking more closely at Gnutella network [Cli01] since this is a fully open network — the protocols are public and most clients are also open-source.

2 State of the art

There are several services that a file-sharing network usually offers: searching for files, storing of files, transport of files and proxying of connections. A file-sharing network does not need all of them for sure, e.g. BitTorrent [Co03] has no search functionality. We concentrate on file transport service in this paper since it usually takes the most of network bandwidth.

There are several techniques to save network bandwidth and use it efficiently. The most obvious method is to compress the data. This, unfortunately, helps only for the control channels but not file transfer channels since the files transported are usually already heavily packed. Another method is to recognize several copies of a file on the network and use this knowledge to make better downloading choices. We can use the fastest server to download the file or to get part of the file from one server and continue from another server when the connection breaks. We can even download several pieces of the same file from different servers at the same time to efficiently utilize our network bandwidth and save time.

For this approach to work, we need to identify the files somehow in order to find the duplicates. The simplest method — using file name and length — works but is not perfect because the users like to use different naming conventions etc. So a better mechanism is needed and the most commonly used solution is provided by checksums. The Gnutella network uses SHA-1 cryptographic hash function [SHS95] as the checksum to uniquely identify files. With each search result, not only the name but also SHA-1 hash is returned. The searches also accept SHA-1 hashes to help finding the duplicates.

3 Problems

There are many open problems in current decentralized file-sharing networks, scalability being one of the most important ones. The searches are performed as broadcasts and this limits the effective size of the networks before getting too slow. Nodes with low network bandwidth are also a problem as they become the points of congestion when faster nodes transfer their information through them. This is currently being solved by changing the structure of the network to put slow nodes as leaves to faster nodes (the latter ones being called ultrapeers).

There is also a persistent problem about the balance between content providers and content consumers. The network becomes inefficient when the percentage of sharing nodes becomes too low. Different networks use different methods to fight this problem, for example the automatic sharing of downloaded or currently downloadable content (BitTorrent), micro-payments in e-cash (Mojo Nation), better connectivity for users who share something (eDonkey [eDon]). Gnutella does not currently force the user to share anything and this might hinder it less efficient.

Another recent problem is poisoning which essentially means flooding the networks with false data. Poisoning may happen because of bugs in file-sharing software but it is usually deliberate. This problem is not a natural phenomenon of the networks, but usually an artificially created one by companies who dislike file-sharing networks. There are different ways of poisoning the network. One way is to serve files full of trash but with names that people want to download (known as file name poisoning). Another way is to pretend to serve the right file but actually send fragments of bad data so the client gets a broken file when it reassembles all the pieces that it has downloaded from different servers. This is called content poisoning and this is the problem we are trying to fight. Chen and Schroeder [CS02] have concluded that poisoning, when combined with other methods, may be quite effective in shutting down peer-to-peer networks, because users grow frustrated and leave.

In order to estimate the quality of our solution, we must first determine the extent of poisoning. If the solution would require more additional bandwidth in the Gnutella network than the amount currently lost because of poisoned data, it would not be practical. We have not done extensive experiments to find out this percentage (and it would be rather complicated to do so), but our subjective experience suggests that about 1% of the traffic in Gnutella is poisoned. Hence we will use this threshold in our future estimates.

4 Solutions to content poisoning

The first solution that already somewhat works is to overlap the different download chunks a little (like 256 bytes). The downloader can then check that the overlapping data matches already downloaded data. This usually works against buggy client programs and false files in antique client programs that do not support SHA-1 hashes for file identification. Still, overlapping is not efficient against deliberate poisoning since the poisoner can send correct overlapping bytes and still send random data in the middle of the chunk.

Hence we need to authenticate the pieces of a file to actually appear in the whole file in the right offset. The main idea of our solution to this problem is to compute a more structured checksum of the file and distribute it along with search results like SHA-1 and provide a proof along with each downloaded chunk that the hash depends on the chunk located in the file at this offset.

It must be hard to fake the proof so that a proof may realistically be given only for correct chunks. The proof must also be small enough as not to cause too much extra network traffic. In essence, we must make the proof size smaller than the current amount of network bandwidth wasted because of poisoning.

The problem we are trying to solve can be seen as an instance of *secure publication* [DGMS01]. Here the clients make queries against a large dataset, the queries are answered by publication authorities who themselves are untrusted, but are in the posession of the whole dataset compiled by a trusted authority. The clients want to have confidence that the results of their queries really come from that dataset. The trusted authority has published a checksum of the dataset, so whenever a publication authority sends out a reply, it has to complement it with a proof that this reply is a result of running the given query against a dataset with given checksum.

In our case, the downloader places its trust onto the filename, together with the checksum of the file (i.e. it assumes that name poisoning has not occurred). The file represents the entire dataset and pieces of the file are replies to the requests for specific fragments of the file. The downloader wants to be sure that each downloaded fragment really comes from a file with a certain checksum. So, in our solution we are basically applying known tools (from the area of secure publication) in a novel context.

5 Authentication trees

Authentication trees were proposed by Merkle [Mer80] and can be used to provide short evidence that some piece of data belongs to a large data collection. An example of an authentication tree is depicted in Figure 1.

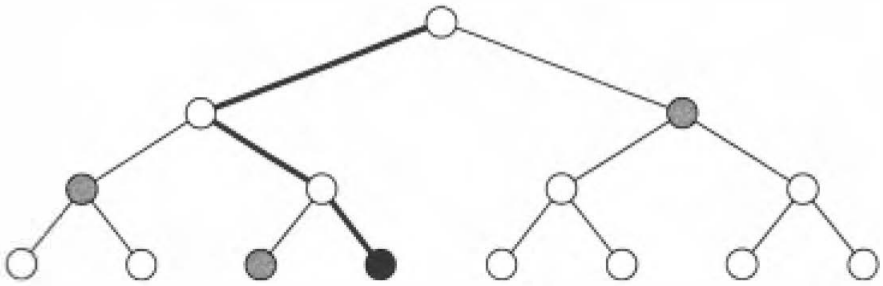


Figure 1: Merkle authentication tree

The leaves of the tree represent individual data items (or their hashes) and each inner node contains the hash of its children (e.g. concatenated or put in some data structure). The hash value at the root of the tree is the checksum depending on all the initial data items. It represents the whole data collection. In order to prove that a particular data item (for example the item colored black in Fig. 1) belongs to the collection represented by that checksum, the values in such nodes of the tree have to be provided, such that the values in nodes on the path from that particular leaf to the root of the tree can be computed. In our example, the values in the three gray nodes have to be provided. Armed with values in the black node and in gray nodes, one can recompute the value in the root and compare it with the known checksum. If these values match, then the inclusion proof is accepted.

In general, if the tree is balanced, the number of additional nodes required is logarithmic in the number of data items. This set of additional nodes will be called the *authentication path* (this name is used for historical reasons, although the nodes in the set do not actually form a path in graph theoretic sense).

6 Authentication trees with Gnutella

Currently, the files in Gnutella are downloaded in chunks. The length of chunks is quite arbitrary and there can be chunks of any length. The client

has usually two chunk lengths to use — a big one like 10M for downloading from fast servers and a small one like 0.5M for downloading from slow servers. Additionally, the chunks get even smaller towards the end of the download of the file. To get efficient chunk authentication, the splitting of the file into chunks must always be done in the same and consistent way between all offerers of the file. To achieve that, we define a block size, make all the chunk sizes multiples of this block size and align the chunks to the block size. An example block size would be 0.5M or 1M.

Now that we can view the file as a sequence of fixed-size blocks, we can start building an authentication tree based on these ordered data blocks. We compute the hashes of all the blocks and create a Merkle tree of these hashes. The hash value at the root of the tree becomes the new hash of the file. This new hash is distributed to the clients with search results alongside (or instead of) the SHA-1 value that is distributed now. When the file is downloaded, the server supplies the authentication path to the root of the tree along with each downloaded block of data. The client then computes the hash of the downloaded block, adds the nodes of the authentication path and recomputes the root value (the computation process is expressed by the solid edges in Figure 1). If the resulting matches the previously published root checksum, then the chunk is authenticated, otherwise it is not.

We also need to authenticate the position of the block in the file, otherwise the server can always send the first block with correct authentication path, no matter which block the client requests. To achieve this, we can use the structure of the authentication path. We note that when recomputing the root hash from a data item, the elements of the authentication path are involved as left or right children of the intermediate nodes. The total number of blocks and the number of the current block are sufficient to establish the left/right property for each node in the authentication path. Hence, adding one L/R bit per element of the authentication path is enough to reconstruct the necessary fragment of the authentication tree and hence to confirm the position of the block in the file.

To supply the authentication path, the server must know the hash values located in the inner nodes of the tree. It would be a bad idea to recalculate them on each request since this would mean rereading the whole source file (like a 650M Linux installation CD image) on each request. So the server should cache the hash values in RAM and maybe even in some persistent cache. This increases the memory requirements of the software somewhat but is feasible, unlike the continuous rereading of files. This also puts a lower limit on the block size — the smaller blocks we have, the more nodes we have in the cached authentication tree. A mixed strategy is also possible — the server caches the hash value at the root of the authentication tree and

also the values in nodes that are in several upper levels of the tree, but it recomputes the values at the lower levels whenever they are requested. To recompute the hash value at some node of the authentication tree, we only need to read the data in all leaves below that node, these leaves are not too numerous if this node is on some low level. This strategy may remove that lower bound on the size of the blocks.

7 Why add a new checksum?

Since it took more than a year to get SHA-1 hashes accepted in Gnutella network, it is easy to see that the addition of a new hash along SHA-1 would also take a lot of time. So it would be great if we could utilize the existing SHA-1 checksum and show that it depends on the specific block being in the file at a specific place.

In the computation of SHA-1 checksum of a file, this file is split to 512-bit blocks $F_1, F_2, \ldots, F_m$ (assume that the size of the file in bits is a multiple of 512, if this is not the case then use some padding). A *round function* H is used to compute 160-bit intermediate values $C_1, \ldots, C_m$ by $C_i = H(F_i, C_{i-1})$. The value C_m is the final checksum, the value C_0 is a constant fixed in the specification of SHA-1. This intermediate values could be used as the elements of authentication paths. Unfortunately, the authentication path proving that F_i is the i-th block of the file is $(C_{i-1}, F_{i+1}, F_{i+2}, \ldots, F_m)$. Hence the expected size of the authentication path would roughly be equal to half of the size of the whole file, which is too much. So in order to save the bandwidth, we really need to add a new checksum that is computed in a non-linear, efficient way (like using a tree).

8 Estimates to extra traffic

If we have 1MB block size and a 700MB file then there would be total 700 blocks. Using a binary Merkle tree, we get 10 as the height of the tree. For computations on each level of the tree we need one hash value as the input from the previous level (which is computed by the client and does not add bandwidth overhead) and one input from the authentication path (20 bytes when we use SHA-1 as the hash function). We also need one L/R bit for each element in the authentication tree. For each 1MB block of data, we get an authentication path of size $20 \cdot 10 + 10 = 210$ bytes. This is about 0.02% of the size of block, which is less than our solution quality threshold 1%. Still we need 140KB for the whole file and we should make it even smaller.

9 Optimizations

The main optimization here is to use an authentication path for a range of blocks (the whole downloaded chunk), not for each block separately. Since many consequent blocks have many values in common for their authentication paths, we can get savings from the subtrees that are fully in the chunk. The hashes for a whole subtree can be computed from the blocks themselves and so we need to have less authentication path elements in total. It turns out that we actually need only two authentication paths for the whole sequence of blocks — one for each end of that sequence [BRW02]. So the size of the authentication information gets down to $2 \cdot 20 \cdot 10 = 400$ bytes for a range of blocks. If a file is downloaded in 10 chunks, the total extra traffic makes up 4KB. This is already good enough to provide a practical solution.

We can improve the result even further because there are more efficient ways to authenticate an interval of values. Interval time-stamps solve the same problem for authenticating items that belong between two other items. So we can make use of the linking schemes used in interval time-stamping. The best known interval time-stamping schemes [Wil02] get the size of authentication paths down from $2c \log_2 n$ (from two paths of size $c \log_2 n$ where c is the length of the hash function output (20 bytes for SHA-1) and n is the number of blocks) asymptotically to about $1.44c \log_2 n$.

10 Discussion

Several open problems remain with our approach. The first of them is the handling of file tails. When most of the file has been downloaded and there are still several servers offering the file, the remaining part of the file is still downloaded from several servers which means that very small chunks of the file are retrieved from a single server. We can authenticate the last block only as a whole, so we cannot separately authenticate the sub-block-sized chunks we downloaded from different servers. The best solution seems to be to ignore the problem and gather all the pieces, check the hash of the whole block and if it does not match, discard the whole block and download the entire block again, this time from a single, randomly chosen server.

The second problem is the deployment. It took a long time to get most servers to support SHA-1 hashes and it would probably take even a longer time for this hash to be accepted for most servers (most clients do not upgrade their software without a good reason).

There are also many proposed additions to the Gnutella protocol. Many programmers and companies have developed enhancements and the process

of making sure that these enhancements do not harm the network overall takes very long. There is no reason to believe that our addition enjoys any better success than the others in the queue.

Finally, it may happen that the problem disappears by itself after some time. Some of the poisoners seem to be from movie companies, they are out to discredit the file-sharing networks to scare the users away from them. We expect this problem to be short-term – it's just a technology shift that has not gotten to everyone and there are not too many regulations about this kind of behaviors yet. But there also seem to be just bad people trying poisoning for fun and it is not expected that they would go away anytime soon.

11 Acknowledgements

We thank the Tiger University Project of Estonian Information Technology Foundation for support.

References

[BRW02] Ahto Buldas, Meelis Roos, Jan Willemson. *Undeniable database queries.* In H-M. Haav, A. Kalja (Eds), Databases and Information Systems II, Selected Papers from the Fifth International Baltic Conference, BalticDB&IS 2002, Kluwer Academic Publishers, 2002.

[CS02] Andrew H. Chen, Andrew M. Schroeder. *A Modified Depensation Model for Peer to Peer Networks: Systemic Catastrophes and Other Potential Weaknesses.* University of Washington, June 13th, 2002. Available from `http://students.washington.edu/achen/papers/p2p-paper.pdf`

[Cli01] Clip2. *The Gnutella Protocol Specification v0.4 (Document Revision 1.2)*, 2001. Available from `http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf`

[Co03] Bram Cohen, *Incentives Build Robustness in BitTorrent*, May 22nd, 2003. Available from `http://bitconjurer.org/BitTorrent/bittorrentecon.pdf`

[DGMS01] Premkumar T. Devanbu, Michael Gertz, Charles U. Martel, Stuart G. Stubblebine. *Authentic Third-party Data Publication.* In Proceedings of IFIP TC11/WG11.3 Fourteenth Annual Working Conference on Database Security, pp. 101–112, Kluwer Academic Publishers, 2001.

[eDon] eDonkey homepage, `http://www.edonkey2000.com/`

[Mer80] Ralph C. Merkle. *Protocols for public key cryptosystems.* In Proceedings of the 1980 IEEE Symposium on Security and Privacy, pp. 122–134, IEEE Computer Society Press, 1980.

[SHS95] *Specifications for Secure Hash Standard.* Federal Information Processing Standards Publication 180-1 (FIPS PUB 180-1), April 17th, 1995.

[Wil02] Jan Willemson. *Size-Efficient Interval Time-Stamps.* Ph.D. Thesis, Tartu University, 2002.

On Server-Aided Computation for RSA Protocols with Private Key Splitting

Anne-Maria Ernvall[1]* and Kaisa Nyberg[2]
[1] Department of Mathematics, University of Helsinki, Finland
[2] Nokia Research Center, Helsinki, Finland

September 23, 2003

Abstract

Server-aided secret computation is a widely investigated topic due to the obvious benefits of such an approach in applications where small constrained terminal equipment is used. Recently, also other ways of using servers to improve the security functionality of end user devices have been proposed. By splitting the private computation functionality between the server and the client, end user devices can be protected from being misused if captured, or the usage of the private key can be controlled. In this contribution the problem of combining different security tasks delegated to the servers by client devices is investigated. The capture resilience and delegated authorisation protocols based on RSA is achieved by composition of the private key as a sum of two partial keys. To reduce the computation of one party, its partial key must be reduced. Previously, continuous fractions were used to break RSA with a short private key. We apply this method to the case where the private RSA key itself is large but it is split into two parts, one of which is small. Our results show that for some typical parameter values the party who knows the longer part of the private key can easily recover the small part.

Key words: E- and M-business security, server-aided secure computation, capture-resilience, delegated authorisation, RSA system, small RSA exponents

1 Introduction

Server-aided computation protocols are protocols between two parties, the client and the server. Cryptographic computations, particularly those related to some public key system, often involve heavy, time and power consuming operations. The problem is how the client can outsource computations that use some private information known to the client, without revealing the private information to the aiding server. Typical applications of server-aided protocols include smart card computations, which may need to be aided using an external processor within the host device. The computations required to be performed by mobile devices often need to be reduced to the minimum because of constrained electric power resources.

*The author's work on this paper was performed during her internship at Nokia Research Center

Numerous protocols for providing server-aided functionality for the common public key systems have been proposed and analysed. For a review of the most important developments in this area see [5] where also a new protocol for server-aided RSA is proposed. Other previous literature highlighting the possible pitfalls include [3] and [11]. For the purposes of this paper we are interested in server-aided RSA protocols, and in particular those not using the Chinese Remainder Theorem.

The goal of this work is to investigate if it is possible to include the server-aided facility to certain new protocols using RSA, which share the private key functionality between a client and a server to provide other security services. In [8] and [9] P. MacKenzie and M. Reiter presented protocols where the private RSA key is split into two parts and given to two parties in such a manner that neither party cannot perform the private RSA operation alone. One of the parties, often an end user device in practise, then produces the result of the private RSA operation using the assistance of the second party. Subsequently, S. Sovio, N. Asokan and K. Nyberg observed that this idea also applies to the case where an owner of a private RSA key wants to delegate the capability of using the RSA key to other devices, while retaining some control of how the key is used by those devices [13]. For example, the owner might want to set a limit to the amount of financial transaction that can be signed using the key. In the Sovio-Asokan-Nyberg protocol the control function is delegated to an assistant, which is a server that is available online, by giving it a share of the private RSA key. Also further delegation is possible using the help of the same assistant.

The main mathematical tool used in this work is the continuous fraction expansion for rational numbers. Recently, this method was used by M. Wiener in [15] to show that the private RSA exponent shall not be too small. Using RSA exponent smaller than $\frac{1}{3}\sqrt[4]{n}$, where n is the RSA modulus, is not secure. For a presentation of this attack we refer to [1] or [14]. Later D. Boneh and G. Durfee increased this bound to $n^{0.292}$ in [2]. They used a method based on the LLL-algorithm [7]. Recently, the feasibility of these attacks was investigated for the multiprime RSA system in [4].

In [8] MacKenzie and Reiter mention the problem of adding server-aided facility to their protocol, but leave it as a problem for future work. In the context of [8] and [13] the client and server roles can also appear to be switched, as it may be desired that the server's workload is reduced. In this paper, we study this question and give some negative results. We use continuous fraction expansions to show that adding server-aided computation facility to the protocols of MacKenzie and Reiter and Sovio et al is not always possible by making one share of the RSA key small. We show that this negative result holds if the public key is small and also if it is sufficiently large. Finally, we also consider the case where the private RSA key is decomposed as a product of two shares, one of which is small. This scenario yields completely under the continuous fractions attack.

The paper is structured as follows. First, the basic server-aided RSA protocol is given. Then the principles of the RSA splitting in the MacKenzie-Reiter and Sovio-Asokan-Nyberg protocols are described. The continued fraction expansion is described in Section 4, where we also show how it was applied to the case of short RSA private key by Wiener. Our results on the insecurity of small RSA private key shares are given in Section 5.

2 Server-Aided RSA Computation

The parameters of the RSA system are two large and different prime numbers p and q, their product $n = pq$ called the RSA modulus, the private exponent d and the public exponent e. The knowledge of the private key is essentially equivalent to the knowledge of the factors p and q. For a probabilistic algorithm for deriving the factors from the knowledge of e, d and n, see, for example, [14].

If the factors p and q are known the RSA computations modulo n can be speeded up using the Chinese Remainder Theorem. However, in the protocols discussed in this paper the factorisation is not known to the parties of the protocols. Therefore, our interest is in server-aided computation protocols that do not assume the use of the Chinese Remainder Theorem. The choice is not large. Essentially all protocols are variations of the basic protocol described below. The most recent protocol of Hong et al [5] uses a different approach, but requires that the client knows the factorisation of n.

The basic approach to server-aided RSA computations [10] is to decompose the private d as follows:

$$d = \sum_{i=1}^{\ell} w_i f_i \mod \varphi(n).$$

Creating such a decomposition requires the knowledge of $\varphi(n)$ unless in each term one of the factors w_i or f_i is equal to 1 or 0.

Client gives the values $w_1, w_2, \ldots, w_\ell$ to the server. The effect of server-aided computation is achieved by selecting all values f_i small. To create a result of a private RSA operation on some message m, the client sends the values $w_1, w_2, ..., w_\ell$ and m to the server. Server computes ℓ values m^{w_i}, $i = 1, 2, \ldots, \ell$, and returns them to the client. It then computes:

$$S = \prod_{i=1}^{k} (m^{w_i})^{f_i}.$$

As shown by Bums and Mitchell [3] it is essential that the client verifies the result S before publishing it. The server may try to find the values of f_i one by one using the following procedure.

When given a message m the server gives the correct values for all but one of m^{w_i}. For one of the indices, say j, the server replaces the value m^{w_j} by tm^{w_j}. If client doesn't verify the result S of the private key operation, it prompts out

$$S = t^{f_j} m^d \mod n.$$

The server computes

$$S^e = t^{ef_j} m^{ed} = t^{ef_j} m \mod n.$$

If this value is equal to m, the server obtains that $f_j = 0$. If this value is not equal to m, the server computes $t^{eh}m$ for different values h until $t^{eh}m = S^e \mod n$. Then $h = f_j$.

3 Protocols Using RSA Splitting

3.1 Capture Resilience

In a series of papers [8] and [9] MacKenzie and Reiter describe methods how to construct "capture-resilient" devices by using a network server. The idea is to protect private key operations by splitting the private key into two parts and storing one part in the device while giving the second part to the network server. Different schemes are presented for RSA encryption, RSA signatures and ElGamal signatures. For the protocols using RSA the private key d is expressed as a sum of two integers $d1$ and d_2 such that $d = d_1 + d_2 \mod \varphi(n)$. The device gives the half-key d_2 to the server while storing d_1 to its own protected memory. When needed, the device can recover d_1 by asking the user to enter a password. Further, the device generates a ticket and gives the ticket and the second half-key to the server. The ticket contains necessary secret information using which the server can authenticate the device. After the initialisation the private RSA values p, q and d are deleted.

When the user wants to generate a signature on a message, it sends a request message that contains the message and the ticket, and the authentication values encrypted using the server's public key. Only after the server has successfully verified that the private key has not been disabled, confirmed the identity of the device and ensured that the user has inserted the correct password, the server agrees to co-operate. The server computes $m^{d_2} \mod n$ and returns this half-signature to the device. The device computes its half-signature $m^{d_1} \mod n$. The device computes the full signature by forming the product of the half-signatures, as

$$m^{d_1} m^{d_2} = m^{d_1+d_2} = m^d \mod n. \tag{1}$$

Since the protocol does not provide explicit authentication of the server to the client device, it is recommended that the device verifies the signature before using it. In such a manner also the server becomes authenticated.

3.2 Delegated Authorisation

The protocol for authorisation delegation in [13] is based on the same principle of sharing the private RSA key as the capture-resilience scheme. But the setting is different. The RSA private key is not deleted but it is kept stored in a device called the master device. The master device generates the splitting of the private exponent d and stores the first part of it to another device, called a slave device, together with some additional authentication information. The second part of the RSA key is sent to the assistant server together with necessary information, using which the server knows how to serve the slave device.

Consider the following scenario. Alice has a bank account that she can access over the Internet using her personal device. She is able to make transactions from her account by issuing digitally signed payment orders. She owns a phone and a PDA, and wants to be able to use the bank service with both. For this purpose she delegates the usage of her RSA signature key from the PDA to the phone, which does not have tamper resistant memory where the full private key could be stored. The assistant server functionality can be offered either by the phone operator or the bank. In another scenario, Bob is Alice's son. Bob is going for a trip over the weekend and needs some

money from Alice. Alice delegates the usage of her account to Bob for the weekend and sets an upper limit to the funds she is willing to grant.

The protocol allows Alice to control the usage of her private key independently. Her PDA will inform the related assisting servers of each new delegation, or revocation of a delegation. The service does not need to know which Alice's device is used to make a transaction, or whether she has authorised somebody else to make the transaction for her. On the other hand, Alice retains the full capability of her private key with her PDA, where it is stored.

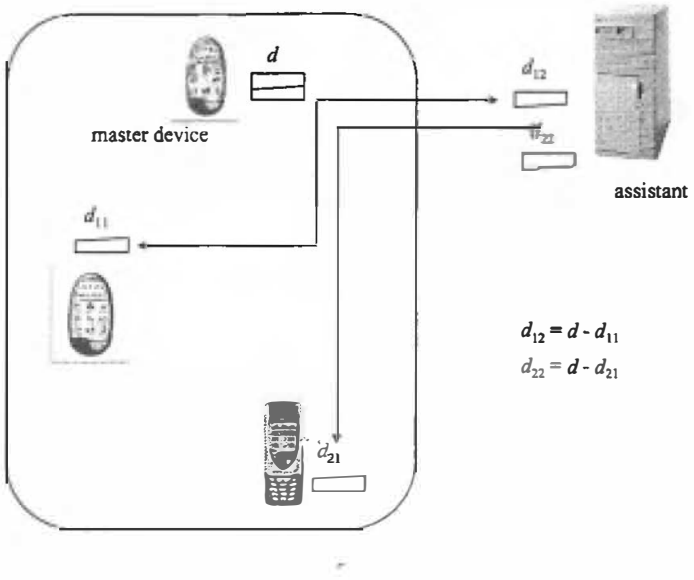


Figure 1: Initialisation of the delegated authorisation protocol

In Figure 1 it shown how a master device delegates the private RSA capability to two slave devices. The two delegations are completely independent. The protocol also allows any of the slave devices to delegate the authorisation further to a third device. If this is not desirable, it can be prevented by the master. The slave does not have access to the full private RSA key or to the $\varphi(n)$ value. Therefore in further delegation the slave's key share is just split into a sum of two integers. Also for the same reason the Chinese Remainder Theorem cannot be used to facilitate the private RSA computations. The private key usage is similar to the case of capture resilient devices explained in Section 3.1.

When considering server-aided computation within the delegated authorisation protocols, two different type of needs may arise depending on the scenario where it is used. First, the client device may be constrained and it would be helpful if the assisting server would do the major part of the computations. A straightforward approach would be to make the client device's half-key small. Since it is also recommended that the client-device should verify the full signature, it is then also necessary to generate the RSA key pair in such a manner that the public key is small. Small public key as such is not known to pose any threat and is frequently used in practise.

In the second scenario, the assistant serves a large number of slave clients, which perform frequent private RSA operations. Then the workload of the assistant can be reduced by making the

assistant's half-key small. In this scenario, the size of the public key need not be reduced.

These scenarios constitute special cases of the server-aided computation protocol of Section 2 with $\ell = 2$. The server's share is $(w_1, w_2) = (1, d_2)$, and the client's share is $(f_1, f_2) = (d_1, 1)$. Neither scenario poses any requirements to the size of the full RSA private key. Below we recall an attack on short RSA private keys, not because it would be applicable to our scenarios, but because it makes use of a mathematical technique that will be later applied to analysing the security of our scenarios.

4 Continued Fractions and Wiener's Attack

In this section we introduce the main mathematical tools used in this paper, which are simple continued fraction expansions and an algorithm for finding convergents. For exact mathematical treatment the reader is referred to, for example, [12]. A simple continued fraction is a chain fractions:

$$x_0 + \cfrac{1}{x_1 + \cfrac{1}{x_2 + \ddots}}$$

where x_i, $i = 0, 1, 2, ...$ are positive integers. This expression is denoted in a more compact form by $[x_0, x_1, x_2, \ldots]$. Each positive real number x can be represented as a continued fraction, which can be formed using the following algorithm: x_0 is the integral part of x, x_1 is the integral part of the inverse of $x - x_0$, x_2 is the integral part of the inverse of $\frac{1}{x-x_0} - x_1$, and so on. Moreover, the simple continued fractions expansion of x is unique. The continued fraction expansion is finite, if there is an index n such that $x_i = 0$, for all $i > n$. Then it is denoted by $[x_0, x_1, x_2, \ldots, x_n]$. The continued fraction expansion of a real number x is finite if and only if x is a rational number. For each positive rational number x there is an n such that $x = [x_0, x_1, x_2, \ldots, x_n]$.

Given a simple continued fraction expansion $[x_0, x_1, x_2 \ldots]$ of a real number x, the sequence of rational numbers c_j, $j = 1, 2, ...$, defined as

$$c_j = [x_0, x_1, x_2, \ldots, x_{j-1}]$$

are called the convergents of x. Each c_j can be written as a rational number $c_j = \frac{a_j}{b_j}$, where the numerator and denominator satisfy the following recurrences:

$$a_j = \begin{cases} 1, & \text{for } j = 0 \\ x_0, & \text{for } j = 1 \\ x_{j-1}a_{j-1} + a_{j-2}, & \text{for } j \geq 2 \end{cases}$$

and

$$b_j = \begin{cases} 0, & \text{for } j = 0 \\ 1, & \text{for } j = 1 \\ x_{j-1}b_{j-1} + b_{j-2}, & \text{for } j \geq 2. \end{cases}$$

The simple continued fraction expansion of a rational number can be computed using the Euclidean algorithm. Given the continued fraction expansion, the convergents can be computed using

the above recurrences. Clearly both algorithms are polynomial in the size of the rational number of x. The size of x is the minimum of the lengths of its numerator and denominator in its reduced form.

The following theorem is very useful when one has to determine whether a rational number is a convergent of some other rational number.

Theorem 1. *Let $\frac{r}{s}$ and $\frac{a}{b}$ be two positive rational numbers such that* $\gcd(r,s) = \gcd(a,b) = 1$. *If*

$$|\frac{r}{s} - \frac{a}{b}| < \frac{1}{2b^2}$$

then $\frac{a}{b}$ is one of the convergents of $\frac{r}{s}$.

The definitive equation of the private and public exponents of RSA is

$$ed - k\varphi(n) = 1, \tag{2}$$

which can be written also as

$$\frac{e}{\varphi(n)} - \frac{k}{d} = \frac{1}{d\varphi(n)}. \tag{3}$$

Wiener observed that $\varphi(n)$ can be approximated using n. Then if d is sufficiently small the condition of Theorem 1 is satisfied. It follows that the fraction $\frac{k}{d}$ can be found as a convergent of $\frac{e}{n}$.

The following Lemma gives an estimate of the unknown $\varphi(n)$ approximated using n.

Lemma 2. *Assume $0 < p < q < n$, where p and q are the prime factors of n, and $q < 2p$. Then $n - \varphi(n) < 3\sqrt{n}$.*

Proof. By the definition of Euler's φ-function,

$$\varphi(n) = (p-1)(q-1) = pq - p - q + 1 = n - p - q + 1.$$

Because $p < q < 2p$, it follows that $p < \sqrt{n}$ and $q < 2\sqrt{n}$, and therefore

$$n - \varphi(n) = p + q - 1 < \sqrt{n} + 2\sqrt{n} - 1 < 3\sqrt{n}.$$

■

Wiener's attack is based on the following theorem. We give the proof here for completeness.

Theorem 3. *Let $d < \frac{1}{3}\sqrt[4]{n}$ be the private exponent and let $e < \varphi(n)$ be the public exponent. Then d can be found in polynomial time in* $\log n$.

Proof. The lefthand side of (3) where $\varphi(n)$ is replaced by n is estimated as follows:

$$\begin{aligned}
|\frac{e}{n} - \frac{k}{d}| &= |\frac{ed - kn}{dn}| \\
&= \frac{1}{dn}|ed - k\varphi(n) + k\varphi(n) - kn| \\
&= \frac{1}{dn}|1 + k(\varphi(n) - n)| \\
&\leq \frac{1}{dn}k(n - \varphi(n)) \\
&\leq \frac{3k}{d\sqrt{n}}
\end{aligned}$$

Because $k\varphi(n) = ed - 1 < ed$ and $e < \varphi(n)$, it follows that $k < d$. Using this and the assumption we get

$$|\frac{e}{n} - \frac{k}{d}| < 3\frac{\frac{1}{3}\sqrt[4]{n}}{d\sqrt{n}} < \frac{1}{2d^2}.$$

Now it follows from Theorem 1 that $\frac{k}{d}$ is a convergent of $\frac{e}{n}$, and can be found in polynomial time. ■

The final step is to check each convergent to find the one that gives the solution for $\varphi(n)$. A complete algorithm to perform Wiener's attack can be found, for example, in [14].

5 Attacks on RSA Splitting with Small Private Exponents

The definitive equation in the RSA splitting scenarios of Section 3 is the following:

$$ed_1 + ed_2 = 1 + k\varphi(n) \tag{4}$$

Let us now consider the scenarios for server-aided computation for the capture-resilience protocol and the delegated authorisation protocol discussed at the end of Section 3. First we investigate the security in the case both e and d_1 are small, and show that it is insecure. In the second case d_1 is small but no requirement is set on e. Using continued fractions we show that also the second case is insecure, if e is sufficiently large. It is possible that our results can still be improved using stronger methods for finding small integer solutions, such as the LLL algorithm, in the similar manner as used by Boneh and Durfee in [2] to improve Wiener's attack.

5.1 Small e and d_1

We prove the following result.

Theorem 4. *Assume that the private exponent d is drawn uniformly at random between 0 and $\varphi(n)$. Assume that $e < \frac{1}{3}\sqrt[4]{n}$ and d_1 is not essentially larger than e. Then d_1 can be found with probability $1 - \frac{1}{t}$ using $t\frac{d_1}{e}$ trials in polynomial time in the length of n .*

Proof. We start by finding the value of k. We estimate the following difference of fractions:

$$\begin{aligned}
|\frac{d_2}{n} - \frac{k}{e}| &= \frac{1}{en}|ed_2 - kn| \\
&= \frac{1}{en}|ed - ed_1 - k\varphi(n) + k(\varphi(n) - n)| \\
&= \frac{1}{en}|1 - ed_1 - k(n - \varphi(n))| \\
&\leq \frac{1}{en}(ed_1 + k(n - \varphi(n)).
\end{aligned}$$

Since $d < \varphi(n)$, we have $k < e$. By this estimate and Lemma 2 we get

$$|\frac{d_2}{n} - \frac{k}{e}| < \frac{d_1 + 3\sqrt{n}}{n} \leq \frac{4}{\sqrt{n}} < \frac{1}{e^2},$$

using the assumption on the size of e. By Theorem 1 it follows that $\frac{k}{e}$ is a convergent of $\frac{d_2}{n}$. Since the attacker knows e, d_2 and n, he can find k in polynomial time in the length of n.

It remains to find d_1. Recall the basic RSA equation (2) from where it follows that $e^{-1} \bmod k = d \mod k$ and therefore, the attacker can compute

$$\delta = d_1 \bmod k = (d - d_2) \bmod k = (e^{-1} \bmod k - d_2) \bmod k.$$

With probability $1 - \frac{1}{t}$ we have $\frac{\varphi(n)}{t} < d$, for any $1 < t$. Therefore $k\varphi(n)$, which is about equal to ed, is larger than $\frac{e}{t}\varphi(n)$, with the same probability. Then $tk > e$ and hence $(t\frac{d_1}{e})k > d_1$. It follows that the attacker can find the correct value of d_1 in at most $t\frac{d_1}{e}$ trials of the form $\delta + sk$, where $0 \leq s < t\frac{d_1}{e}$. ■

The last step in the attack described in the above proof can be made infeasible if e is chosen much smaller than d_1. For example if the lengths of e and d_1 are 16 and 128 respectively, then the number of trials is $t2^{112}$.

5.2 Small d_1 and Large e

Now we present a result which demonstrates the insecurity of uneven splitting of the private RSA exponent in the case when the public exponent is large.

Lemma 5. *In the setting of Equation (4)*

$$d_2 \leq k \text{ if and only if } (\varphi(N) - e)d_2 < ed_1.$$

Proof. Assume first that $d_2 \leq k$. Then

$$ed_1 = k\varphi(n) + 1 - ed_2 \geq (\varphi(n) - e)d_2 + 1 > (\varphi(n) - e)d_2.$$

Assume then that $ed_1 > (\varphi(n) - e)d_2$. We get

$$(d_2 - k)\varphi(n) = d_2\varphi(n) + 1 - ed_1 - ed_2 = d_2(\varphi(n) - e) + 1 - ed_1 < 1$$

using the assumption. It follows that $d_2 - k \leq 0$. ■

Theorem 6. *Assume that in the setting of (4) $d_1 \leq \frac{1}{3}\sqrt[4]{n}$ and that e is sufficiently large satisfying the conditions*

$$\max\{\frac{n}{6d_1^2}, \varphi(n) - \frac{n}{6d_1d_2}\} < e < \varphi(n).$$

Further it is assumed that $\gcd(d_1, k - d_2)$ *is small. Then, given d_2, e and n, the secret d_1 can be recovered in polynomial time in the size of n.*

Remark. From the assumed lower bounds for e the second one is typically the maximum.

Proof.

$$\begin{aligned}
|\frac{e}{n} - \frac{k-d_2}{d_1}| &= \frac{|ed_1 - nk + nd_2|}{nd_1} \\
&= \frac{1}{nd_1}|1 + k\varphi(n) - ed_2 - nk + nd_2| \\
&= \frac{1}{nd_1}|1 - (k-d_2)(n-\varphi(n)) + d_2(\varphi(n) - e)|.
\end{aligned}$$

By assumption

$$d_2(\varphi(n) - e) < \frac{n}{6d_1} < ed_1.$$

Hence it follows from Lemma 5 that $d_2 \leq k$. Since $e < \varphi(n)$, it follows from 2 that $d > k$. Therfore $d_1 \geq k - d_2 \geq 0$. We use this information, the assumption that $d_2(\varphi(n) - e) < \frac{n}{6d_1}$ and the estimate $n - \varphi(n) \leq 3\sqrt{n}$ given by Lemma 2 to estimate further:

$$\begin{aligned}
|\frac{e}{n} - \frac{k-d_2}{d_1}| &\leq \frac{1}{nd_1}((k-d_2)(n-\varphi(n)) + d_2(\varphi(n) - e) \\
&\leq \frac{1}{nd_1}(3d_1\sqrt{n} + \frac{n}{6d_1}) \\
&= \frac{1}{6nd_1^2}(18d_1^2\sqrt{n} + n) \\
&= \frac{1}{6d_1^2}(2(3d_1)^2\frac{1}{\sqrt{n}} + 1) \leq \frac{1}{2d_1^2}
\end{aligned}$$

for $d_1 \leq \frac{1}{3}\sqrt[4]{n}$. It follows that the reduced form of $\frac{k-d_2}{d_1}$ is a convergent of $\frac{e}{n}$. Given the convergents $\frac{a_j}{b_j}$ of $\frac{e}{n}$, computed in polynomial time in the size of n, the secret part d_1 can be found by testing all numbers of the form sb_j, where $1 \leq s \leq \gcd(d_1, k - d_2)$. ■

The solution given in the proof of the theorem above becomes infeasible if $\gcd(d_1, k - d_2)$ is large. Such parameters can be generated by selecting d_2 such that $k - d_2$ is a multiple of a large factor of $d - k$ and at the same time $d_1 = d - d_2$ is sufficiently small. On the other hand, if generated at random, the shares d_1 and d_2 will satisfy $\gcd(d_1, k - d_2) = 1$ with probability about 0.608 (see [6], Section 4.5.2).

5.3 Decomposition as a Single Product

Finally let us consider the case, where the private exponent is split into two half-keys d_1 and d_2 in such a manner that $d = d_1 d_2 \mod \varphi(n)$. Such a splitting can be considered as a special case $\ell = 1$, for the server-aided computation scheme presented in Section 2. Due to the attack of Burns and Mitchell [3] it is necessary that the client verifies the result of the private key operation, before publishing the result. Therefore it is required that the public key is small.

Clearly, it would be possible to generate the RSA parameters such that e and d_1 are of limited size, but only if $\varphi(n)$ is known. Therefore further delegation without knowledge of $\varphi(n)$ is not possible. But this scenario is always insecure, as shown by our next result.

Theorem 7. *Let in the context of (2)* $d = d_1 d_2 \mod \varphi(n)$. *If* $e < \frac{1}{2}\sqrt[6]{n}$ *and* $d_1 < \frac{1}{2}\sqrt[6]{n}$, *then* d_1 *can be found in polynomial time in the length of* n.

Proof. The proof proceeds by showing that $\frac{k}{d_1}$ is a convergent of $\frac{ed_2}{n}$. For this purpose we estimate their difference as follows:

$$\begin{aligned} |\frac{ed_2}{n} - \frac{k}{d_1}| &= \frac{1}{nd_1}|ed_1d_2 - kn| \\ &= \frac{1}{nd_1}|ed - k\varphi(n) + k\varphi(n) - kn| \\ &< \frac{1}{nd_1}k(n - \varphi(n)) < \frac{3k}{d_1\sqrt{n}}. \end{aligned}$$

Since $d_2 < \varphi(n)$ it follows that $k < ed_1$. Using the assumption we estimate that $6ed_1^2 < \sqrt{n}$. Hence

$$|\frac{ed_2}{n} - \frac{k}{d_1}| < \frac{3e}{\sqrt{n}} < \frac{1}{2d_1^2}.$$

We see that the condition of Theorem 1 is satisfied and that d_1 can be found as the denominator of one of the convergents of $\frac{ed_2}{n}$ in polynomial time in the length of n. ■

For example, if n is a 1024-bit value, which is typical in today's applications, the splitting of the private RSA exponent is insecure already when e and d_1 are 170-bit values. This is in the typical range of parameter values for server-aided computations.

6 Conclusion

In this paper we identified some insecure decompositions of RSA private keys in server-aided scenarios. We used continued fractions expansions. Some cases were left open. For example, we cannot tell if the case with an arbitrary public key allows secure splitting of the private key as a sum of two parts, one of which is small. An open problem to be studied is whether our results can be strengthened using methods such as the LLL algorithm used by Boneh and Durfee [2]. Such investigations would be necessary if one wants to identify the secure decompositions of the private RSA key. Although our results are negative, they reveal new properties of the RSA system.

Another approach to solving the server aid problem would be to use existing server-aided secure RSA computation protocols to facilitating the computations of either of the parties in the capture resilience protocol and the authorisation delegation protocol. This means that the server-aided protocol is applied to the private half-key, say d_1. However, certain additional conditions must then be satisfied. The server-aided computation must not use the corresponding public key, that is, the inverse of d_1 modulo $\varphi(n)$, since the knowledge of it would allow the client to factor n. Therefore, it may not be possible to verify the result before it is delivered. Also the Chinese Remainder Theorem cannot be used. The study of suitable server-aided protocols applicable to RSA half-keys, and the exact requirements for such protocols is also left for future work.

References

[1] D. Boneh. Twenty Years of Attacks on the RSA Cryptosystem. *Notices of the American Mathematical Society*, 46 (1999), 203-213.

[2] D. Boneh, G. Durfee. Cryptanalysis of RSA with Private Key d Less Than $N^{0.292}$. *IEEE Transactions on Information Theory*, 46 (2000), 1339-1349.

[3] J. Burns, C. Mitchell. Parameter selection for Server-Aided RSA computation Schemes. *IEEE Transactions on Computers*, 43 (1994), 163-174.

[4] M. J. Hinek, M. K. Low and E. Teske. On some attacks on Multi-prime RSA. In K. Nyberg and H. Heys (Eds.) *Selected Areas in Cryptography, SAC 2003*, St. Johns, Newfoundland, Canada, August 2002, LNCS 2595, Springer-Verlag 2003, 385-404.

[5] S. M. Hong, J. B. Shin, H. Lee-Kwang and H. Yoon. A New Approach to Server-Aided Secret Computation, *Proceedings of ICISC'98*, Seoul, Korea, December 1998, 33-45.

[6] D. E. Knuth. *The Art of Computer Programming, Vol. 2, Seminumerical Algorithms*. Addison-Wesley, 3rd edition, 1997.

[7] A. K. Lenstra, H. W. Lenstra, Jr. and L. Lovasz. Factoring Polynomials with Rational Coefficients, *Mathematische Annalen*, 261 (1982), 515-534.

[8] P. MacKenzie and M. K. Reiter. Networked cryptographic devices resilient to capture. In *Proceedings of the 2001 IEEE Symposium on Security and Privacy*, May 2001, 12-25.

[9] P. MacKenzie and M. K. Reiter. Delegation of Cryptographic Servers for Capture-Resilient Devices. In *Proceedings of the 2001 ACM Conference on Computer and Communication Security*, November 2001, 10-19.

[10] T. Matsumoto, K. Kato and H. Imai. Speeding up secret computations with insecure auxiliary devices. In S. Goldwasser (Ed.) *Advances in Cryptology - Crypto '88*, LNCS 403, Springer-Verlag 1990, 497-506

[11] P. Pfitzmann, M. Waidner. Attacks on Protocols for Server-Aided RSA Computation. In: R. Rueppel (Ed.) *Advances in Cryptology - Eurocrypt '92*, LNCS 658, Springer-Verlag 1993, 153-162.

[12] K. H. Rosen. *Elementary Number Theory and its Applications*. Third edition. Addison-Wesley, 1993.

[13] S. Sovio, N. Asokan and K. Nyberg. Defining Authorization Domains Using Virtual Devices. In *IEEE 2003 Symposium on Applications and the Internet Workshops (SAINT Workshops 2003)*, January 27-31, 2003, Orlando, Florida, IEEE Computer Society Press (2003), 331-336.

[14] D. Stinson. *Cryptography - Theory and Practice*. Second edition. Chapman&Hall/CRC, 2002.

[15] M. Wiener. Cryptanalysis of short RSA secret exponents. *IEEE Transactions of Information Theory*, 36 (1990), 553-558.

Author Index

Al Meaither, Mansour 95
Almegren, Henrik 83
Almgren, Magnus 57
Arnesen, Ragni R. 13

Boldt, Martin 51
Boudriga, Noureddine 25, 71
Buan, Aslak 151

Carlsson, Bengt 1, 51

Danielson, Jerker 13

Ernvall, Anne-Maria 195

Gjerde, Marius 151
Gjøsæter, Terje 109

Hamdi, Mohamed 25, 71
Hansen, Frode 129
Haslum, Kjetil 109

Jacobsson, Andreas 1
Jonsson, Erland 57, 83

Koshutanski, Hristo 143
Krichene, Jihene 25
Kylänpää, Markku 117
Køien, Geir 161

Laud, Peeter 185
Lundin, Emilie 57

Massacci, Fabio 143
Mjølsnes, Stig F. 151
Mäntylä, Janne 117

Nyberg, Kaisa 195

Oleschuk, Vladimir 129, 161

Pöial, Jaanus 175

Rantala, Aarne 117
Roos, Meelis 185

Suomalainen, Jani 117
Söderström, Ola 83

Tolvanen, Jarkko 117
Tounsi, Mahmoud 25

Virtanen, Teemupekka 37

Wieslander, Johan 51
Willemson, Jan 175, 185

Yan, Zheng 37

Zaitseva, Jelena 175
Zhang, Peng 37

ISBN 82-993980-4-5