A Cost Optimizing Model for IT Security

Jüri Kivimaa

Doctoral Thesis in Management | No. 17 | Tallinn 2013

Estonian Business School

Estonian Business School

A COST OPTIMIZING MODEL FOR IT SECURITY

Thesis of the Degree of Doctor of Philosophy
by
Jüri Kivimaa

Tallinn 2013

Department of Information Technology, Estonian Business School, Estonia

The dissertation was accepted for the defense of the degree of Doctor of Philosophy in Management by the Research Council of Estonian Business School on June 20, 2013.

Supervisor: Professor Peeter Lorents, Ph.D., Department of Information Technology, Estonian Business School, Tallinn, Estonia.

Opponents: Professor emeritus Leo Võhandu, Ph.D.
Tallinn University of Technology, Tallinn, Estonia.

Professor Leo Mõtus, Ph.D.
Tallinn University of Technology, Tallinn, Estonia.

Public Commencement on November 6, 2013, Estonian Business School, Lauteri 3, Tallinn, Estonia

Language editor: Rain Ottis

ISBN 978-9949-9447-3-6 (trükis)
ISBN 978-9949-9447-4-3 (pdf)

EBS Print, 3 Lauteri Street, Tallinn

ACKNOWLEDGEMENTS

I would like to thank EBS Education for the chance to enter the doctoral study program in Estonian Business School.

I am also very grateful to the people and organizations that have contributed to my research:

- My deepest gratitude belongs to Professor Peeter Lorents for supervising my academic journey. His encouragement and support over the years have made this thesis possible.

- I wish to thank major Chris Fellows for his support to my research. He was one of the first to understand the value of IT security costs optimization topic.

- I am very grateful to the co-authors of my articles for their amazing ideas that have been instrumental in developing the GSES:
  Enn Tõugu, Andres Ojamaa and Pavel Grigorenko from the Institute of Cybernetics at the Tallinn University of Technology,
  Toomas Kirt from the University of Tartu, and
  Geert Albergs from the Ecole Supérieure d'Informatique Electronique Automatique, Paris, France.

- My special thanks to the IT managers and IT security experts from the Swedbank Baltic and SEB Estonia for cooperation and sharing their expert knowledge about IT security.

- My appreciation to my opponents, Dr. Leo Mõtus and Dr. Leo Võhandu, and all anonymous peer reviewers of my publications for their comments and suggestions.

- And finally, I wish to express my deepest gratitude to my wife Irene, who supported me through all the years of working on my dissertation!

CONTENTS

LIST OF TABLES.

LIST OF FIGUR

LIST OF ABBREVIATONS

AIM	*Astmeline infoturbe mudel* (GSM)
AIES	*Astmeline infoturbe ekspertsüsteem* (GSES)
ALE	Annual Loss Expectancy
ARO	Annual Rate of Occurrence
DSS	Decision Support System
GS	Graded Security
GSM	Graded Security Model
mb_GSM	matrix-based Graded Security Model
g_AIM	*graafipõhine astmeline infoturbe mudel* (gb_GSM)
gb_GSM	graph-based Graded Security Model
GS	Graded Security
GSES	Graded Security Expert System
GUI	Graphical User Interface
HW	Hardware
ICT	Information Technology
IS	Information System
ISKE	Three-level IT baseline security system ISKE / ***I**nfo**S**üsteemide **K**olmeastmeline **E**talonturbe süsteem*
ISO	International Organization for Standardization
ITS	IT System
ITIL	IT Infrastructure Library
LAN	Local Area Network
NISPOM 2006	National Industrial Security Program Operating Manual (February 2006), US, CSA (Cognizant Security Agency denotes the Department of Defense - DoD, the Department of Energy - DOE, the Nuclear Regulatory Commission - NRC, and the Central Intelligence Agency – CIA).
NIST	National Institute of Standards and Technology, US
NIST SP 800-53r4	NIST Special Publication 800-53 Revision 4
Rc	Redundancy coefficient
RIA	Estonian Information System's Authority / *Riigi infosüsteemi amet*
ROI	Return On Investment
S_I	Study I
S_II	Study II
S_III	Study III
S_IV	Study IV
S_V	Study V
S_VI	Study VI
S_VII	Study VII
SE	Security Effectiveness
SLA	Service Level Agreement

SME	Small and Medium Enterprise
SW	Software
WAN	Wide Area Network

ABSTRACT

Nowadays it is impossible to manage any business effectively without information systems. IT has become ubiquitous and practically all companies have to view IT as a common, yet very critical, resource to their success. However, common resources generally do not provide any substantial competitive advantage. Therefore, the new rules for IT management are to spend less and to focus on lowering residual risks.

As a result, all organizations have to be optimal in IT and IT Security. The competitive advantage from IT is ensured mainly by the price and security of IT and their impact on the net cost of the service(s) and product(s) provided by the company.

This thesis describes the development of a graph-based Graded Security Model for IT Security and the cost optimization software prototype called Graded Security Expert System. This is a new and dynamic decision support system that allows IT and IT Security management to make reasoned urgent managerial decisions based on calculated values of interest – the maximum possible IT Security effectiveness or minimum IT Security Total Costs as a function from IT/IT Security Budget in a given budget range. A Graded Security Model is proposed, which binds security measures with their costs and security effectiveness. In addition, the Graded Security Expert System (a software tool/utility) is proposed, in order to realize bi-objective optimization to calculate the Pareto-optimal curve for IT security costs and achieved security level – providing information to managers in a visual and easily understandable form.

The GS Model and the GS Expert System will allow IT Security experts to customize the IT Security measures to meet their specific requirements in a way that is optimal for their organization. It will also be easier to justify security expenses to management - i.e. the gap between managers and IT Security experts can be substantially narrowed.

GSM/GSES can be used as a decision support tool for IT governance – to make justified decisions about future IT security investments in order to achieve optimal security, i.e. to achieve the required or optimal security level with the minimum total cost. In short, this approach saves money.

I.e. Decision Support Systems (such as ours) could help the management to make better management decisions and thereby provide a competitive advantage to the institution.

However, GSES is also suitable for any business process optimization, if:

- is able to describe a corresponding graph-based business process model and
- the sub-processes can be described and implemented with grades, and if the grades costs and effectiveness values can be defined.

Keywords: IT security, IT Security graph-based model, Graded Security Model, Graded Security Expert System, IT security costs optimization.

List of original articles in proceedings

S_I. Kivimaa, Jyri; Ojamaa, Andres; Tyugu, Enn (2008). Graded security expert system. *In: CRITIS 2008 : Third International Workshop on Critical Information Infrastructure Security, Villa Mondragone, Monte Porzio Catone, Rome, October 13-15, 2008, (Pre-Proceedings):* AIIC, ENEA, 2008, 333–339. (3.4)
Kivimaa, Jyri; Ojamaa, Andres; Tyugu, Enn (2009). Graded security expert system. *In: Critical Information Infrastructure Security: Third International Workshop, CRITIS 2008, Rome, Italy, October 13-15, 2008, Revised Papers.* Berlin: Springer, 2009, (Lecture Notes in Computer Science; 5508), 279–286. (3.1)

S_II. Ojamaa, Andres; Tyugu, Enn; Kivimaa, Jyri (2008). Pareto-optimal situation analysis for selection of security measures. *In: MILCOM 2008: Assuring Mission Success: Unclassified Proceedings, November 17-19, 2008.* San Diego: 2008, 3224–3230. (3.1)

S_III. Kivimaa, Jyri (2009). Applying a Cost Optimizing Model for IT Security. *In: Proceedings of the 8th European Conference on Information Warfare and Security: Lisbon, Portugal, July 6–7, 2009.* UK: Academic Conferences Limited, 2009, 142–153. (3.1)

S_IV. Kivimaa, Jyri; Ojamaa, Andres; Tyugu, Enn (2009). Managing evolving security situations. *In: MILCOM 2009: Unclassified Proceedings, October 18–21, 2009, Boston, MA:* Piscataway, NJ: IEEE, 2009, 1–7. (3.1)

S_V. Kirt, Toomas; Kivimaa, Jyri (2010). Optimizing IT security costs by evolutionary algorithms. *In: Conference on Cyber Conflict Proceedings 2010: Conference on Cyber Conflict; Tallinn, Estonia; June 15-18, 2010.* Tallinn: Cooperative Cyber Defence Centre of Excellence Publications, 2010, 145–160. (3.4)

S_VI. Kivimaa, Jyri; Kirt, Toomas (2011). Evolutionary Algorithms for Optimal Selection of Security Measures. *In: Proceedings of the 10th European Conference on Information Warfare and Security: Tallinn, Estonia, July 7–8, 2011.* UK: Academic Conferences Limited, 2011, 172–184. (3.1)

S_VII. Alberghs, Geert; Grigorenko, Pavel; Kivimaa, Jyri (2011). Quantitative system reliability approach for optimizing IT security costs in an AI environment. *In: 12th Symposium on Programming Languages and Software Tools, SPLST'11: Tallinn, Estonia, October 5–7, 2011, Proceedings.* Tallinn: TUT Press, 2011, 219–230. (3.2)

My contribution to the development of the GSM/GSES method is:

- development of graph-based GSM,
- selecting a suitable metrics for a IT security costs optimization,
- enhancing the Business Process overall availability (alias effectiveness) calculation algorithms for a IT security – added was IT security specific not-full-redundant alias Rc-redundant parallelism,
- defining the required functionality for GSES, and
- conducting the case study to test the method.

Excellent ideas from my co-authors:

Enn Tõugu and Andres Ojamaa from the Institute of Cybernetics:

- to use CoCoViLa as development platform for GSES,
- discrete dynamic programming algorithm to calculate the optimal Pareto-frontier for a Dependency Matrix-based version of GSM.

Toomas Kirt from the University of Tartu:

- developing evolutionary algorithms for optimization in GSM/GSES,
- optimizing the parameters for the evolutionary optimization algorithm.

Geert Albergs from the Ecole Superieure d'Informatique Electronique Automatique, Paris, France:

- adding capabilities for calculation and optimization of indefinetly complex graphs (allowing bridge and star structures),
- developing the SW for GSES for graph-based GSM.

INTRODUCTION

The Relevance of the Topic

Nowadays it is impossible to manage any business effectively without information systems. IT has become ubiquitous and practically all companies have to view IT as a common, yet very critical, resource to their success. New rules for IT management are to spend less and focus on risks but not on competitive advantage opportunities (Carr 2003).

General solutions do not provide competitive advantages. For a ubiquitous service or product the competitive advantage mainly depends on the price and security of supply and their impact on the net cost of the business service(s) or product(s). A good example from the past is the introduction of electricity - there was a period where electricity was a rare commodity and gave a very significant competitive advantage. Today, however, electricity is so common that no competitive advantage comes from it.

The same situation can now be seen in IT and IT Security:

- IT have become unavoidable and very important for all businesses,
- it is no longer possible to gain a relevant competitive advantage through IT solutions alone,
- when a resource becomes essential to competition but inconsequential to strategy, the risks it creates become more important than the advantages it provides (Carr 2003) – i.e. IT security have become very important for all businesses,
- in general the most IT systems in real world are not secure because security is presumed as too expensive,
- in order to compete based on the price of the service(s) or product(s), the IT and IT Security spending must be optimized.

Therefore, the relevance of the IT and IT security costs optimization has significantly increased and nowadays they must be included in all business plans and business analysis documents. This means that up-to-date IT Security standards, Best Practices and methodologies must have the IT Security cost optimization functionality.

Research Aim and - Tasks

PDCA (plan–do–check–act or plan–do–check–adjust) is an iterative four-step management method used in business for the control and continuous improvement of processes and products.[1]

This thesis addresses the plan-phase for Information Security process – to establish the objectives and processes relevant to managing risks and to deliver IT Security in accordance with the expected output – the required or needed IT Security level. An important question in information security is how to allocate the budget among all possible security measures/activities in order to achieve the maximum security level or effectiveness possible. To answer this question, we developed an economic model.

Currently, the most popular information security standards, Best Practices, etc. fail to adequately address the problem of information security optimization. The choice of security measures is often very subjective and the security gains are several times lower than what is possible to get for the same amount of money.

The research question for this thesis is:
How to determine for enterprise optimal IT Security– i.e. the optimal cost of security measures for IT Security and/or Cyber Security, to achieve the required (by law, by contracts) and business needed IT security level?
NB! In essence, optimal IT security costs mean the optimal list of IT security measures that need to be implemented.

The research aim is:
to develop a decision support system for IT Governance, in order to make reasoned and optimal decisions about investments to IT security with the volume of work which is also acceptable to small and medium-sized businesses.

More concretely the aim of the thesis is:

1. **To develop the optimization method for information security spendings/costs.**
2. **The optimal IT security Cost found will also define the optimal security profile – i.e. the optimal security measures list.**
3. **All previous can be done with tolerable work-capacity** (i.e. with tolerable labor cost) **for SME's (» 1-2 man-months).**

[1] http://en.wikipedia.org/wiki/PDCA

The hypothesis:
the described Graded Security Model (GSM)

- ***is sufficiently detailed for accurately optimizing information security costs,***
- ***is also sufficiently simple, so that information collection, optimization and analysis tasks require an order of magnitude less work than the widely used detailed risk analysis (implementing GSM requires 1-2 man months of work; the equivalent work with the detailed risk analysis requires 1-2 man years).***

The Graded Security Model, which describes information security as a process, is based on approaches that are as simple and as widely applicable as possible, including:

- viewing information security strictly within the frame of best practices and standards for determining IT system security measures (this is considerably simpler than, for example, the business risk based view) and
- describing IT and information security processes with the People-Process-Technology approach (Figure 3) is considerably simpler than, for example, the ISACA BMIS (Figure 2).

Consider an expert system that is based on a simple (easy to understand) graph model, which dwells from the IT view on information security.[2] The expert system saves an order of magnitude in work that would be needed to get the information the company's management needs for decision making in the field of information security investments. This information visualizes the dependency between security effectiveness and the resources allocated for security - *SecurityEffectiveness=f(SecurityCosts)*. The main point is that an order of magnitude less granular model generates the required information with an order of magnitude less work. In order to develop the decision-support-system and to be sure that it adequately describes the real situation the following steps must be taken:

1. building a descriptive model,
2. testing the model, and
3. implementing the model.

Building a descriptive model

There are several sub-problems that need to be solved:

1. a model that supports information security specific aspects and describes information security as a process,
2. a metric that is suitable for making information security decisions,
3. algorithms for calculating information security effectiveness, as well as an algorithm suitable for optimization,
4. requirements for a software implementation, choosing a suitable software platform.

[2] View based on security goals (mainly CIA – confidentiality, integrity, availability) and security measures from Best Practices (standards are Best Practices too) needed to attain them – What has to be done?

These problems are addressed in detail in the theoretical part of the thesis. The final result is the description of the Graded Security Model (GSM) and its implementation in the Graded Security Expert System (GSES).

Testing the model

A critical case is defined as having strategic importance in relation to the general problem. A critical case allows the following type of generalization: "If it is valid for this case, it is valid for all (or at least many) cases."[3]

In order to be sure that a model and its assumptions are correct, one needs to be able to gather relevant expert data and do an *a posteriori* check of the model. In other words, theoretical ideas should be verifiable and verified by practice. Therefore, I undertook a case study from the IT Security front line – from two biggest banks in Estonia. The case study gave two key results:

1. it is possible to get the relevant information from experts,
2. the model produces a result that corresponds sufficiently with reality.

Used is the *a posteriori* method, which takes the information security situation of the Bank on year X and compares it to the results of the model for year X (mainly the losses from security incidents – real and calculated). If the two are sufficiently close, then we can assume that the model is good enough to optimize the information security investments for the year X+1. Obviously, the model must take into account the changes in IT and information security (at institutional level and global level) that have occurred during the year.

Implementing the graph-based Graded Security Model (gb_GSM)

While implementation in companies is not directly the subject of this thesis, it is the next logical step if the test results are satisfying. In order to effectively implement gb-GSM, the accounting and risk analysis systems of the company need to be adjusted, so that we can get the most important inputs (Costs from accounting, and Confidence and Losses from risk management) with a click of the mouse, thus reducing the work load considerably.

[3] http://en.wikipedia.org/wiki/Case_study

Methods Used in the Research

The main research methods used in the fields of Economics and Commerce are:

- **Empirical and experimental research-based projects**. These include surveys, statistics, questionnaires or fieldwork.
- **Theoretical projects**. These tend to look mainly at conceptual issues.
- **Case studies**. These involve analysis of real world problems of which one has experience or is able to observe. (CASE STUDIES: Research Methods)

The method used for developing a descriptive model of IT Security

The Graded Security Model (GSM) was developed using the theoretical method. Existing models were compared and analyzed, and a novel graph-based general IT and information security model for enterprises and the corresponding optimization algorithms were synthesized.

The method for testing and that will be used in future implementations of the model

The method that was used in testing and will be used in future implementations of the GSM is case study.

Information security is very enterprise specific – somewhat similar to the uniqueness of human fingerprints. There are approximately to 4^{40} or 10^{26} different realistic variations for implementing strong (at the level of the banking sector) information security (Figure 5). For SME's, there are significantly less variations (approximately 10^{10}). Therefore, implementing the model in a real company can only be based on the specific case study of that company – it is only possible to optimize the specific information security of the specific company. However, existing IT security cost optimization case studies are very useful as guides for new case studies – allowing significant (up to an order of magnitude, if the cases are similar) savings in work hours.

It is highly recommended to review theoretical scientific ideas in their real-life context:

1. **Can we collect the necessary source information?**
2. **Do we get a result that matches the reality?**

This research uses the *a posteriori* look at information security as it is being performed in real life in selected organizations. **This involves describing the information security situation of previous years (in hindsight, we can be very accurate, precise and smart).**

3. **Is the model generally applicable?**

To be more confident in generalizing from case studies, a critical case is defined as having strategic importance in relation to the general problem.

A critical case allows the following type of generalization, 'If it is valid for this case, it is valid for all (or at least many) cases.' In its negative form, the generalization would be, 'If it is not valid for this case, then it is not valid for any (or only for few) cases.'[4]

"**A case study** is an empirical inquiry that investigates a contemporary phenomenon within its real life context, especially when the boundaries between phenomenon and context are not clearly evident". (Yin 1994)

The GSM/GSES model is intended as a prototype model that can be adapted to concrete enterprises on a case by case basis – a concrete optimization for a particular organization.

Assumptions

Assumptions and their rationality

The work includes two types of assumptions:

- Rational assumptions – the main goal is to ensure a lower work load when implementing the method.
- Forced assumptions – mostly caused by the lack or quality of the corresponding information. The underlying problems in such cases remain unsolved and may merit separate research in the future.

Rational assumptions:

1. We base our work on the IT view on information security – i.e. view based on security goals (mainly CIA – confidentiality, integrity, availability or Security Effectiveness) and security measures needed to attain them (more details on page 26).
2. We base our work on the simplest and most widely used People-Process-Technology business process model (more details on page 38-39).
3. We exclude decisions, which are clearly bad.
   Therefore, we have possibility first develop the prototype GSES, which performs the effectiveness calculations and optimization, but does not address the problems of searching for and excluding technical and human errors (more details on page 72).
4. We often use the term *"function"* in a simplified fashion – meaning that we only look at the dependency from the variable that interests us most. From the costs perspective, this is mainly a'la *f(SecurityCosts)* or *f(SecurityBudget)* (more details on page 48-49).

[4] http://en.wikipedia.org/wiki/Case_study

Forced assumptions:

1. In most cases it is impossible to find or specify numeric values for the cyber attack probabilities. Since we do not have that information, we must assume that unprotected valuable information will definitely be attacked.
2. In the Case Study, the costs are based on the total IT costs of the Bank, since it is very difficult to separate the IT and IT Security costs. I.e. we have to describe an IT and IT security Business Model (more details on page 37).
3. In the public sector (including the military) there is no real information about probable losses from security incidents in terms of money (at least for Estonia). It means that in the public (including military) sector we must be content with the second stage of optimization – the "Do things right" stage (i.e. maximal security effectiveness with the money we have) (more details on page 50).

The Originality of the Research and Its Practical Merit

The cyber security field of research is rapidly developing. Over the last decade, there have also been developments in the area of IT security cost optimization (multi-layer models, etc.), but there is still a lack of understanding of the principles of IT security optimization. Specifically, there is no systematic and consistent treatment of information security as an important and expensive business process.

The main contribution of my research is as follows:

- Adding new knowledge to the field of IT and Cyber Security by offering a new graph-based Graded Security Model to efficiently handle institution-level IT Security.
- Development of the GSM/GSES method that, based on IT Security graph-model and evolutionary optimization algorithms, will allow IT and IT Security cost optimization at the planning stage by estimation of the quantitative behavior of the IT Security system, and could be used as a decision-support system for IT Governance. The optimization is achieved by using mathematical models in the strategic management of the organization's resources (through the decision-making process).
- Decreasing the gap between IT Security experts and the management by allowing the experts express their thoughts in visual and understandable form.
- Illustrating the need to address cost optimization in IT Security standards, models, regulations and policies.

I hope that the proposed IT and IT security costs (investments) optimization will improve the IT Security level in institutions especially in situations where there is a shortage of IT Security resources, meaning that the optimal use of the resources is required.

PART I. A BRIEF OVERVIEW ABOUT PREVIOUS IT SECURITY OPTIMIZATION APPROACHES

If we want to measure and optimize the information security in an enterprise (basically, information security spending), then we must consider the real (already in place or planned) security measures. Therefore, we are interested in the information security cost optimization models that also aim to define the necessary security measures.

So we were looking for an information security model:

1. That allows IT security costs optimization.
2. Where the optimal IT security cost found will directly define the optimal security profile – i.e. the optimal security measures list.
3. Where all previous can be done with tolerable work-capacity (i.e. with tolerable labor cost) for SME's (» 1-2 man-months).

I.1. IT Security cost optimization problematic in popular security standards and models.

In 2009 we made a considerable effort to find such a model/method, investigating about 800 sources in cooperation with the security experts from SEB Estonia.

A short summary of the literature-based analysis

1. **Is many interesting standards, Best Practices and models, that are not helpful for actual IT security cost optimization and are not involved in defining needed security measures too (i.e. interesting, but not met the requirements 1 and 2):**
   - COBIT (IT Governance Institute) – a collection of international Best Practices that regulates IT management and auditing.
   - ISACA (Information Systems Audit and Control Association), membership for IT Audit, Security, Governance and Risk Professionals.
     The Business Model for Information Security: ISACA, 2010.
   - SSM CMM (Systems Security Engineering Capability Maturity Model v3) and ISM (Information Security Management Maturity Model v2) - Maturity Models.
   - Common Criteria (CC) for IT Security Evaluation (ISO/IEC 15408) - successor of Orange Book, TCSEC and ITSEC, it allows many different software applications to be integrated and tested in a secure way.
   - ITIL (Information Technology Infrastructure Library for IT Service Management) planning and managing IT services, service delivery, control, continuity, budgeting, accounting, problem management, configuration and change management, continual service improvement, ITIL is frequently used as a method of preparation for achieving ISO/IEC 20000 certification.

- Attack Trees, think-like-an-attacker models:
  - Rinku Dewri, Nayot Poolsappasit, Indrajit Ray and Darrell Whitley, *Optimal Security Hardening Using Multi-objective Optimization on Attack Tree Models of Networks*.
  - Ahto Buldas, Peeter Laud, Jaan Priisalu, Märt Saarepera, and Jan Villemson, *Rational Choice of Security Measures via Multi-Parameter Attack Trees.*
  - Aivo Jürgenson and Jan Willemson, *Processing Multi-Parameter Attack trees with Estimated Parameter Values.*
- Attack models based on game theory – very interesting models to handle the attack component, but plagued with lack of necessary source information and expert knowledge:
  - Schlicher, Bob G., and Abercrombie, Robert K. *Information Security Analysis Using Game Theory and Simulation*. Oak Ridge National Laboratory, Oak Ridge, TN 37831 USA.
  - Grossklags, J., Christin, N., and Chuang, J. 2008. *Secure or Insure? A Game-Theoretic Analysis of Information Security Games.*
- Ontology based models – new and very interesting type of information security models, which allow the creation of more understandable detailed risk assessment models:
  The ontology can be used as a general vocabulary, roadmap, and extensible dictionary of the domain of information security. With its help, users can agree on a common language and definition of terms and relationships. In addition to browsing for information, the ontology is also useful for reasoning about relationships between its entities, for example, threats and countermeasures. The ontology helps answer questions like: Which countermeasures detect or prevent the violation of integrity of data? Which assets are protected by SSH? Which countermeasures thwart buffer overflow attacks? (Herzog Shahmehri Duma 2007)
  - Security Ontology Aurum, ttp://securityontology.securityresearch.at/aurum/.

2. In the following section we will take a deeper look at the models that deal with defining security measures (to meet the requirement 2).

Two different methods or viewpoints are used for that. These two views on information security are very nicely included in the ISO/IEC 27000 series of standards:

- From the business viewpoint, i.e. view based on risks and controls to avoid them:
  Information security is the protection of information from a wide range of threats in order to ensure business continuity, minimize business risk, and maximize return on investments and business opportunities. (ISO 27002: 2008)

- From the IT viewpoint, i.e. view based on security goals (mainly CIA – confidentiality, integrity, availability) and security measures needed to attain them:
  Information security is the preservation of confidentiality, integrity and availability of information; in addition, other properties, such as authenticity, accountability, non-repudiation, and reliability can also be involved. (ISO 27000: 2008)

Risk analysis based IT Security methods (standards, Best Practices, models)

These methods are based on analysis of business risks and mainly used for control and audit. The main question is whether discovered business risks are reduced with appropriate security measures to residual risks of appropriate level?

- ISO/IEC 27005:2010 Information technology - Security techniques - Information security risk management.
- ISO/TR 13569:2005 Financial services. Information security guidelines.
- Information Security Forum (ISF) methods for risk assessment and risk management.
- Most used business risks and their detailed analysis based models [5]:
  - **CRAMM** (CCTA Risk Analysis and Management Method), UK, is a Qualitative Risk Analysis and Management Tool, developed by UK government's Central Computer and Telecommunications Agency (OGC since April 2001) in 1985.
    Pricing and licensing models (December 2005): CRAMM expert £2950 per copy plus £875 annual license, CRAMM express £1500 per copy plus £250 annual license, sectors with free availability or discounted price : UK Government, NATO, UK local authority, NHS, Academic;
  - **EAR/Pilar Magerit**, Spain, EAR/Pilar is the software that implements and expands Magerit RA/RM Methodology, first released in 2004. EAR is commercial and PILAR is public administration restricted, its functionalities include mainly: quantitative and qualitative Risk Analysis and Management, and quantitative and qualitative Business Impact Analysis & Continuity of Operations.
    Pricing and licensing models (December 2005): EAR 1500€, sectors with free availability or discounted price: Educational world-wide, Spanish Public Administration.
  - **Octave v2.0** (and Octave-S v1.0 for Small and Medium Businesses); USA; initiators of the product are Carnegie Mellon University (USA) and CERT (Computer Emergency Response Team); defines a risk-based strategic assessment and planning technique for security. Price: Free.

[5] http://rm-inv.enisa.europa.eu/methods

Detailed risk analysis based security models are quite complicated. There are ~ thousand risks to avoid, which leads to ~ thousand possible incident trees and ~ thousand security measures to choose and implement. As a result that model is very complicated and labor intensive, especially when one considers the developing, maintenance and implementation workload in man-years.

A detailed risk analysis (driven by business risk) is commonly used to determine necessary security measures in large enterprises, which are typically located in large countries. In smaller countries like Estonia, companies typically do not conduct a full scale detailed risk analysis, since it is too expensive. As a result, SME's are forced to use some other security model. However, it should be noted that the frugal solutions available to SMEs can in certain cases be very useful (for example, determining the optimal security spending profile) or absolutely necessary (for example, in crisis situations where there are not enough time or people) for large enterprises as well.

In addition, detailed (business) risk analysis based models are not well suited for general optimization of security costs. The widely used Return on Investment (ROI) focuses on single security solutions.

Information Security activities-centric methods (Standards and Best Practices[6])

The security measures and activities in the following list have led to a good information security status (meaning the needed confidentiality, integrity and availability of information) in many enterprises. In any enterprise, this list should be followed (by complying) in order to achieve a good information security status.

Baseline models – comply or explain:

- ISO/IEC 27002, „Information technology - Security techniques – Code of practice for information security management" is a baseline IT Security model and has adaptations in several other countries as national equivalent standards, such as Australia and New Zealand (AS/NZS ISO/IEC 17799:2006), Netherlands (NEN-ISO/IEC 17799:2002), Denmark (DS484:2005), Sweden (SS 627799), Japan (JIS Q 27002), Spain (UNE 71501), United Kingdom (BS ISO/IEC 27002:2005), and others.
  ISO/IEC 27000-series (also known as the "Information Security Management Systems Family of Standards' or "ISO27k" for short) from International Organization for Standardization (ISO, the world's largest developer of standards) is very capacious (~25 different IT Security standards) and popular. The series provides best practice recommendations on information security management, risks and controls within the context of an overall information security management system (ISMS).

[6] All Standards are in essence Best Practices too.

- IT-Grundschutz / IT Baseline Protection Manual, BSI (Bundesamt für Sicherheit in der Informationstechnik, German Federal Office for Information Security), Germany:
  - the most up-to-date-version in German is: IT-Grundschutz Catalogues - IT-Grundschutz-Kataloge,12. Ergänzungslieferung - September 2011;
  - the English version of the IT-Grundschutz Catalogues is available in the pdf-format: IT-Grundschutz Catalogues 2005 (BSI).
- PCI Security Standards Council, Payment Card Industry Data Security Standard v2.0, October 2010.

None of these baseline models address the prudence and optimality of costs. Instead, they address the ROI of single security solutions.

Multi-level models.

High-level risk analysis, i.e. security objectives (at least C, I, A) and their needed/required levels, is the foundation for a defined necessary security measures list.
In recent years, three-level models (High/Medium/Low) have become quite popular. For example, the US NIST sp800-53 (currently the "Final Public Draft of NIST Special Publication 800-53 Revision 4") and the Estonian ISKE v6.00 (Table 1).

There is also one multi-level model that has 27 levels of security – the NISPOM 2006 model for Critical Infrastructure IT Systems in USA (Table 2).

- ISKE v6.00

A model that has 64 levels of security requirements, but for some reason has reverted to three (High/Medium/Low) levels for defining security measures (Table 1).

Security Goals:
- **Confidenciality,**
- **Integrity,**
- **Availability.**

Security Goals Levels:
0 ÷ 3.

Security Levels:
- **Low - L,**
- **Medium - M,**
- **High - H.**

		A0	A1	A2	A3
I0	C0	L	L	M	H
	C1	L	L	M	H
	C2	M	M	M	H
	C3	H	H	H	H
I1	C0	L	L	M	H
	C1	L	L	M	H
	C2	M	M	M	H
	C3	H	H	H	H
I2	C0	M	M	M	H
	C1	M	M	M	H
	C2	M	M	M	H
	C3	H	H	H	H
I3	C0	H	H	H	H
	C1	H	H	H	H
	C2	H	H	H	H
	C3	H	H	H	H

Table 1. The Dependency Matrix of ISKE linking security goals levels and security levels

- **NISPOM 2006** (National Industrial Security Program Operating Manual), DoD, USA

This standard (with 27 levels) has made a significant step in addressing the prudence of costs. Our original idea was to base GSES on the Graded Security matrix model developed (DoE 1999) and updated (NISPOM 2006) in USA. The Graded Security Model (GSM) in NISPOM (National Industrial Security Program Operating Manual) 2006 approach is used to express the relations between information security goals and security activities domains (logical groups of security controls and measures).

	Confidentiality Protection Level		
Requirements (Paragraph)	P L 1	PL 2	PL 3
Audit Capability (8-602)	Audit 1	Audit 2	Audit 3 Audit 4
Data Transmission (8-605)	Trans 1	Trans 1	Trans 1
Access Controls (8-606)	Access 1	Access 2	Access 3
Identification & Authentication (8-607)	I&A 1	I&A 2,3,4	I&A2,4,5
Resource Control (8-608)		ResrcCtrl 1	ResrcCtrl 1
Session Controls (8-609)	SessCtrl 1	SessCtrl 2	SessCtrl 2
Security Documentation (8-610)	Doc 1	Doc 1	Doc 1
Separation of Functions (8-611)			Separation
System Recovery (8-612)	SR 1	SR 1	SR 1
System Assurance (8-613)	SysAssur 1	SysAssur 1	SysAssur 2
Security Testing (8-614)	Test 1	Test 2	Test 3

	Integrity Level of Concern		
Requirements (Paragraph)	Basic	Medium	High
Audit Capability (8-602)	Audit 1	Audit 2	Audit 3
Backup and Restoration of Data (8-603)	Backup 1	Backup 2	Backup 3
Changes to Data (8-604)		Integrity 1	Integrity 2
System Assurance (8-613)		SysAssur 1	SysAssur 2
Security Testing (8-614)	Test 1	Test 2	Test 3

	Availability Level of Concern		
Requirements (Paragraph)	Basic	Medium	High
Alternate Power Source (8-601)		Power 1	Power 2
Backup and Restoration of Data (8-603)	Backup 1	Backup 2	Backup 3

Table 2. The Tables from NISPOM 2006 linking measure-groups and CIA security goals.

These multi-level models are the first significant steps towards optimizing information security costs. All these models directly address the prudent relationship between information security measures and the corresponding costs. However, they are quite far from information security cost optimization (that is – achieving the maximum or required security effectiveness with minimum costs).

The main problem of information security is to find the general optimum of costs, or in other words, to "allocate existing information security resources in a way that ensures the maximum information security level across the enterprise".

In summary, standards, Best Practice and compliance security models have not adequately addressed the security optimality question – not methods for IT and information security overall system level optimization. Not methods to calculate the *"function" Security Effectiveness = f (Security Costs)*.

3. However, we should note the economic models sub-group, which includes several general economic models about investments in Information security.

The most interesting of these are:

- Olovsson T. (1992), "A structured Approach to Computer Security"– an interesting basic idea about optimal IT Security Costs: we have to find 'minimal TotalCost = SecurityCosts+SecurityLosses'.
- CyberProtect ver 1.0 (1999) and ver 2.0 (2010), DOD Information Assurance Training & Awareness Product (interactive training exercise) to select and optimize security controls for a particular organization. They present the effectivenesses and costs of nine defensive security tools. It is a very nice basic idea and example, although quite simplified (in reality there are at least 30-40 fields of activity in cyber security). These products are basically educational games, but they do not explain or explore the theoretical side. It is also not possible to update or improve these models. Therefore, they represent an interesting idea, but are not suitable for real life implementation.
- Gordon, Lawrence P., and Loeb, Martin P. (2002), "The Economics of Information Security Investment", ACM Transactions on Information Systems Security, November 2002, ppg 438-457. According to their paper, we should use no more than 37% of potential loss for security costs. However, this approach is not very useful in finding the optimal costs, since the latter are generally up to an order of magnitude smaller than the proposed value.
- Duffany J.L. (2007), "Optimal resource allocation for securing an enterprise information infrastructure." The approach involves analyzing attacks versus defence (the user needs expert knowledge about both).
  An economic model is developed which will indicate the cost or penalty from not adopting any countermeasure and the resulting mitigation factor which results from adopting a particular countermeasure or combination of countermeasures. (Dyffany 2007)

This is a very interesting theory, but model will be very complicated and labor intensive. Also practically there is not enough initial information to implement it in practice. Especially, there is not enough available expert knowledge from the attacker perspective. Hackers do not approach their attacks in a very systematic way. They are more interested in breaching the system than documenting, analyzing and publishing their processes (especially in the scientific literature).

In conclusion:

We did not find an IT security cost optimization model/method with all the desired three properties, but the analysis raised several good ideas that could be adopted for future research and development of our own version. In particular:

1. Good ideas to adopt from General economic models:
   - from CyberProtect the concept of graded/leveled security measures/ activities groups and to use their Cost and Effectiveness values as basic for bi-dimensional optimization, and
   - the principle/"function" from Olovsson that in IT security 'TotalCosts = SecurityCosts + SecurityLosses'.
2. Good models to adopt from IT security standards and Best Practices area:
   - to transfer from the optimal IT security Cost and the corresponding IT security profile to the optimal security measures list we can use IT-centric multilevel (graded) standards/models such as NISPOM 2006, NIST sp800-53 r4, ISKE v6.0 (or other similar models; at least the three level H/M/L-models have become very popular). Of course can we also describe a specific IT-centric multilevel (graded) model for our concrete institution ourselves, and hopefully we will have a model that describes the real situation more accurately, but the necessary volume of work is going to be especially large.

We should also note about quite popular but improper approach for enterprise information security cost optimization - namely, ROI analysis of IT Security measures/activities is insufficient for IT Security cost optimization. Important security activities follow the same logic as the strength of the chain – it is as strong as the weakest link. Similarly, the overall security is strong only when all the relevant security components are strong.

PART II. THE THEORETICAL PART

The labor-intensiveness of defining detailed risk analysis based information security measures and the fact that practically all standards and Best Practices are meant for large enterprises (they are very far from optimal from the perspective of SME's, meaning that the information security costs would be too high compared to the value of the protected information) has led to a situation where the information security situation is especially weak at SME's. Considering that SME's are responsible for 90-95% of GDP (depending on the country) and that cyber crime is becoming more and more automated (allowing the criminal to earn significant sums by performing many attacks that individually would bring in a small amount of money), it is very important to substantially raise the general information security level of SME's. At the same time, even large enterprises (with plenty of resources) may sometimes prioritize time in their information security decision making process (reaction speed). Therefore, they would need a quick alternative solution for determining information security measures and costs.

The main goals for the thesis are:

1. **to develop an enterprise information security model, which is less labor intensive than existing solutions (and therefore more suitable for SME's),**
2. **to define the suitable security metrics, security effectiveness calculation and optimization algorithms for this model,**
3. **and to develop the corresponding software prototype for concrete optimizations.**
   NB! In essence, optimal IT security cost means the optimal list of security measures that need to be implemented – i.e. practically we are developing a security cost optimization tool/utility for a graded IT security standards or Best Practices.

As example NISPOM 2006, NIST SP 800-53 r4, and ISKE v6 are very suitable for that. Of course is possible to self-describe such model too, but the workload is going to be a major.

To explain the goals a little, the thesis aims to provide a simple and easy to understand graph model (based on IT view on information security), and to implement it as an expert system, in order to save an order of magnitude in work (compared to detailed risk analysis) that would be needed to get the information the management needs for decision making in the field of information security investment. Mainly information that visualizes the dependency between security effectiveness (SE) and the resources allocated for security - *SE=f(Costs)*.

In order to achieve the main goals, several sub-problems have been solved, including the creation, argumentation and presentation of:

1. **a model that supports information security specific aspects and describes information security as a process,**
2. **a metric that is suitable for making information security decisions,**
3. **algorithms for calculating information security effectiveness, as well as an algorithm suitable for optimization,**
4. **requirements for a corresponding expert system, choosing a suitable SW platform, developing the SW.**

These problems are addressed in detail in the theoretical part of the thesis. The final result is the description of the Graded Security Model (GSM) and its implementation in the Graded Security Expert System (GSES).

We have used an approach, which has not yet been described in detail – the IT view on information security. This is grounded in necessary security measures (not business risks) and it creates the possibility to use an order of magnitude less granular models and to have an order of magnitude lower workloads.

The collection and updating of necessary data for the graded information security model is an order of magnitude less work (approximately a few man-months) than the corresponding detailed risk assessment. In fact, it is possible to achieve the first (so-called rational) result in a few man-days, using a simplified and optimized approach.

The requirements for a good enterprise IT Cost Optimization Model are:

1. The ability to get the necessary source information (statistical data or expert assessment).
2. The model calculates the specific optimum for two criteria (multi-objective optimization) for a given enterprise – meaning the Pareto-optimal (Pareto-effective) distribution, thus finding maximum results with minimum cost.
3. The model should give as the answer the integral security level of the entire information security system and to follow two key principles:
    - for main fields of activity (relevant) – the strength of the chain is determined by the weakest link,
    - for supporting fields of activity – information security is multi-level: so-called Multi-Level Security (MLS), layered defense or Defense in Depth (DiD). The latter is a key concept in IT security. It posits that no single defense is adequate for IT security. Progress towards improved security posture involves understanding threats and vulnerabilities and arraying a multiple layered (and evolving) defense.
4. Less labor intensive compared to other analogous models.
5. We must be able to verify the model by *a posteriori* solving the optimization task for previous year(s) and comparing the model results with the real

results. Since this involves extensive calculations, a software tool/expert system (prototype) must be developed.

Our research is a part of economic models subgroup and uses Cost-Effectiveness analysis as a metric. The results allow the developers of information security standards to comprehend the economic importance of information security cost optimization and to use economic models as tools/utilities to economize resources for IT security, and thereby ensure the overall economic success for the enterprise.

II.1. A model that describes information security as a process

Based on the analysis in A BRIEF OVERVIEW ABOUT PREVIOUS IT SECURITY OPTIMIZATION APPROACHES, we determined the good starting ideas for an easily understandable and less labor-intensive IT security cost optimization model:

1. base on IT-centric multilevel standards as NISPOM 2006, NIST SP 800-53 r4 or ISKE v6.0 – makes possible to transfer from the optimal IT security Cost and corresponding IT security profile to the optimal security measures list,
2. to adopt good ideas from General economic models:
   - from CyberProtect the concept of graded/leveled security measures/ activities groups and to use their Cost and Effectiveness values as basic for bi-dimensional optimization, and
   - the principle/"function" from Olovsson that in IT security 'TotalCosts = SecurityCosts + SecurityLosses'.

IT and information security centric models are considerably simpler and easier to understand than various alternatives:
- in case of matrix models, the dependency matrix between the three (confidentiality, integrity, availability) to seven security goals (and their corresponding three or four security levels) and the 10 to 30-40 security activities (and their corresponding three or four security levels) is used (as example see Table 3),
- in case of a process based graph of 30-40 information security activities there are » 10 nodes and » 50-60 edges (see Figure 5).

Therefore, information security centric models can be demonstrated on one A4 page and can be understood by a single glance (both by management and by IT Security experts). In addition the workload is at least an order of magnitude lower and decreases with experience (at the given enterprise) – first implementation will take man-months, the next one will take man-weeks, or even man-days. Changes/ updates can also be done in man-days.

Risk analysis based models are much more complex:
- » 1000 security incident risks-> » 1000 possible attacks -> » 1000 sets of necessary security measures, or
- out of the » 1000 necessary security measures some are not implemented to the required level -> » 1000 possible attacks -> » 1000 possible corresponding security incidents.

The corresponding workload for the first implementation is in man-years.

It is the workload or labor-intensiveness, which motivated us to focus on IT and information security centric models, since information security experts are typically too busy and forced to avoid time-consuming projects. Time shortage is

often also a problem for the management and they typically prefer short and easy to understand explanations.

Estonia has adopted a mandatory information security model (ISKE) for government institutions, which is an important step forward in information security cost optimization. However, while ISKE contains 64 possible levels or states of information security, the actual security measure determination process is limited to three levels (H/M/L). It is improbable that of the trillions of possible information security solution variants, one of the three is the optimal one for a given enterprise or institution.

The US national information security model (NISPOM), however, is persistently logical. There is a dependency matrix (see Table 3), where 27 classes of security requirements are mapped to the required levels of information security activities. The model is easy to grasp with a glance. Now, if the security measure groups levels in NISPOM are complemented with the corresponding cost and effectiveness values, it is possible to start optimizing. The US Military Academy interactive information security training game CyberProtect is a good example here (see Figure 1).

Therefore, we combined two great ideas in our initial matrix based GSM:

1. DoE Dep Matrix – a matrix model developed (DoE, 1999) and updated (NISPOM 2006) in US. GSM is used to express the relationship between information security goal levels and controls and/or measures grouped by the possible implementation levels of security domains.

		Confidentiality Protection Level			Integrity Level of Concern			Availability Level of Concern		
		PL 1	PL 2	PL 3	Basic	Medium	High	Basic	Medium	High
Section 6. Protection Requirements										
1.	Alternate Power Source (8-601)								Power 1	Power 2
2.	Audit Capability (8-602)	Audit 1	Audit 2	Audit 3, 4	Audit 1	Audit 2	Audit 3			
3.	Backup and Restoration of Data (8-603)				Backup 1	Backup 2	Backup 3	Backup 1	Backup 2	Backup 3
4.	Changes to Data (8-604)					Integrity 1	Integrity 2			
5.	Data Transmission (8-605)	Trans 1	Trans 1	Trans 1						
6.	Access Controls (8-606)	Access 1	Access 2	Access 3						
7.	Identification & Authentication (8-607)	I&A 1	I&A 2,3,4	I&A2,4,5						
8.	Resource Control (8-608)		ResrcCtrl 1	ResrcCtrl 1						
9.	Session Controls (8-609)	SessCtrl 1	SessCtrl 2	SessCtrl 2						
10.	Security Documentation (8-610)	Doc 1	Doc 1	Doc 1						
11.	Separation of Functions (8-611)			Separation						
12.	System Recovery (8-612)	SR 1	SR 1	SR 1						
13.	System Assurance (8-613)	SysAssur 1	SysAssur 1	SysAssur 2		SysAssur 1	SysAssur 2			
14.	Security Testing (8-614)	Test 1	Test 2	Test 3	Test 1	Test 2	Test 3			
15.	Disaster Recovery Planning (8-615)							DRP 1	DRP 2	DRP 3

Table 3. The Dependency Matrix from NISPOM 2006 linking security goals and security activities.

Behind each Dependency Matrix cell is a list of security measures from NISPOM 2006.

2. Cost/Effectiveness values for security activity areas from the U.S. Military Academy information security educational tool/interactive game CyberProtect 2.0.

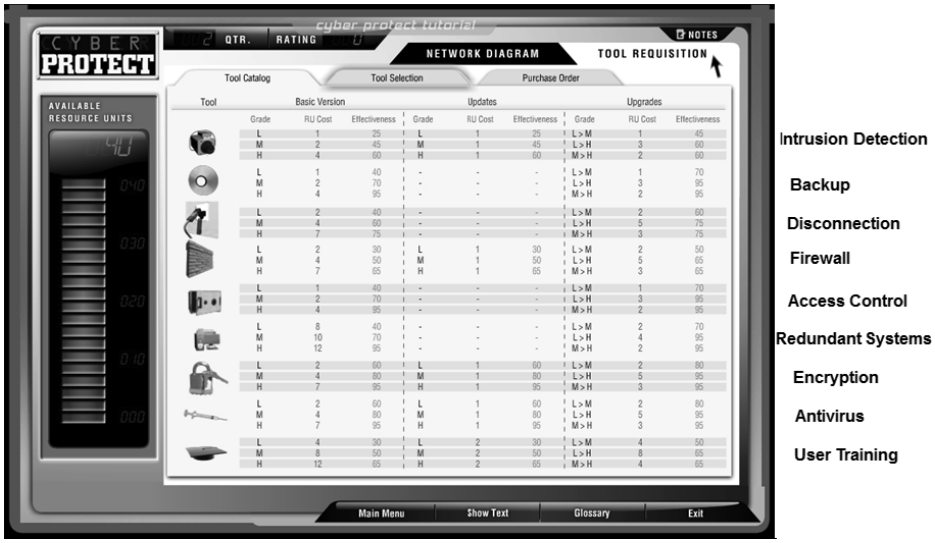


Figure 1. Cost and Effectiveness information from CyberProtect.

Behind each cell in Dependency Matrix will be these Cost/Effectiveness values. One of the original goals of this work was a multi-level GSM: 64 or 256 or 1024 (etc.) levels. With more levels we are likely to find an information security profile that is closer to the optimal for any given enterprise or institution. Thanks to the pervasiveness of IT and availability of computing resources, such increase in complexity is practically irrelevant. We will get the answer with a click of the mouse.

The information security goals are often based on the CIA paradigm (Confidentiality, Integrity and Availability). When each of these categories is assigned 4 possible levels (a'la ISKE in Estonia), then we get a model of 4x4x4=64 levels. However, for cyber security, 7 categories may be more suitable, resulting in a realistic model with 4096 levels. The seven realistic security goals/objectives for Cyber Security model (for CII) could be: Confidentiality, Integrity, Availability, Non-repudiation, Authenticity, Resilience (the ability of a system to provide and maintain an acceptable level of service in the face of various faults and challenges to normal operation), and Mission Criticality.

Our initial development let to an up-to-date matrix based GSM, where the corresponding GSES optimization criteria was the weighted average of information security activity effectiveness (Study I 2008), the optimization solution was based on Pareto-frontier and Dynamic Programming (most detailed explanation available in Study III, 2009). However, by that time (2009) we began to understand the

problems with the matrix model. Specifically, the first case study in Banking in 2009 brought to light several very significant deficiencies.

Ideally the matrix model would require that there would be no dependence between security activities/measure groups, while it would be possible to add their values. However, in IT security:

1. IT and IT security activities cannot be considered in isolation, because practically all IT activities include the information security component (and cost) as well. For example, take HW - the computers are typically considered as an IT cost. However, the purchased servers are often at least ten times more expensive than the cheapest option – indicating that about 90% of IT costs to HW are actually information security costs. Or take SW development – much more programming work is needed to find, fix, and avoid the possible technical and human activities and/or errors, compared to creating the needed business functionality of the IT system (i.e. about 90% of the work (cost) is associated with information security). In other words, we have to describe an IT and IT security Business Model. It became very clear from the Bank case study – we had to base our analysis on the Bank's IT budget, i.e. IT spending (costs).
2. There are very important relationships and dependencies between security activities. For example, the perimeter defense of a system could be more affected by firewall administrator training than buying new hardware or software.
3. Information security activities are not equal in importance (effect). Some are relevant (weakest link logic applies) while others are supporting services to make relevant(s) more secure (Multilevel Security, Defense-in-Depth).
4. The main substantial difference between relevant and supporting activities:
   - If some relevant IT service does not work, the IT as a service does not either.
     For example, if information system hardware as a service/activity does not work (effectiveness is 0), then it is irrelevant how well the other information security activities are implemented (software configuration, power supply, anti-virus, etc.), since the entire system does not function (effectiveness is 0). Therefore, we can look at relevant services as connected in series.
   - If a support service does not function or functions only partially, then it affects only those relevant activities that it is designed to support, while many/most relevant activities may not be affected at all.
     For example, the end user training does not affect the effectiveness of hardware, software, power supply, etc., but it does affect the effectiveness of the end user himself. Therefore, we can look at supporting systems as connected in parallel to the relevant services they support.

The potential solution to this was offered in Study III:
In the future we plan to cover these problems in more detail - use (find or work out)

the information security requirements levels and information security activities areas realization levels dependency graph. (S_III).

While doing a more detailed analysis, it became clear that the matrix model is probably too simplified to describe information security. Describing information security as a system through the average or weighted average value of information security activities (based on the matrix model) does not provide an adequate picture of the actual information security situation. However, it is possible to include the dependency functions between the rows and columns of the matrix, but by doing so we would lose the general easy to grasp nature of the model.

Considering the various deficiencies of the matrix model, we needed to develop a new model that does not share those problems (because we did not find an appropriate existing model/method – in more detail in PART I). We followed the example of models that describe information security as a process.

Process – a set of interrelated or interacting activities which transforms inputs into outputs (ISO 9000: 2005).

People, Process and Technology have been the cornerstone of descriptions of the business processes. The IT Security as a process is very nicely presented in *Defense in Depth* (TISN 2008) and *The Business Model for Information Security* (ISACA 2010).

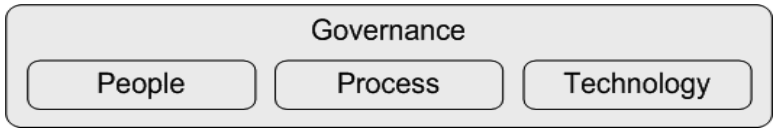


Figure 2. Governance, people, process and technology (TISN 2008)

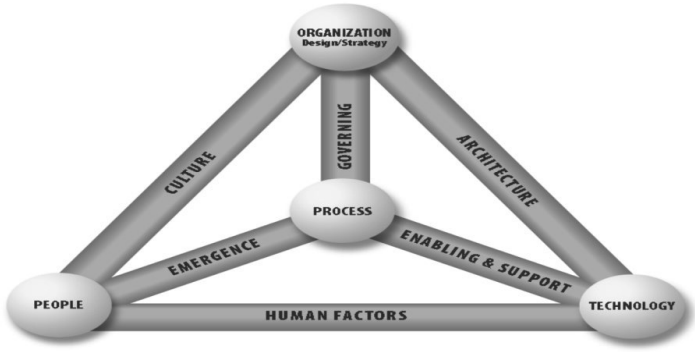


Figure 3. The Business Model for Information Security (ISACA)

We base our work on the simplest and most widely used People-Process-Technology // Governance (Figure 2) business process model. More IT specifically it can be understandable as People-SoftWare-HardWare//Organization.

A potential alternative is the considerably more complex ISACA BMIS model (Figure 3), which includes so-called star and bridge topologies. There has also been an attempt to develop an even more complex model based on ISACA BMIS. However, such a model presumes an order of magnitude more information and, consequently, an order of magnitude more work to get the information. The possibility of errors and the volume of checks are also an order of magnitude greater. For these reasons, this attempt was abandoned – there just was not enough will to spend several man-years' worth of work for the expected result.

Bearing in mind that modern information security contains 30 to 40 activity areas, the following rules are used to decide if components should be placed in the Business Model in series or parallel:

- If failure of a part (security domain) leads to the entire IT system becoming inoperable, the part is considered to be relevant, and all relevant parts are connected in serial (a'la chain links).
- If failure of a part leads to the other part(s) becoming less secure/effective while remaining operable (many or even most parts may not be affected at all), the part is considered to be supporting - i.e. the supporting part is mainly to make relevant part(s) more secure/effective (Multilevel Security or Defense-in-Depth), and all supporting parts are considered to be operating in parallel to the relevant part(s).

Information security as a process can be described in more detail (Figure 4, from our Banking use case) by depicting the main IT and IT security activities:

- Relevant (serial must-be) activities:
  - People: IT Systems Users and their IT Workstations.
  - Process: Software, Environment, Physical Security, AntiMalWare, IT Maintenance.
  - Technology: Power, Data Center; LAN, WAN.
  - IT Organization/IT Governance (look at Figure 2 – parallel to all relevant(s), but by itself at least relevant for big institutions).
- Supporting (parallel to relevant) activities:
  - Awareness and Training.
  - Access Rights Management, Network Access Control.
  - Business Continuity Management, Crisis Management, IT Services Recovery.
  - Logging, Monitoring, Help Desk.
  - Backup, Archive.
  - Security Testing, SW Testing.

- Asset Management.
- High Level Security Documentation (Security Strategy, - Policy, …), Security Audit, Security Compliance, Risk Management (supporting to IT Governance).

The measure group relationship diagram solution proposed in Figure 4 was in GSES extended to a real graph structure (Figure 5), where the relevant measure groups are the edges of the graph connecting the circular nodes and represented as red boxes. The nodes are considered being fully reliable (effectiveness equal to 1). The green boxes, connected to the relevant measure groups represent the supporting security measure groups.

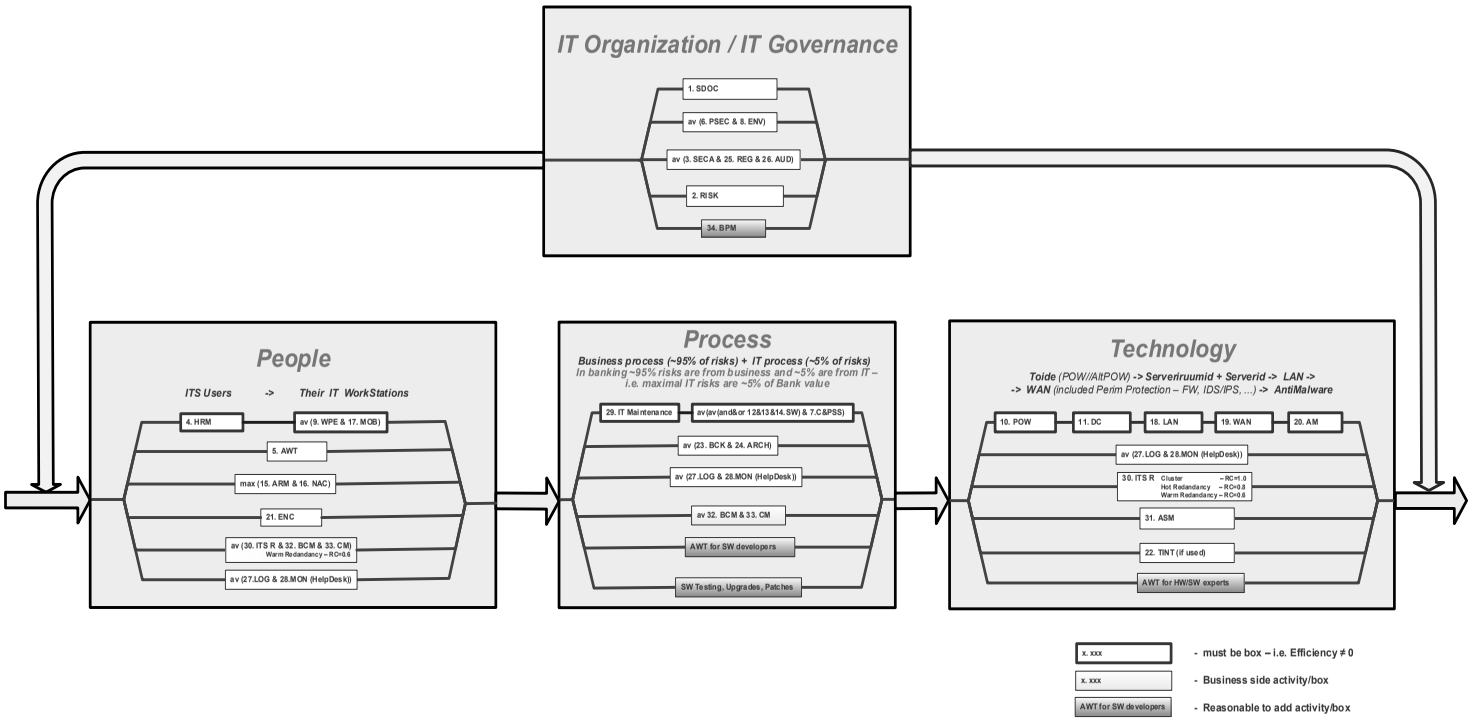


Figure 4. Business Model for IT and IT security (based on Case Study in Banking).

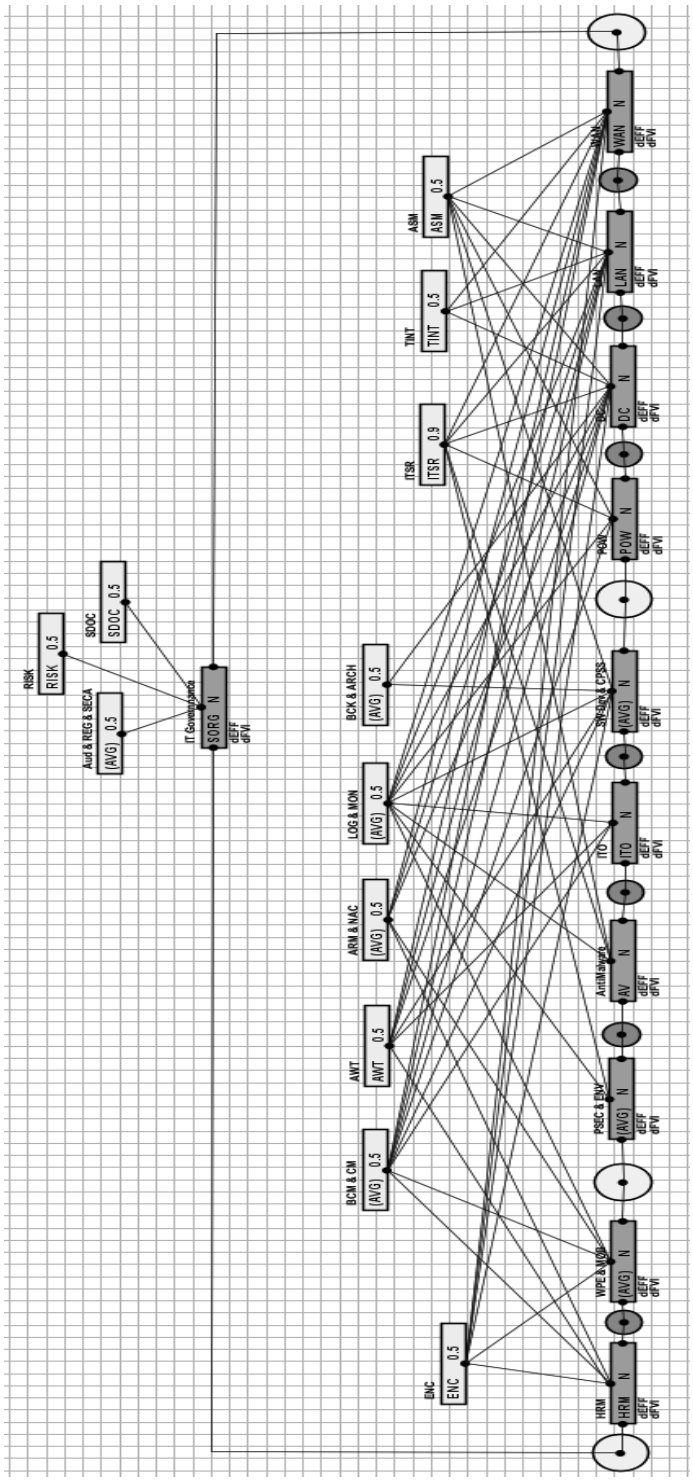


Figure 5. Graph based Business Model for IT and IT security (based on Case Study in Banking)

It should be remembered that the final view of the IT and IT security graph will be institution specific. For example, consider the activity “IT Governance”:

- for large and IT-critical bodies (eg the Bank) it is very important - ie, parallel relevant,
- however, for some medium or non-IT-critical body it is not so important, and therefore, supporting,
- while for small bodies possibly it may not be specifically needed at all.

We named this new version the “graph based Graded Security Model” (gb_GSM). System Availability is calculated by modeling the system as an interconnection of parts in series and parallel. The new model solves all the previously listed (see page 37) problems with the matrix based model (mb_GSM). At the moment it seems that gb_GSM describes enterprise information security well enough to enable cost optimization.

It means that the Effectiveness functions evolved from a weighted average (in Study I, Study II, Study III, Study IV) to the serial – parallel Effectiveness graph (in Study V, Study VI, Study VII). For more details see II.3.

II.2. IT Security Metrics appropriate for IT Security Costs Optimization

You can't manage what you don't or can't measure.

In order to guide some process in the desired direction, we must be able to measure it. In terms of information security cost optimization, we must be able to measure the effects of decreasing or increasing the security budget.

Suitable metrics allow us to measure the progress from the current situation towards a more optimal solution. Therefore, the first sub-goal is to define metrics that are suitable for solving information security cost optimization problems.

The bottom line is that metrics are like goals.

In order to optimize information security costs, we must be able to:

1. collect the necessary information at the level of (security) activities, and
2. calculate the total (encompassing the entire IT system) security effectiveness of the entire information system process.

Why is implementing adequate security measures so important?

- We need to protect our organization from direct (e.g. broken HW or SW, leak of confidential information) and indirect losses (e.g. reputational damage), and thereby ensure that our business goals are achieved.
- We need to be compliant with national and international laws.
- We need to show our organization's commitment to guarantee security for our customers, co-partners and other interested parties.

The selection of the right security measures is a complex problem, because multiple objectives need to be achieved at the same time. IT Security must primarily assure required CIA-levels and acceptable losses from security incidents (residual risks) within the available IT budget.

Wanted is to get the best results for our money - to minimize the costs and to maximize the overall security effectiveness for the available budget, with minimal total IT security cost (total cost includes IT Security investment and maintenance costs, plus losses from security incidents).

We must be able to measure and calculate the total effectiveness of the entire information system, because in information security we are more concerned with the strength of the entire chain, not the strength (effectiveness and economic viability – ROI) of an individual component (security activity). This means dividing the existing and possible information security resources optimally between all security activities, in order to achieve the maximum total security effectiveness.

However, while a specific information security solution may be very effective in economic terms, there is always a possibility that there are other solution(s) that would provide a better result.

For IT Security cost optimization the logical choice is to use Cost-Effectiveness as a metric.

How to define Effectiveness? Effectiveness must meet two requirements in the context of GSM/GSES:

1. We must be able to determine concrete values for the effectiveness of all IT security activities (using expert assessment or real statistical values).
2. There must be a theory for calculating the total effectiveness of the system (because information security is only effective, if all important activities are effective).

In the matrix model, we originally used weighted average effectiveness, since the matrix did not contain the interdependencies between information security activities.

Security effectiveness is defined as follows:
The security effectiveness of a measure group indicates how confident[7] we are, that our security measure group implemented at a certain level, will not be the underlying reason of any security incident. This confidence is expressed as a value between 0 and 1.

However, the four deficiencies of the matrix model (outlined on page 37) also apply to the IT Security weighted average effectiveness metric used in this model. Since we introduced interdependencies between security activities into our model (primarily serial and parallel connections of security activities), then obviously we must base our definition of effectiveness on business process management theory – i.e. System Reliability or Availability, which allows us to calculate the total overall (integral) Reliability or Availability of a process.

Should we use Availability or Reliability as basis for IT Sec Costs optimization?

1. Availability is the probability (or the likelihood or the percentage of times) that a given system (or component) will be operational at any random time, t.

   Reliability is the probability (or the likelihood) that a system will function at the given time, t.
2. Focus on MTTR (mean time to repair).
   If your system is reasonably well-designed and you didn't cut a bunch of corners, your MTTF (mean time to failure) is almost exclusively determined

[7] In our case the notion of confidence can be considered as the exact opposite of the term likelihood used in risk management:
Likelihood (that security activity will be the reason of a successful attack)= 1–Confidence (that security activity performs its protective task)

> by the architecture and the intrinsic characteristics of the components (hardware and software). This means it's difficult to make significant changes in MTTF without either re-architecting, spending lots of money on more redundancy (which can cause its own reliability problems by adding complexity), or changing to an underlying platform with different reliability characteristics. By contrast, there are often large gains to be realized in MTTR without nearly as much investment by relatively simple changes in tools, techniques, and organization. (Guth 2011)

The quote above means that availability captures both reliability (MTTF) and another critical concept – mean time to repair (MTTR). And this is exactly what we need to keep in mind in IT Security optimization.

For us the most relevant is a steady state average availability in a year:

A_{av} =MTTF/(MTTF + MTTR)
MTTF = Mean time to failure, MTTR = Mean time to repair

For IT Security cost optimization, Availability is a more intuitive and understandable metric – Availability is Uptime and Unavailability is Downtime, and Availability(%) + Unavailability(%) = 100%.

Availability – 4.7 - the property of being accessible and usable upon demand by an authorized entity. (ISO 27000 : 2008)

But IT Security system can fail to perform as required due to:

- technical unavailability
- protection functionality is not adequate – upgrade/update is required.

In order to highlight this important difference compared to regular availability, we use the term "effectiveness" – i.e. IT Security Effectiveness essentially means not failing against C and/or I and/or A (and/or whatever functionality or security goal of the IT and IT Security System).

Effectiveness (ISO 27000 : 2008) - 4.13 -
extent to which planned activities are realized and planned results achieved [ISO 9000: 2005]

Effectiveness - the degree to which objectives are achieved and the extent to which targeted problems are solved. [8]
By applying the ISO 27000 definitions of Availability and Effectiveness the **IT Security Effectiveness** for the GSM can be reformulated:

[8] Read more: http://www.businessdictionary.com/definition/effectiveness.html#ixzz2RkYXG4YD

The ability (probability) of the IT Security measures group/domain or overall IT Security system to perform a required security function, under given conditions (vulnerabilities, threats – i.e. missing capabilities in defense, sophistication and intensity of attacks) over a stated period of time (in our case a year). The security effectiveness is expressed as a probability value between 0 and 1.

Number of failures per year (annual failure rate, AFR, or annual rate of occurrence, ARO) is usually the only available statistic, which gives us MTTF=1/AFR (year). In our case the steady state average security Effectiveness in a year:

E_{av} =MTTSI/(MTTSI + MTTR + MTTU)
MTTSI - Mean time to security incident,
MTTR - Mean time to repair,
MTTU - Mean time to upgrade/update.

I.e. Effectiveness is IT security specific availability – the system must function normally and at the same time be up-to-date with recent security malware and exploits.

The failure rate λ typically evolves according to a so-called bathtub curve (see Fig. 6).

The formula for failure rate is:
λ= 1/MTBF = R/T, where R is the number of failures and T is total time.

Initially λis higher, due to undiscovered defects in the component which mostly show up when the component is put in use. After the burn-in period, λ stabilizes. At the end of its lifetime the component gets worn out and λincreases again.

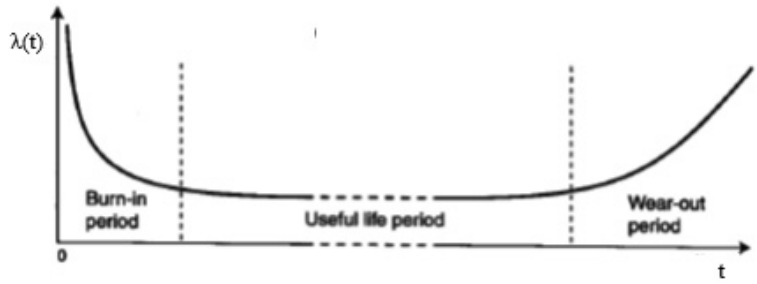


Figure 6. The bathtub curve.

We assume that a well maintained and updated measure group can be normally used during its entire useful lifetime, where is approximately a constant. Extra caution is needed during implementation and retirement phases of security measures. And it is a good idea to avoid the burn-in and wear-out periods in real work.

The greatest problem is that for better optimization we need better – meaning more accurate and complete – information. However, collecting this information is very time consuming and labor intensive. Correct statistical information requires years of collecting, which unfortunately has not been done so far (at least with the necessary level of detail). Hopefully, the cost optimization model presented in this thesis is enough to motivate detailed information security incident analysis and collection of the corresponding statistical information in the future.

For determining the real desired goals, two principles - "Do things right" and "Do right things" - are very widely used, ranging from road construction (Moore 2005) to IT security - in COBIT 5.0 (2.0 GOVERNANCE). The case is quite similar in IT security cost optimization.

With the GSM/GSES we have three stages of IT security costs optimality (depending on expert data we have been able to collect)**:**

- **Do rational things** – do not secure more (it is wasting money) and not less (too many security incidents – i.e. security losses will be too big) than needed.
- **Do things right** – use resources optimally to achieve the best result – i.e. maximal efficiency for the security system with resources we have (time, experts, money).
- **Do right things** – sum of security investments and security losses must be minimal – i.e. in general, it makes sense to increase the level of security until losses from security incidents decrease more than the growth of the corresponding information security spending ($\Delta Loss$ or $\Delta Risk/\Delta Budget \geq 1$).

In order to make decisions we must have appropriate information. Based on morevad detailed and complete information we can make more accurate decisions. In our work we have taken the pragmatic premise that an expert system must allow the optimization at various levels. If a company does not have all the necessary information for a (more) complete optimization, then it is better to optimize partially, than not at all.

The main goal of the thesis is the optimization of information security costs, which basically means that we should be able to describe the *"function"* of *Security Goal = f(Security Costs)* (see the explanation about *"function"* on page 20), where the Security Goal can be Security Class ($C_iI_iA_i$), Security Effectiveness (SE or just E), mitigation Rate (mRate) or IT security Total Cost (TC) depending on the stage of optimization. This *"function"* visualizes the dependency between a *Security Goal (SG)* and the resources allocated for security *(SC) – SG=f(SC)* – i.e. it provides bi-objective optimization to IT and IT security managers, which lets them achieve the needed IT Security goal with minimal total costs.

It is clear that information Security Costs (and correspondingly the information Security Goal) generally depend on many other factors as well. For example, the

additional cost for increased system performance and for the increased system complexity. It is possible to achieve a higher level of security by removing most or perhaps all functionality from a system. The effectiveness of all costs depends substantially on how motivated the IT security experts are to perform well, and so on.

The simplification we have made is justified, if we ensure (meaning – if we assume that we ensure):

- that the necessary security measures that we have identified will be correctly implemented, and
- that the other factors that affect security costs stay constant for the time period in question.

Needed information and optimization criterias for these three stages of optimality:

Do rational things

The so called rational information security is quite wide spread. According to this stage, the goal is the security level (meaning the corresponding levels of confidentiality, integrity and availability) that is derived from the business process or required by law/by contract. A higher level would waste resources, while a lower level would result in too many losses from security incidents. The first stage of IT security activities cost optimization answers the question *"How much should be spent to deliver Desired Information Security?"* We call this stage reasonable or rational (not optimal) because the basis for this, i.e. Dependency Matrix (as example Table 1, Table 2), is quite subjective. However, rational approach is quite widely used, since it ensures reasonable information security spending (not more or less than required).

The Security Class (CIA–level) is a high level expert opinion on information security risks: secure IT systems and their information according to data security requirements - no more (if achieved activities security level(s) are higher than required then security expenses are consequently higher than needed) and no less (too many security incidents and accordingly too high security losses) than needed. To specify this rational level the business side must provide the required/needed security goal levels (e.g CIA) and the IT Security experts must assess the security activities costs that are needed to achieve the required security levels (based on Dependency Matrix).

This rational approach is widely used in the state level. Laws and regulations set the required Security Class for enterprises and government agencies, as well as the rules that define the required security measures.

Internationally, the best known model for such information security cost optimization is the US NISPOM 2006, which applies to highly critical infrastructure

(Department of Energy with its nuclear power stations, Department of Defense, Central Intelligence Agency, etc.). In Estonia the analog is ISKE, which is mandatory for the government sector. Both models are based on dependency matrices of information security goals and security activities.

"Do rational things" stage information security metrics:

- input is the required Security Class of the information in question
- output is the corresponding necessary security measures and their costs – the rational costs for information security.
  At that stage the optimization *"function"* (in this thesis) is:
  $C_iI_iA_i = f(SC)$, where $C_iI_iA_i$ is the required/needed Security Class and SC (Security Costs) represents the corresponding costs for information security.

At the "do rational things" stage the necessary information is:

- Required Security Class,
- Dependency Matrix,
- The costs necessary to achieve the required Security Class on information security activities.

At the "do rational things" stage the security goal is to ensure the required Security Class (meaning the required levels of confidentiality, integrity and availability).

However, quite often the enterprise does not have enough resources to achieve the required Security Class. Then the optimization task changes to "do things right", which seeks to get the maximum possible Security Class with the existing resources.

Do things right

At this level we encounter actual optimization – trying to achieve the maximum information security effectiveness with the existing resources. This presumes a lot more information up front: the bases for cost optimization are the effectiveness and cost of all alternative levels of possible implementing of the information security activities/security measure groups.

In our security effectiveness engineering, all information about a security measure group's effectiveness can then be described based on collected statistical information about annual rate of occurrence (ARO) of failures and Mean Time To Failure (MTTF) or by expert opinions. The next step is to combine several security system components (measure groups) into one overall IT Security system (gb_GSM) and to use quantitative system effectiveness (availability) analysis techniques.

The easiest option for determining the total effectiveness of an information security system is to look at the weighted average of the effectiveness of the security activities (Studies I-IV). However, this thesis describes the more realistic graph model of information security as a business process, as well as the calculation (based on the model) of total integral effectiveness of the information security system (Studies V-VII).

"Do things right" stage information security metrics:

- input is the existing information security resources/budget and
- output is the maximum effectiveness of information security as a system. The optimization is based on the *"function": SE = f(SC*),
  where SE - Security Efficiency and SC – Security Costs, i.e. investment and maintenance costs to IT Security (i.e. SC is basically IT Security Budget).

This *"function"* visualizes the dependency between security effectiveness (SE) and the resources allocated for security (SC) - *SE=f(SC)* – i.e. to provide to IT and IT security managers bi-objective optimization, which lets them achieve the maximum IT Security effectiveness with minimal total costs.

At the "do things right" stage the necessary information is:

- the existing information security resources/budget (or its possible range), and
- the costs of all possible/alternative security levels of all information security activities.

At the "do things right" stage the goal of security is to ensure the maximum possible information security effectiveness with the existing resources.
In practice, the effectiveness based information security level is often expressed as the rate of risk mitigation or annual residual loss:

- the rate of risk mitigation: mRate = 1 / (1-SE) = ALE/AL
- the potential residual annual loss: AL = ALE /mRate,
  where ALE (Annual Loss Expectency) is the potential annual risk without any implemented security measures, and AL is the calculated potential residual risk or the previous year's actual loss due to information security incidents.

The cost optimization information for management would then be:
mRate = f(SC) or *AL = ALE /mRate = f(SC)*

and the maximum possible security effectiveness (for the existing budget) based on the formulas, as well as the corresponding optimal security profile – i.e. the list of security measures to implement.

Do right things

The drawback of the previous optimization stage is that information security is generally very resource intensive and the possible losses from information security incidents (especially at higher security effectiveness levels) are not large enough to justify the security costs.

The main goal of information security cost optimization is to determine the minimum information security total cost (and the corresponding security profile, meaning corresponding security measures), in order to minimize the sum of information security costs and the losses from security incidents.

For this we need additional information about potential information security incident losses and their dependence on implemented information security measures.

Unfortunately, such statistical information is practically non-existent (at least in Estonian government sector). In the private sector, it is possible to collect expert assessments from the business side (very time consuming, measureable in man-months), but public sector entities are typically unable to provide such an assessment.

In general, this is understandable, since the price of products and services, as well as associated risks, are easy to measure in financial terms on the private side. However, on the public side the main goal is to guarantee services where the price and risks are very difficult to express in financial terms. Usually, a qualitative assessment is assigned, such as L – light trouble with providing the service, M – serious trouble with providing the service, H – service could not be provided. However, it is practically impossible to optimize information security costs based on qualitative assessments.

"Do right things" stage information security metrics:

The general goal for IT Security is to achieve the minimum sum of security investments and security losses, i.e. the optimization is based on the *"function": minTC= f(SC)*,
where *TC= AL+SC*, TC – Total Costs/Expenses, AL – Annual Loss, SC – Security Costs (basically the IT Security Budget)

At the "do right things" stage the necessary information is:

- the cost and effectiveness of all possible/alternative security levels of all possible information security activities and
- the potential losses from information security incidents, depending on the implemented information security profile – i.e. *Losses = f(Security Costs)* or *Losses = f(achieved Security Class)*.
-

At the "do right things" stage the security goal is to ensure the minimal information security total cost and to determine the corresponding optimal security profile.

In conclusion:

It is possible to collect the basic statistical information for GSM – i.e. recording security incidents and finding out which measure group(s) have failed. Number of failures per year (i.e. annual failure rate, AFR, or annual rate of occurrence, ARO) are typically all that is collected in information security statistics, and MTTF=1/AFR (year).

Acquiring the information remains a non-trivial task, however. Also, for each measure group we need to define what would be considered an incident. This definition has a significant effect on the measure group's failure rate.

If no security incident statistics are available, then the effectiveness values can still be estimated by security experts.

In our security effectiveness engineering, all information about a measure group's effectiveness's can then be described based on collected statistical information about annual rate of failures (ARF) or by expert opinions. The next step is to combine several security system activities (measure groups) into one overall IT Security system for an organization. Only then can quantitative system availability (alias effectiveness) analysis techniques be used.

In this chapter the IT System's security effectiveness has been defined and made quantifiable. The next chapter describes the calculation of the total integral effectiveness of the system.

II.3. IT Security System effectiveness (alias availability) calculation for gb_ GSM

The next step involves computing the overall availability of the entire IT Security system.

Generally availability refers to how often something works correctly when you try to use it. In IT Security the term "effectiveness" is typically used, meaning the extent to which planned activities are realized and planned results achieved (ISO 27000 : 2008) – i.e. how often IT and IT Security System works correctly.

These calculations have been mainly based on serial and parallel availability calculation formulas from Business Process Theory. However, in this thesis we have introduced information security specifics – i.e. very common situations in IT security where a supporting security activity is parallel to the relevant activity.

Availability in Series
If any one of the system components fails in a system configured in series, the entire system fails. Conceptually, a series system is as weak as its weakest link. A graphical description of a series system is shown in Figure 7.

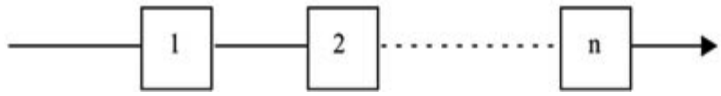


Figure 7. Representation of a Series System of "n" components.

Engineers are trained to work with system availability [A_S] concepts using "blocks" for each system element, each block having its own availability for a given mission time T:

$$A_{Serial} = \prod_{i=1}^{n} A_i = A_1 \times A_2 \times \ldots \times A_n \text{ , if the components availabilities differ, or}$$

$A_{Serial} = [A_i]^n$ (if all i = 1, ... , n components are identical),
where A_i – system component availability and i = 1, ... , n.
A set of n blocks connected in series can be replaced with a single block with the Availability A_S (or Effectiveness – $E_{S)}$.

Availability in Parallel
In a system that is configured in parallel, as long as one component works, the entire system also works. Conceptually, in a parallel configuration the total system availability is higher than the availability of any single system component. A graphical description of a parallel system of "n" components is shown in Figure 8.

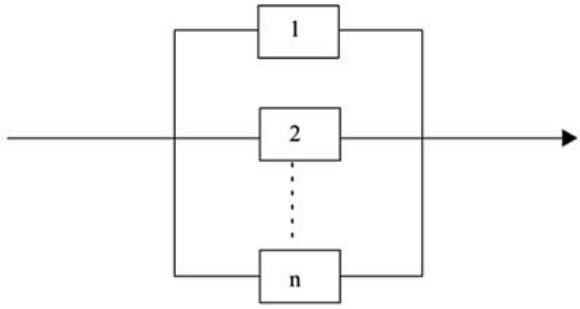


Figure 8. Representation of a Parallel System of "n" components.

Availability engineers are trained to work with parallel systems using block concepts:

$$V_{Parallel} = \prod_{i=1}^{n} V_i = V_1 \times V_2 \times \ldots \times V_n$$

where V_i – system component vulnerability and i = 1, ... , n.
And if to keep in mind that Availability = 1 – Vulnerability :

$$A_{Parallel} = 1 - \prod_{i=1}^{n} (1 - A_i) = 1 - (1 - A_1) \times (1 - A_2) \times \ldots \times (1 - A_n)$$

if the component availabilities differ, or
$A_{Parallel} = 1 - [1 - A]^{n}$, if all "n" components are identical: [$A_i = A$; i = 1, ..., n].
A set of n blocks connected in parallel can be replaced with a single block with the Availability $A_{Parallel}$ (alias Effectiveness – E_{p}).

Therefore, it is clear that even though Parts with very low availability were used, the overall availability of the system will be much higher. Parallel operation provides a very powerful mechanism for making a highly reliable system from low reliability Parts. The principle is as long as not all of the system components fail, the entire system works. For this reason, all mission critical systems are designed with redundant components.

Availability in Not-Full-Redundant Parallel or Rc-Redundant System Availability (alias Rc-Redundant System Effectiveness) – i.e. Availability for Systems where supporting activities are parallel to relevant ones for graph-based GSM (Figure 9).

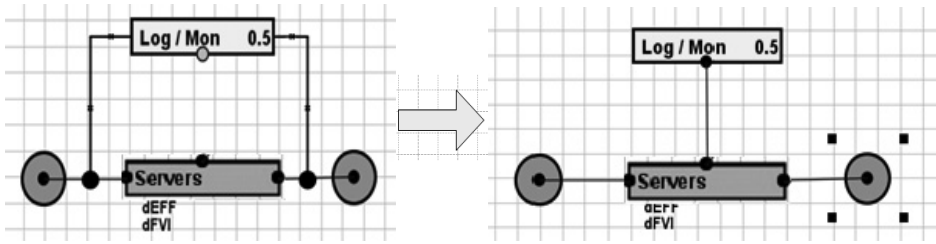


Figure 9. A Parallel System of relevant and supporting components.

This problem was first covered in S-VI.
In gb_GSM we have the so called relevant serial boxes and the logic of "if any one of the system components fails, the entire system fails" is exact and perfect.
NB! If any $E_{Relevant}$=0 then E_{System}=0.

But in IT security the situation is a bit more complicated than in typical parallel systems – in most cases the relevant parallel supporting activities in IT security are not identical. For fully redundant security activities (for example, relevant "HW" and supporting "Redundant HW") the principle is: "as long as not all of the system components fail, the entire system works". However, supporting activities in IT security are typically improving just one aspect of the supported relevant component (for example, relevant HW and supporting Logging/Monitoring). In such cases we do not have full redundancy and we must introduce the Redundancy Coefficient ($R_{C)}$.

The use of Rc ensures that we can separately consider the effectiveness of Relevant and Supporting areas (i.e., the effectiveness of measure groups implementation levels), and the level of support that the Supporting areas provide to the Relevant areas.

Partial parallelism can be accounted for as an expert assessment on the Effectiveness of the Supporting area. However, it would not show the real problem - that the supporting security measure may be poorly implemented, or that the Supporting area (although well implemented) does not support the Relevant area well - i.e. the Supporting solution is not useful.

Practically, $R_C = 0{,}1 \div 1$. In case of full redundancy, $R_C = 1$. A parallel supporting activity with Redundancy less than 0,1 is practically a waste of resources.
If situation for full redundancy is as in usual Availability in Parallel:
$E_{1//2} = 1 - (1 - E_1)(1 - E_2)$,

then for Partially-Redundant parallel situations with redundancy R_C for relevant and supporting activities**:**
$E_{relevant//supporting} = 1 - (1 - E_{relevant})(1 - R_C * E_{supporting})$.

NB! It must be kept in mind that if $E_{relevant}$=0 then $E_{relevant//supporting}$=0 – i.e. supporting parts can only make relevant parts better. In fact non-existing relevant parts can not be improved.

And $R_C = (E_{relevant//supporting} - E_{relevant}) / E_{supporting} (1- E_{relevant})$.
If relevant subsystem is supported by "n" Partially-Redundant parallel subsystems with redundancy R_{Ci}:

$(1 - E_R) \times (1 - R_{C\_1} \times E_{S\_1}) \times (1 - R_{C\_2} \times E_{S\_2}) \times \ldots \times (1 - R_{C\_n} \times E_{S\_n})$,

and again if E_R=0 then $E_{R//S\_1..n}$=0.

And as explanation (see Figure 9):

In order to eliminate hardware problems in computers, the following three (simplified) activities are needed:

1. Find Defect.
2. Fix Defect.
3. Restart System.

Let us assume that Restart happens very quickly and can therefore be ignored.
Computer HW (relevant Servers) effectiveness (availability) is often improved by the information security solution Logging/Monitoring.

Let the service breaks (without supporting security measures) be: Find=2h and Fix=2h, i.e. overall service break is 4h, and the proportion between Find and Fix is 50/50; the Rate of Occurrence=1 per week and E_r=164/168=0,9762.

Thanks to the Logging/Monitoring supporting system we can complete the Find Defect step quite quickly – i.e. the Find Defect step is improved and there is no influence on the Fix Defect step. For example, if Effectiveness of the Logging/Monitoring System is E_s=0,9 then Find Defect=0,2h and Fix Defect stays 2h, i.e. Find + Fix=2,2 and
$E_{relevant//supporting} = E_{Servers//Log\text{-}Mon} = (168\text{-}2,2)/168 = 0,9869$,

and the we get the same result, if we take R_C=0,5 for Log-Mon System in supporting the relevant Servers:
$E_{Servers//Log\text{-}Mon} = 1 – (1\text{-}0,9762)(1\text{-}0,5*0,9) = 0,9869$.

NB! If we do not have functioning HW (for example, servers are not working for a whole year) then good Logging/Monitoring supporting system does not help us – $E_{Servers//Log\text{-}Mon}$=0. Fortunately, this is mostly a theoretical possibility.

Calculating System Effectiveness (Availability)
By recursively replacing the series and parallel subsystems by single equivalent elements we can obtain the Availability (Effectiveness) A_{System} (E_{System}) for the entire graph-system and the new graph model enables us to calculate the wanted Availability (Effectiveness) for a specific IT security System, as well as the Losses mitigation Rate (mR), which is very interesting for managers. The value of mR can be expressed as
mR = Maximal Risks / Real Annual Losses = $1 / (1- E_{System})$ or
Real Annual Losses = Maximal Risks / mR.
Things to note:

- Availability of software is usually higher, even though hardware MTBF is higher. The main reason is that software has a much lower MTTR. In other words, the software does fail often but it recovers quickly, thereby having less impact on system availability.
- The input and output nodes for relevant activities in the graph have fairly high availability, and thus fairly high availability can be achieved even without redundant components. These nodes are considered fully reliable (effectiveness is 1).

A potential problem is that relations between measure groups may not be only serial or parallel. More complicated models may include bridge, star or other topologies. While this is currently considered as a mostly theoretical problem, we will immediately encounter it with ISACA BMIS, which would introduce both bridge and star topologies.

In Study VII we found a solution to this problem: the coherent graph can always be replaced by an equivalent series structure of its Minimal Cut Sets (MCSs). The MCSs search algorithm is based on the findings of Librizzi, Sansavini and Zio (2006).

A methodology based on a combination of Cellular Automata (CA) [9] and Monte Carlo (MC) sampling [10] or Union-find cut set search algorithm [11] is used to identify the MCSs of our coherent graph.

Therefore, the GSES is now able to calculate the Effectiveness function for every possible graph (all types of topologies are acceptable), by only using two formulas: the serial components' availability formula and the parallel components' availability formulas described above.

This creates an opportunity to describe even more complex models in the future.

[9] http://en.wikipedia.org/wiki/Cellular_automata

[10] http://en.wikipedia.org/wiki/Monte_Carlo_method

[11] http://en.wikipedia.org/wiki/Disjoint-set_data_structure

II.4. A relevant optimization algorithm for gb_GSM

The main need for optimization in an institution is to achieve:

- maximal IT Security Effectiveness with minimal Costs or somewhat more concretely – maximal IT Security with the money (IT Security budget) that we have,
- minimal IT Security Total Costs or again somewhat more concretely – the money (budget) needed for IT Security that would lead to minimal Total Costs.

  Since we have a bi-dimensional optimization then the Pareto set based Pareto-frontier is a very useful solution for explaining the situation to the management – it is a visual solution that can be understood with a glance. The question is about optimization algorithms to calculate the Pareto set.

  We have used a very pragmatic approach to find the right optimization algorithm for our GSM – without a broader theoretical analysis we just used the first fits where experts were immediately available.

  To calculate Pareto set/curve for GSM we have tested three optimization techniques (see Figure 10 and Table 4):
- Brute Force (all possible variations of information security implementations are calculated and then the minimal or maximal one is chosen) (described in S_I)
- Dynamic Programming (developed in Institute of Cybernetics at the Tallinn University of Technology) (described in S_II, S_III)
- Evolutionary Algorithms (more detail in S_V, S_VI).

  Evolutionary algorithms are popular approaches to solving multiobjective optimization. Nowadays, most evolutionary optimizers apply Pareto-based ranking schemes. Genetic algorithms have become standard approaches. [12]

[12] http://en.wikipedia.org/wiki/Evolutionary_algorithm

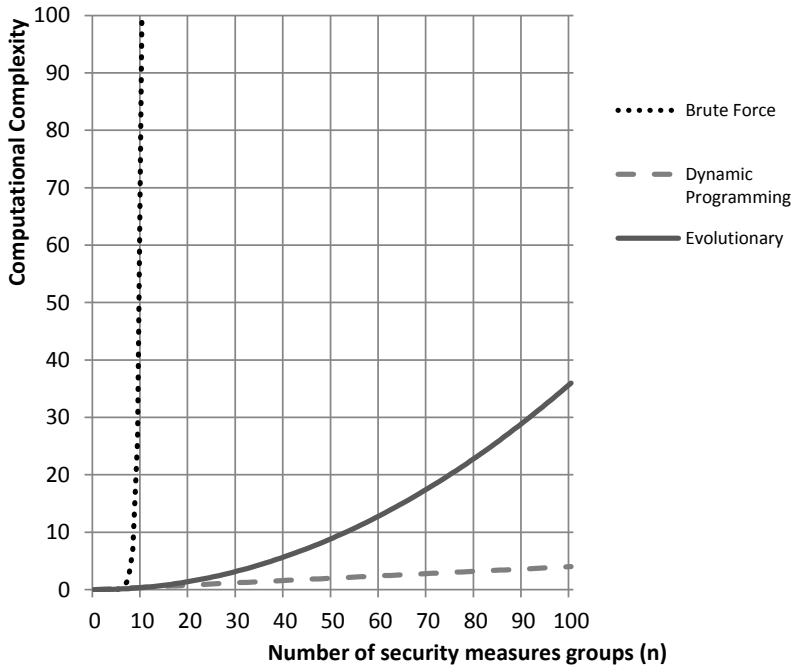


Figure 10. Computational complexities for optimization algorithms used.

	Brute Force	Discrete Dynamic	Evolutionary
Calculations complexity	qk^n (calculations time in years)	q^2kn (calculations time ~ 1-2 minutes)	$36qn^2$ (calculations time ~ 1-2 minutes)
"C" – function	matrix & graph	matrix only	matrix & graph
Hypothesis	None	*Independence of security activities areas is required, all alternative solutions are not found*	None
Usefulness	IMPOSSIBLE	LIMITED	ALWAYS

„q" is the number of resource levels between min and max,
„k" is the number of security levels
„n" is the number of security measures groups

Table 4. Comparison of optimization algorithms used.

Our contribution to the evolutionary optimization is to optimize the parameters of an evolutionary algorithm for our specific task – for an IT security cost optimization.

As the evolutionary optimization process is based on randomness it makes the speed of the problem solving task rather variable. There are no hard and fast rules for choosing appropriate values for the parameters (Cicirello & Smith, 2000). The first scientist, who put a considerable effort into finding parameter values, was De Jong (1975). He tested different values experimentally and concluded that the following parameters give reasonable performance for his test functions: population size 50, crossover 0.6 and mutation rate 0.001 (see also for details Eiben, Hinterding, & Michalewicz, 1999). But those values are suitable for the problem that he had at hand. It has been shown that it is not possible to find parameter values which are optimal for all problem domains (Wolpert, & Macready, 1997) therefore each problem need its own approach and different set of parameters. (Kivimaa, Kirt 2011)

Topic is discussed in more detail in the publication S_VI. Based on the measurements we were able to generate formulas to specify the parameters of evolutionary optimizer for our IT security costs optimization approach. Evolutionary algorithms variation operators (e.g., crossover, mutations, swap, inversion, insertion, displacement) are applied to the individuals that modify the population of solutions dynamically. Every variations operator has its own probability of occurrence and therefore in the further optimizations the following set and values for variation operators could be used:

- Specific values for IT security costs optimization:

repeat	3
population size	n * 3
tournament size	50
generations	n * 4

where n is the number of security activity areas.

- Probabilities of occurrence for variations operators:

crossover rate	0.9
mutation rate	0.8
swap rate	0.6
inversion rate	0.1
insertion rate	0.07
displacement rate	0.11

Number of variants required to calculate and compare by this algorithm is:
q * Population size * Number of Generations * Number of Repeats.
As based on results of meta-level optimization 'Population size' = n*3, 'Number of Generations' = n*4 and 'Number of Repeats' = 3, where
q is the number of possible values of security budget / - costs,
n is the number of security measure groups/security activities areas) and
optimal number of variations to calculate/compare is $\mathbf{36*q*n^2}$.

As example:

- For 10 security activities areas is required testing of $36*100*10^2=0{,}36*10^6$ variations,
- For 30 security activities areas is required testing of $36*100*40^2=3{,}24*10^6$ variations.

For more detailed IT security handling optimization time increase is quadratic (n^2) and consequently is quite important to use optimal parameters in optimization. Thanks to meta-level optimization of Evolutionary Algorithms for our specific task, IT security costs optimization, we succeeded in our banking case study calculations to reduce the calculation time spent on the ASUS notebook N76V more than 10 times - from ~ hour to a few minutes.

In conclusion:

- in selecting the optimization algorithm the calculations time is the critical factor;
- Brute Force optimization method is inappropriate for more complex (for more than 12 measure groups) cases – i.e. unsuitable for contemporary systems;
- Dynamic Programming based optimization method does not have any problems related to calculation time and is excellent for mb_GSM, but it is inappropriate for gb_GSM – independence of security activities areas is required and all alternative solutions are not found – i.e. it is of limited use;
- in the Evolutionary method it is important to use the optimal optimization parameters (S_VI). In such a case it is proper for current models.

II.5. About losses in IT security

The rationale behind information security is to reduce losses from IT security incidents. In order to optimize the security costs for longer periods (couple of years) we must take into account the security losses – the loss that is not taken into account is equivalent to a case where security losses = 0, which would imply that spending resources on information security is pointless.

To specify the real optimal IT Security we must look for the minimum sum of security costs and security losses (Olovsson 1992) – i.e. minimal Expected total cost(SE) = Cost for security enhancing mechanisms(SE) + Expected total cost for violations(SE) (Figure 11).

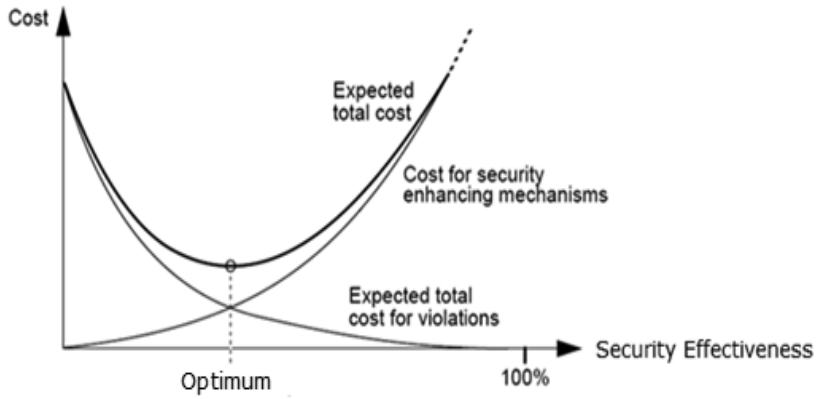


Figure 11. The security cost "function" (Olovsson 1992).

Therefore, in addition to prior first level optimization information (the Cost and Effectiveness values of all possible levels of all security activities, as assessed by IT security experts) we must have (from business side experts) the values of all possible security losses – i.e. we need security Losses and security Costs as functions of security Effectiveness.

In previous chapters we have thoroughly described the Cost and Effectiveness functions. In order to implement Olovsson's optimization criteria, we therefore need the Loss curve SL(Effectiveness).

Unfortunately, this kind of statistical information is not available and we must base our work on risk assessments.

The purpose of a risk assessment can be more broadly defined as identifying and evaluating the following:

- the probability of attacks;
- the vulnerabilities in operations, assets, or individuals;

- the threats to operations, assets, or individuals;
- the impact or consequence (losses).

The risk related to a security incident is defined as a function of two components – probability and impact of the security incident. The relationship of these risk components can be described with the following formula:

Risk = Probability * Loss, (very often **Loss** is replaced with the term **Impact)**
And in IT or Cyber Security probability for security incident is:
$\mathbf{P = P_A * P_{V\&T} * P_I,}$ where
$\mathbf{P_A}$**- Interest and/or probability of Attack,** i.e. IS or information in it is essential for the attacker;
$\mathbf{P_{V\&T}}$**- Possibility and/or probability of Vulnerabilities and Threats -** i.e. if IS has weakness(s) and at the same time attackers have knowledge how to exploit it/ them, then there will be a threat of attack;
$\mathbf{P_I}$ **- Probability of Interruption,** i.e. needed protective security measures are not implemented and the attack will be successful.

$\mathbf{P_N}$ **- Probability of attack's Neutralization,** and $\mathbf{P_I + P_N = 1,}$ thus
$\mathbf{P = P_A * P_{V\&T} * (1 - P_N).}$
Some remarks:

- the concept that is generally called "Probability of attack's Neutralization" in risk analysis, is referred to as "security Effectiveness" in IT security (and in this work),
- about $P_{V\&T}$ - practically all ISs have exploitable weaknesses. In order to find exploitable and unprotected weaknesses, the attacker only needs knowledge. New V&T's – i.e. zero-day exploits and APTs, are discovered all the time. In most cases it is impossible to find numeric values of the probabilities and losses. Still, this formula can be used for evaluation of relative risks.

In private sector the business people have no trouble estimating possible losses in terms of money. However, in the public sector (including the military) this is so far (at least in Estonia) uncharted territory. In essence, this is a solvable problem. For example, there are theoretical solutions for this in the Public-Private-Partnership management theory. Nevertheless, we will not tackle this problem in this thesis.

We base our work on the risk assessments from the banking sector. We assume that bankers can count money and that their assessments are sufficient for verifying the model (more detail in the chapter "Fault tolerance of the GSM/GSES-method").

Hackers are quite interested in hacking banks (for direct monetary gain). Therefore, we can assume that P_A is roughly 1 – meaning that the Bank will definitely be attacked. We also assume that the Bank will be attacked by very skillful hackers – meaning that $P_{V\&T}$ is also roughly equal to 1. Hence, the security of the bank is practically only dependent on security Effectiveness – E or P_N.

For example, a possible way to specify security losses is the "SLA-questionnaire about Info System Security risks" (Table 5) in SEB bank. A completed questionnaire for one IS, the Core Banking System, is shown in Table 7. Such questionnaires must be filled out for every (or at least all relevant) business IS. By summing up their individual values we get the potential total loss for the bank (see Table 6).

Five Potential Losses levels for IT security incidents :

Incident's Potential loss	Estimated potential loss of IT security incident
1. Extremely L*ow* (ExtL)	Less than 10 000 € in a year
2. Low (L)	From 10 000 to 100 000 € in a year
3. Medium (M)	From 0,1 million to 1 million € in a year
4. High (H)	From 1 million to 10 million € in a year
5. Extremely H*igh* (ExtH)	From 10 million to 100 million € in a year

Five estimated Annual Rate of Occurrence (ARO) levels for potential IT security incidents:

Incident's Annual Probability	Estimated likelihood of potential major IT security incident	ARO (times a year)
1. Extremely L*ow* (ExtL)	The presence is very unlikely – i.e. still not appeared or potential of incident occurring is ~ once in ten years	0,1
2. Low (L)	The presence is unlikely – i.e. potential of incident occurring is ~ once in four years	0,25
3. Medium (M)	The presence is medium – i.e. a potential incident occurring ~ once in a year	1
4. High (H)	The presence is likely – i.e. a potential incident occurring ~ once in a quarter	4
5. Extremely H*igh* (ExtH)	The presence is a very likely – i.e. a potential incident occurring ~ once in a month	10

Annual Probable Loss= S(Incident's Potential Loss x Probable Annualized Rate of Occurrence)

Table 5. SLA questionnaire about Information System Security risks

	A security level decrease X3-> X0, meaning the sum of risks X3 -> X0, X2 -> X0 and X1 -> X0	**A security level decrease X2 -> X0,** meaning the sum of risks X2 -> X0 and X1 -> X0	**A security level decrease X1 -> X0,** meaning the risks X1 -> X0	**Potential risks at the required security level X3**
C	1 057,80	117,95	1,70	0,20
I	2 825,75	1 384,35	441,20	4,50
A	2 024,20	734,45	116,30	0,50

5 907,75 million
EEK ≈ 400 million
€

Table 6. The sum of business process risks due to a decreased CIA level by a security incident.

IT Security SLA

IT System : ***Core Banking System*** ***Owner :*** _____

Missioncritical (yes / no) (according to ... Bank Business Continuity Plan)***:*** ***yes***

IT Security goals	Definition	x / - yes/no	IT Security Classificator	Define information value *			
				Security Level's downfall	Single Loss Expectancy (in millions € per year)	Annualized Rate of Occurrence	Annual Loss Expectancy (in millions € per year)
Confidentiality	**Public information:** no access restrictions to information (i.e., read-right to all interested parties, the right to amend as defined by the integrity requirements); information disclosure will not cause material or moral damages.		C0				
	Information for internal use only: access to information is permitted in the case of the legitimate interest of the person requesting – only Bank employees or, information disclosure may cause material (in hundreds thousands of €) or moral damages		C1	C1⇒C0			
	Secret information: access to information is permitted in the case of the legitimate interest of the person requesting – only to certain concrete users groups, information disclosure disturbs the functioning of the institution or violates personal privacy, damages reach into millions €.	X	C2	C2⇒C1	up to 0,01	1	up to 0,01
				C2⇒C0	1 – 10	0,25	0,25 – 2,5
	Top Secret information: access to information is permitted in the case of the legitimate interest of the person requesting – only to certain concrete users, information disclosure is a danger to personal or Bank's safety/security, may cause uncontrollable changes in important systems of the Bank and/or the losses take shareholders equity below Estonian Bank's required level of capital (damages reach into tens millions €), information leakage can lead to uncontrolled direct access to the Bank or the Bank's client money.		C3	C3⇒C2			
				C3⇒C1			
				C3⇒C0			
Integrity	Information accuracy is not important. Verification of information changes or source is not important.		I0				
	Special periodic (monthly, quarterly, yearly) accyracy checks are needed. Information changes must be recognizable (even in case the systems manager made them in his regular work).		I1	I1⇒I0			
	Daily information accyracy checks are needed. Information changes must be recognizable and information changer and source must be identifiable.	X	I2	I2⇒I1	1 – 10	1	1 – 10
				I2⇒I0	10 – 100	0,25	2,5 – 25
	Real time (before information usage) information accuracy checks. Required non-repudiation - implies that one party of a transaction cannot deny having received a transaction nor can the other party deny having sent a transaction. Must have value in court.		I3	I3⇒I2			
				I3⇒I1			
				I3⇒I0			
Availability	Information delay does not cause any complications.		A0				
	Required IS's 90% availability (ensured practically for all modern computers) - total permissible outage in a month ~ 72 hours or ~ 3 days.		A1	A1⇒A0			
	Required IS's 99% availability (ensured practically for all server versions of computers for well-known computer companies) - total permissible outage in a month ~ 8 hours. Generally information must be available only on a working hours – i.e. 5x8, max 6x12 hours.		A2	A2⇒A1			
				A2⇒A0			
	Required IS's 99,9% availability - total permissible outage in a month ~45 minutes. Information mostly must be available 7x24.	X	A3 7x24	A3⇒A2	0,01 – 0,1	1	0,01 – 0,1
				A3⇒A1	1 – 10	0,25	0,25 – 2,5
				A3⇒A0	1 – 10	0,25	0,25 – 2,5

Table 7. SLA form (as example for a Core Banking Information System).

Next we will determine the best order for reducing risks (meaning increasing security effectiveness). By starting with C0I0A0 the first step should be to eliminate the greatest decrease in security level by increasing the system's security level from I0 to I1 (to security class C0I1A0). The next step should be to eliminate the next greatest decrease by raising the security level from A0 to A1 (to security class C0I1A1), etc. See Table 8 for more details.

	Reduction in risk	**Recommended order of increasing the security class**	**Security class**
C0 -> C1	939,85	**3**	C1I1A1
C1 -> C2	116,25	**8**	C2I3A3
C2 -> C3	1,70	**9**	C3I3A3
I0 -> I1	1 441,40	**1**	C0I1A0
I1 -> I2	943,15	**4**	C1I2A2
I2 -> I3	441,20	**6**	C1I3A2
A0 -> A1	1 289,75	**2**	C0I1A1
A1 -> A2	618,15	**5**	C1I0A2
A2 -> A3	116,30	**7**	C1I3A3

Table 8. The desired order of security classes based on the risk reduction amounts.

Assuming that the Bank's risk probabilities practically depend on only security Effectiveness, we can use GSES to calculate the security Effectiveness for each security class (meaning all possible CIA security classes) (See Table 9).

CIA	**Budget**	**Effectiveness**	**Max Total Risk –levels of risk reduction 1÷9**
000	0	0	5 907,75
010	3250	0,156	4 466,35
011	4500	0,356	3 376,60
111	5250	0,3825	2 636,75
121	8250	0,77	1 293,60
122	9500	0,856	775,45
132	16000	0,972	234,25
133	18750	0,986	117,95
233	19000	0,988	1,7
333	19500	0,99	0,2

Table 9. Total Risk = maxTotal Risk – the graded decrease of risk at the desired security level.

Therefore we have determined *SL = F(E)*. By using *SC = F(E)* that we found in previous chapters, we can find the Total Optimum = min TC (see Figure 12).

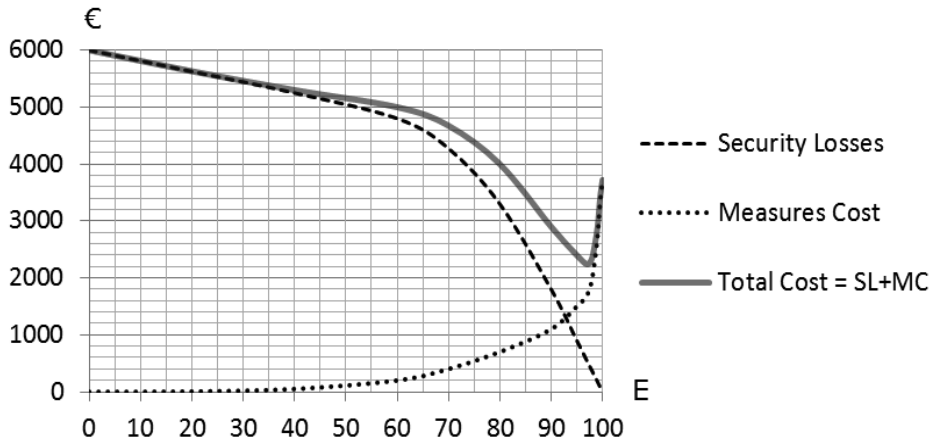


Figure 12. *The "functions" SL = f(E), SC = f(E) and TC(E) = SC(E) + SL(E).*

It should be noted that the resulting Loss curve is significantly different from the 'Expected total cost for violations' curve offered by Olovsson. Therefore, the assessments from the private sector confirm that the most significant decreases in losses occur only at the higher levels of security. This, in turn, is a good proof for the idea that information security effectiveness will be good only when the effectiveness of all relevant activities is good – meaning that the logic about the strength of the chain applies to information security and that it is the total security effectiveness that matters. It also confirms that the ROI analysis of single information security activities is not very useful.

Our Loss curve makes the overall picture of costs + losses much more interesting from the perspective of optimization (compare Figures 11 and 12).

However, for the public sector (including the military) there is no real information about probable losses from security incidents (in terms of money). This field has not been sufficiently researched (at least in Estonia), which means that:

- in the public (including military) sector we must be content with the second stage of optimization – the "Do things right" stage (i.e. maximal security effectiveness with the money we have) or
- we define the area, where $\Delta Risk/\Delta Budget \geq 1$,or in another way: $maxALE/mRate_i - maxALE/mRate_{i+1} \geq Budget_i - Budget_{i+1}$, for optimization budget points i and i+1.

The Security Budget, where $\Delta Risk = \Delta Budget$, would be the last optimal Security Cost (Budget).

The Bank Case Study shows that the difference between the losses from maxALE/mRate-calculated results and the expert assessments was approximately 20% (Appendix 8).

Therefore, if we only know the maximum possible ALE, we can determine the optimum, but the error is relatively high. The topic requires further research.

In conclusion:
It is not likely to find existing, suitable and systematic information about an institution's (even when it is a Bank) losses due to information security incidents. We have described one possible way to obtain such information. We have used it to get usable information about losses due to an information security incident from a Bank. It was not perfect – the main problem was that the error rate (accuracy) of the first questionnaire was ±82% (see Table 5). For example, if the category of Loss is in the millions (1 to 10 million at the High level in Table 5), then we use the average in our calculations – 1+(10-1)/2=5,5 million.

- Therefore, the possible error at the lower limit (1 million) is: (1–5,5)/5,5= -82% and
- the possible error at the upper limit (10 million) is: (10–5,5)/5,5= +82%.
  In following questionnaires it is realistic to improve the accuracy and get sufficiently accurate information about information security losses for the model. As we found out, the fault tolerance of the model is ±20% (see Table 10), which is sufficient and achievable. Therefore, it is possible to get a sufficient expert assessment of the potential information security losses for a private enterprise.
  Two ways of defining monetary damage are presented and compared:
- questioning business experts, and
- calculating the achieved level of risk mitigation (mRate).
- There are currently quite significant weaknesses in both approaches, which also shows in the relatively large difference of the results (~ 20%):
- in the Bank use case the accuracy (± 82%) of the questionnaire was still inadequate, and,
- at the same time, the previously proposed *"function"* AL = maxALE/mRate was perhaps too simplified, and should likely include some additional dependencies and variables.

The problem of determining losses in monetary terms remains, and will require further research.

II.6. A proper SW-platform for GSES

We based the development of GSES for our GSM on a visual simulation and decision-making environment called CoCoViLa, which is a compiler compiler for visual languages, and a very useful tool for developing expert systems, developed in the Institute of Cybernetics at the Tallinn University of Technology.

The system includes knowledge modules (rule sets) in the form of decision tables for handling expert knowledge of security costs and gains (security effectiveness). Other components are an optimization program for calculating Pareto optimality curve parameterized by available resources, and a visual user interface for graphical specification of the secured system, visual control of the solution process through a GUI, and visualization of the results. These components are connected through a visual composer that builds a Java program for each optimization problem, as well as compiles and runs it on the request of the user.

CoCoViLa has an intelligent Graphical User Interface - i.e. visual specification and programming for input problem tasks and a quickly understandable visual output for decision makers. The visual GSES development interface and visualization for analysis and presenting optimization results make good decision support possible for IT Security cost management.

The CoCoViLa platform is a very good choice to develop a decision support system:

- CoCoViLa works on Windows, Linux, Mac platforms – the applications does not require any changes to work in any of the platforms.
- CoCoViLa is developed in public and free to use (GNU *General Public License*, GNU GPL).
- The widely used Java programming language is used for making the applications.
- The expert uses visual programming to describe his specific model or system. Visual programming does not require special programming skills and is basically a user friendly graphical user interface (GUI) – experts can describe their problem set without programming, by using just as ‘paint the graph’ or Lego block approach.
- The model can be adjusted with little time and effort to describe and optimize a specific institution’s information security system and costs. This is important because optimization is always institution-specific. The user friendly GUI and visual programming in the implementation of the model ensure that it is easy to make necessary changes quickly.
- The visual output is easy to understand and is perfectly suited for explaining the usefulness and optimality of the security solutions to the management.

It should be stressed that optimization process is computation-heavy. In order to be able to verify one’s model and data, a suitable software solution is needed.

We must be able to perform the optimization calculations, if we want to be able to verify our model and data based on calculated results, as well as to compare them to the actual results. At this phase of GSM/GSES development, a software prototype is sufficient, although it may not be up to commercial standards (the required work for a prototype and a commercial solution differs by an order of magnitude) in terms of user friendliness, ability to detect human errors, etc.

II.7. Fault tolerance of the GSM/GSES-method

As with any expert system, our tool is only as good as the experts are who have provided the assessments on the cost and effectiveness. At this point we must rely on expert assessments to a great extent and these are known to be rather inaccurate - a result of ±20% would be relatively good. Therefore, the fault tolerance of the model is very important. The hope is that a real ±20% uncertainty of the raw data does not lead to significant changes in the result.

Error calculations are very important, but they are very work intensive, even more so than the main topic of optimization. However, there is an alternative option for determining the fault tolerance of the GSM/GSES method. We compare the real results with the results from our model – calculated with the theoretically correct data (0% error) and with ± errors. For example, we calculate the optimum with 0% error (and assume it is correct), then by assuming a +20% error on the initial data, then with -20% error, etc. For losses, even an error of up to ±82% (see explanation on page 70). Therefore, we should consider errors of ±20'st% to ±82% (» ±80%).

Fault tolerance is handled in S_III for mb_GSM, but we will also look at the situation with the new version of the model, the graph-model (gb_GSM/GSES).

We ask expert assessments about security measure-group Costs, Effectiveness and about possible Losses from possible security incidents. Next, we assess the effect of the associated errors. The ±20% error (very good for expert assessments) can reasonably and immediately be achieved for Costs, but is problematic for Effectiveness and Losses. The real possible error in the Loss assessments by business experts is up to ±82%.

We can see from Figures 13 - 16 that with +20% and –20% error (grey curves) the optimal minimum is within the range of only a few percent.

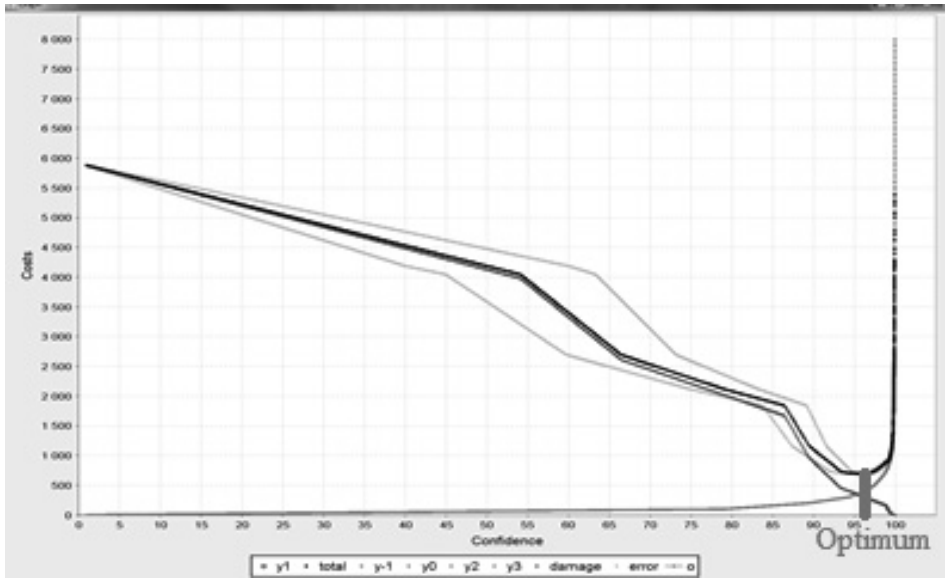


Figure 13. The Total Cost *"function"* if Confidence (Effectiveness) +20% and –20% (grey curves).

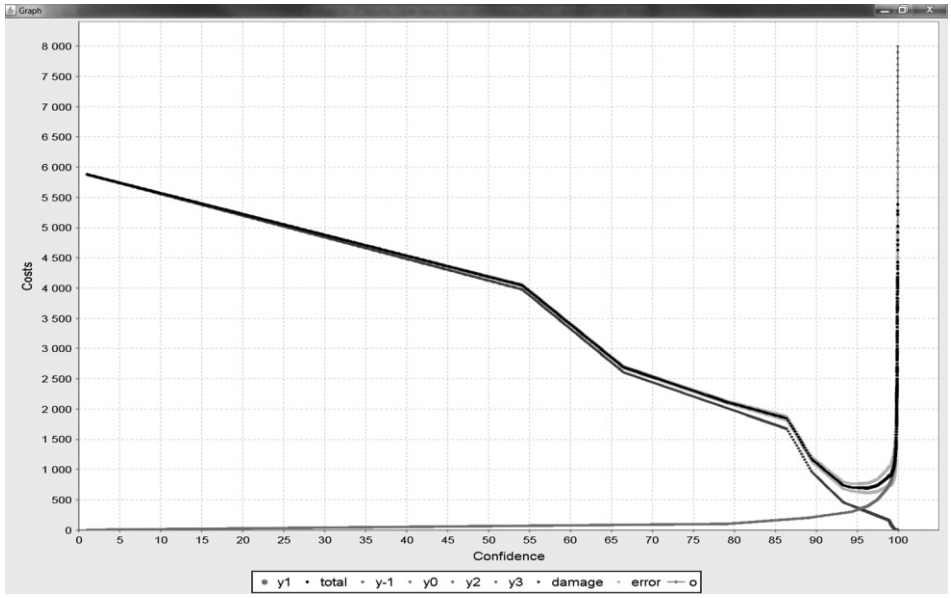


Figure 14. The Total Cost *"function"* if Cost +20% and –20% (grey curves).

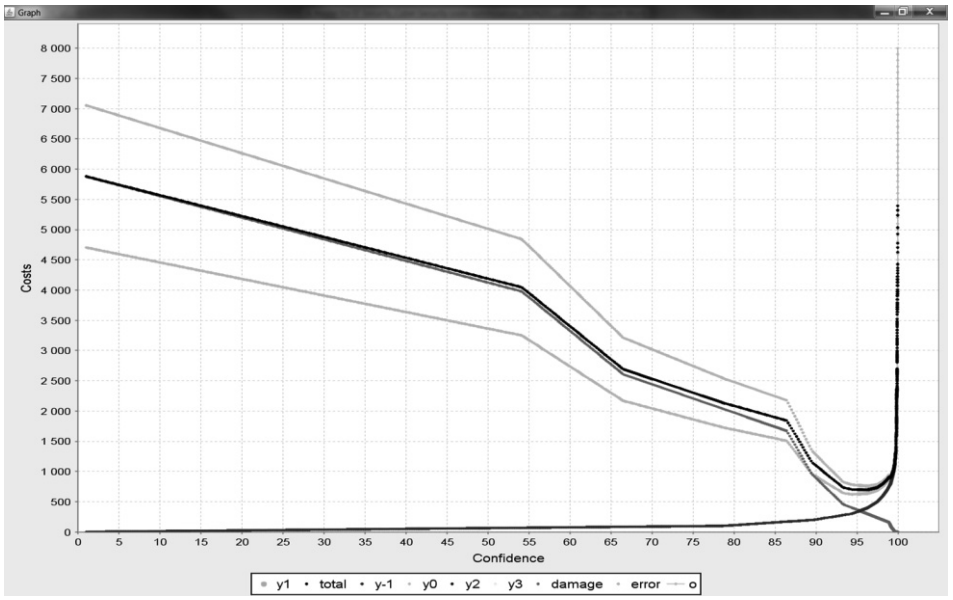


Figure 15. The Total Cost *"function"* if Losses +20% and –20% (grey curves).

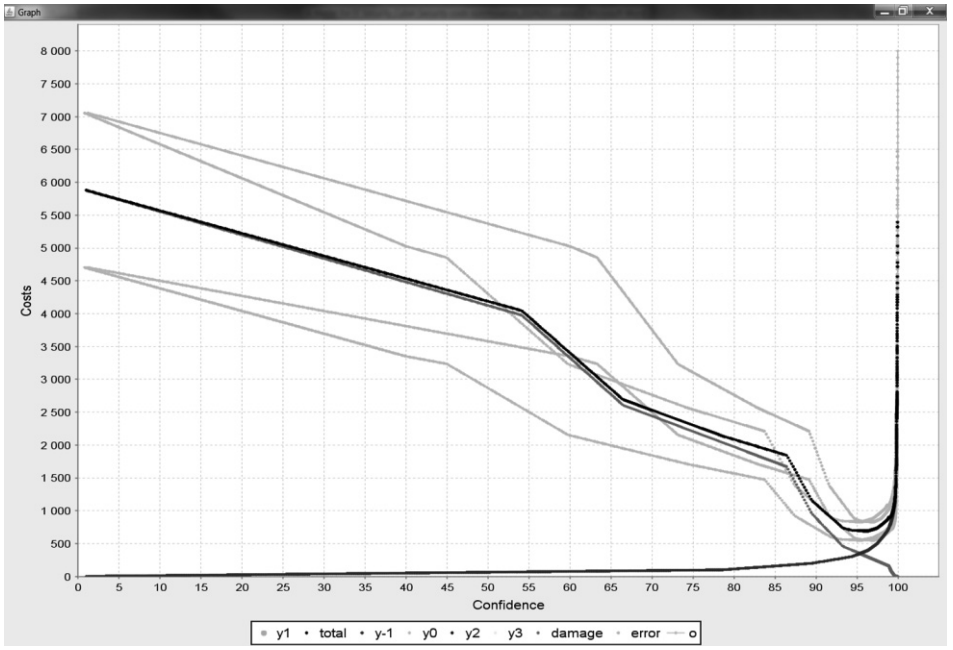


Figure 16. The Total Cost *"function"* if Confidence&Costs&Losses +20% and –20% (grey curves).

However, the greatest potential source for error in expert assessments is the Loss assessment.

Concrete error calculations of Losses from the Bank CASE STUDY:

- We assume that the Loss=f(Effectiveness) curve that we got from the questionnaire is correct and calculated the optimal information security cost was – 13000 (thousand €)
- We assume -20% error – meaning Loss=0,8*f(Effectiveness) – optimum is 12750 (thousand €).
- We assume +20% error – meaning Loss=1,2*f(Effectiveness) – optimum is 12750 (thousand €).

……

- We assume -80% error – meaning Loss=0,2*f(Effectiveness) – optimum is 10000 (thousand €).
- We assume +80% error – meaning Loss=1,8*f(Effectiveness) – optimum is 15250 (thousand €).

Error %	**Opt Budget**		**Losses**		**TC**		**mR**		**E**	
-80	10000	-23%	3420	-71%	13420	-46%	101	-34%	0,99013	-0,34%
-40	11500	-12%	8300	-30%	19800	-20%	126	-18%	0,9923	-0,12%
	12250	-6%	7525	-36%	19775	-20%	139	-10%	0,9928	-0,07%
-20	**12750**	**-2%**	9238	-22%	21988	-11%	147	-5%	0,9932	-0,03%
0	**13000**	**---**	**11800**	**---**	**24800**	**---**	**154**	**---**	**0,9935**	**---**
+20	**13350**	**3%**	13012	9%	26362	6%	162	5%	0,99384	0,03%
+40	13500	4%	14865	26%	28365	14%	162	5%	0,99384	0,03%
+80	15000	15%	17838	51%	32838	32%	176	14%	0,99432	0,08%
	15250	17%	17583	51%	32833	32%	178	15%	0,99439	0,09%

Table 10. The error calculations for the Bank model (Figure 5).

The topic requires more detailed analysis, but we can say as a first rough assessment that, for example, in GSES for the ±20% presumed errors result in optimum Budget(optimum IT security cost) values in the range of ±2÷3% (see Figure 13 ÷ 16, Table 10).

In general, we can say that our model's fault tolerance is good if we guarantee the accuracy of expert assessments within ±20%.

At the same time, an interesting theoretical problem arises – alternative optimal solutions (see Table 10):

- For example, in case of +80%, if we definitely spend 250000€ less (Opt Budget 15000 thousand € vs 15250 thousand €), then the probable Loss

increases by 255000€ (Losses 17838 thousand € vs 17583 thousand €) and the potential Total Cost would be 5000€ greater (32838 thousand € vs 32833 thousand €).
This means there are two choices:

1. definitely save 250000€ and maybe lose 5000€ in Total Cost or
2. definitely spend 250000€ more and maybe gain 5000€ in Total Cost.

Instinctively the first option seems better, i.e. to definitely save 250000€ and then later maybe lose 5000€.

- An analogous situation exists with -40%: if we definitely spend 750000€ less (Budget↓), then the probable Loss increases by 775000€ and the probable Total Cost would increase by 25000€.

It is clear that for 10^{26} possible different security profile variations there are tens of alternative profiles that have a very similar value to the optimal solution that we have calculated. If we also take into account that a good accuracy for expert assessments is ±20%, then it is very difficult to claim with full certainty that some close alternative may not be the actual optimum.

The existence of alternative solutions is a peculiarity of information security cost optimization – something that must be acknowledged, as well as researched in more detail.

PART III PUBLICATIONS

STUDY I

GRADED SECURITY EXPERT SYSTEM

Kivimaa, Jüri; Ojamaa, Andres; Tyugu, Enn

Kivimaa, Jüri; Ojamaa, Andres; Tyugu, Enn (2008). Graded security expert system. CRITIS 2008 : Third International Workshop on Critical Information Infrastructure Security, Villa Mondragone, Monte Porzio Catone, Rome, October, 13-15, 2008, (Pre-Proceedings). AIIC, ENEA, 2008, 333 – 339.
Classification: 3.4

Kivimaa, Jüri; Ojamaa, Andres; Tyugu, Enn (2008). Critical Information Infrastructure Security: Third International Workshop, CRITIS 2008, Rome, Italy, October 13-15, 2008, Revised Papers: (Eds.) Setola, Roberto; Geretshuber, Stefan. Berlin: Springer, 2009, (Lecture Notes in Computer Science; 5508), 279 – 286.
ISSN: 0302-9743
ISBN: 978-3-642-03551-7
Classification: 3.1

ABSTRACT.

A method for modeling graded security is presented and its application in the form of a hybrid expert system is described. The expert system enables a user to select security measures in a rational way based on the Pareto optimality computation using the dynamic programming for finding points of Pareto optimality curve. The expert system provides a rapid and fair security solution for a class of known information systems at a high comfort level.

Keywords: Graded security, coarse-grained security analysis, Pareto optimal security evaluation

1 Introduction

Graded security model have been in use for a long time in the high-risk areas like nuclear waste depositories, radiation control etc. [1]. Also in cyber security, it is reasonable to apply a methodology that enables one to select rational security measures based on graded security, and taking into account the available resources, instead of using only hard security constraints prescribed by standards.

It is well known that complete (100%) security of an information system is impossible to achieve even with high costs. A common practice is to prescribe the security requirements that have to be guaranteed with a suffciently high degree of confidence for various classes of information systems. This is the approach of most security standards, e.g. [2]. However, a different approach is possible when protecting a critical information infrastructure against the cyber attacks – one may have a goal to provide the best possible defense with given amount of resources (at the same time considering the standard requirements). This approach requires a considerable amount of data that connects security measures with required resources and security measures with provided degree of security.

Practically, only a coarse-grained security can be analyzed in such a way at present, using a finite number of levels (security classes) as security metrics. This is a basis of the graded security methodology. This approach has been successfully applied in the banking security practice and included at least in one security standard [3]. The ideas of graded security are based on the US Department of Energy security model from 1999 [4] and its updated version from 2006 [5].

The graded security model itself is intended for helping to determine a reasonable set of needed security measures according to security requirements levels. However, in practice it can be the case that there are not enough resources to achieve the baseline. In this case it is still desirable to invest the limited amount of resources as effectively as possible, i.e. to find and apply an optimal set of security measures.

The data required for estimating required resources and security measures can be presented in the form of expert knowledge in an extendable expert system. At present, this expert system can include at least the data that have been used in the banking security design, in particular in a branch of the Swedish bank SEB. Using an expert system has the advantage that it provides flexibility in selecting the required values for the security analysis – the values can be selected based on various input data, and even default values can be used in some noncritical places.

The present paper is organized as follows: the graded security model is presented in Section 2, the optimization method for finding a Pareto optimal curve depending on available resources is described in Section 3, and Section 4 gives a brief overview of the whole software system together with a demo example of security analysis.

2 Graded Security Model

In the present section we briefly explain the basic concepts of the graded security model: security goals, classes and measures as well as costs related to the security measures. We use integrated security metrics for representing the overall security of a system. We explain the way these entities are related.

Conventional goals of security are confidentiality, integrity and availability. In this presentation, that is based mainly on banking security, we use the following four slightly different security goals: confidentiality (C), integrity (I), availability (A) and satisfying mission criticality (M). The model can be extended by including additional security goals. A finite number of levels are introduced for each goal. At present, we use four levels 0, 1, 2, 3 for representing required security, but the number of levels can vary for different measures. The lowest level 0 denotes absence of requirements.

Security class of a system is determined by security requirements that have to be satisfied. It is determined by assigning levels to goals, and is denoted by respective tuple of pairs, e.g. C2I1A1M2 for the system that has second level of confidentiality C, first level of integrity I etc.

To achieve the security goals, some security measures have to be taken. There may be a large number of measures. It is reasonable to group them into security measures groups. Let us use the following nine groups in our simplified examples which are based on an educational information assurance video game CyberProtect [6]:

1. – user training,
2. – antivirus software,
3. – segmentation,
4. – redundancy,
5. – backup,
6. – firewall,
7. – access control,
8. – intrusion detection,
9. – encryption.

The number of possible combinations of security levels for all security goals is $4^4 = 256$. This is the number of different security classes in our case, see Fig. 1. A security class determines minimal required security levels for each group of security measures. Abstract security profile is an assignment of security levels (0, 1, 2 or 3) to each group of security measures. Hence, in the present example, we have totally $4^9 = 2621144$ abstract security profiles to be considered. The number of security measures groups may be larger in practice, e.g. 20. This gives a big number of abstract security profiles – 4^{20} for 20 groups. Knowing the costs required for implementing security measures of any possible level, one can calculate the costs of implementing a given abstract security profile.

After selecting security levels for a security measures group, one can find a set of concrete measures to be taken. For example, in the case of the security level 1 for the group "user training" the following measures have to be taken:

- New employees must be instructed for security – procedures and practice must be explained.
- An employee must know security related rights and obligations, must understand security practice, know about handling of passwords and keys.
- An employee must be instructed about security regulations and should be motivated to follow the regulations. Help about security must be available for all users of information systems.

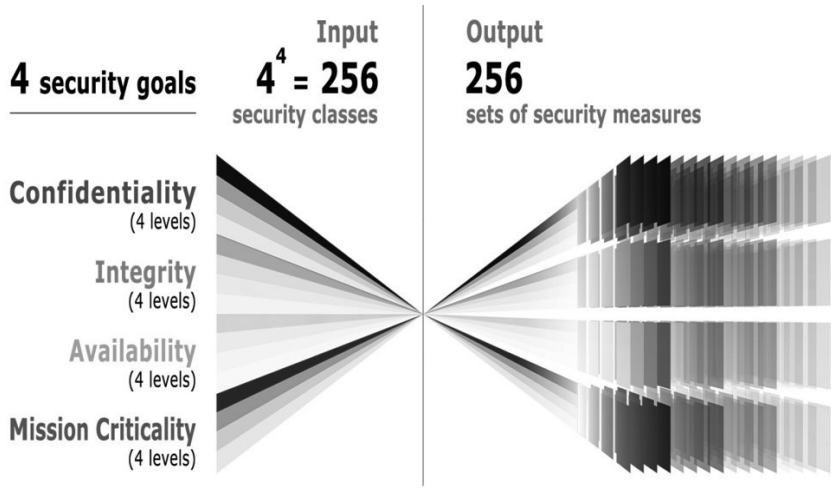


Fig. 1. Security classes of graded security model

This information is kept in the knowledge modules of the expert system of security measures, see Section 4.
It is assumed that, applying security measures, one achieves security goals with some confidence. The security confidence li is described by a numeric value between 0 and 100 for each group of security measures i = 1,...,n, where n is the number of groups.
We describe overall security of a system by means of an integrated security metrics – the security is evaluated by weighted mean security confidence S:

$$S = \sum_{i=1}^{n} a_i l_i \ ,$$

where li is security confidence of i-th security measures group, ai is a weight of the i-th group, i = 1,...,n, and

$$\sum_{i=1}^{n} a_i = 1 \ .$$

Information about costs, required security measures and confidence levels needed for calculations is presented in the expert system that will be described in Section 4.

3 Optimization Technique

Finding optimal amount of resources to be spent for security is considerably more complex problem than calculating resources required for implementing security measures of a given security class. First, a security class prescribes security requirements and respectively – spending of some minimally required amount of resources rmin. Applying expert knowledge, it is easy to calculate also resources rmax that can be optionally spent for achieving the maximal possible security level –

$$S_{max} = \sum_{i=1}^{n} a_i l_{max\,i} \ ,$$

where lmax i is maximal security confidence of the i-th group of security measures. Applying some resources between the values rmin and rmax, one can get better security in a rational way. We have an optimization problem with two goals: to minimize resources on the interval [rmin,rmax] and to maximize security, preferably guaranteeing the levels prescribed by a given security class. We are going to solve this problem by finding the abstract security profile that has maximal value of a fitness function given by the weighted mean security for a given value of resources. Repeating this calculation for suffciently many values of resources on the interval [rmin,rmax], we get a Pareto optimal solution of the problem expressed by a Pareto optimality tradeoff curve of the form shown in Fig. 2. Finally, the calculated optimal abstract security profile is compared to the concrete security profile prescribed by the security class – security levels should not be less than prescribed by security requirements.

The exhaustive search of optimal solutions for q possible values of resources, n security measures groups and k security levels requires testing (calculating weighted mean) of qk^n points. Building optimal solutions gradually, for 1,2,...,n security measures groups enables us to use discrete dynamic programming, and to reduce considerably the search time. Indeed, the fitness function S defined on intervals from j to k as

$$S(j,k) = \sum_{i=j}^{k} a_i l_i$$

is additive on the intervals, because from the definition of the function S we have

$$S(1,n) = S(1,k) + S(k,n) \ .$$

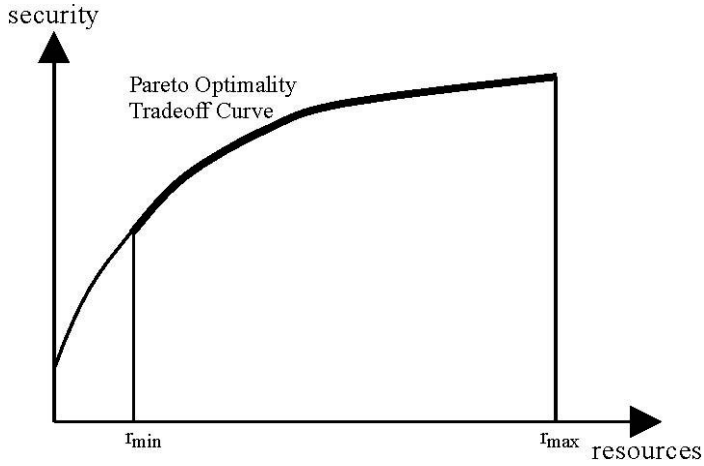


Fig. 2. Search of optimal security along resource dimension

This means that one can build an optimal resource assignment to security measures groups gradually, as a path in the space with coordinates x1, x2, where x1 equals to the number of security measures groups that have got resource (i.e. x1 = k) and x2 equals to the amount of used units of resources (1,2,...,1000 in our example). Figure 3 shows a search step, where known optimal partial solutions (assignments of resources to already tested security measures groups) are the paths from initial state (where no resources are assigned) to intermediate states s1,...,sn. The aim is to find one step longer optimal paths from a to the states t1,...,tm that follow the states s1,...,sn. This can be done for each security measures group i = 1,...,n by trying out all possible continuations of the given partial optimal paths to s1,...,sn as shown in Fig. 3. This algorithm requires testing of q^2kn points (q is the number of possible values of resources, k is the number of security levels, n is number of security measures groups).

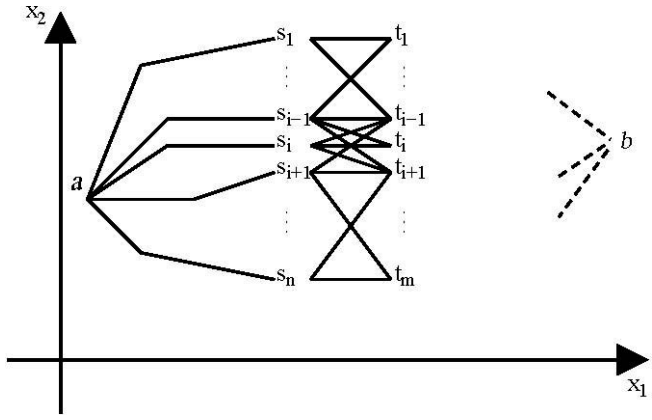


Fig. 3. Resource assignment by means of discrete dynamic programming

4 Security Expert System

A hybrid expert system with visual specification language for security system description has been built on the basis of a visual programming environment CoCoViLa [7]. The system includes knowledge modules (rule sets) in the form of decision tables for handling expert knowledge of costs and gains, as well as for selecting security measures for each security group depending on the required security level. Other components are an optimization program for calculation Pareto optimality curve parameterized by available resources, and a visual user interface for graphical specification of the secured system, visual control of the solution process through a GUI, and visualization of the results. These components are connected through a visual composer that builds a Java program for each optimization problem, compiles and runs it on the request of the user, see Fig. 4.

Let us explain the usage of the expert system on the following simplified example. We have nine security measures groups as given in Section 2. Two groups – "user training" and "encryption" – have specific values of cost and confidence related to security levels that must be given as an input. We can use standard values of cost and confidence given in the expert knowledge modules for other groups. We have to solve the problem in the context of banking and can use resources measured in some units on the interval from 1 to 70. The security class C2I1A1M2 is given as an input. The expected outcome is a graph that shows the weighted mean security confidence depending on the resources that are used in the best possible way. The graph should also indicate whether the security goals specified by the security class can be achieved with the given amount of resources. Besides that, the curves showing security confidence provided by user training and redundancy must be shown.

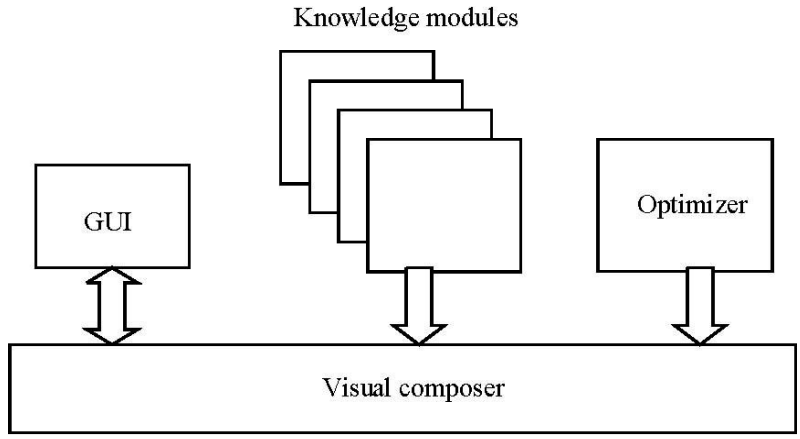


Fig. 4. Graded security expert system

The visual composer is provided by the CoCoViLa system that supports visual model-based software composition. The main window of the expert system shown

in Fig. 5 presents a complete description of the given problem. It includes also visual images of components of the expert system and a toolbar for adding new components, if needed. In particular, new security measures groups can be added by using the third and fourth button of the toolbar. Besides the security measures groups there are three components – Optimizer, SecClass and GraphVisualizer – shown in the window. The components in the main window can be explicitly connected through ports. This allows us to show which values of security should be visualized ("user training" and "redundancy" in the present case) etc. There are two different views of security measures groups – "user training" and "encryption" that have explicit values of costs and confidence given as an input. Other groups use the standard values of costs and confidence given in the expert knowledge modules as specified in the problem description. The SecClass component is used for specifying security goals. During computations the component also evaluates the abstract security profiles calculated by the Optimizer against the actual security requirements using a knowledge module from the expert system.

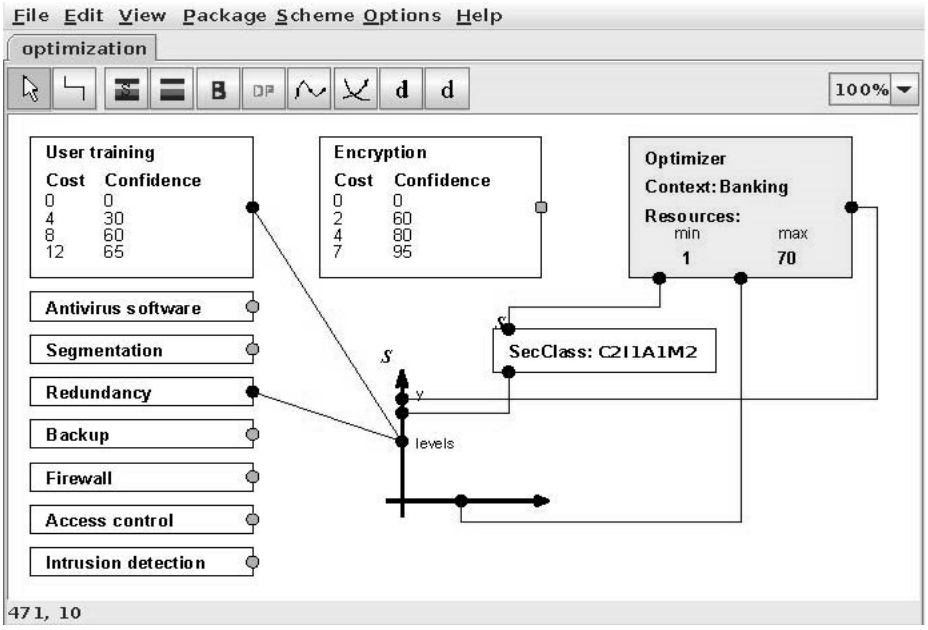


Fig. 5. Problem specification window

In Fig. 6 there is a window showing the optimization results. The first curve (Confidence) represents the optimal value of weighted mean security confidence depending on the resources that are used in the best possible way. This curve is further divided into four parts to visualize to which degree the optimal result satisfies the security requirements given by the security class. The first part (thin black line) indicates the interval of resources where none of the four (in our example) security goals can be achieved. The second part (thin grey line, three

separate segments) shows that at least one of the security goals is satisfied while also at least one is not. The third part (thick black line) represents the amount of resources that, when used optimally, would result in satisfying the requirements exactly. One should note that this coincidence of the optimal security profile and the security requirements does not always exist. The last part of the graph (thin black line, again) shows the amounts of resources that are more than is strictly needed to satisfy the requirements. It is interesting to notice that on the interval of costs from 36 to 45 units it is possible to satisfy all security goals, because already spending 34 units enables one to do this. However, the solutions with highest values of the weighted mean security confidence do not satisfy all security goals on this interval.

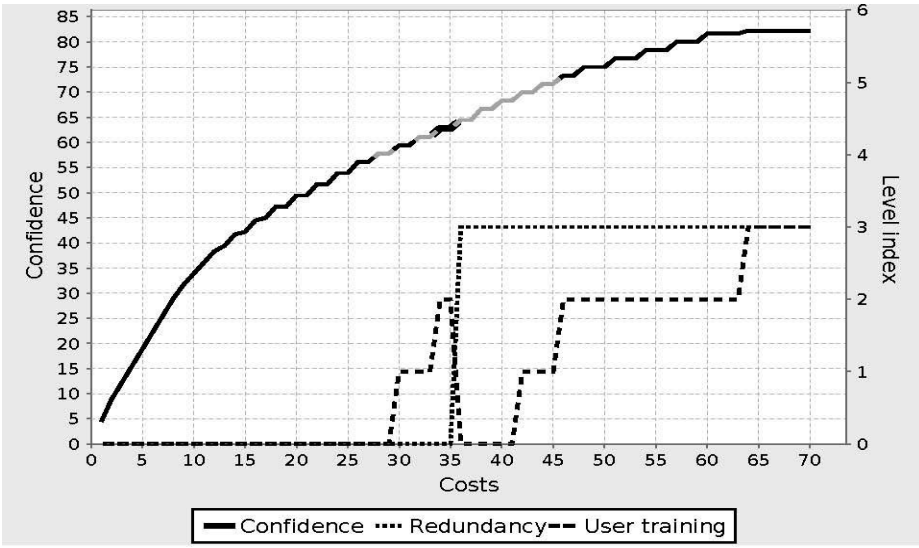


Fig. 6. Solutions window

The lower graphs indicate (on the right scale) the optimal levels of two measures groups corresponding to the given amount of resources. These graphs are not necessarily monotonic as can be seen in this example at the resource values 35 and 36. When there are 35 units of resources available it is reasonable to apply the measure “user training” at level 2. Having one more unit of resources better overall security confidence level is achieved by taking all resources away from “user training” and investing into the “redundancy” measures group to achieve level 3.

5 Conclusions

The advantage of the expert system of the graded security is that it provides a rapid security solution at a suffciently high although not 100% confidence level. Based on our previous experience, the graded security expert system allows a typical security solution to be developed within approximately 8 hours, with about half the time spent on security class identification and the other half on analyzing available resources, accepted security risks, attack costs and other optimization variables. Our method reduces the time for analysis and provides a Pareto optimal solution. It includes:

- graded security selection procedure that yield the security measures for a given security class;
- high-level analysis of usage of resources for information security and accepted risks based on advanced optimization technique.

We understand that wider application of this method will depend on the availability of expert knowledge that binds costs and security confidence values with taken security measures. This knowledge can be collected only gradually, and will depend on the type of the critical infrastructure that must be protected.

References

Kang, Y., Jeong, C. H., Kim, D. I. Regulatory approach on digital security of instrumentation, control and information systems in nuclear power plants. Korea Institute of Nuclear Safety. Daejeon, Korea. http://entrac.iaea.org/I-and-/TM_ IDAHO_2006/CD/

German Federal Office for Information Security (BSI). IT Baseline Protection Manual. 2005. http://www.bsi.de/gshb/

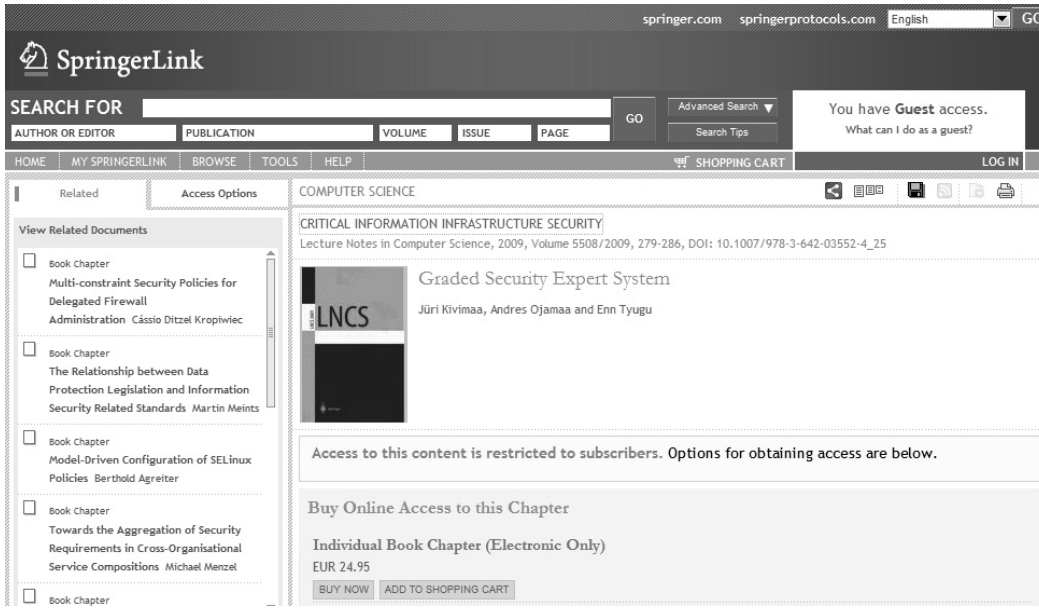
Estonian Information Systems Three-Level Security Baseline System – ISKE ver. 1.0.

U. S. Department of Energy, Office of Security Affairs. Classified Information Systems Security Manual. 1999.

U. S. Department of Defense. National Industrial Security Program Operating Manual (NISPOM). 2006.

U. S. Department of Defense, Defense Information Systems Agency. CyberProtect, version 1.1. July 1999. http://iase.disa.mil/eta/product_description.pdf

Grigorenko, P., Saabas, A., Tyugu, E. Visual tool for generative programming. ACM SIGSOFT Software Engineering Notes, 2005, 30, 5, 249-252.


springer.com
springerprotocols.com
English
GO
SpringerLink
SEARCH FOR
AUTHOR OR EDITOR
PUBLICATION
VOLUME
ISSUE
PAGE
GO
Advanced Search
Search Tips
You have Guest access.
What can I do as a guest?
HOME
MY SPRINGERLINK
BROWSE
TOOLS
HELP
SHOPPING CART
LOG IN
Related
Access Options
COMPUTER SCIENCE
View Related Documents
Book Chapter
Multi-constraint Security Policies for Delegated Firewall Administration Cássio Ditzel Kropiwiec
Book Chapter
The Relationship between Data Protection Legislation and Information Security Related Standards Martin Meints
Book Chapter
Model-Driven Configuration of SELinux Policies Berthold Agreiter
Book Chapter
Towards the Aggregation of Security Requirements in Cross-Organisational Service Compositions Michael Menzel
Book Chapter
CRITICAL INFORMATION INFRASTRUCTURE SECURITY
Lecture Notes in Computer Science, 2009, Volume 5508/2009, 279-286, DOI: 10.1007/978-3-642-03552-4_25
LNCS
Graded Security Expert System
Jüri Kivimaa, Andres Ojamaa and Enn Tyugu
Access to this content is restricted to subscribers. Options for obtaining access are below.
Buy Online Access to this Chapter
Individual Book Chapter (Electronic Only)
EUR 24.95
BUY NOW
ADD TO SHOPPING CART

STUDY II

PARETO-OPTIMAL SITUATON ANALYSIS FOR SELECTION OF SECURITY MEASURES

Ojamaa, Andres; Tyugu, Enn; Kivimaa, Jyri

Ojamaa, Andres; Tyugu, Enn; Kivimaa, Jyri (2008).
Pareto-optimal situation analysis for selection of security measures.
MILCOM 2008.

MILCOM 2008: Assuring Mission Success: Unclassified Proceedings, November 17-19 San Diego:, 2008, 3224 – 3230.
ISBN: 978-1-4244-2677-5
Classification: 3.1

Andres Ojamaa and Enn Tyugu, Institute of Cybernetics of Tallinn University of Technology Tallinn, Estonia.
Jyri Kivimaa, Estonian Defence Forces Training and Development Centre of Communication and Information Systems Tallinn, Estonia.

ABSTRACT

A methodology of selection of security measures is presented and a prototype implementation in the form of a hybrid expert system is described. This expert system is applicable, first of all, in the security management. It enables a user to select security measures in a rational way based on the Pareto optimality computation using a discrete dynamic programming method. This enables one to select rational countermeasures taking into account the available resources instead of using only hard constraints prescribed by standards. The prototype expert system is presented that provides a rapid security solution for a class of known information systems. Coarse-grained security can be analyzed in such a way at present, using a finite number of levels (security classes) as security metrics. This is a basis of the graded security methodology.

1. INTRODUCTION

Selection of security measures is a complex problem due to the fact that multiple objectives must be achieved at the same time. Considering data security, the security goals can be confidentiality, integrity and availability. Besides that, a security officer may want to keep costs reasonably low from one side, and reach the security goals with as high confidence as possible. Low cost and high confidence are two universal goals. The complexity has been an obstacle to finding optimal solutions for the security management problem. Another obstacle has been the absence of reliable metrics for measuring the said goals.[13*]

Graded approach has been applied earlier in standards covering areas other than information security [3]. In re- cent years a graded security method has been developed and used in a number of areas, not necessarily in information assurance [4]. This method relies on a coarse-grained metrics for the security goals and achieved confidences. It is successfully applied as a basis for security standards that prescribe concrete security measures for achieving a required level of confidence for each security goal [5, 6]. The method is not immediately applicable for finding an optimal solution of the security problem.

[13] "Good metrics are those that are SMART, i.e. specific, measurable, attainable, repeatable, and time-dependent, according to George Jelen of the International Systems Security Engineering Association [1]. Truly useful metrics indicate the degree to which security goals, such as data confidentiality, are being met, and they drive actions taken to improve an organization's overall security program [2]."

We are going here to use the metrics of the graded security method and build a model that binds taken security measures with costs and confidences of achieving the goals. We introduce a fitness function that presents by one numeric value the integral confidence of achieving the security goals. This allows us to formulate a problem of selecting security measures as an optimization problem in precise terms. However, we still have two goals: to minimize the costs and to maximize the integral security confidence. This problem will be solved by means of building a Pareto optimality tradeoff curve that explicitly shows the relation between used resources and security confidence. Then, knowing the available resources, one can find the best possible security level that can be achieved with the resources and find the security measures to be taken. From the other side – if the required security level is given one can find the resources needed and the measures that have to be taken. This requires solving an optimization problem for each value of resources. As the number of possible security measures (that are in principle the independent variables of the optimization problem) is large, we have grouped the measures into security measures groups that will be characterized by security confidence levels. Taking the confidence levels of the groups as independent variables, we get an optimization problem of a reasonable size that can be solved by means of a discrete dynamic programming method.

The presented method of finding optimal security measures is in principle applicable in different situations, in particular, for designing overall security of a communication network, for designing a security of a critical information infrastructure of a bank etc. However, the method requires considerable amount of data that bind costs and confidences with security measures groups as well as expert knowledge that binds concrete security measures with a selected security confidence requirements level of a group. In the end of the present paper we give an example of an expert system developed for banking security that has the data and has been used for experimenting. Most of the expert knowledge of this kind can be extracted from standards or internal security policies of the bank or other organization that must have them before trying to optimize the security.

2. GRADED SECURITY MODEL.

In the present section we briefly explain the basic concepts of the graded security model that gives functional dependencies for our optimization method. We are go- ing to use integrated security metrics for representing the overall security of a system. Conventional goals of security are *confidentiality* (C), *integrity* (I), and *availability* (A). The model can be extended by including additional security goals. A finite number of security levels are introduced for each goal. This is a coarse-grained metrics, but the only available in this context at present. We use four levels 0, 1, 2, 3 for representing required security, but the number of levels can vary for different measures. The lowest level 0 denotes absence of

special protective measures. *Security class* of a system is determined by security requirements that have to be satisfied. It is determined by assigning levels to goals, and is denoted by a respective tuple of pairs, e.g. C2I1A1N2 for the system that has second level of confidentiality C, first level of integrity I and availability A and second level of non-repudiation N.

To achieve the security goals, proper *security measures* have to be taken. There may be a large number (hundreds) of measures. It is reasonable to group them into *security measures groups* g_1, g_2, . . . , g_n. The grouping should be done in such a way that measures of one and the same group will be always used for achieving one and the same level of security. We will need a function *f* that produces a set of required security measures *f(l, g)* for a given security measures group g and a security level l of the group.
A security class determines the required security level for each group of security measures. Let us denote by s a respective function that produces a security level s(c, g) for a group g when the security class is c. Abstract security profile is an assignment of security levels (0, 1, 2 or 3) to each group of security measures. This can be expressed by the tuple $p = (s(c, g_1), s(c, g_2), \ldots, s(c, g_n))$, where p denotes the abstract security profile an the elements of the tuple p are indexed and appear in the tuple in the same order as the groups of security measures.

For n security measures groups we have totally 4^n abstract security profiles to be considered. The number of security measures groups may be in practice up to 20 or even more. This gives a number of abstract security profiles: 4^{20}. (If we had considered all security measures without grouping them, then we had got an incomprehensibly large number of security profiles – 4k, where k is several hundreds.)

Knowing the cost function h that gives the costs *h(l, g)* required for implementing security measures of a group g for a level l, one can calculate the costs of implementing a given abstract security profile:

$$costs(p) = \sum_{i=1}^{n} h(l_i, g_i),$$

where $p = (l_1, l_2, \ldots, ln)$.
Our goal is to keep the value *costs(p)* as low as possible.

The information for calculating values of functions *f, h, c* and *s* should be kept in the knowledge modules of an expert system of security measures.

It is assumed that, applying security measures, one achieves security goals with some confidence. The security confidence *q* of a group *g* that satisfies the security level *l* is given by a function *q(l, g)* and it is a numeric value between 0 and 100 for each group of security measures.

We describe overall security of a system by means of an *integrated security metrics* that is a weighted mean security confidence *S*, called *also integrated security level:*

$$S = \sum_{i=1}^{n} a_i q_i ,$$

where q_i is security confidence of the *i*-th security measures group, *ai* is a weight of the *i*-th group, and

$$\sum_{i=1}^{n} a_i = 1 .$$

In the simplest case $a_i = 1/n$, and the integral security confidence is the average confidence of security measures groups. The information about the weights ai, as well as about the costs, required security measures and confidence levels needed for calculations must be presented in an expert system.

3. OPTIMIZATION TECHNIQUE

Now we can formulate an optimization problem as follows: "find the abstract security profile *p* with the best (highest) value of *S* for given amount of resources *r*, so that $costs(p) \leq r$." We have introduced all functions needed for calculating *S* and *costs* in the previous section. Independent variables whose values have to be found by optimization are the security levels assigned to security measures groups: $l_1, l_2, \ldots, l_n$. If the security class c is given, then the solution has to satisfy also the constraints

$l_i \geq s(c_i, g_i), \; i = 1, 2, \ldots, n$.

Remark. The graded security model presented in Section 2 is usually used for finding (for a given security class) the required security levels of security measures groups and respective costs and concrete measures to be taken. This problem is considerably simpler that the optimization problem considered here.

Let us solve the optimization problem in the general case when also a security class is given. First, a security class prescribes only minimal security requirements and respectively – spending of some minimal amount of resources *rmin*. It is easy to calculate also resources *rmax* that can be reasonably spent for achieving the maximal possible integrated security level –

$$S_{max} = \sum_{i=1}^{n} a_i q_{max\, i} ,$$

where $q_{max\, i}$ is maximal security confidence of the i-th group of security measures.

Applying some resources between the values r_{min} and r_{max}, one can get better security in a rational way. Now we have an optimization problem with two goals: to minimize resources on the interval $[r_{min}, r_{max}]$ and to maximize security, guaranteeing at least the levels prescribed by a given security class. We are going to solve this problem by finding a function that gives the abstract security profile that has maximal value of a security.

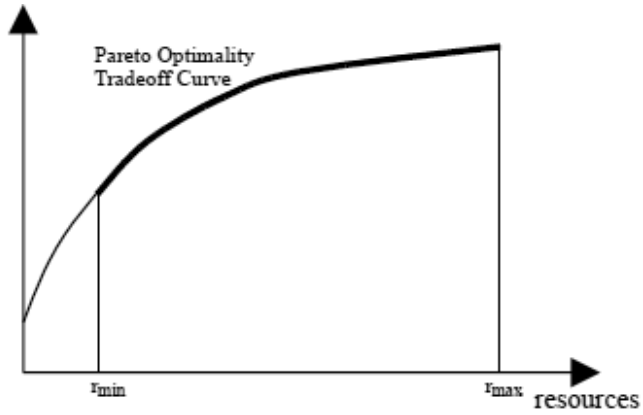


Figure 1. Search of optimal security along resource dimension

fitness function S given by the weighted mean security for any given value of resources on the interval $[r_{min}, r_{max}]$. This gives us a Pareto optimal solution of the problem expressed by a *Pareto optimality tradeoff curve* of the form shown in Fig. 1. In the case when the minimal security requirements are not strict for security measures groups, then it is reasonable to compute Pareto optimality even for resources less than r_{min}. This can be done, if the optimization procedure is sufficiently fast, like in our case.

The exhaustive search of optimal solutions for m possible values of resources, n security measures groups and k security levels requires testing (calculating weighted mean confidentiality) of mk^n points.

Building optimal solutions gradually, for 1, 2, . . . , n security measures groups enables us to use discrete dynamic programming, and to reduce considerably the search. Indeed, the fitness function S defined on intervals from j to k as

$$S(j,k) = \sum_{i=j}^{k} a_i l_i \,,$$

is additive on the intervals, because from the definition of the function S we have

$$S(1, n) = S(1, k) + S(k, n) \,.$$

This means that one can build an optimal resource assignment to security measures groups gradually, as a path in the space with coordinates x_1, x_2, where x_1 equals to the number of security measures groups that have got resource (i.e. $x_1 = k$) and x_2 equals to the amount of used units of resources (1, 2, . . . , 1000 in our example). The discrete dynamic programming method requires using of a finite number of values of a resource (x_2). This number of values depends on the precision that is required. A precision that can be achieved using expert knowledge is not very high, usually a hundred points is sufficient. As our optimization procedure works sufficiently fast we are using 1000 points. Fig. 2 shows a search step, where known optimal partial solutions (assignments of resources to already tested security measures groups) are the paths from initial state (where no resources are assigned) to intermediate states $s_1, \ldots, s_n$. The aim is to find one step longer optimal paths from a to the states $t_1, \ldots, t_m$ that follow the states $s_1, \ldots, s_n$. This can be done for each security measures group $i = 1, \ldots, n$ by trying out all possible continuations of the given partial optimal paths to $s_1, \ldots, s_n$ as shown in Fig. 2. This algorithm requires testing of $m^2 n$ points (m is number of possible values of resources, n is number of security measures groups).

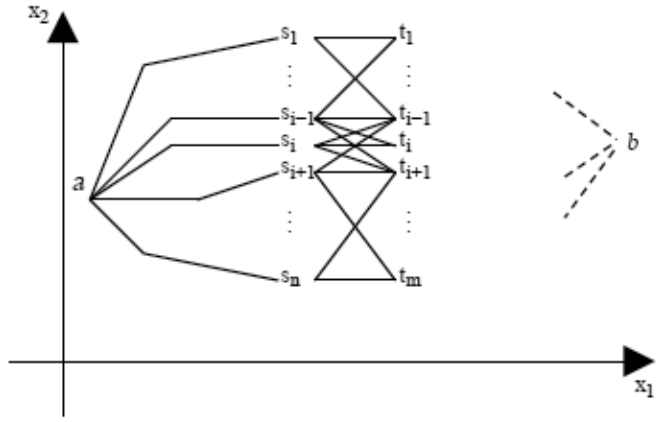


Figure 2. Resource assignment by means of discrete dynamic programming.

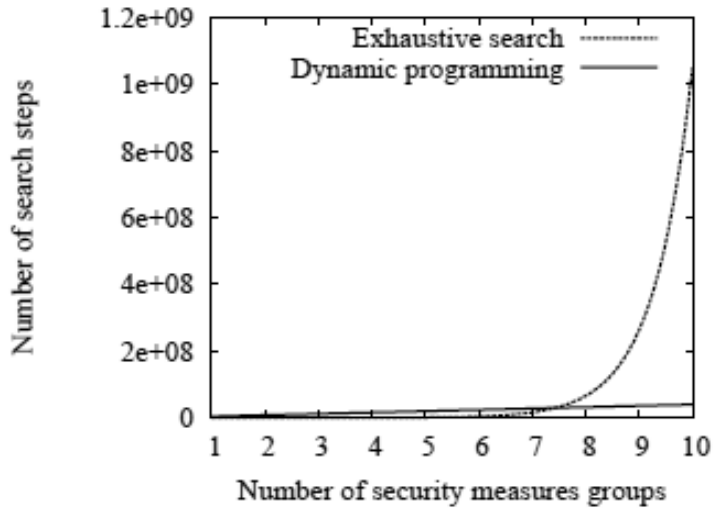


Figure 3. Complexity of search.

In Figure 3 it is shown how the number of search steps (and consequently search time) depends on the number of security measures groups for the number of groups 10. Our method has linear complexity, the search time grows linearly with the number of groups. The exhaustive search used initially grows exponentially.

4. APPLICATION EXAMPLE

We have developed a prototype of a security expert system for selecting security measures in banking. This expert system has been developed in a visual programming environment CoCoViLa [8]. Let us explain its functioning on an example. Here we use the following four security goals: confidentiality (C), integrity (I), availability (A) and satisfying mission criticality (M). We use the following nine security measures groups in our simplified example which are based on an educational information assurance video game CyberProtect [7]:

- firewall,
- access control,
- intrusion detection,
- encryption,
- user training,
- antivirus software,
- segmentation,
- redundancy,
- backup.

After selecting security levels for a security measures group, one can find a set of concrete measures to be taken. For example, in the case of the security level 1 for the group "user training" the following measures can be found from the expert system:

- New employees must be instructed for security – procedures and practice must be explained.
- An employee must know security related rights and obligations, must understand security practice, know about handling of passwords and keys.
- An employee must be instructed about security regulations and should be motivated to follow the regulations. Help about security must be available for all IS users.

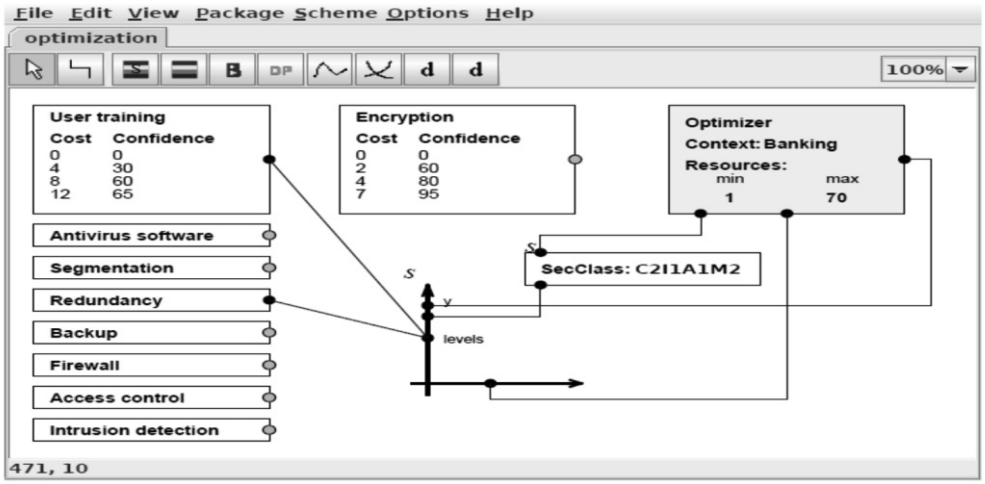


Figure 4. Main window of security expert system

The main window of the expert system is presented in Fig. 4. It includes visual specification of our problem. The specification is a scheme where components are security measures groups and other software components that are used for solving the problem. The usable components are represented by buttons in the menu bar on top of the scheme. In the scheme we see images of all security measures groups. Besides the security measures groups there are three components *Optimizer, SecClass* and *GraphVisualizer* shown in the scheme. Two groups "user training" and "encryption" have specific values of cost and confidence related to security levels that are explicitly given as an input. We use standard values of cost and confidence given in the expert knowledge modules for other groups. We have to solve the problem in the context of banking and can use resources measured in some units on the interval from 1 to 70 that is shown in the Optimizer block. The security class C2I1A1M2 is given as a separate block as well. The blocks in the main window are connected through ports. This allows us to show which values of security should be visualized ("user training" and "redundancy" in the present case) etc. The expected outcome is a graph produced by the *GraphVisualizer* that

shows the weighted mean security confidence depending on the resources that are used in the best possible way. The graph should also indicate whether the security goals specified by the security class can be achieved with the given amount of resources. Besides that, the curves showing security confidence provided by user training and redundancy will be shown, see the respective connection lines between the visual images.

In Fig. 5 there is a window showing the optimization results. The first curve (Confidence) represents the optimal value of weighted mean security confidence depending on the resources that are used in the best possible way. This curve is further divided into four parts to visualize to which degree the optimal result satisfies the security requirements given by the security class. The first part (thin black line) indicates the interval of resources where none of the four (in our example) security goals can be achieved. The second part (thin grey line, three separate segments) shows that at least one of the security goals is satisfied while also at least one is not. The third part (thick black line) represents the amount of resources that, when used optimally, would result in satisfying the requirements exactly. One should note that this coincidence of the optimal security profile and the security requirements does not always exist. The last part of the graph (thin black line, again) shows the amounts of resources that are more than is strictly needed to satisfy the requirements. It is interesting to notice that on the interval of costs from 36 to 45 units it is possible to satisfy all security goals, because already spending 34 units enables one to do this. However, the solutions with highest values of the weighted mean security confidence do not satisfy all security goals on this interval.

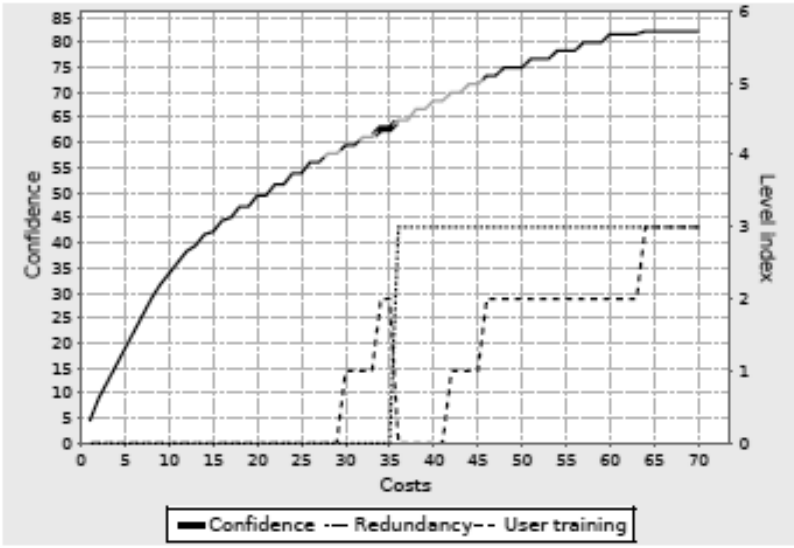


Figure 5. Solution of the problem.

The lower graphs indicate (the scale shown on the right) the optimal levels of two measures groups corresponding to the given amount of resources. These graphs are not necessarily monotonic as can be seen in this example at the resource values 35 and 36. When there are 35 units of resources available it is reasonable to apply the measure "user training" at level 2. Having one more unit of resources better overall security confidence level is achieved by taking all resources away from "user training" and investing into the "redundancy" measures group to achieve level 3.

5. CONCLUDING REMARKS

In the present work we have developed a method for systematic design of a security solution of an information or communication system, and the method is explained on an example from the banking security. The method relies on a graded security model used in practice in different applications. The novelty of the method is, first, the usage of an advanced optimization technique based on discrete dynamic programming and, second, the output of many alternative solutions in the form of a Pareto optimality tradeoff curve that enables the user to select the best security solution depending on availability of resources.

Another novelty is introduction and usage of an integral security measure in the form of a weighted mean security confidence. The method performs security situation analysis using coarse-grained metrics for security levels of partial solutions (security measures groups) from one side, and an integrated security metrics in the form of weighted mean security confidence from the other side. A tool developed as a prototype supports visual presentation of a general view of a security situation and enables one to perform the situation analysis on different levels of details, e.g. using standard functions of confidences and costs or presenting them as additional inputs. Time required for automated analysis, when a set of input data is given, is only a few seconds. This enables one to perform the analysis rapidly for many different assumptions.

We understand that wider application of this method will depend on the availability of expert knowledge that binds costs and security confidence values with taken security measures. This knowledge can be collected only gradually, and will depend on the type of the critical infrastructure that must be protected. However, our expectation is that more expert knowledge will be collected when interactive analysis applications with graphical user interface such as the prototype presented in this paper become available

ACKNOWLEDGEMENTS

The first author would like to thank the Estonian Information Technology Foundation and the Tiger University program for partial support of this work.

REFERENCES

[1] G. Jelen. SSE-CMM Security Metrics. NIST and CSSPAB Workshop, Washington, D.C., 13-14 June 2000. http://csrc.nist.gov/csspab/june13-15/jelen.pdf (10 July 2001).

[2] S. C. Payne. A Guide to Security Metrics. SANS Reading Room, 2006. http://www.sans.org/reading room/whitepapers/

[3] C. E. Pasterczyk. A graded approach to ISO 9000 implementation for records managers. Association of Records Managers and Administrators international annual conference, Toronto (Canada), 25–29 Sep 1994.

[4] Y. Kang, C. H. Jeong, and D. I. Kim. Regulatory approach on digital security of instrumentation, control and information systems in nuclear power plants. Korea Institute of Nuclear Safety. Daejeon, Korea. http://entrac.iaea.org/I-and-/TM IDAHO 2006/CD/

[5] German Federal Office for Information Security (BSI). IT Baseline Protection Manual. 2005. http://www.bsi.de/gshb/

[6] U. S. Department of Defense. National Industrial Security Program Operating Manual (NISPOM). 2006.

[7] U. S. Department of Defense, Defense Information Systems Agency. CyberProtect, version 1.1. July 1999. http://iase.disa.mil/eta/product description.pdf

[8] P. Grigorenko, A. Saabas, E. Tyugu. Visual tool for generative programming. ACM SIGSOFT Software Engineering Notes, 2005, 30, 5, 249-252

http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4753520

Pareto-optimal situaton analysis for selection of security measures

This paper appears in:
Military Communications Conference, 2008. MILCOM 2008. IEEE
Date of Conference: 16-19 Nov. 2008

Author(s): Ojamaa, A.
Inst. of Cybern., Tallinn Univ. of Technol., Tallinn
Tyugu, E. ; Kivimaa, J.

Page(s): 1 - 7
Product Type: Conference Publications

ABSTRACT

A methodology of selection of security measures is presented and a prototype implementation in the form of a hybrid expert system is described. This expert system is applicable, first of all, in the security management. It enables a user to select security measures in a rational way based on the Pareto optimality computation using a discrete dynamic programming method. This enables one to select rational countermeasures taking into account the available resources instead of using only hard constraints prescribed by standards. The prototype expert system is presented that provides a rapid security solution for a class of known information systems. Coarse-grained security can be analyzed in such a way at present, using a finite number of levels (security classes) as security metrics. This is a basis of the graded security methodology.

STUDY III

APPLYING A COST OPTIMIZING MODEL FOR IT SECURITY

Kivimaa, Jüri

Kivimaa, Jüri (2009).
Applying a Cost Optimizing Model for IT Security. 8th ECIW, 2009.

Proceedings of the 8th European Conference on Information Warfare and Security: Lisbon, Portugal, 06-07.07.2009. Reading, UK: Academic Conferences Limited, 2009, 142 – 153.
ISBN: 978-1-906638-34-4 Cd
Classification: 3.1
Jyri Kivimaa
Cooperative Cyber Defence Centre of Excellence, Tallinn, Estonia
jyri.kivimaa@mil.ee

Abstract

In real life good solution today is quite often better than perfect solution after month(s). That's the reason why we are developing IT Security/Cyber Security Graded Security Expert System - for quick and economically rational/optimal specifying needed security measures to protect concrete information accordingly to its concrete needed/required security goals/goals levels.

Graded Security Expert System is based on the high level risk analysis (gives mainly a required levels of information security goals), on the Graded Security methodology (DOE 1999, NISPOM 2006) and on an IT security costs optimizing function/model.

Keywords: graded security model, Pareto optimal security evaluation, high level risk analysis, information security metrics, information security requirements.

1. Introduction

Information security is a growing priority for organizations, many of which are struggling to decide the appropriate amounts of investments to counter threats to availability, confidentiality and integrity of information systems that put interlinked business processes at risk. The investments in security countermeasures usually have the characteristics of externalities since one entity's investment decision affects the utility of other entities that are connected to it. Despite information security being a priority issue for many enterprises, the evaluation of investments in information security as well as how to determine company's policies is poorly understood. Effective countermeasures exist for many of the security threats, but are often not optimally deployed. Deciding how best to invest resources in information security is not straightforward. The difficulty is compounded by multiple uncertainties about threats and vulnerabilities, about the consequences of a successful attack, and about the effectiveness of mitigation measures. Given the challenge of ensuring information security under conditions of uncertainty, how can organizations determine appropriate measures to enhance cyber security and allocate resources most efficiently?

To define the security measures a high level security model is needed. It should be noted that security models are too complex to be developed in a particular enterprise – from this follows that investigations to develop a generic models

is needed. The generic model could be adapted for the specific enterprise. And using an expert system that is based on the generic model has the advantage that it provides flexibility in selecting the required and optimal security solution to secure concrete data in concrete information system in a concrete enterprise.

The important issue in defining and implementing security measures is the economical efficiency of security activities, that is − we want to get the best results for our money. Using a well-defined security model we can assure that the approach based on this model is effective, that is – we can specify minimal costs to achieve the needed security level and guarantee the cost-efficiency for our IT security investments (the best security/maximal security confidence level for the enterprise). Accordingly - a cost optimizing model/utility for our security model should be developed for the optimal allocation of resources to achieve the best possible security goals for the enterprise.

Our objective is to improve the consistency of the Risk Assessment methods which are currently being used (mainly detailed risk analysis and baseline security methodologies). We have found two good ideas – the US DoD/DoE/CIA/... graded security methodology (*Best Practice* security methodology to specify needed security measures for needed security levels) and Estonian governmental data classification (metrics to specify needed security level) – and connecting them we have made our version of Graded Security.

Our main ideas are:

- use metrics to determine information systems security requirements - i.e. use high level risk analysis (levels of security goals) as IT security metrics;
- secure IT systems and their information in an economically rational/optimal manner – i.e. accordingly to data security requirements;
- have fair and satisfactory security solution *today* - i.e. we must be able to specify the list of needed data security measures for the ICS the day we need them.

Fields of use:

- for small and medium enterprises (SME) - it is practically the only usable/executable model for SME, because usually they lack resources for IT security in the needed quantity;
- quickly find out customers/co-partners IT security compliance to our security requirements;
- quickly find out reasonable IT security costs for budget.

The present paper is organized as follows. In the next section we present briefly the graded security method that provides the functional dependencies needed for calculations. A separate section (Section 3) is devoted to the discussion of the integral security metrics needed for comparing the solutions. The following Section 4 includes a brief description of the software used for making calculations,

limitations to optimization based on high level risk analyze results and model's precision.

2. Graded security model

Graded approaches have been applied earlier and in areas other than information security – as example by Pasterczyk for ISO 9000 in 1994. In information security this method relies on coarse-grained metrics for the security goals and required security measures to assure these goals (from 1999 - Classified Information Systems Security Manual, U. S. Department of Energy, Office of Security Affairs). It is successfully applied as a basis for security standards that prescribe concrete security measures for achieving a required security level for each security goal. Look tables 1-3 from NISPOM (2006: 8-4-3 and 8-4-4) as examples how achievable security goals levels (Low/Middle/High for CIA) depend on engaged levels in security activities areas – i.e. on executed/realized security measures in these areas. However, this method is not immediately applicable for finding an optimal solution of the security problem.

Table 1: Protection Profile Table for Confidentiality

Confidentiality Protection Level			
Requirements (Paragraph)	P L 1	PL 2	PL 3
Audit Capability (8-602)	Audit 1	Audit 2	Audit 3, Audit 4
Data Transmission (8-605)	Trans 1	Trans 1	Trans 1
Access Controls (8-606)	Access 1	Access 2	Access 3
Identification & Authentication (8-607)	I&A 1	I&A 2,3,4	I&A2,4,5
Resource Control (8-608)		ResrcCtrl 1	ResrcCtrl 1
Session Controls (8-609)	SessCtrl 1	SessCtrl 2	SessCtrl 2
Security Documentation (8-610)	Doc 1	Doc 1	Doc 1
Separation of Functions (8-611)			Separation
System Recovery (8-612)	SR 1	SR 1	SR 1
System Assurance (8-613)	SysAssur 1	SysAssur 1	SysAssur 2
Security Testing (8-614)	Test 1	Test 2	Test 3

Table 2: Protection Profile Table for Integrity

Integrity Level of Concern			
Requirements (Paragraph)	Basic	Medium	High
Audit Capability (8-602)	Audit 1	Audit 2	Audit 3
Backup and Restoration of Data (8-603)	Backup 1	Backup 2	Backup 3
Changes to Data (8-604)		Integrity 1	Integrity 2
System Assurance (8-613)		SysAssur 1	SysAssur 2
Security Testing (8-614)	Test 1	Test 2	Test 3

Table 3: Protection Profile Table for Availability

Availability Level of Concern			
Requirements (Paragraph)	Basic	Medium	High
Alternate Power Source (8-601)		Power 1	Power 2
Backup and Restoration of Data (8-603)	Backup 1	Backup 2	Backup 3

As security metrics (information security goals/requirements levels to specify the needed security activity levels) in our expert system we use the Estonian governmental data classification – i.e. more concrete levels for information security requirements/goals (as example for CIA). Shortly – levels High/Middle/Low are not concrete enough on country level (what is *high* for one institution, is *middle* for second and *low* for third). As example, quite concrete and similarly understandable for all institutions *availability* (A) levels are *not important, 90%, 99%* and *99.9%.*

We are going to use the metrics of the graded security method and build a model that binds taken security measures with costs and confidence levels to achieve the goals. We introduce a fitness function that presents an integral confidence of achieving the security goals by one numeric value. This allows us to formulate a problem of selecting security measures as an optimization problem in precise terms. However, we still have two goals: to minimize the costs and to maximize the integral security confidence. This problem will be solved by means of building a Pareto optimality trade-off curve that explicitly shows the relation between used resources and security confidence. Then, knowing the available resources, one can find the best possible security level that can be achieved with the resources and find the security measures to be taken.

In the present section we briefly explain the basic concepts of the graded security model that gives functional dependencies for our optimization method. We are going to use integrated security metrics for representing the overall security of a system. Conventional goals of security are confidentiality (C), integrity (I) and availability (A). The model can be extended by including additional security goals. As example non-repudiation, authenticity, mission criticality will be added

for Information Assurance/Cyber Security. A finite number of security levels are introduced for each goal. This is a coarse-grained metric, but the only one available in the present context. We use four levels (0, 1, 2, 3) for representing required security, but the number of levels can vary for different measures. The lowest level 0 denotes absence of special protective measures. Security class of a system is determined by security requirements that have to be satisfied. It is determined by assigning levels to goals, and is denoted by a respective tuple of pairs, e.g. C2I2A1 for the system that has second level of confidentiality C, second level of integrity I and first level of availability A.

Practically SecClass is high level expert opinion to information security risks: secure IT systems and their information according to data security requirements - no more (if achieved security level(s) are higher than required then security expenses are consequently higher than needed) and no less (too many security incidents and accordingly too much security loss) than needed.

A security class is variation with recurrences and a finite number of possible different security classes/ a number of possible different security grades is:

$VR_n^m = n^m$, where n is a number of possible different security goals l

evels and m is a number of possible different security goals.
For m security goals and 4 levels we have a total of 4^m abstract different security grades to be considered – for conventional CIA (m=3) 64 grades, for Cyber Security (Information Assurance) is realistic m=5 (or 6) and correspondingly 1024 (or 4096) grades.
Graded model gives us the reasonable/rational levels for security activities– i.e. reasonable/rational security costs.

3. Optimization technique

(This chapter is mainly based on Kivimaa, J., Ojamaa, A. and Tyugu, E. “Pareto-optimal security situation management”.)

To achieve the security goals, proper security measures have to be taken. There is a large number (hundreds, in several standards/methodologies even roughly a thousand) of possible measures. It is reasonable to group them into groups by security activity areas (and corresponding security measures) $g_1, g_2, \ldots, g_n$ (as example in IT groups are perimeter protection, access control, encryption etc.). We will need a function *f* that produces a set of required security measures *f(l; g)* for a given security measures group *g* and a security level *l* of the group.

A security class determines the required level (possibly the same 4 levels as for security goals) for each group of security measures (Figure 6). Let us denote by *s*

a respective function that produces a security level *s(c; g)* for a group *g* when the security class is *c*. Abstract security profile is an assignment of security levels (0, 1, 2 or 3) to each group of security measures. This can be expressed by the tuple $p = (s(c; g_1); s(c; g_2); : : : ; s(c; g_n))$, where *p* denotes the abstract security profile and the elements of the tuple *p* are indexed and appear in the tuple in the same order as the groups of security measures.

For *n* security measures groups we have totally 4^n abstract security profiles to be considered. The number of security measures groups may be in practice up to 20 or even more. This gives a number of abstract security profiles: 4^{20}.

Knowing the cost function h that gives the costs h(l; g) required for implementing security measures of a group *g* for a level *l*, one can calculate the costs of implementing a given abstract security profile:

$$costs(p) = \sum_{i=1}^{n} h(l_i, g_i), \text{ where } p = (l_1, l_2, \ldots, l_n) .$$

The information for calculating values of functions *f*, *h*, *c* and *s* should be kept in the knowledge modules of a graded security expert system.

It is assumed that applying security measures, one achieves security goals with some confidence. The security confidence *q* of a group *g* that satisfies the security level *l* is given by a function *q(l, g)* and it is a numeric value between 0 and 100 for each group of security measures.

We describe overall security of a system by means of an integrated security metrics that is a weighted mean security confidence S, called also integrated security level:

$$S = \sum_{i=1}^{n} a_i q_i ,$$

where q_i is security confidence of the i-th security measures group, a_i is a weight of the i-th group, and

$$\sum_{i=1}^{n} a_i = 1 .$$

The weight of the security measures group depends of the security goals guaranteed by this group (for example encryption can help to protect information security and integrity, but not availability) and the importance of guaranteed goals to the concrete enterprise's concrete information system (for example in banking information/ICS integrity is the most important part for main business information systems , but for ISP's it is availability).

In the simplest case $a_i = 1/n$, and the integral security confidence is the average confidence of security measures groups. The information about the weights a_i, as

well as about the costs, required security measures and confidence levels needed for calculations must be presented in an expert system.

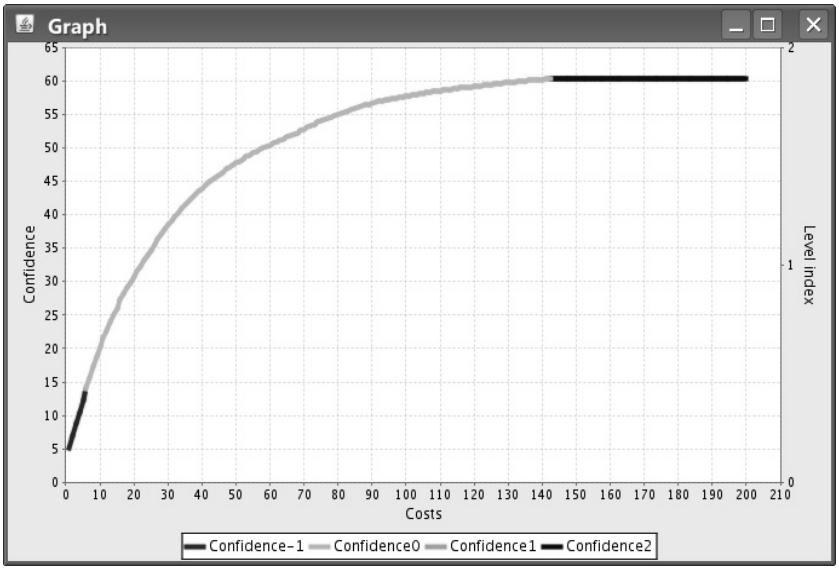
Remark. Using weighted mean approach is first version on view of information security activities areas/security measures groups dependencies. It gives possibility to trim our model to specific needs of the concrete IS's of the concrete institution (as example in banking the most important is the integrity of information, but for medicine and ISP's may-be availability and so on).

Now we can formulate an optimization problem as follows: "find the abstract security profile p with the best (highest) value of S for given amount of resources r, so that costs(p) $\leq$ r ". We have introduced all functions needed for calculating S and costs in the previous section.

We have an optimization problem with two goals: to minimize resources on the interval $[r_{min}; r_{max}]$ and to maximize security, guaranteeing at least the levels prescribed by a given security class. We are going to solve this problem by finding a function that gives the abstract security profile that has maximal value of a security confidence function *S* given by the weighted mean security for any given value of resources on the interval $[r_{min}; r_{max}]$.

The task of the optimization application is to find the best combination of security measure levels which provide the maximum confidence at possible cost. For example, one can get better confidence by lowering the security level of one security measure and for the cost saved by this increase the level of another security measure, provided the security measure level which was lost provided less confidence than the security measure level which was gained.

This optimization is performed at each budget level, as if asking - „For every possible budget level, what is the maximum confidence one can expect?" Plotting the increasing budget levels with the optimal confidence levels will give us a graph, visualizing the possibilities of expenditure.



Red line – all security activities area's security levels are ≤ and at least one is < than required
Green point/line – all security goals/their required levels are exactly achieved.
Yellow line - at least one security level is less and at least one security level is more than required.
Blue line – all security levels are ≥ and at least one security level is > than required.

Figure 1: Costs/confidence optimality curve.

There are mainly two optimization algorithms to solve our task – one is a brute force optimizer and the other is based on a Pareto optimality (Pareto frontier or Pareto set) and discrete dynamic programming method.

With brute force we must do qk^n computations and with the dynamic programming method q^2kn (q is number of possible values of resources, k is the number of security levels, n is number of security measures groups).

In developing our security costs optimization utility we use 9_security_areas–based on cost/efficiency data from CyberProtect 1.1 and there are no serious problems to optimize using both algorithms.

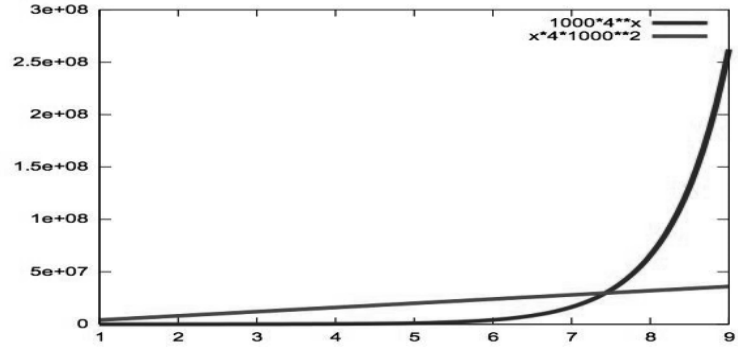


Figure 2: Computation comparison for BruteForce and Pareto for 9 areas.

It is obvious that this 9-area version is quite simplified - in CyberProtect 1.1 these cover only one of the six main IT security activity areas (others are administrative, personnel, physical, media and comsec&tempest controls/protections).

NISPOM (pages 8-4-3 and 8-4-4) has divided security into 14 activities/security measures areas (Tables 1-3).

Nowadays it is realistic to have more than 20 security activity areas, if grouping IT security measures to IT security activities areas is tied to security costs and expert working areas.

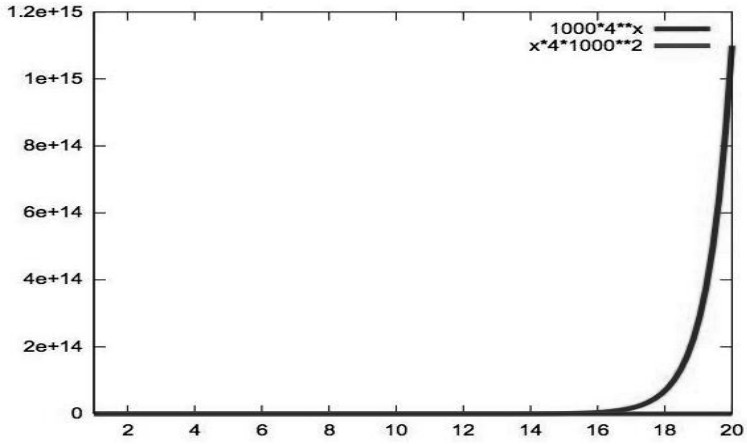


Figure 3: Computations comparison for BruteForce and Pareto for 20 areas.

To compare: if with the Pareto optimality & dynamic programming we have a curve for 100 budget points in ~3 seconds then *Brute Force* would take ~10 years to calculate it - i.e. that in up-to-date security costs optimization model/expert system it is only feasible to use the Pareto optimality computation with discrete dynamic programming.

Building optimal solutions gradually, for 1, 2, . . . , n security measures groups enables us to use discrete dynamic programming, and to reduce the search considerably. Indeed, the fitness function S defined on intervals from j to k as

$$S(j,k) = \sum_{i=j}^{k} a_i l_i \,,$$

is additive on the intervals, because from the definition of the function S we have S(1, n) = S(1, k) + S(k, n).

I.e. – to use dynamic programming in optimization presume that security activities areas/security measures groups must be not dependent from each other's. Independency between IT security activities areas is quite problematic, but in first approximation it is acceptable (if for example IT security experts/specialists training costs are included into the costs of concrete security activities areas/areas levels and some other analogical principles must be followed). In the future we plan to cover these problems in more detail - use (find or work out) the information security requirements levels and information security activities areas realization levels dependency graph.

4. Application example

We base the development of optimization functions to our graded security system on a visual simulation and decision-making environment with Intelligent User Interface (i.e. input-problem specification and visual output) called CoCoViLa. The system includes knowledge modules (rule sets) in the form of decision tables for handling expert knowledge of costs and gains, as well as for selecting security measures for each security group depending on the required security level – in development stage from CyberProtect 1.1 (Figure 5). Other components are an optimization program for calculating Pareto optimality curve parameterized by available resources, and a visual user interface for graphical specification of the secured system, visual control of the solution process through a GUI, and visualization of the results. These components are connected through a visual composer that builds a Java program for each optimization problem, compiles and runs it on the request of the user (Figure 4).

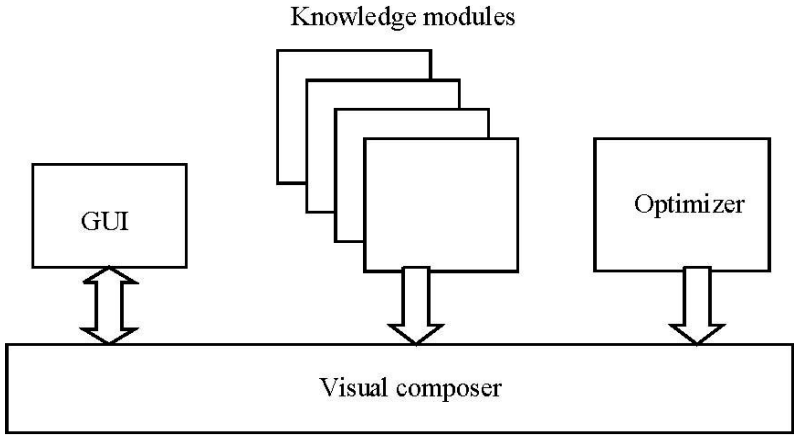


Figure 4: Graded security expert system

Let us explain the usage of the expert system on the following simpe example – in development stage we secure our hardware/software/firmware based on nine security activity/measure groups, their high/middle/low level realization costs and effectiveness's from CyberProtect 1.1 (Figure 5).

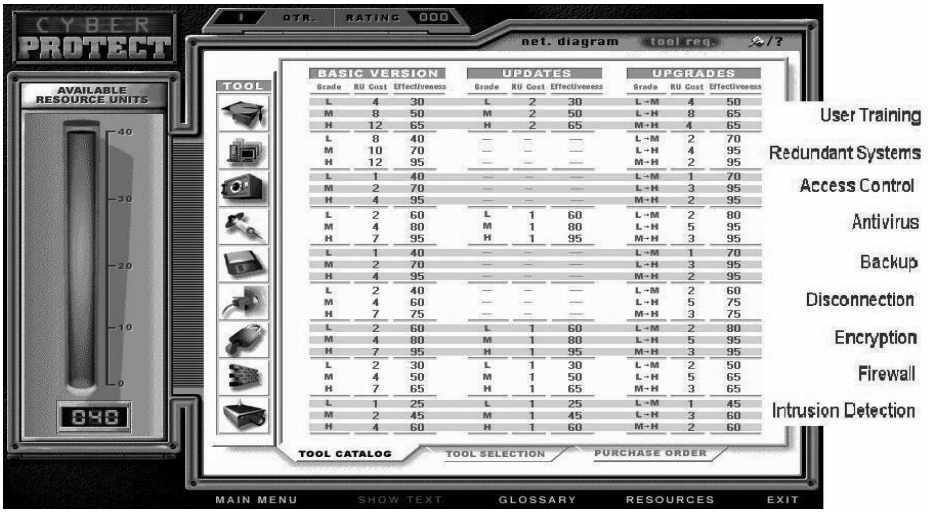


Figure 5: IT security costs/confidence data from CyberProtect 1.1

Expert knowledge is lead into expert system by decision tables (in our case the information security requirements levels and information security activities areas realization levels dependency matrix) - i.e. basic ideas of graded security are presented as a decision table. For example, a decision table of relations between security requirement levels and security activity area levels.

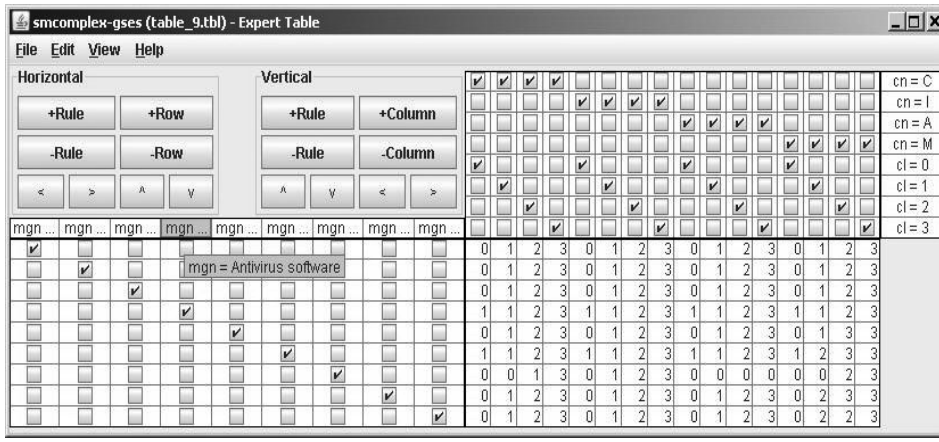


Figure 6: Knowledge Modules as Decision Tables

The visual composer is provided by the CoCoViLa system that supports visual model-based software composition. The main window of the expert system shown in Figure 7 presents a complete description of the given problem. It includes also visual images of components of the expert system and a toolbar for adding new components, if needed. In particular, new security measure groups can be added by using the third and fourth button of the toolbar. Besides the security measure groups there are three components – Optimizer, SecClass (in detail 4.1) and GraphVisualizer – shown in the window. The components in the main window can be explicitly connected through ports. This allows us to show which values of security should be visualized ("user training" and "redundancy" in the present case). There are two different views of security measures groups – "user training" and "encryption" that have visualized explicit values of costs and confidence given as an input. Other groups use the values of cost and confidence given in the expert knowledge modules as specified in the problem description. The SecClass component is used for specifying security goals. During computation the component also evaluates the abstract security profiles calculated by the Optimizer against the actual security requirements using a knowledge module from the expert system.

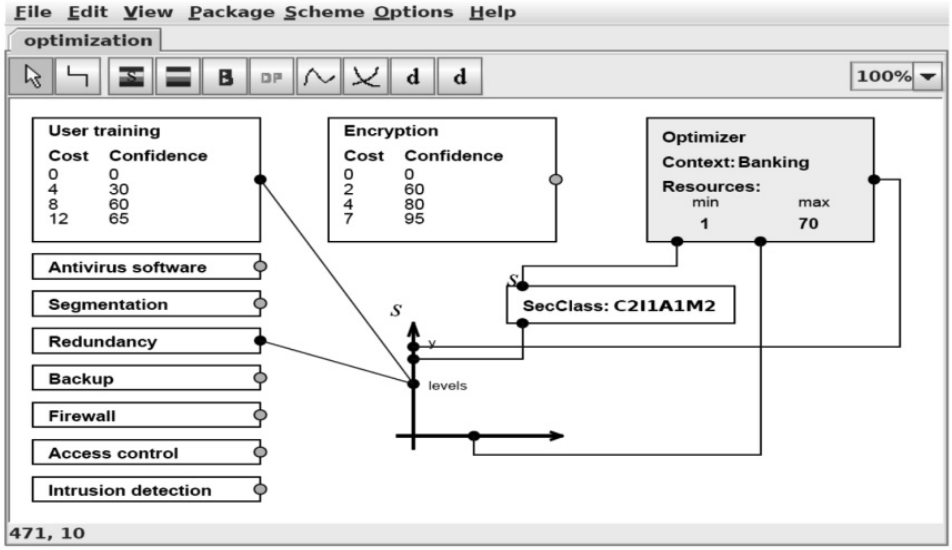
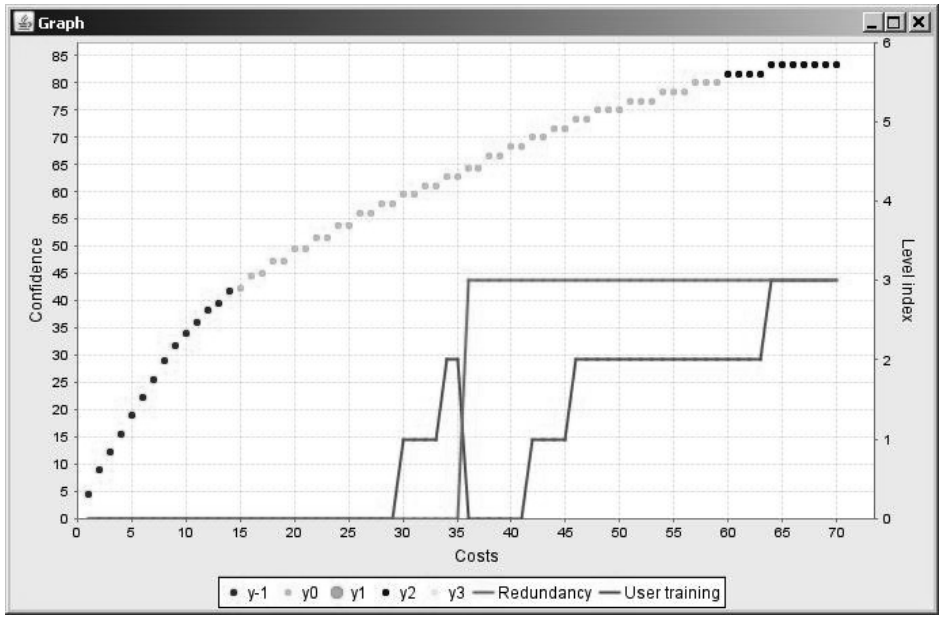


Figure 7: Visual problem specification window

In Figure 8 there is a window showing the optimization results. The curve (Confidence) represents the optimal value of weighted mean security confidence depending on the resources that are used in the best possible way. This curve is further divided into four parts to visualize to which degree the optimal result satisfies the security requirements given by the security class.

One should note that this coincidence of the optimal security profile and the security requirements does not always exist. The last part of the graph (blue line) shows the amounts of resources that are more than is strictly needed to satisfy the requirements.

The lower graphs indicate (on the right scale) the optimal levels of two measures groups corresponding to the given amount of resources. These graphs are not necessarily monotonic as can be seen in this example at the resource values 35 and 36. When there are 35 units of resources available it is reasonable to apply the measure “user training” at level 2. Having one more unit of resources better overall security confidence level is achieved by taking all resources away from “user training” and investing into the “redundancy” measures group to achieve level 3.



Red line – all security activities area's security levels are ≤ and at least one is < than required
Green point/line – all security goals/their required levels are exactly achieved
Yellow line - at least one security level is less and at least one security level is more than required
Blue line – all security levels are ≥ and at least one security level is > than required

Figure 8: Solutions window

The original algorithm of the optimization application simply calculated the optimal levels for a predefined range of budget points, assuming the desire for absolute maximum confidence level. The levels of each of the security measures were fluctuating wildly between all four levels, just to provide the absolute maximum confidence level. Even at the quit high budget, some security measures might have been left at level zero (i.e. no real security) since the first level might have had very high cost with very little confidence provided (see Figure 8).

The graded security theory accepts that there is only a limited budget to spend on increasing the security measure levels of the information systems. Also, the importance of each information system of the organization will dictate the need for its security, which might be above any cost to confidence ratio. In other words - some information systems are important enough to necessitate high expenditure without highest confidence provided, while other information systems are so unimportant that spending any considerable budget on their security is pointless.
The importance of information systems is expressed in their security classifier.

In an essence, security classifier defines the level of each security measure that is needed to reach the required security profile. While spending more is possible and will increase the security confidence, mostly it is reasonable to spend enough to meet the required levels of security measures - no more (usually too expensive, at least needs ROI analysis) and no less (too much security incidents and usually that means too much security losses) than needed.

4.1 Limitations to optimization

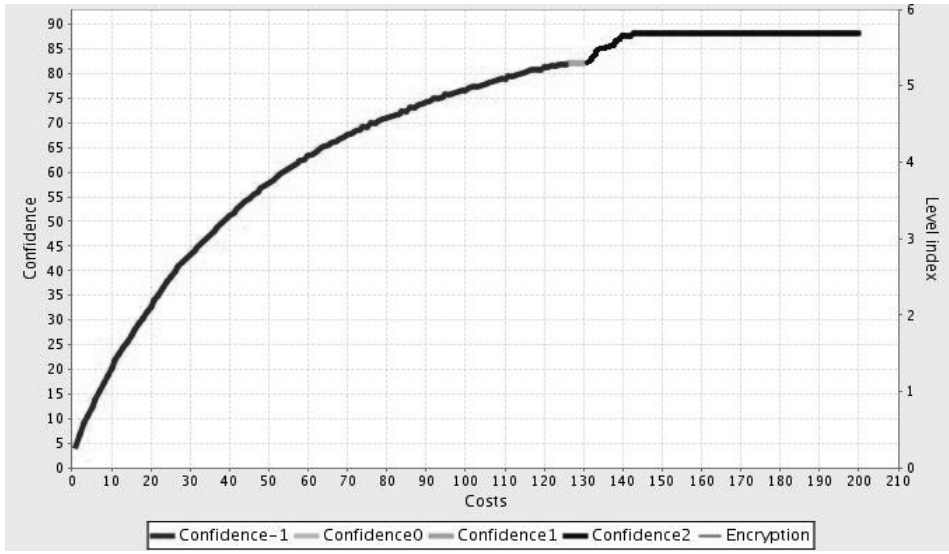
A problem is, that if in IT security costs management we only follow the costs/confidence optimality (Figure 1 – all interesting/significant/relevant part of optimality-curve is yellow), then we probably never find the optimal (green) point/segment.

The refined theory states, that at each budget point in which the required levels of the security measures are still out of reach, it is unwise (i.e. too expensive) to spend on security measure levels which are above the required ones. Only after moving with the budget beyond the point where all of the security measures have reached their required levels, any higher than required levels of security measures can be obtained. It would be equally unwise to let any security measure level drop below the required level once all of the required levels are obtained.

It means that in costs optimization we must use the required SecClass - result of high level risk analyze.

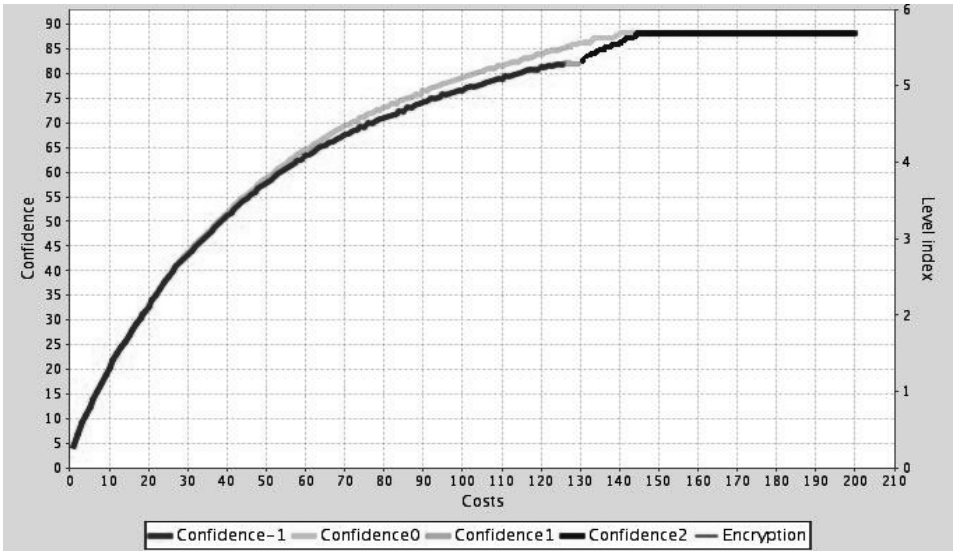
This was the first of the additions added to the original CoCoViLa application – instead of one single continuous budget expenditure graph, divide the graph into two, the first part covering the budget points before reaching all of the required security measure levels specified by the security classifier and the second part covering the rest of the budget points once the required levels are reached.

Shortly – we must optimize IT security costs/confidence but with two limitations (Fig. 9): that at each budget point in which the required levels of all of the security measures are still out of reach (i.e. too expensive), it is unwise to spend on security measure levels which are above the required ones; and that after moving with the budget beyond the point where all of the security measures have reached their required levels, only ≥ than required levels of security measures can be accepted.



Red line – all security activities area's security levels are ≤ and at least one is < than required
Green point/line – all security goals/their required levels are exactly achieved
Yellow line - at least one security level is less and at least one security level is more than required
Blue line – all security levels are ≥ and at least one security level is > than required

Figure 9: Costs/confidence optimality curve using security-class limitation.



Red line – all security activities area's security levels are ≤ and at least one is < than required
Green point/line – all security goals/their required levels are exactly achieved
Yellow line - at least one security level is less and at least one security level is more than required
Blue line – all security levels are ≥ and at least one security level is > than required

Figure 10: Costs/confidence optimality curves with and without limitation.

Without limitation case practically describe situation when needed information security requirements are maximal – SecClass= C3I3A3.

4.2 Model's precision

One of the biggest concerns was our model's sensitivity to experts estimations. It is quit good if we get experts estimations in the limits of ±10-20% and came out that generally our models fail-safety is quit good.

As with any expert system, our tool is only as good as the experts are who have provided the assessments on the costs and confidence. Hence we have computed two additional graphs which represent the best and worst case scenario within a given error margin.

With the budget cost value, it is easy to applying the error margins, the minimum value being 20% less and maximum being 20% more than given value.

The difficulty is in the ambiguity of the method in using the error margin of the confidence level, which is a percentage value itself. As implemented currently, there are several possible algorithms to do this. For the beginning, we use the simplest – add or subtract the error margin of the total average confidence value, but clip the value to 100% boundary, e.g. 90% confidence with 20% error margin will have the plus and minus points at 100% and 72% respectively (the plus point is clipped).

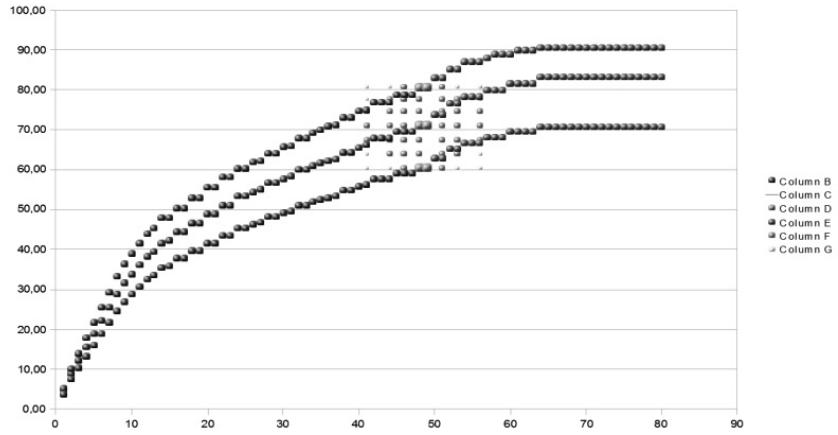


Figure 11. Confidence ± 20%.

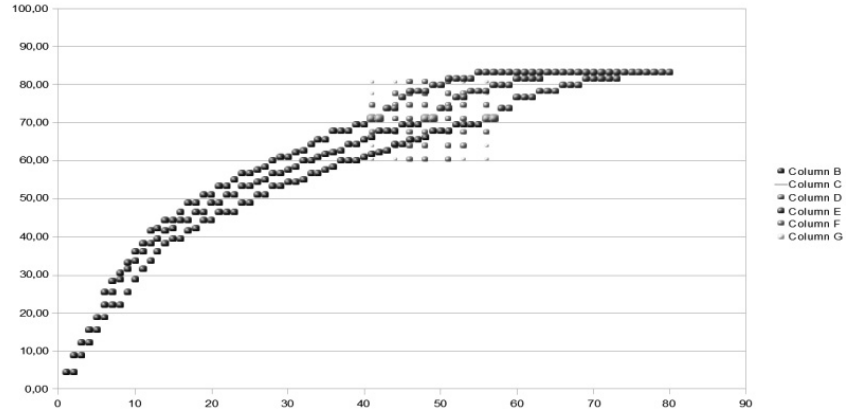


Figure 12: Costs ± 20%.

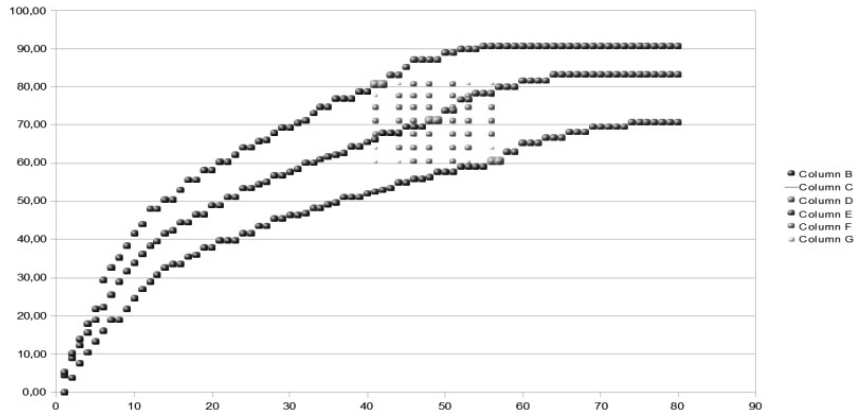


Figure 13: Confidence & costs ± 20%.

Based on Figures 11 – 13, we can conclude that our model's precision is quite good - on the most important optimality (green) point, despite the roughness of experts' estimations, we hold the optimality status (stayed green).

NB! Important is to keep optimistic or pessimistic style in expert estimations.

5. Acknowledgements

The author would like to thank Enn Tyugu and Andres Ojamaa from the Tallinn University of Technology Institute of Cybernetics for their support of this work.

6. Concluding remarks

In developing our IT security costs optimizer the present results are quite encouraging – in development Graded Security Expert System we based on year 1999 expert knowledge (CyberProtect 1.1), and opinions from information security experts with 10-20 years practice in this area are good – solutions proposed would have been realistic for that time. It seems reasonable to continue its development – mainly to collect expert knowledge for the up-to-date model – i.e. up-to-date information security requirements levels and information security activities areas realization levels dependency matrix and up-to-date theirs levels realization costs and effectiveness's.

We understand that wider application of this method will depend on the availability of expert knowledge or statistics that binds costs and security confidence values with the security measures. This knowledge could be gathered only gradually, and will depend on the type of the infrastructure where information must be protected,

are different for different countries, are different for different economy areas (as example different for banks and for ISP and so on). The only realistic solution is an expert system that experts can adjust to suit concrete situations.

However, our expectation is that more expert knowledge will be collected when interactive analysis applications with graphical user interface such as the prototype presented in this paper become available.

References

Classified Information Systems Security Manual. (1999) U. S. Department of Energy, Office of Security Affairs, 1999.

CoCoViLa - a compiler compiler for visual languages. Available from www.cs.ioc.ee/~cocovila/

Cyber Protect, version 1.1. U. S. Department of Defense, Defense Information Systems Agency. Available from: http://iase.disa.mil/eta/product description.pdf

Kivimaa, J., Ojamaa, A. and Tyugu, E. (2008) "Pareto-optimal security situation management". In *MILCOM:08, ASSURING MISSION SUCCESS*, San Diego.

NISPOM, National Industrial Security Program Operating Manual. (2006) U. S. Department of Defense.

Pasterczyk, C. E. (1994) "A graded approach to ISO 9000 implementation for records managers". In *Association of Records Managers and Administrators international annual conference*, Toronto.

STUDY IV

MANAGING EVOLVING SECURITY SITUATIONS

Kivimaa, Jyri; Ojamaa, Andres; Tyugu, Enn

Kivimaa, Jyri; Ojamaa, Andres; Tyugu, Enn (2009).
Managing evolving security situations. MILCOM 2009.
MILCOM 2009: Unclassified Proceedings, October 18-21, 2009, Boston, MA.
Piscataway, NJ: IEEE, 2009, 1 – 7.
ISBN: 978-1-4244-5239-2
Classification: 3.1

Jyri Kivimaa, Cooperative Cyber Defence Centre of Excellence, Tallinn, Estonia
Andres Ojamaa, Institute of Cybernetics at Tallinn University of Technology, Tallinn, Estonia
Enn Tyugu, Cooperative Cyber Defence Centre of Excellence, Tallinn, Estonia

ABSTRACT

A method is described that takes into account the investments done in the security and/or achieved security confidence in planning new security measures. The method uses new integral security metrics and the well-known graded security model. A precondition for the application of this method is the availability of expert knowledge or statistical data for the model in use that describes a class of situations where the analyzed security situation belongs to. For a number of situations at present, this information has been extracted from standards of graded security. For specific military communications applications the data must be collected from a log analysis of characteristic attacks and security reports, as well as by the traditional knowledge acquisition means.

1. INTRODUCTION

The security situation in cyber space is changing rapidly. This requires continuous analysis of security situations and continuous security management: selection of security measures, planning of investments for security measures groups. Our goal is to provide a method for planning security measures not only for a fixed time point, but to do this for a longer time period, possibly, in- vesting into the security gradually. This paper presents a method that is an extension of the Pareto-optimal security situation analysis implemented in an expert system [4]. It takes into account the legacy systems and security levels achieved by means of former investments. This enables one to plan the usage of resources considering evolving security situations over a longer time period.

Comprehensive security planning is a complex task. This can be seen from the complexity of standards and requirements like Common Criteria [7] or ISKE [1]. Standards prescribe minimal required measures, and usually do not include economic parameters—the costs of implementing the security measures. A detailed cost- benefit analysis of cyber security [2] may require months. An alternative approach is to manage security on the basis of security requirements. It is efficient, if reason- ably good expert knowledge of security requirements and goals is available. We have taken this approach.

A well-known graded security methodology [6, 8] is based on a comprehensive but coarse grained model, and provides a way of planning security and calculating costs. In our paper [4] we have shown how to use the graded security model for

finding optimal solutions depending on the given security situation. However, a description of a situation there reflects neither the investments already done into security nor the levels of security already achieved. Based on the application of a discrete dynamic programming method described in [5], one can solve rather complex security optimization problems on ordinary PCs and laptops. This enabled us to extend the optimization method for longer time intervals, solving the optimization problem stepwise.

This paper is organized as follows. In the next section we present briefly the graded security method that provides the functional dependencies needed for calculations. A separate section (Section 3) is devoted to the discussion of the integral security metrics needed for comparing the solutions. These metrics were introduced for the first time in [4]. The following Section 4 includes a brief description of the software used for making calculations. Section 5 includes a discussion of the influence of the legacy security on new security solutions. It presents formulas needed for planning evolving security mea- sures. Section 6 includes descriptions of solvable legacy security problems and some solutions.

2. GRADED SECURITY MODEL

Here we briefly introduce variables and functions used in the graded security model. The overall security of a system is described by a *security class*. It shows how the security goals (confidentiality, integrity, availability, . . .) are satisfied. It is determined by assigning *security levels* to *security goals*, and is denoted by a respective tuple of pairs, e.g., C2I1A1M2 for the system that has the second level of confidentiality C, the first level of integrity I etc.

To achieve the security goals, proper security measures have to be taken. There may be a large number (hundreds) of measures. It is reasonable to group them into security measures groups $g_1, g_2, \ldots, g_n$. The grouping should be done in such a way that measures of one and the same group will always be used for achieving one and the same level of security. One uses a function f that produces a set of required security measures $f(l, g)$ for a given security measures group g and a security level l of the group. A security class determines the required security level for each group of security measures. Let us denote by s a respective function that produces a security level $s(K, g)$ for a group g when the security class is K . An *abstract security profile* is an assignment of security levels (0, 1, 2, or 3) to each group of security measures. This can be expressed by the tuple $p = (s(K, g_1), s(K,g_2), \ldots, s(K,g_n))$, where p denotes the abstract security profile and the elements of the tuple p are indexed and appear in the tuple in the same order as the groups of security measures $g_1, g_2, \ldots, g_n$ have been indexed. Knowing the cost function $h(l, g)$ that gives the costs r required for implementing security measures of a group g for a level l, one can calculate the costs of implementing a given abstract security profile:

$$costs(p) = \sum_{i=1}^{n} h(l_i, g_i)\,,$$

where $p = (l_1, l_2, \ldots, l_n)$.

The goal is to keep the value *costs(p)* as low as possible, guaranteeing a required security. It is assumed that by applying security measures, one achieves security goals with some confidence. The security confidence c of a group g that satisfies the security level l is given by a function *e(l, g)* and it is a numeric value between 0 and 100 for each group of security measures.

3. INTEGRAL SECURITY METRICS

The graded security model uses coarse-grained metrics differentiating three or four security levels for each security goal. To compare security situations in general, one needs a more precise metric that expresses the quality of a security situation by one numeric value. It is reasonable to take into account influences of all security measures on the overall security of the system. The simplest choice would be to calculate the mean security confidence of all groups. However, the influence of groups on the overall security is different. Therefore, the best solution would be to use partial derivatives of the security measure depending on the security confidences of the groups. These derivatives could be used as coefficients of the security confidences when calculating their mean value. Unfortunately, these derivatives are hard to determine. Instead of the derivatives, one can use empirically found weights of the security confidences.

We have introduced a security metric in [4] that evaluates a security situation on the basis of security confidences provided by the security measures groups. We describe the overall security of a system by means of an integrated security metric S that is a *weighted mean security confidence*, called also *integral security confidence*:

$$S = \sum_{i=1}^{n} a_i c_i\,,$$

where c_i is security confidence of the i-th security measures group, a_i is the weight of the i-th group, and

$$\sum_{i=1}^{n} a_i = 1\,.$$

Using a linear combination of security confidences of measures groups is reasonable as long as a security situation does not change too rapidly. (The gradient of the integral security confidence in the space of confidences of security measures groups can be estimated in such a case and its components used as the required coefficients.)

4. VISUALIZING A SECURITY SITUATION

In this section we very briefly present a tool for making calculations on graded security models. This is a software package with a visual language for specifying security situations and problems. The package has been developed on the basis of the visual software development environment CoCoViLa [3], and it has been described in more detail in [4] and [5]. The package includes expert knowledge for a particular class of security situations. This expert knowledge is usable only for demonstrating the method – it has been taken mainly from [9].

Fig. 1 shows a specification of a security planning problem. The toolbar has buttons for defining components that will constitute a specification. It includes two buttons for defining security measures groups: one for groups with standard values of parameters, and another for groups with parameters defined as inputs. It includes also buttons for defining a security class, for selecting an optimization method and for defining a graphical output. All these components are also visible on the scheme in Fig. 1. This scheme is a specification of a problem for finding a Pareto-optimal solution for a security class C2I1A1M2 and specific parameters given for two security measures groups: *User training* and *Encryption*. Each security measures group has a pop-up window. This window is shown for the *Encryption* group in Fig. 1.

We use this package for all calculations on the graded security model. The package is extended with new components for solving the legacy security problems described in the following sections, see Sections 5 and 6.

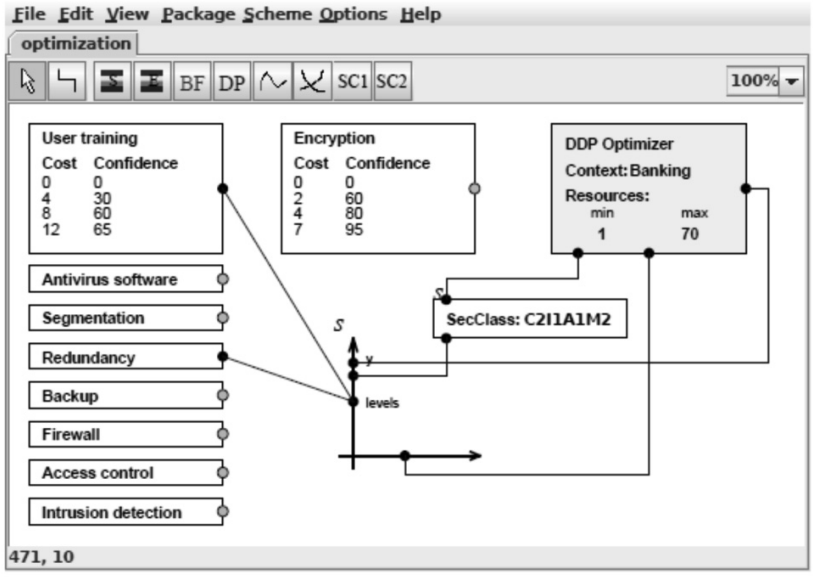


Figure 1. Visual specification of a security situation.

5. LEGACY SECURITY INFLUENCE

The widely used graded security model is based on the assumption that former investments into the security and already existing security situation do not influence the outcome of the investments planned. The former investments are sometimes included in the total amount of investments calculated. These investments may be included with a factor less than one, but this is still a rough approximation. We propose here an approach that more precisely takes into account the already achieved security.

Let us fix a security measures group and consider only one group of security measures here. Then we can use a simplified form of the functions h and e for calculating costs r and security confidence c – without showing explicitly the security measures group:

$$r = h(l),$$
$$c = e(l).$$

We use also a function for calculating security level l for invested costs, which is an inverse function of h:

$$l = h^{-1}(r).$$

We need data for already existing security:

l' – existing level of security,
c' – existing security confidence.

To continue analysis of security investments, we need a function H that calculates the needed additional investments r depending on the existing security level l' and the required security level l:

$$r = H(l, l').$$

It may seem that instead of the function H one can use a function h^* that calculates the required resources for increasing security level by Δl, where $\Delta l = l - l'$:

$$r = h^*(\Delta l).$$

It is easy to see that in the case when no investments in the security have been done before, i.e. when $l' = 0$, the function h^* coincides with the already known function h. However, in the case of $\Delta l = 0$ and $l' > 0$ we have to consider the degradation of security as well – the security level will decrease with time. This shows that the usage of h^* instead of H would be quite a rough approximation.

This analysis is valid for all security measures groups. But in the general model, we have to introduce an argument g (group number) in each function considered here. This gives us the functions:

$$r = H(l, l', g),$$
$$r = h^*(\Delta l, g).$$

These functions should be obtained from expert knowledge.

Another approach would be to use security confidence c instead of security level. These variables are bound by the function e in the graded security model:

$$c = e(l).$$

The relation between costs and security confidence is expressed by the formulas:

$$r = h(e^{-1}(c)), \text{ and}$$
$$c = e(h^{-1}(r)).$$

Knowing the already achieved security confidence, one can ask to calculate additional investments for achieving the new security confidence (or keeping the required confidence level). This requires the knowledge of a new function E that gives the costs r for achieving required security confidence c by upgrading the given security confidence c':

$$r = E(c, c').$$

As discussed above, one can sometimes assume that the costs depend only on the difference Δc of security confidences:

$$\Delta c = c - c',$$

and use the function e^* that calculates the costs:

$$r = e^*(\Delta c).$$

Again, in the general model we have to introduce an argument g (group number) in each function considered here. This gives us the functions for calculating costs in the general case:

$$r = E(c, c', g),$$
$$r = e^*(\Delta c, g).$$

Concluding the analysis here we can say that, for taking into account the legacy security measures in calculating resources required for achieving a given security confidence, we need one of the functions H, h^*, E or e^*. It is preferable to use H or E, because these describe the security situation more precisely. In practice, these functions are represented in a tabular form as expert knowledge. One would like to solve an inverse problem – calculate achievable security confidence for given resources. This is done by using one of the inverse functions H^{-1} or E^{-1} as H and E:

$$l = H^{-1}(r, l', g),$$
$$c = E^{-1}(r, c', g).$$

Let us call the functions H, h^*, E, e^*, H^{-1} and E^{-1} legacy functions.

The legacy values of l and r are bound by the functions h and h^{-1} as follows:

$$r' = h(l'), \text{ and}$$
$$l' = h^{-1}(r').$$

Therefore we can use legacy resources r' instead of l' as inputs of the calculations. We use this in an example in Section 6.

6. OPTIMIZING EVOLVING SECURITY

Security planning can be performed in two different ways. The traditional way is to decide somehow which security levels are required, and to calculate the required resources, using a function H or E. This is an application of the well-known *graded security method* [6]. The security levels are usually prescribed by some standards in this case.

Another way is to solve the *inverse problem*: for given resources find the best assignment of the resources to different security measures groups. This is an optimization problem that can be solved by means of discrete dynamic programming as shown in [5]. The quality of a solution is evaluated by the integral security metric S introduced in [4] and described in Section 3. Fig. 2a shows a solution of the inverse problem: the value of S for given resources r, and also selected security levels of security measures groups. The levels for the groups numbered from 1 to 9 are shown on the right side scale.

Besides the value of S, one may have to consider constraints put on the solution by the security class K, if it is given—all security goals prescribed by K must be satisfied. If priorities are assigned to the security goals, then it is possible to solve a more general problem: find the best possible security solution that satisfies the goal with the highest priority and, if possible, then satisfies also a goal with the next higher priority etc.

Our experiments have shown that the dynamic programming method is fast enough for solving even a more general problem: finding a Pareto-optimal set of security solutions for a given range of resources. Simply speaking, this means that the problem above must be solved for many values of resource r and the result must be plotted as a curve as shown in Fig. 2b.

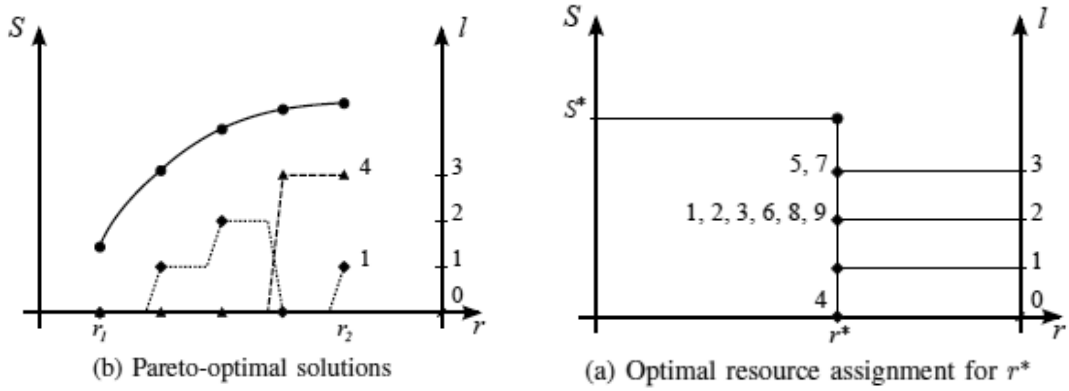


Figure 2. Solutions of the optimization problem of finding the best assignments of resources to different security measures groups.

Fig. 3 shows such a curve for resources from 1 to 70 units. It is obtained by using the expert system described in [4] for the problem specified in Fig. 1. We can see that the security class is C2I1A1M2 and that two security measures groups (*User training* and *Encryption*) get specific input values for the functions *h* and *e*. Other measures groups use the values from the built-in expert system.

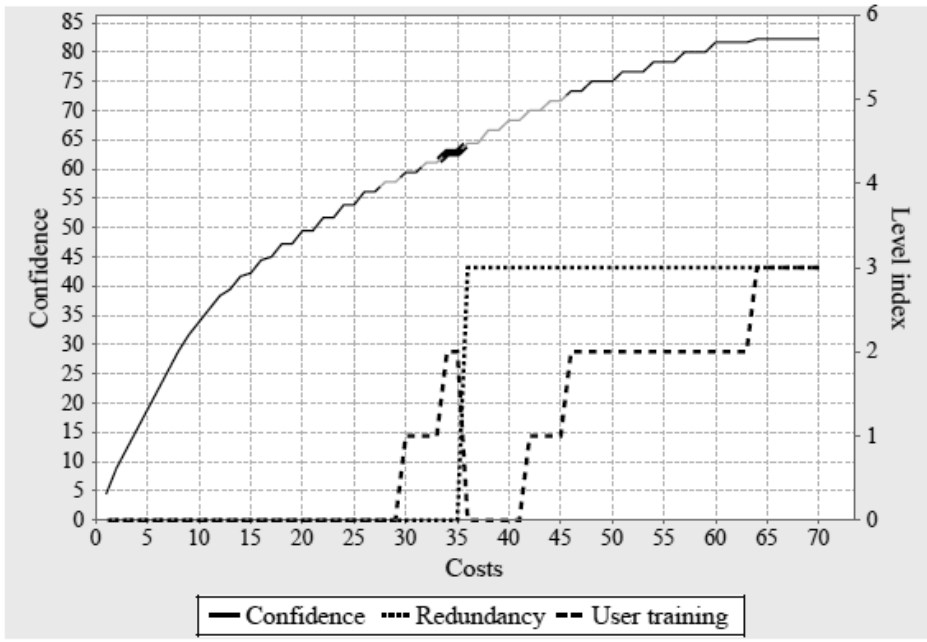


Figure 3. Solution of the problem.

In Fig. 3, the lower graphs indicate (on the scale shown on the right) the optimal levels of two measures groups (*Redundancy* and *User training*) corresponding to the given amount of resources. These graphs are not monotonic as can be seen in this example at the resource values 35 and 36. For a more detailed explanation see [5].

Let us consider now the inverse problem considering also the legacy security: given a security class K, resources r, existing security levels l' and a legacy function H^{-1}, find the security solution with the highest value of mean weighted security confidence S that satisfies all security goals of K. This problem may or may not have a solution. Even if it does not have a solution, the problem without the constraint K (without the requirements on security goals) will have a solution. It is interesting to notice that, in the case when the problem has a solution, this solution may be different from the solution obtained without the constraint K.

Fig. 4 shows a solution for both cases: the red curve presents a solution for the problem with a constraint K = C3I1A1M2, and the green curve presents a solution for the unrestricted problem. We can see the cases where prescribing K gives worse values of S.

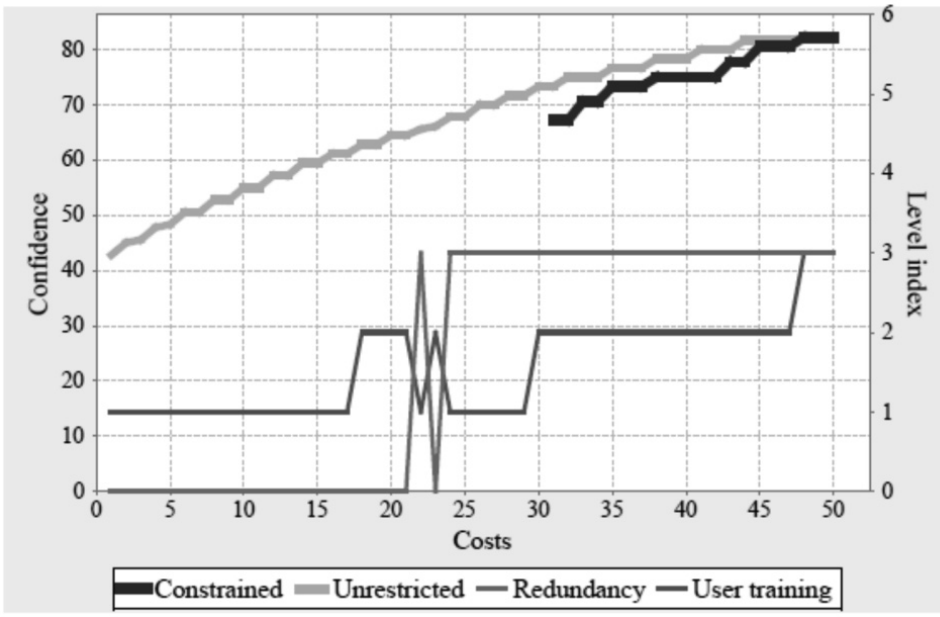


Figure 4. Solutions with and without a constraint.

For solving the legacy problems we have extended the expert system by adding the legacy information to the components representing security measures groups, and adding the calculation of the legacy function

$$c = e(h^{-1}(r0)),$$

where $r_0 = r + (1 - q)r'$ is an effective resource that takes into account both current resource r and decayed value of the legacy resource r'; q is a decay of a resource, $q < 1$. We have used the values of legacy resource r' and decay q given in Table 1.

Table 1. Values of legacy resource and decay.

g	r'	q
User training	4	0.3
Antivirus software Segmentation	4	0.6
Redundancy	4	0.2
Backup	0	0.3
Firewall	4	1.0
Access control	4	0.5
Intrusion detection Encryption	4	0.2

Knowing the legacy function, we can plan optimal security measures for a number of time intervals (years) in advance. The values l' of existing security levels must be given as initial data. The values of l' for each following year must be taken equal to the values of l of the previous year. The Pareto-optimal set is a surface in a multidimensional space with coordinates $r, y, l_1, \ldots, l_n$ and S, where y is the year number in this case.

Even if we consider Pareto-optimal solutions only for one year, visualization of the Pareto-optimal set is possible only in a special case when all security levels of all security measures groups are equal. In this case, the Pareto-optimal set is a surface in the three-dimensional space r, S, l, where l is the confidence level of all measures groups, and this can be visualized.

7. CONCLUDING REMARKS

The software developed in the present work for analyzing security situations is easy to use for security experts. The developed experimental tool has a simple graphical interface and a visualization component that supports its usage by security managers of all levels. The experiments have also shown that stability of optimal solutions found by the presented method is good. However, the practical applicability of the software will depend on the availability of good expert data representing the legacy function as well as functional dependencies of the graded security model. The developed software has been designed as an expert

system. It supports easy inclusion of new expert knowledge, but expert knowledge acquisition is always a complicated task. For specific military communications applications the data must be collected from a log analysis of characteristic attacks and security reports, as well as by the traditional knowledge acquisition means.

Finally, the contemporary security landscape is dynamic and rapidly changing. This is the main reason for developing agile methods of security situation management. The presented method of managing evolving security situations is one of these.

ACKNOWLEDGMENTS

We thank the Cooperative Cyber Defence Centre of Excellence, the Estonian Defence Forces Training and Development Centre of Communication and Information Systems, and the Estonian Ministry of Defence for the support of this work.

REFERENCES

[1] Estonian Informatics Centre. *Estonian Information Systems Three-Level Security Baseline System – ISKE version 4.01*. http://www.ria.ee/27220 (10 Apr 2009).

[2] L. A. Gordon, M. P. Loeb. *Managing Cybersecurity Resources: A Cost-Benefit Analysis*. McGraw-Hill, 2006.

[3] P. Grigorenko, A. Saabas, E. Tyugu. *Visual tool for generative programming*. ACM SIGSOFT Software Engineering Notes, 2005, 30, 5, 249–252.

[4] J. Kivimaa, A. Ojamaa, E. Tyugu. *Graded security expert system*. CRITIS 2008: Third International Workshop on Critical Information Infrastructure Security, Rome, October 13–15 2008. Springer, LNCS, 2009.

[5] A. Ojamaa, E. Tyugu, J. Kivimaa. *Pareto-optimal situation analysis for selection of security measures*. MILCOM 08: Assuring Mission Success: Unclassified Proceedings, San Diego, November 17–19 2008, 7p.

[6] C. E. Pasterczyk. *A graded approach to ISO 9000 implementation for records managers*. Association of Records Managers and Administrators international annual conference, Toronto (Canada), 25–29 September 1994.

[7] *The Common Criteria*. http://www.commoncriteriaportal.org/ (10 Apr 2009)

[8] U.S. Department of Defense. *National Industrial Security Program Operating Manual (NISPOM)*. 2006.

[9] U.S. Department of Defense, Defense Information Systems Agency. *CyberProtect, version 1.1*. July 1999. http://iase.disa.mil/eta/product description.pdf (10 Apr 2009).

STUDY V

OPTIMIZING IT SECURITY COSTS BY EVOLUTIONARY ALGORITHMS

Kirt, Toomas; Kivimaa, Jyri

Kirt, Toomas; Kivimaa, Jyri (2010).
Optimizing IT security costs by evolutionary algorithms. CyCon 2010.

Conference on Cyber Conflict Proceedings 2010: Conference on Cyber Conflict; Tallinn, Estonia; June 15-18, 2010. Tallinn: Cooperative Cyber Defence Centre of Excellence Publications, 2010, 145 – 160.
ISBN: 9789949904013
Classification: 3.4

Toomas Kirt, University of Tartu, Estonia,
Jüri Kivimaa, CCD COE, Tallinn, Estonia

Abstract

One of the most critical issues in IT security is to establish a cost-effective framework for cyber protection against possible threats. The overall security framework is divided into security activity areas, which can have a number of protection levels. Each level of one security activity area provides certain confidence and also requires some expenditure. As the budget level is predefined a critical question remains how to find out an adequate security profile for a certain cost level. As the behavior of cyber attackers and cyber security threats are continuously changing therefore there should not be just one model to construct an effective security mechanism but rather a variety of changing alternatives. Several methods have been proposed for cost optimization but they are limited by providing only one alternative. In this paper we propose an evolutionary approach as an alternative for optimizing IT security costs and for finding variants of security profiles for every cost level. Higher variability of security profiles will make the security organization more resistant to changing cyber attacks.

Keywords: graded security model, information security metrics, information security requirements, evolutionary computing, genetic algorithms

Introduction

We have the challenge of ensuring information security under conditions of uncertainty, how can organizations determine appropriate measures to enhance cyber security and allocate resources most efficiently? For finding out an optimal amount of resources a security costs function is proposed, where the total cost of security for a system is based on the cost of system security investments plus the cost of damage and cost of recovery from any security incidents (Olovsson, 1992). Despite the cost function includes also indirect cost in this study we take into account only direct costs of security investments. Usually, available resources are limited and therefore it is needed to optimize applied security measures to achieve the highest attainable confidence level. The security framework is divided into several security activity areas that can have a number of levels providing certain confidence. As the number of security activity areas increases the number of different combinations of security measures or profiles grows exponentially. For finding an optimal security profile several optimization methods are used like a brute force optimizer and a discrete dynamic programming method (Kivimaa, 2009; Ojamaa, Tyugu, & Kivimaa, 2008).

It is argued that the dynamic programming may have some problems related to independency of security activity areas and additivity and therefore the solutions may not to be optimal (Kivimaa, 2009). The restriction also limits the search space and it is difficult to find out alternative security profiles that provide the same level of confidence. Therefore our aim is to apply an additional method to find out whether the solutions are adequate and also identify alternative security profiles for a certain cost level. We decided to use an evolutionary algorithm as a universal method for complex optimization in many fields. Genetic algorithms are also actively used in IT security and intrusion detection systems (e.g., Li, 2004; Sinclair, Pierce, & Matzner, 1999).

Evolutionary algorithms are based on a Darwinian natural selection process and form a class of population-based stochastic search algorithms (Dracopoulos, 2008; Eiben & Smith, 2003; Holland, 1975). In the evolutionary process for all the individuals representing candidate solutions some perturbations (e.g., crossover, mutations) are applied to generate variation and thereafter a selection procedure, based on the value of a fitness function, is enforced. The selection mechanism prefers individuals that are the best candidates for the solution of the optimization problem. To maintain the variation in population in our experiments the population was divided into subgroups and the selection process was performed within a group. This measure helped to avoid the optimization process to fall into a local optimum and provided better results. To solve the optimization task we have established an evolutionary framework and applied it to the IT security cost/confidence data consisting of 9 security areas (CyberProtect, see Table 1). In the following optimization tasks we had two goals: to minimize the costs and to maximize the integral security confidence.

This paper is divided into four main parts. In the first part the security model and the data is described that we use in our optimization tasks. Next we introduce the basis of evolutionary algorithm. Thereafter the results of optimization are given. Finally the results are discussed and conclusions are made.

Security Model

The main challenge in IT security is to ensure required information security under conditions of uncertainty. To achieve the goal an organization has to define adequate security levels and to determine appropriate measures for increasing cyber security and allocating resources most efficiently. Usually certain risk assessment methods are used for performing detailed risk analysis. For small and medium size enterprises the detailed risk analysis is relatively expensive and also the available resources for IT security are limited. Therefore a simpler version of the security model is needed which provides possibility to achieve maximum possible confidence with limited resources.

In this research we rely on the graded security model, which is an improved and combined version of two security methodologies: the US DoE graded security methodology (best practice security methodology to specify needed security measures for needed security levels; DOE, 1999) and Estonian governmental data classification (metrics to specify needed security level; ISKE, 2009). The system includes knowledge modules (rule sets) in the form of decision tables for handling expert knowledge of costs and confidence, as well as for selecting security measures for each security group depending on the required security level. Basic ideas of graded security are presented as a decision table – information security activities areas/their realization levels and information security requirements/their levels in a dependency matrix. As an example a very simple (9 security subareas) decision table/dependency matrix is given in Appendix.

Tabel 1. IT security costs/confidence data. 9 security measures

Security measure \ level		Level 0	Level 1	Level 2	Level 3
1. User Training	Cost	0	4	8	12
	Confidence	0	30	50	65
2. Redundant Systems	Cost	0	8	10	12
	Confidence	0	40	70	95
3. Access Control	Cost	0	1	2	4
	Confidence	0	40	70	95
4. Antivirus	Cost	0	2	4	7
	Confidence	0	60	80	95
5. Backup	Cost	0	1	2	4
	Confidence	0	40	70	95
6. Disconnection	Cost	0	2	4	7
	Confidence	0	40	60	75
7. Encryption	Cost	0	2	4	7
	Confidence	0	60	80	95
8. Firewall	Cost	0	2	4	7
	Confidence	0	30	50	65
9. Intrusion Detection	Cost	0	1	2	4
	Confidence	0	25	45	60

The example used in the experiments of this paper is an educational security framework CyberProtect version 1.1 (CyberProtect, Table 1). It determines how hardware/software/firmware can be secured based on nine security activity/measure groups and their high/middle/low level realization of costs and confidence. The cost in this example covers only the costs of security investments and is given in conventional units. The confidence level is in the scale of 0…100 and the value is provided as an expert opinion. Each security measure can have a certain level which determines required resources to achieve confidence. The baseline security methodologies define conventional goals of security as confidentiality (C),

integrity (I), availability (A), and mission criticality (M). For each goal a finite number of security levels have been determined. For example, four levels 0, 1, 2, 3 for representing required security and protection can be used, where the lowest level 0 denotes unnecessary of special protective measures.

We can formulate an optimization problem as follows: find the abstract security profile with the best (highest) value of confidence for given amount of resources. As we have a limited amount of available resources r our goal is to achieve a maximal security level

$$S_{\max} = \sum_{i=1}^{n} a_i q_{\max i}$$

where $q_{\max i}$ is maximal security confidence of the i-th group of security activity areas and a_i is the weight of the i-th group

$$\sum_{i=1}^{n} a_i = 1$$

We have an optimization problem with two goals: to minimize resources on the interval $[r_{\min}; r_{\max}]$ and to maximize security, guaranteeing at least the levels prescribed by a given security class. We are going to solve this problem by finding a function that gives an abstract security profile that has maximal value of a security confidence function given by the weighted mean security for any given value of resources on the interval $[r_{\min}; r_{\max}]$. The task of the optimization application is to find the best combination of security activities levels which provides the maximum confidence at a cost level.

In previous experiments mainly two optimization algorithms were used to solve our task – one of them was a brute force optimizer and the other one was based on a Pareto optimality (Pareto frontier or Pareto set) and discrete dynamic programming method (Ojamaa, et al., 2009). This problem can be solved by means of building a Pareto optimality trade-off curve that explicitly shows the relation between used resources and security confidence. Then, knowing the available resources, one can find the best possible security level that can be achieved with the resources and specify the security measures to be taken.

For n security measures groups and k levels for information security requirements/ goals we have totally k^n abstract security profiles to be considered. The number of security measures groups may be in practice up to 30 or even more and in Estonian data classification 4-levels version for security goals is used. This gives a number of abstract security profiles: 4^{30}.

With the brute force method we must do rk^n computations and with the dynamic programming method r^2kn (r is number of possible values of resources, k is the number of security levels, n is number of security measures groups). For example,

if we have a 100 budget points curve for 25 security subareas then it takes ~10 seconds to calculate it with the Pareto optimality & dynamic programming and by Brute Force method it would take ~10 years to calculate (Kivimaa, 2009).

To use Pareto optimality and dynamic programming in optimization security activities areas/security measures groups must be not dependent from each other's and their security measures to realize their levels must be additive. Independency in IT security activities is quite problematic for some security areas, but in first approximation it is acceptable if we use certain specific logic of description (for example, the IT security experts/specialists training costs are included into the costs of concrete security activities areas/areas levels and some other analogical principles might be followed).

The second weakness of dynamic programming is that it has some difficulties in finding alternative security profiles for a certain optimal cost/confidence level. To get over of those weaknesses and to measure adequacy of the dynamic programming we decided to use evolutionary algorithm as an alternative method. We expect that the evolutionary approach is not stuck to such limitations and can provide results with a quite reasonable time.

Evolutionary Algorithms

An evolutionary algorithm is a population-based stochastic search algorithm. The basic principle is to iteratively generate random variation within individuals of population, that represents the candidate solution to the problem, and to select the fittest candidates that provide the best solution to the task in hand. The view, that random variation provides the mechanism for discovering new solutions (Michalewicz & Fogel, 2004), was inspired by the process of natural evolution. The idea of using Darwinian principles of evolution to solve some combinatorial optimization problems arose with the invention of electronic computers. Afterwards several approaches were developed like evolutionary programming (Fogel, Owens, & Walsh, 1966) and genetic algorithms (Holland, 1975) in the early stage of the study of evolutionary algorithms. Now there are a wide variety of approaches that can be described as belonging to the field of evolutionary computing. The algorithms used in the field are termed as evolutionary algorithms (Dracopoulos, 2008). The most important characteristics of evolutionary algorithms are as follows:

- *Representation*. Each candidate solution to the problem in hand is represented as an individual. The characteristics of the individual are encoded by genes. The set of individuals form a population.
- *Fitness*. The quality of a candidate solution is measured by a fitness function. The fitness function is used to measure how good an individual is. Fitter solutions have a higher probability to survive and to contribute their characteristics to offspring.

- *Variation*. Variation operators (e.g., crossover, mutations) are applied to the individuals that modify the population of solutions dynamically.
- *Selection*. The average fitness is improved over time as a selection mechanism is applied and the fittest individuals are selected for the next generation (survival of the fittest).

The basis of an evolutionary algorithm is simple. First, a population of initial candidate solutions has to be generated randomly. Thereafter iteratively a number of variation generation operators are applied and new generations are selected based on the fitness values of individuals.

Algorithm

There are proposed several modifications to the basic algorithm and we have adapted some aspects of cooperative co-evolutionary algorithms (see Machado, Tavares, Pereira, & Costa, 2002; Potter & De Jong, 2000). In this approach the problem is decomposed into subcomponents that represent potential components to the global problem (see more details in Selection). As the problem in hand was not very complex we decided to decompose a population P into S subpopulations Ps instead of decomposing a problem. The aim was to maintain variety within the population as a whole.
The algorithm can be defined then as follows:
– for each subpopulation S do:
– Initialize population Ps(0)
– Evaluate all individuals from Ps(0)
– While termination condition not met repeat:
– For each subpopulation S do:
– Apply crossover and mutation operators to individuals of Ps(t) and obtaining a set of offspring Os(t)
– Evaluate individuals from Os(t)
– Combine Ps(t) and Os(t) obtaining Ps(t+1)
– During the evaluation the fitness value (average confidence level) of an individual is found. The fittest from the ordered set of parents and offspring are selected for the next generation.

Representation

How to choose a suitable genetic representation of an individual is a key issue in evolutionary computing. Each individual has two representations: phenotype (outside) and genotype (inside). Object forming possible solutions within the original problem context are referred as phenotypes, while their encoding, that is, the individuals within the evolutionary algorithm, are called genotypes (Eiben & Smith 2003). Phenotypic characteristics of the candidate solution are encoded

by individual's genotype. The genes are the functional units to carry inherited information and they can be arranged in chromosomes. In evolutionary algorithm a chromosome can be a string of symbols or a vector of numerical variables (Gen & Lin, 2008). The complete inherited information is called genome.

Genotype contains inherited information to build an individual in phenotype space. In the natural systems the mapping from genotype to phenotype is not direct. In the context of evolutionary algorithms three classes of possible mappings are defined: direct, developmental and implicit (Floreano, Dürr, & Mattiussi, 2008). In a direct representation, there is a one-to-one mapping between the parameter values of the task in hand and the genes that compose the genetic string. In developmental representations which are used mostly in case of large problems the specification of a developmental process is genetically encoded which in turn constructs the desired phenotype. In case of implicit encoding like in biological gene networks, the interaction between the genes is not explicitly encoded in the genome, but follows implicitly from the physical and chemical environment in which the genome is immersed.

In this paper direct mapping is used and each candidate solution is represented as a chromosome consisting of the same amount of genes as the number of security activity areas. Each gene denotes a security level of one security activity area. For example, if there are 3 security levels plus one for the lowest level 0 denoting absence of special protective measures four possible values for one gene (0, 1, 2, 3) can be defined. If there are 9 security activity areas then a chromosome can be G = {1 0 3 2 3 1 2 1 3}.

Fitness

The goal of the evolutionary search is defined as a user-specified measure of the quality or the fitness of the individuals. The algorithm is expected to find in the search space an individual with maximum quality or fitness. In our experiments the fitness is measured as a weighted average of confidence levels of security activity areas.

Variation

The initial population is usually generated by random and therefore it is highly variable. The movement in the search space is based on random changes in chromosomes generated by reproduction and applying several variation operators. The reproduction is carried out with some stochastic mutation and recombination of the parents in order to explore new regions the search space and combine the information carried by each parent (Gen & Lin, 2008).

The main operator to generate variation in population is the crossover. There are introduced several approaches to select parents and to recombine their genetic information. Recombination, the process whereby a new individual is created from the information contained within two parents, is considered to be one of the most important feature in evolutionary algorithms. In the experiments we use the crossover operator called n-point crossover, where the value of n is 2. The basic steps of applying crossover operator are as follows: first, to select two parents based on some restrictions (if there are) and next, select segments of genes from both parents to form the genes of an offspring. The second parent is selected by random from the whole population. An example is illustrated in Figure 1. A segment {4, 3} is taken from one parent and is transferred to the other parent's genetic code.

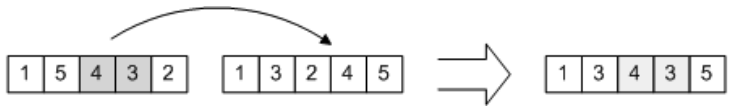


Figure 1. n-point crossover. n = 2.
Several variation operators are used to make variation in population and to move in the search space.

Random mutation is the change of the value of one gene. For example, the value of the first gene {1} is replaced by the new value {3}.

Figure 2. Random mutation of a single gene
Swap operator: selects two genes and swaps them. For example, genes {5} and {3} are selected and swapped.

Figure 3. Swap mutation
Inversion operator: selects a segment of genetic code and reverses order of the genes belonging to it. For example, genes {1 5} are reversed {5 1}.

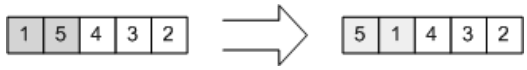


Figure 4. Inversion mutation
Insertion operator: selects a gene and inserts it in another place. For example, gene {1} is moved to the end of the genetic code.

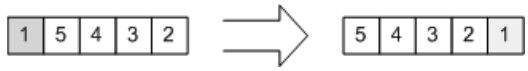


Figure 5. Insertion mutation
Displacement operator: selects a segment of genetic code and inserts it in another place. For example, genes {1 5} are moved to the end of the genetic code.

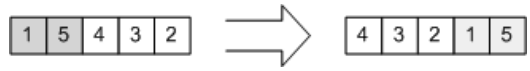
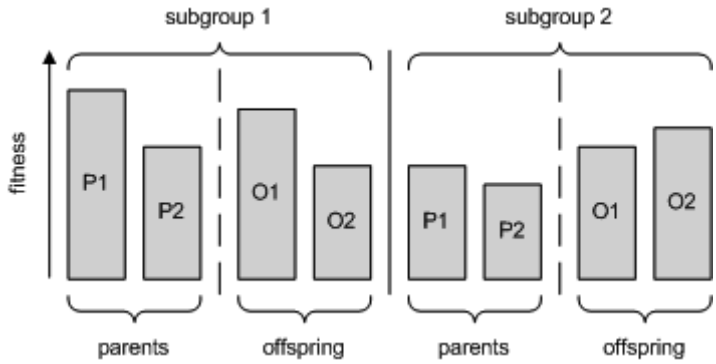


Figure 6. Displacement mutation

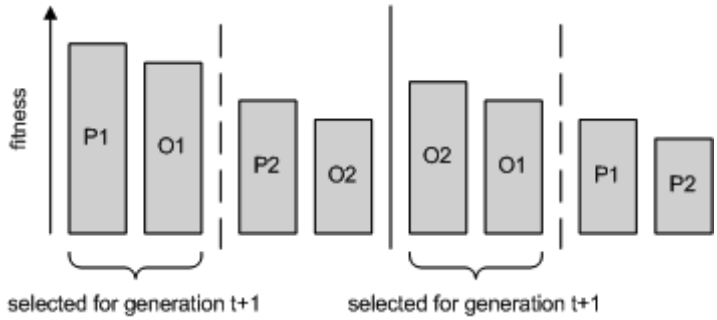
When mutation operators are applied, the genes are validated whether they are in accordance to the restrictions of the task in hand. When the code does not meet the restrictions it is not used in the further processing.

Selection

The selection is a process to select survivals for the next generation. During each generation, the chromosomes are evaluated, using some measures of fitness. A new generation is formed by selecting some parents and offspring, according to their fitness values, and rejecting others to keep the population size constant.



a) After reproduction and mutation a new sets of individuals (offspring) are formed in each subpopulation



b) For selection the parents and offspring within a subgrouop are ordered based on the fitness value and the fittest are selected for the next generation

Figure 7. An example of a tournament selection of 2 sub-population consisting of 2 individuals. 4 candidates (2 parents P and 2 offspring O) are competing for selection for next generation within a sub-population

In this study the selection method is based on the tournament selection strategy that is deterministic. The tournament selection is effective, because it does not require any global knowledge of the population and it also avoids falling into a local optimum by maintaining variety in the population. This strategy also enhances the search space and allows exploring it parallel. To perform tournament selection we have to define the tournament size k. The members of a tournament are usually selected by random, but we use a deterministic strategy where the competing sub-populations are predefined. For example, the tournament or subpopulation size is defined as 2 (Figure 7). After reproduction and mutation phase (Figure 7a) 4 candidates (2 parents and 2 offspring) compete for being selected for the next generation (Figure 7b). The selection is performed locally and therefore the winning members of one tournament may have weaker fit value than the least-fit members of the other tournament. Further mutations in such weak subpopulation may reveal some properties of an individual that are needed to reach global optimum and are not represented in other subgroups.

Experiments

For experiments we had the IT security cost/confidence data consisting of 9 security activity areas (CyberProtect; see Table 1). The aim of the optimization was to find highest average confidence level for a given amount of resources. The optimization task is formed as a question (Kivimaa, 2009): "For every possible budget level, what is the maximum confidence one can expect?" In the optimization tasks the amount of resources (budget) was predefined form 1 to max+1. The max value equals the costs of the security measures of the highest level. The first task was to measure the mean computational time to solve the optimization problem. The second task was to find the cost/confidence optimality curve. The third task was to find out the cost/confidence optimality curve when the optimality was restricted by a security class. The fourth task was to identify adequate and equivalent security profiles for every cost level.

For the results presented in this section we used the following experimental settings: crossover rate 0.49, mutation rate 0.2, swap rate 0.1, inversion rate 0.1, insertion rate 0.1, and displacement rate 0.1. The number of generations was set as 30 and population size 80, and the tournament or subpopulation size was 5. The cost of the highest security level (C3I3A3M3) was 64 units and the optimization was performed for the cost levels from 1 to 65 units. With each cost level 5 experiments were performed. The rates for crossover and mutation operators were selected as the best practice of solving other optimization problems. Despite the optimization tasks are similar the rates might not to be the best for solving the security optimization task. Additional computation time is required either the variation rate is very low or high, as there are needed to perform unnecessary calculations.

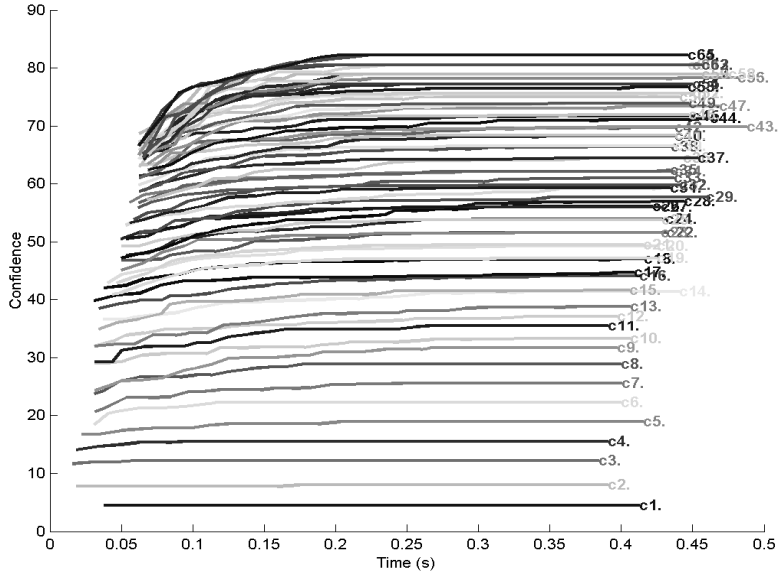


Figure 8. Mean computational time to find optimal confidence value for 9 security areas (mean value of 5 experiments)

As a result the average time for optimization was between 0.4 and 0.45 seconds (Figure 8). The task two was to find the cost/confidence optimality curve (yellow dots in Figure 9). For interpretation a color coding of dots in the curve is used as follows: red dots – all security activities area's security levels are ≤ and at least one is < than required; green dots – all security goals/their required levels are exactly achieved; yellow dots – at least one security level is less and at least one security level is more than required; blue dots – all security levels are ≥ and at least one security level is > than required. The curve represents the optimal value of weighted mean security confidence depending on the resources that are used.

Table 2. The experimental dependency matrix of 9 security measures

Security measure	C0	C1	C2	C3	I0	I1	I2	I3	A0	A1	A2	A3	M0	M1	M2	M3
1. User Training	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
1.Redundant Systems	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
3. Access Control	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
4. Antivirus	1	1	2	3	1	1	2	3	1	1	2	3	1	1	2	3
5. Backup	0	1	2	3	0	1	2	3	0	1	2	3	0	1	3	3
6. Disconnection	1	1	2	3	1	1	2	3	1	1	2	3	1	2	3	3
7. Encryption	0	0	1	3	0	1	2	3	0	0	0	0	0	0	2	3
8. Firewall	0	1	2	3	0	1	2	3	0	1	2	3	0	2	3	3
9. Intrusion Detection	0	1	2	3	0	1	2	3	0	1	2	3	0	2	2	3

Next experiments were performed when the limitation of security class was applied. In this study an experimental dependency matrix of connections between security measures and conventional goals of security was used (Table 2). For example, as the security class is defined C1I1A1M1 then the highest level of a security measure is selected and the security configuration is (1, 1, 1, 1, 1, 2, 1, 2, 2). The comparison of confidence values between the security classes C3I3A3M3 and C1I1A1M1 is given in Figure 9. As there are more available resources than are needed to satisfy the restrictions caused by a security class the security measures cannot be weaker than determined by the security class.

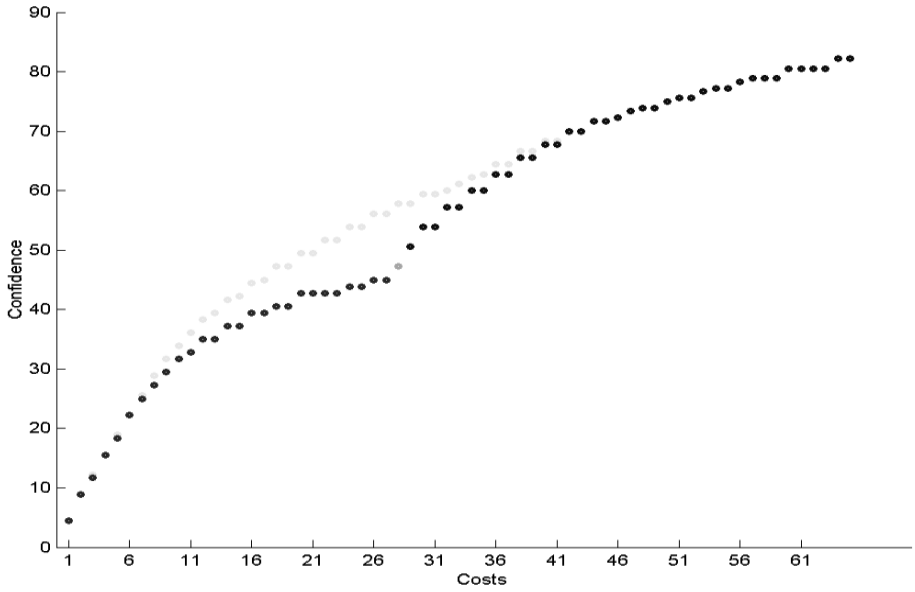


Figure 9. Costs/confidence optimality curve using security-class limitation. Security class C3I3A3M3 versus C1I1A1M1.
Optimal security configuration (1, 1, 1, 1, 1, 2, 1, 2, 2)

The final task was to obtain different security profiles. To find out different security profiles we ran experiments 35 times for every cost level. An extract of the results is given in Table 3. For example, when 34 unit of money was available (budget restriction) then 5 equivalent security profiles were found.

Table 3. Equivalent security profiles for every cost/confidence level in case of 9 security measures.

				Security measure								
No	Money	Costs	Confidence	1	2	3	4	5	6	7	8	9
…..												
88	34	34	62,22	1	4	4	2	4	2	3	3	3
89	34	34	62,22	1	4	4	3	4	2	2	3	3
90	34	34	62,22	1	4	4	3	4	3	2	2	3
91	34	34	62,22	1	4	4	2	4	3	3	2	3
92	34	34	62,22	1	4	4	2	4	3	2	3	3
93	35	35	62,78	2	1	4	3	4	4	3	3	4
94	35	35	62,78	2	1	4	3	4	3	4	3	4
95	35	35	62,78	2	1	4	3	4	3	3	4	4

				Security measure								
No	Money	Costs	Confidence	1	2	3	4	5	6	7	8	9
96	35	35	62,78	2	1	4	4	4	3	3	3	4
97	36	36	64,44	1	4	4	3	4	3	3	2	3
98	36	36	64,44	1	4	4	2	4	3	3	3	3
99	36	36	64,44	1	4	4	3	4	2	3	3	3
100	36	36	64,44	1	4	4	3	4	3	2	3	3
…..												

Conclusions

The aim of the study was to evaluate whether the evolutionary approach is applicable to security the cost/confidence optimization task and whether it allows us to generate equivalent security profiles for every cost level. As a result we could conclude that the evolutionary approach is viable for such task. The results indicated that the evolutionary algorithm was fast enough to provide results and turned out to be more flexible than the discrete dynamic programming method. The evolutionary approach provided results within a reasonable time limit and the cost/confidence optimization of 9 security activity areas took 0.4-0.45 seconds (Figure 7).-The main advantage of the evolutionary algorithm was that it provided several adequate and equivalent security profiles for every cost level with reasonable time (see Table 3). As it is noted, there should not be just one model to construct an effective security mechanism but several simple security mechanisms that are attuned to the needs of differing applications and organizations (Wulf & Jones, 2009). Thereby the evolutionary approach might help us to provide a better confidence level.

References

CyberProtect, version 1.1. U. S. Department of Defense, Defense Information Systems Agency. Retrieved from http://iase.disa.mil/eta/.

DOE (1999). Classified Information Systems Security Manual. Retrieved February 1, 2010, from https://www.directives.doe.gov/directives/archive-directives/471.2-DManual-2/at_download/file.

Dracopoulos, D. C. (2008). Evolutionary Learning. In B. Wah (Ed.), Wiley Encyclopedia of Computer Science and Engineering. New York: John Wiley & Sons.

Eiben, A. E., & Smith, J. E. (2003). Introduction to Evolutionary Computing. Berlin: Springer.

Floreano, D., Dürr, P., & Mattiussi, C. (2008). Neuroevolution: from architectures to learning. Evolutionary Intelligence, 1(1), 47–62.

Fogel, L. J., Owens, A. J., & Walsh, M. J. (1966). Artificial Intelligence Through Simulated Evolution, John Wiley & Sons: New York.

Gen, M., & Lin, L. (2008). Genetic Algorithms. In B. Wah (Ed.), Wiley Encyclopedia of Computer Science and Engineering. New York: John Wiley & Sons.

Holland, J. H. (1975). Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. Cambridge, MA: MIT Press.

ISKE, (2009). ISKE - three-level IT baseline protection system (Version 5.00). Retrieved February 1, 2010, from http://www.ria.ee/public/ISKE/iske_rakendusjuhend_5_00.pdf.

Kivimaa, J. (2009). Applying a costs optimizing model for IT security. In H. Santos (Ed.), Proceedings of the 8th European Conference on Information Warfare and Security (pp. 142–153). Reading, UK: Academic Publishing Limited.

Li, W. (2004). Using Genetic Algorithm for network intrusion detection. In Proceedings of United States Department of Energy Cyber Security Group 2004 Training Conference (pp. 1-8). Kansas City, Kansas.

Machado, P., Tavares, J., Pereira, F. B., & Costa, E. (2002). Vehicle Routing Problem: Doing it the Evolutionary Way, In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002) (p. 690). New York, USA.

Michalewicz, Z., & Fogel, D. B. (2004). How To Solve It: Modern Heuristics. Berlin: Springer.

Ojamaa, A., Tyugu, E., & Kivimaa, J. (2008). Pareto-optimal situation analysis for selection of security measures. In Military Communications Conference: Unclassified Proceedings (pp. 3224–3230). Piscataway, NJ: IEEE.

Olovsson, T. (1992). A Structured Approach to Computer Security. Technical Report No. 122. Göteborg, Sweden: Chalmers University of Technology.

Potter, M. A., & De Jong, K. (2000). Cooperative Coevolution: An Architecture for Evolving Coadapted Subcomponents. Evolutionary Computation, 8(1), 1–29.

Sinclair, C., Pierce, L., & Matzner, S. (1999). An application of machine learning to network intrusion detection. In Proceedings of the 15th Annual Computer Security Applications Conference (pp. 371-377). Phoenix, AZ.

Wulf, W. A., & Jones, A. K. (2009). Reflections on Cybersecurity. Science, 326, 943–944.

Appendix

The dependency matrix of 9 security measures

No	Information Security Goals / Information Security activities areas	Confidentiality Requirements				Integrity Requirements				Availability Requirements				Mission Criticality				Final levels
		C0	C1	C2	C3	I0	I1	I2	I3	A0	A1	A2	A3	R0	R1	R2	R3	
		Public Data	Data for Internal Use	Confidential Data	Highly Confidential Data	Protection/Detection of Changes is Not Important	Protection/Detection of Unauthorized Changes	Inputter/Changer must be detectable	Inputter/Changer Must be Provable (in court)	Delay of Data Will Not Cause Problems	Availability 90% - allowed ~ one day delay in week	Availability 99% - allowed ~ one hour delay in week	Availability 99,9% allowed one ~10 minutes delay in week	Data Delay Will Not Cause Significant Consequences	Data Delay Causes Damages in Hundreds of Thousands	Data Delay Causes Damages in Millions of Kroons	Data Delay Causes Hundreds of Millions of Damages	
1.	**Access Control**	AC-0	AC-1	AC-2	AC-3	AC-0	AC-1	AC-2	AC-3								AC-4	
2.	**User Training**	UT-0	UT-1	UT-2	UT-3													
3.	**Disconnection** (Data Communications)	DC-0	DC-1	DC-2	DC-3					DC-0	DC-1	DC-2	DC-3				DC-4	
4.	**Encryption**	CR-0	CR-1	CR-2	CR-3	CR-0	CR-1	CR-2	CR-3									
5.	**Intrusion Detection** (Monitoring)	ID-0	ID-1	ID-2	ID-3													
6.	**FireWall** (Perimeter Protection)	FW-0	FW-1	FW-2	FW-3	FW-0	FW-1	FW-2	FW-3									
7.	**Antivirus**	AV-0	AV-1	AV-2	AV-3	AV-0	AV-1	AV-2	AV-3									
8.	**Backup and Recovery**					BR-0	BR-1	BR-2	BR-3	BR-0	BR-1	BR-2	BR-3				BR-4	
9.	**Redundancy** (IT Recovery)									R-0	R-1	R-2	R-3				R-4	

STUDY VI

EVOLUTIONARY ALGORITHMS FOR OPTIMAL SELECTION OF SECURITY MEASURES

Kivimaa, Jyri; Kirt, Toomas

Kivimaa, Jyri; Kirt, Toomas (2011).
Evolutionary Algorithms for Optimal Selection of Security Measures.
10th ECIW 2011.

Proceedings of the 10th European Conference on Information Warfare and Security: Tallinn, Estonia, 7-8 July 2011. Reading, UK: Academic Conferences Limited, 2011, 172-184.
ISBN: 9781908272072
Classification: 3.1

Jüri Kivimaa [1], Toomas Kirt [2]

[1] Cooperative Cyber Defence Centre of Excellence, Tallinn, Estonia, jyri.kivimaa@mil.ee

[2] University of Tartu, Tallinn, Estonia, Toomas.Kirt@ut.ee

Abstract

A very important issue in IT Security or Cyber Security management is to provide cost-efficient security measures to achieve needed or required security goals (mainly CIA - Confidentiality, Integrity, Availability levels). For providing an optimal solution an optimization task with two goals have to be solved – to minimize needed resources and to maximize achievable security. The computational complexity of the optimization task is very high.

In previous work a matrix based security model and an optimization framework based on the Pareto optimality and the discrete dynamic programming method has been used. But that solution has a quite important imperfection – there was required independence between security activity areas. That is not appropriate for IT security, as this solution does not follow the quite important principle in IT security – security is like a chain that is only as strong as the weakest link of layered security or defence in depth. The evolutionary optimization, as an alternative optimization tool, removed the independence restriction of the matrix based security model and the dynamic optimization method, but the first implementation of it was slightly slower than the other methods. For improving the performance of the evolutionary optimization we have performed a meta-level optimization of parameters of the algorithm and as a result the speed of optimization is comparable to other optimization techniques. As the evolutionary optimization is independent for all possible budget levels it lead to possibility to use a graph based security model. The graph based security model is a new and dynamical framework for security management.

This paper presents how implementation of an evolutionary optimization technique removed the restrictions of independence of security measures and lead to implementation of an efficient graph based security model.

Keywords: graded security model, information security metrics, genetic algorithms, evolutionary optimization,

1. Introduction

One of the most important tasks for IT security management is the optimal use of existing resources and the main idea for our R&D work is to propose to IT Security decision-makers a Graded Security Model (GSM) and a decision support

system for this. In papers (Kivimaa, 2009; Kivimaa, Ojamaa, and Tyugu, 2009; Ojamaa, Tyugu, and Kivimaa, 2008) it was shown how to use the GSM for finding optimal solutions based on the Pareto-optimal situation analysis, the discrete dynamic programming method for optimization calculations and weighted average confidence of security activities areas was used as optimization criteria. As it turned out the computational complexity of the optimization task is very high. For example, if to consider that an IT security model has 30-40 activity areas and in each of them has 4 possible implementation levels then there are $4^{30} \div 4^{40}$ possible solutions within to select an optimum. The Brute Force optimization technique requires a couple of years to calculate even one possible budget point.

In (Kivimaa 2009) was also brought up some weaknesses caused from the dynamic programming method. Namely, using dynamic programming in optimization of security activities areas must be independent from each other and their levels must be additive. To achieve better solutions in the future it is reasonable to continue GSM development – mainly to collect expert knowledge for the up-to-date model – that is, up-to-date information about security goals, their levels and information security activities areas and their realization levels dependency matrix and up-to-date theirs levels realization costs and effectiveness's. And, as requirement for independence of IT security activities is source for quite serious problems, to cover IT security problems in more detail and correct way we have to accept dependencies between lines in Dependencies Matrix - to describe these dependencies in addition to Dependencies Matrix use (find or work out) the IT security or IT security activities areas Dependencies Graph.

Because the independence of security activity areas was required by the Dynamic Programming (DP) method our aim was to apply an alternative method for optimization and we decided to use an evolutionary algorithm as a universal method for complex optimization in many fields. The evolutionary algorithm starts each optimization process from the beginning and therefore it does not have any problems related to independence and additivity.

As the evolutionary optimization is independent for all possible or interesting budget levels and intervals it leads to possibility to use a graph based security model. The graph based security model is a new and dynamical framework for security management. The new graph model gives us possibility to calculate the most needed/wanted reliability for a specific IT security System (also often named as Confidence) and Security Efficiency (SE), which value can be expressed as SE = Annual Loss Expextancy / Real Losses = 1 / (1 - Confidence).

Our main ideas are:

- use metrics to determine information systems security requirements - i.e. use high level risk analysis (levels of security goals) as IT security metrics;
- secure IT systems and their information in an economically rational/optimal manner – i.e. accordingly to data security requirements;

- the important issue in defining and implementing security measures is the economic efficiency of security activities, that is: we want to get the best results for our money - to minimize the costs and to maximize the integral security confidence.

2. Graded security model

The graded security model has been in use for a long time in the high-risk areas like nuclear waste depositories, radiation control etc. (DOE 1999, see also Kivimaa 2009 for details). In IT security is also reasonable to apply a methodology that allows one to select rational security measures based on graded security, and taking into account the available resources, instead of using only hard security constraints prescribed by standards that usually do not include economic parameters - the cost and efficiency of implemented security measures.

The ideas of graded security were used on the US Department of Energy security model (DOE 1999) and on its updated NISPOM version (NISPOM 2006).

In the NISPOM model 14 graded security activities areas are defined and 15÷20 left only on base levels. As the NISPOM model is meant for protection of critical information infrastructure it is obvious that these base levels are the highest possible implementation levels. But for institutions having less critical IT security these NISPOM areas on the base level have different possible implementation levels too – i.e. theoretically they are graded too (look Figure 1).

But the matrix based model has one quite serious limitation – in table we have no good possibilities to consider dependencies between table columns and rows – that is, there is not any good way to describe really existing additive and dependent nature in IT security goals and activities areas (Kivimaa 2009).

2.1 Graph based security model

It is possible to write dependencies between the matrix rows as functions into cells, but much more understandable and comprehensive results (understandable in one look) if we represent collection of rules as a graph structure. At the same we are no more limited to weighted average only, with graph we get possibility to calculate for decision makers some very interesting and important parameters about achieved security level - confidence and security efficiency (in more details look 2.2).

The graded IT security graph is based on the main ideas from the "(People - Process – Technology) and Organization" Business Model for IT security (ISACA 2009). Based on this and the IT security Dependency Matrix (Figure 1), containing security areas and their levels, a Bank IT security Graph (Figure 2) is formed.

Security Activities				Confidentiality				Integrity				Availability			
				C0	C1	C2	C3	I0	I1	I2	I3	A0	A1	A2	A3
Identified Security Goals	Reference			Public	Confidential	Secret	Top secret	Identification not needed	Non-authorized usage identified	Authorized usage identified	Identification usable in court	Delay measured in days	Delays measured in hours	Delays measured in minutes	Delays measured in seconds
G1: Organization of information security															
1	Security Documentation	SDOC	CIA	1	2	3	4	1	2	3	4	1	2	3	4
2	Risk Assessment and Treatment	RISK	CIA	1	2	3	4	1	2	3	4	1	2	3	4
3	Security Accrediting	SECA	CIA			1	2			1	2			1	2
G2: Human resources security															
4	IT Human Resource Management	HRM	CA	0	2	3	4					0	2	3	4
5	Awareness Training	AWT	CA	1	2	3	4					1	2	3	4
G3: Physical and environmental security															
6	Perimeter Security(Physical Security)	PSEC	C		2	3	4								
7	Communication & Process Support IT Systems (DBserver, failserver, printserver, meilisüsteem, e-	C&PSS	CIA	0	2	3	4	0	2	3	4	0	2	3	4
8	Personnel Working Environment(Physical Security)	ENV	C		2	3	4								
9	Personnel Workplace Equipment	WPE	A									0	1	2	3
10	Power	POW	A									0	2	3	4
11	Data Centres	DC	CIA	0	2	3	4	0	2	3	4	0	2	3	4
G4: Information systems acquisition, development and maintenance															
12	Outsourcing (incl. 3rd parties support)	OUTS	CIA	0	1	2	3	0	2	3	4	0	2	3	4
13	Purchased SoftWare	SW	CIA	0	2	3	4	0	2	3	4	0	2	3	4
14	Self-developed SoftWare	DEV	CIA	0	1	2	3	0	1	2	3	0	1	2	3
G5: Access control															
15	Access Rights Management	ARM	CA		1	2	3						1	2	3
16	Network Access Control	NAC	CI	1	2	3	4	1	2	3	4				
17	Mobile computing and teleworking	MOB	CA	0	2	3	4					0	2	3	4
G6: Communications and operations management (ISO 17799)															
18	Internal network security	LAN	CA	0	1	2	3					0	2	3	4
19	External network security (incl. PerimProt, IDS/IPS, ...)	WAN	CA	1	2	3	4					1	2	3	4
20	Malware Handling	AM	CI	1	2	3	4	1	2	3	4				
21	Information exchange policies and procedures	ENC	CIA		2	3	4		2	3	4		2	3	4
22	Transaction Integrity	TINT	I							1	1				
23	Data backup and Restoration	BCK	IA					1	2	3	4	1	2	3	4
24	Data Archiving	ARCH	IA					1	2	3	4	1	2	3	4
G7: Compliance															
25	External Regulations	REG	CI		1	1	2		1	1	2				
26	Audit Capability	AUD	CI	1	2	3	4	1	2	3	4				
G8: Information security incident management															
27	Audit Trail	LOG	C	1	2	3	4								
28	Monitoring(Help Desk)	MON	A										2	3	4
29	IT Governance(IT quality, fiduciarity & security manage	IT Gov	CIA	1	2	3	4	0	1	2	3	0	1	2	3
G9: Business continuity management															
32	Business Continuity Management(main input to ITS R)	BCM	A										1	2	3
30	IT Systems Recovery(Redundancy jms, sisend BCM'ist	ITS R	A										2	3	4
33	Crisis Management(tagab, et max IT riskid ~5% Panga k	CM	A										1	2	3
G10: Asset Management															
31	Asset Management	ASM	A										2	3	4
35															

Figure 1: IT security Dependency Matrix for a Bank

There are two important principles in IT security that are based on the graph much more visible and understandable:

- A chain is only as strong as the weakest link – in some IT security areas we must have valid reliability level otherwise overall reliability of security system will be 0 (look Figure 2 – mainly people, SW, Power, HW, LAN and AntiMalware) - so called *must-be elements* in the graph (look Figure 2).
- Layered security / defence in depth – we have a lot security activities areas that are parallel to so called must-be areas that make possible to raise reliability of these must-be areas (Figure 2).

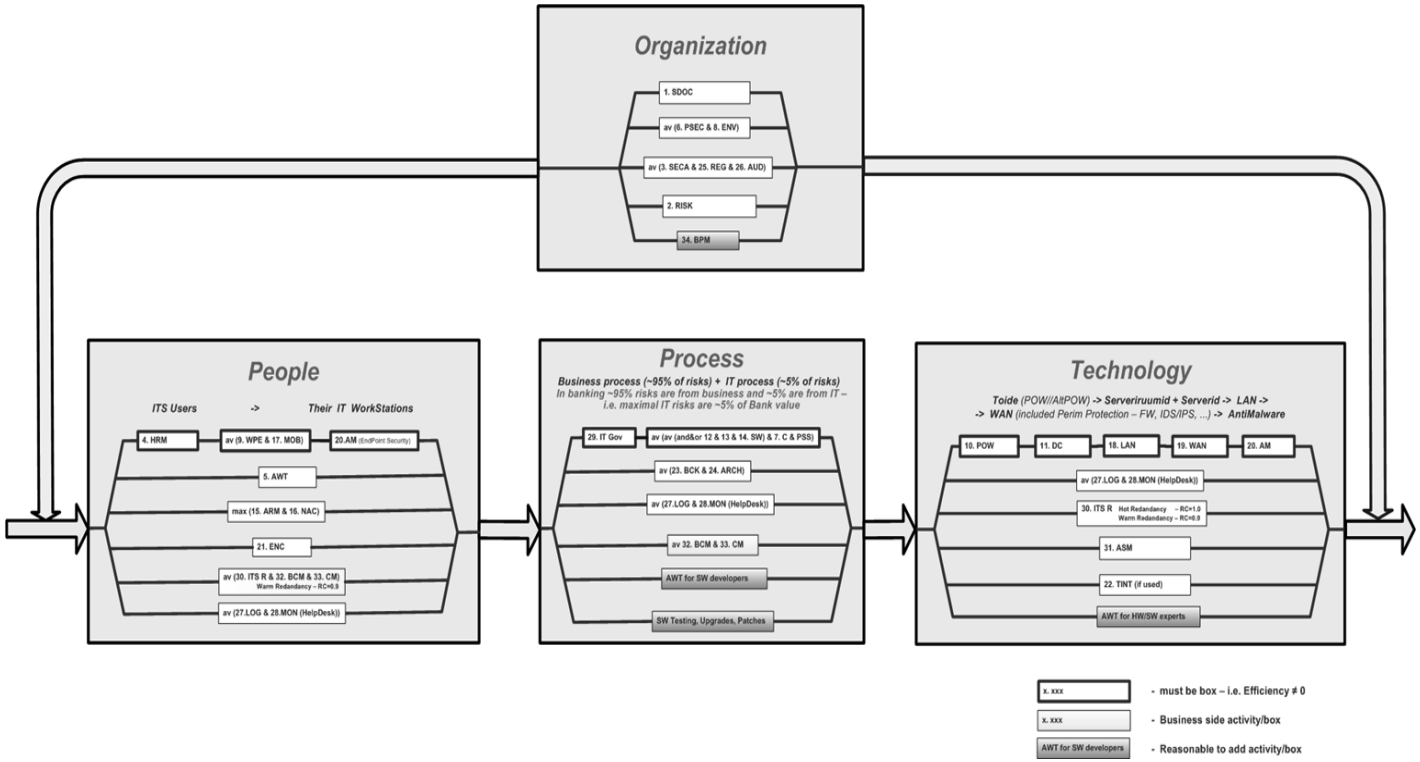


Figure 2: IT security Dependency Graph for a Bank

2.2 Model optimization

We are building a model that binds security measures (grouped by security activities areas) costs and confidences with achieved the security goals and their levels. We introduce a fitness function that presents by one numeric value the integral confidence of achieved security level. This allows us to formulate a problem of selecting security measures as an optimization problem in precise terms. However, we still have two goals: to minimize the costs and to maximize the integral security confidence. This problem will be solved by means of building a Pareto optimality trade-off curve that explicitly shows the relation between used resources and security confidence (Figure 3).

Knowing the available resources, we can find the best possible security level that can be achieved with the available resources and find the security measures to be taken. From the other side – if the required security level is given we can find the resources needed and the measures that have to be taken. This requires solving an optimization problem for each value of resources.

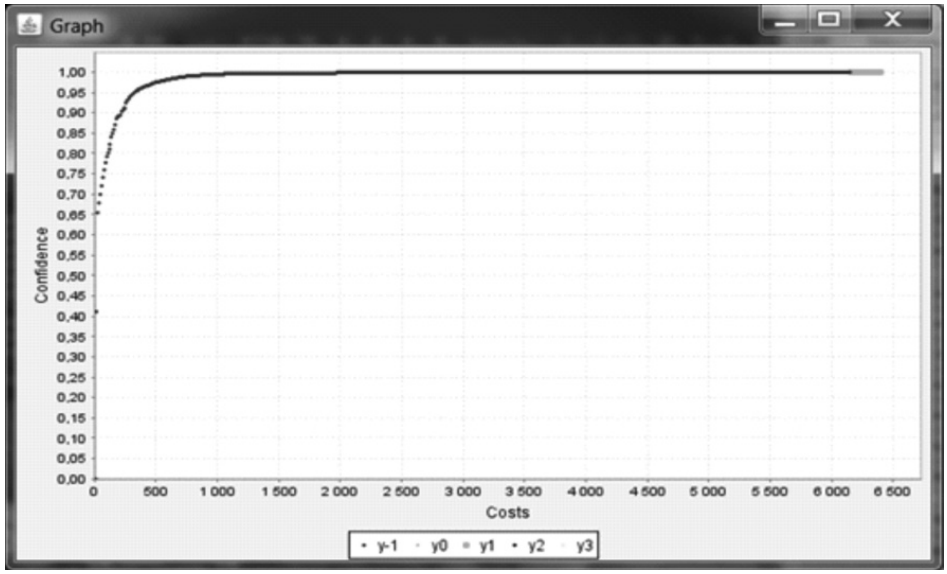


Figure 3: Search of optimal security along resource dimension – Pareto optimality trade-off curve

To calculate Pareto set/curve for GSM we have used/tested three possible optimization techniques:

- Brute Force
- Dynamic Programming
- Evolutionary Algorithms

And all approaches have their pluses and minuses. The first area for problems is calculations time needed for optimization (in more detail look 2.2.1).

Although the Dynamic Programming method is very good way to become free from calculation time problems (optimizations time for medium consumer desktop PC is excellent – minute or two), the DP has quite serious other limitations:

- security activities areas/security measures groups must be not dependent from each other
- their levels/security measures to realize their levels must be additive
- practically impossible to specify alternative and very close optimization results.

The best capabilities has the evolutionary algorithm – it has no problems with dependency/independency, additive/non-additive and matrix/graph, it finds all alternative or very close results for all possible and interesting cost-levels and the main advantage is that evolutionary optimization starts optimization for all possible and/or interesting budget points from the very beginning. The only possible problem is related to calculations time - the parameters for optimization have to be optimal (in more detail look 2.2.1 and 3.1).

2.2.1 The computational complexity of the optimization task

For comparing three optimization methods we will find calculation times for all three optimization methods for small and medium not IT-critical enterprises (~10 security activities areas) and for bigger IT-critical enterprises (for the Bank ~30 security activities areas):

1. **Brute force**

   We have to calculate and compare $\boldsymbol{qk^n}$ possible variations (**q** is the number of possible values of security budget levels, **n** is the number of security measure groups or security activities areas, **k** is the value of possible implementation levels for security measure group/security activities area, quite prevalently used 3 or 4):

   - For 10 security activities areas is required testing of $100*4^{10}=\sim 100*10^6$ variations,
   - For 30 security activities areas is required testing of $100*4^{30}=\sim 100*10^{18}$ variations,

   In more detailed IT security handling (n) optimization time increase is exponential and if to consider that medium consumer PC can perform optimization for 10 security activities areas (for small and not IT-critical institution, $\sim 100*10^6$ calculations and comparisons) in a minute then Brute Force optimization for bigger and IT-critical institution will take hundreds years.

2. **Dynamic programming**

   We have to calculate and compare $\boldsymbol{q^2kn}$ possible variants (**q** is the number of

possible values of security budget levels, **n** is the number of security measure groups or security activities areas, **k** is the value of possible implementation levels for security measure group/security activities area, quite prevalently used 3 or 4):

- For 10 security activities areas is required testing of $100*100*4*10=0,4*10^6$ variations,
- For 30 security activities areas is required testing of $100*100*4*30=1.2*10^6$ variations.

  In more detailed IT security handling optimization time increase is linear and consequenly n rise even the magnitude does not lead to any calculations time problems.

3. Evolutional

The number of variants required to calculate/compare by this algorithm is:

q * Population size * Number of Generations * Number of Repeats.

And as based on results of meta-level optimization (see 3.1.2) 'Population size' = n*3, 'Number of Generations' = n*4 and 'Number of Repeats' = 3 (**q** is the number of possible values of security budget levels, **n** is the number of security measure groups or security activities areas) and optimal number of variants to calculate and compare is $\mathbf{36*q*n^2}$:

- For 10 security activities areas is required testing of $36*100*10^2=0,36*10^6$ variations,
- For 30 security activities areas is required testing of $36*100*40^2=3.24*10^6$ variations.

For more detailed IT security handling optimization time increase is quadratic and consequently is quite important to use optimal parameters in optimization.

In conclusion:

- optimization time is critical,
- the Brute Force optimization method is inappropriate for more complex cases,
- the Dynamic Programming based optimization method has not any problems related to calculations time,
- for the Evolutionary method it is important to use the optimal optimization parameters.

2.2.2 GS graph-based model reliability/confidence calculations.

The main idea for optimization is to achieve graph's maximal Confidence with minimal Costs – i.e. Pareto set or Pareto frontier for GSM Costs or Confidence.

2.2.3 Reliability (alias Confidence) of series systems of "n" Identical and Independent components

A series system is a configuration such that, if any one of the system components fails, the entire system fails. Conceptually, a series system is one that is as weak as its weakest link. A graphical description of a series system is shown in Figure 4.

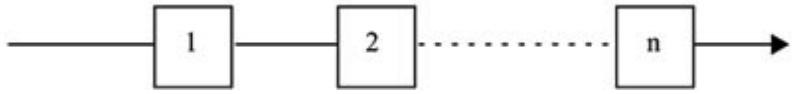


Figure 4: Representation of a Series System of "n" Components

Engineers are trained to work with system reliability [R_S] concepts using "blocks" for each system element, each block having its own reliability for a given mission time T:
$R_S = R_1 \times R_2 \times ... R_n$ (if the component reliabilities differ, or)
$R_S = [R_i]^n$ (if all i = 1, ... , n components are identical)

A set of n blocks connected in series can be replaced with a single block with the Reliability/Confidence R_S/C_S.

2.2.4 Reliability (alias confidence) of parallel systems

A parallel system is a configuration such that, as long as not all of the system components fail, the entire system works. Conceptually, in a parallel configuration the total system reliability is higher than the reliability of any single system component. A graphical description of a parallel system of "n" components is shown in Figure 5.

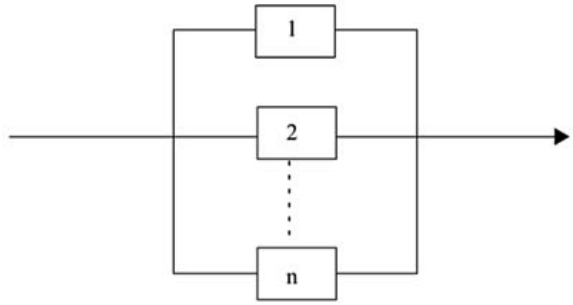


Figure 5: Representation of a Parallel System of "n" Components

Reliability engineers are trained to work with parallel systems using block concepts:
$R_S = 1 - (1 - R_i) = 1- (1 - R_1) \times (1 - R_2) \times ... (1 - R_n)$; if the component reliabilities differ, or
$R_S = 1 - [1 - R]^n$; if all "n" components are identical: [$R_i = R$; i = 1, ..., n].

A set of n blocks connected in parallel can be replaced with a single block with the reliability/Confidence R_S/C_S.

By recursively replacing the series and parallel subsystems by single equivalent elements we can obtain the Reliability/Confidence R_S/C_S for entire graph/system.

2.2.5 Specifics for GS graph-based model confidence calculations.

In GSM we have the only so called must-be serial box's and logic „if any one of the system components fails, the entire system fails" is exact and perfect.

But with parallel components is situation a bit more complicated. For full redundant security activities (for example, HW and Redundant HW) is principle „as long as not all of the system components fail, the entire system works exact, but if we have in parallel must-be security activity area with activities areas trying to improve the must-be activity Confidence (as example HW and Logging/Monitoring) then we have not fully redundant situation – we must bring in Redundancy Coefficient R_C. Practically $R_C = 1 \div 0{,}1$ - for full redundancy $R_C = 1$ and parallel to must-be activity with less Redundancy than 0,1 is pointless.

If for full redundancy $C = 1 - (1 - C_{1\_mb})*(1 - C_2) = C_{1\_mb} + C_2 (1 - C_{1\_mb})$
then bringing in Redundancy Coefficient R_C for Not-Full-Redundant parallel situations
$C = 1 - (1 - C_{1\_mb})*(1 - R_C * C_2)$ or $C = C_{1\_mb} + R_C * C_2 * (1 - C_{1\_mb})$

By recursively replacing the series (must-be) and parallel subsystems by single equivalent elements we can obtain the Reliability/Confidence R_S/C_S for entire graph/system and the new graph model gives us possibility to calculate for IT managers/decision makers the most needed/wanted values for IT Security optimization: reliability for a specific IT security System (also often named as Confidence) and Security Efficiency (SE), which value can be expressed as
SE = Max Annual Losses Expectancy / Real Losses = $1 / (1 - C_S)$.

For example, on Figure 6 SE is produced as a function from IT security activities and measures of costs.

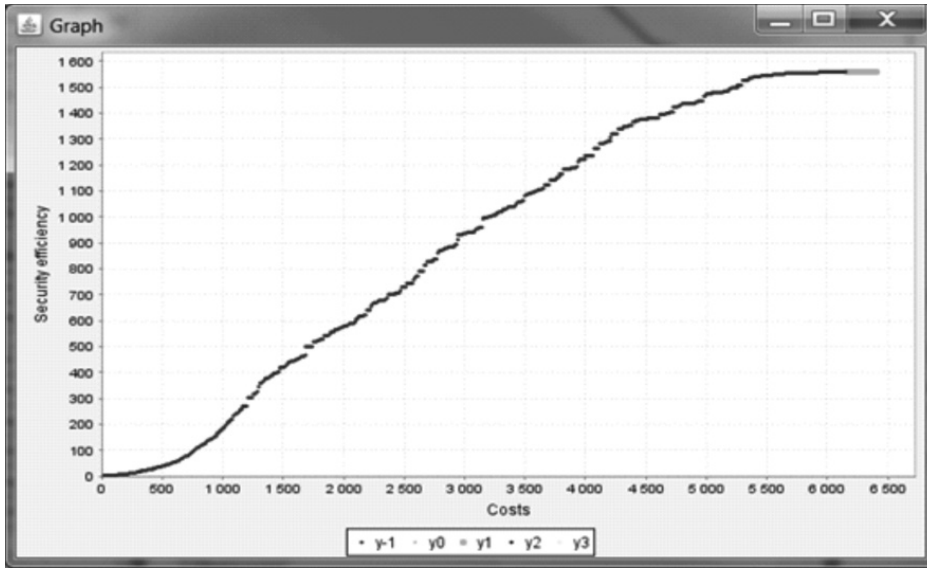


Figure 6: SE = f (Costs)

3. Evolutionary algorithms

Evolutionary algorithms are based on a Darwinian natural selection process and form a class of population-based stochastic search algorithms (Dracopoulos, 2008; Eiben & Smith, 2003; Holland, 1975). The view, that random variation provides the mechanism for discovering new solutions (Michalewicz & Fogel, 2004), was inspired by the process of natural evolution. The idea of using Darwinian principles of evolution to solve some combinatorial optimization problems arose with the invention of electronic computers. Now there are a wide variety of approaches that can be described as belonging to the field of evolutionary computing. The algorithms used in the field are termed as evolutionary algorithms (Dracopoulos, 2008).

The most important characteristics of evolutionary algorithms are as follows:

- Each candidate solution to the optimization problem is represented as an individual. The set of individuals are named as a population.
- The quality of a candidate solution is measured by a fitness function. Fitter solutions have a higher probability to survive and to contribute their characteristics to offspring (next generation).
- Variation operators (e.g., crossover, mutations) are applied to the individuals that modify the population of solutions dynamically.
- The average fitness is improved over time as a selection mechanism is applied and the fittest individuals are selected for the next generation (survival of the fittest).

The basis of an evolutionary algorithm is simple. First, a population of initial candidate solutions has to be generated randomly. Thereafter iteratively a number of variation generation operators are applied and for the new generations the fittest individuals are selected.

3.1 Meta-level optimization of evolutionary algorithms

The aim of this work is to optimize the parameters of an evolutionary algorithm. As the optimization process is based on randomness it makes the speed of the problem solving task rather variable. There are no hard and fast rules for choosing appropriate values for the parameters (Cicirello & Smith, 2000). The first scientist, who put a considerable effort into finding parameter values, was De Jong (1975). He tested different values experimentally and concluded that the following parameters give reasonable performance for his test functions: population size 50, crossover 0.6 and mutation rate 0.001 (see also for details Eiben, Hinterding, & Michalewicz, 1999). But those values are suitable for the problem that he had at hand. It has been shown that it is not possible to find parameter values which are optimal for all problem domains (Wolpert, & Macready, 1997) therefore each problem need its own approach and different set of parameters.

A widely practised approach to identify a good set of parameters for a particular class of problem is through experimentations and using the trial-and-error approach. As the evolutionary approach is mostly based on the trial-and-error to move through the search space therefore it would be reasonable to use the evolutionary algorithm itself to optimize its parameters and such approach is called as a meta-level optimization (Cicirello & Smith, 2000). The main weakness of this approach is that it is computationally expensive and takes a lot of time.

There are two ways to improve the performance of the evolutionary algorithm. The strategy can either be static or adaptive (Aine, Kumar, & Chakrabarti, 2006). For static framework, the parameter values are decided at the start of the algorithm and the decision is not revised during runtime. The static model works well when there is little or no uncertainty about the progress of the algorithm. For algorithms where the progress is not predictable and different parameter settings are suitable at different stages, a dynamic monitoring based strategy is preferred. In the dynamic case, the control decision is updated during runtime by monitoring the progress of the algorithm for a particular run. As the IT security costs optimization task is rather stable and does not include many uncertainties, we decided to find out a static set of parameters rather than develop a dynamic framework for parameter changes.

3.1.1. Meta-level optimization set-up

An individual in the optimization task was represented as a vector consisting of 10 elements. The elements represented the adjustable set of parameters: Repeat – how many times to repeat optimization process, Population – population size, Tournament – tournament size (number of individuals in a subset), Generations – a predefined number of generations, Crossover – probability of applying crossover operator (value 0.49 means that in 49% cases the crossover occurs), Mutate – probability of mutation, Swap – probability of swapping, Inversion – probability of inversion, Insertion – probability of insertion, and Displacement – probability of displacement. During the meta-level optimization process a candidate solution was optimized based on these parameters.

An important question was how to measure the fitness of the meta-level evolutionary optimization. We had two optimization goals, first, to find maximum level of confidence and second, to find it as fast as possible. Therefore we had to combine the measure of confidence and time. As each optimization was repeated **r** times the value of meta-level fitness function F was calculated as average of fitness of original task minus time:

$F = \text{sum}(c_i - t_i) / r$

where c_i is the confidence level and t_i is the calculation time in seconds of i-th experiment (see curve in Figure 7).

Results of meta-level optimization

We performed experiments with the data (Figure 1) consisting of 33 security activity areas. From the original data we formed 6 sets consisting of 13, 17, 21, 25, 29 and 33 areas. The parameters for meta-level optimizer were as follows: population size 75, tournament size 15 and the number of generations 75, crossover rate 0.9 and mutation rate 0.7.

The optimization process took almost two and half days. As we could see on the detailed graph (Figure 7) the fine tuning of the meta-level optimization took some time to find the optimal level.

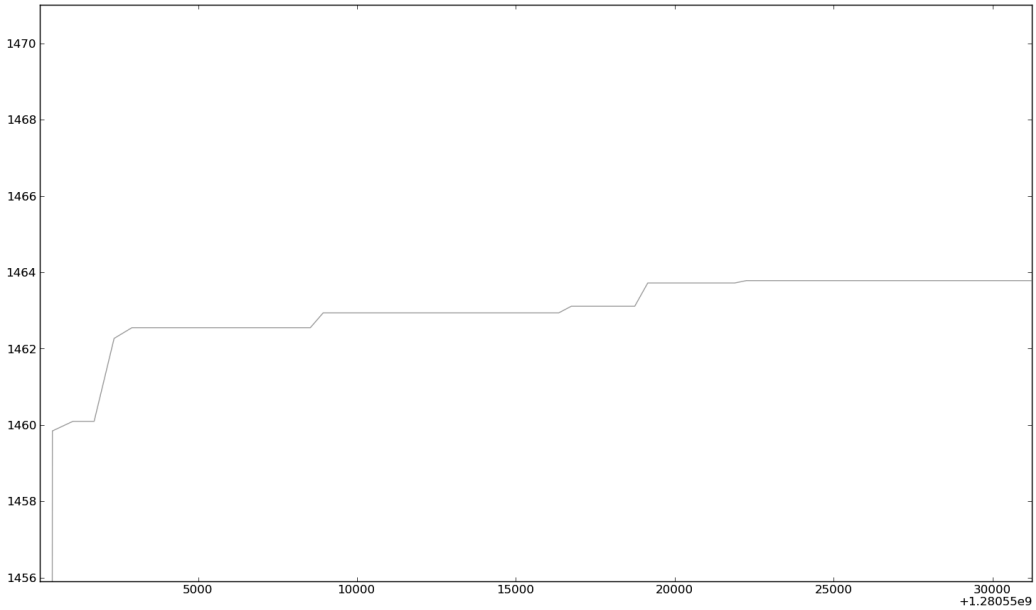


Figure 7: The fitness value of the meta-level optimization task (upper part of the fitness curve)

Average results of the optimization process are given in Table 1.

Table 1: Average values of parameters as a result of meta-level optimization

No	Pop	Tourna-ment	Gene-rations	Cross-over	Muta-tion	Swap	Inver-sion	Inser-tion	Displace-ment
13	28.86	41.43	42.86	0.82	0.7	0.58	0.19	0.15	0.15
17	35.57	69.14	67.43	0.85	0.89	0.63	0.14	0.16	0.12
21	46.57	40.71	70.71	0.8	0.88	0.53	0.1	0.13	0.12
25	43.43	31.86	95.86	0.85	0.77	0.61	0.08	0.13	0.15
29	48.86	65.29	92.43	0.8	0.89	0.74	0.07	0.1	0.16
33	61.43	37.71	96.43	0.91	0.74	0.72	0.13	0.06	0.13

As we calculated correlation coefficients (Table 2) we could see that there is strong linear correlation between the number of security activity areas (the size of task) and the number of individuals in a population (r=0,95) and the number of generations (r=0.92). There is also positive correlation between the size of task and crossover probability (0.45). With the most other probability values the correlation is negative.

Table 2: Correlation coefficients of all 35 selected results

	No	Pop.	Tourna-ment.	Gen.	Cross-over	Muta-tion	Swap	Inver-sion	Inser-tion	Dis-place-ment
No	1	0.95	-0.13	0.92	0.45	0.06	0.73	-0.64	-0.92	0.16
Population	0.95	1	-0.21	0.82	0.48	0.08	0.57	-0.53	-0.93	-0.12
Tournament	-0.13	-0.21	1	-0.1	-0.29	0.7	0.37	-0.06	0.24	-0.04
Generations	0.92	0.82	-0.1	1	0.4	0.18	0.62	-0.8	-0.71	0.16
Crossover	0.45	0.48	-0.29	0.4	1	-0.47	0.4	0.2	-0.51	-0.28
Mutate	0.06	0.08	0.7	0.18	-0.47	1	0.05	-0.56	0.18	-0.3
Swap	0.73	0.57	0.37	0.62	0.4	0.05	1	-0.29	-0.7	0.38
Inversion	-0.64	-0.53	-0.06	-0.8	0.2	-0.56	-0.29	1	0.33	-0.21
Insertion	-0.92	-0.93	0.24	-0.71	-0.51	0.18	-0.7	0.33	1	-0.12
Displacement	0.16	-0.12	-0.04	0.16	-0.28	-0.3	0.38	-0.21	-0.12	1

In Figure 8 we could see that the probabilistic values of variation operators (Crossover, Mutation and Swap) had quite high values and the others value was rather small and even diminished as the problem grows. Probably their computational cost was relatively high comparing the gain of fitness.

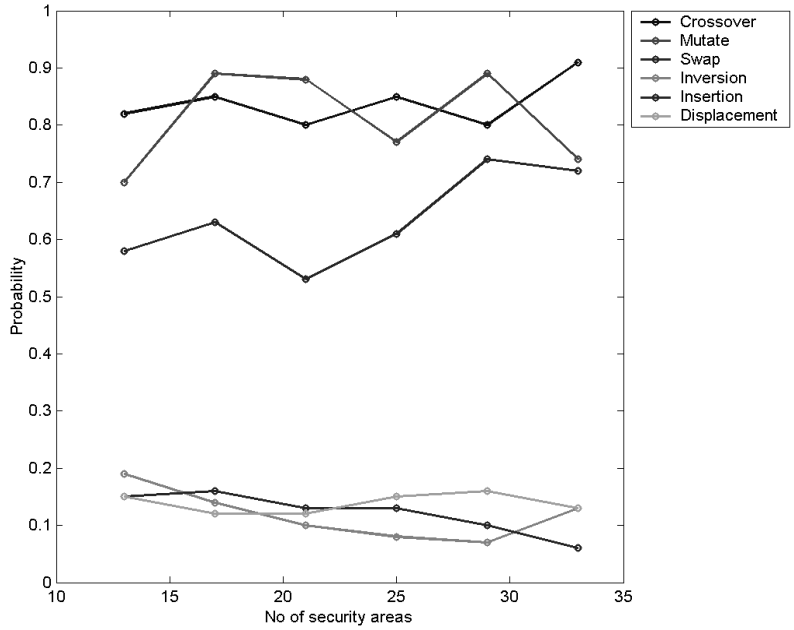


Figure 8: Change of probability of variation operators

In Figure 9 we could see that there is a clear linear relation between the problem size and the population size and the number of generations.

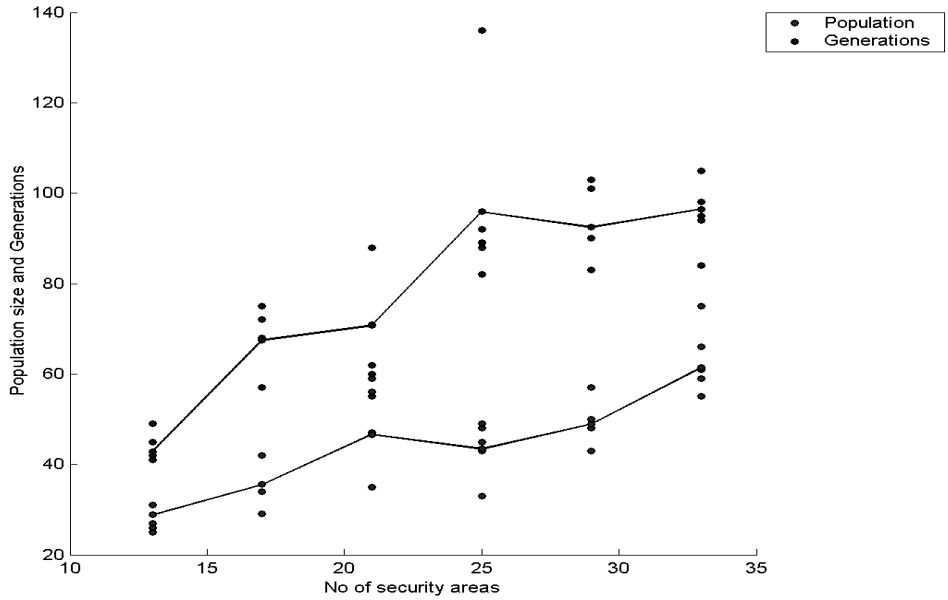


Figure 9: Distribution of population and generation values and their mean value (line)

Based on the measurements we were able to generate formulas to specify the parameters of evolutionary optimizer. As we added to the mean value and the standard deviation $\mu + \sigma$ to get rough estimate for the population related values (e.g., based on the mean value of Generations / Number security activity areas $\mu = 3.429$, standard deviation $\sigma = 0.5688$, we can calculate the coefficient $3.429 + 0.5688 \approx 4$).

The results could be as follows:

repeat	3
population size	N * 3
tournament size	50
generations	N * 4

where N is the number of security activity areas and the number of security levels is 4.

As there was a tendency to move closer to certain values we decided to use in further optimizations the following parameter set for variation operators:

crossover rate	0.9
mutation rate	0.8
swap rate	0.6
inversion rate	0.1
insertion rate	0.07
displacement rate	0.11

As we could predict optimal population related parameters and also identified optimal values for probability operator values we could estimate optimization time and to perform optimization tasks much faster.

4. Conclusions

We have performed an analysis to identify linear coefficients for estimating the parameter values of the evolutionary algorithm. As a result we have found a way to calculate the value for population size and the number generations that are based on the problem size and also identified optimal parameter set for variation operators. It makes the use of evolutionary algorithm more efficient and enables us to increase the optimization speed. As there are certain restrictions related to the other optimization techniques the evolutionary approach also enables us to enhance the IT security methodology and a new graph-based model is proposed.

But wider application of the graph-based model will depend on the availability of expert knowledge or statistics that binds costs and security confidence values with the security measures. This expert data will depend on the type of the infrastructure where information must be protected - different for different countries and economy areas. The only realistic solution is an expert system that can be adjusted by experts to suit concrete situations. Therefore some further work is needed to enhance the model and provide appropriate expert knowledge to turn the model more accurate.

References

Aine, S., Kumar, R., and Chakrabarti, P.P. (2006) “Adaptive Parameter Control of Evolutionary Algorithms Under Time Constraints”, in A., Tiwari, J. Knowles, E. Avineri, K., Dahal, and R., Roy (Eds.), Applications of Soft Computing, Berlin, Springer, pp. 373–382.

Cicirello, V. A., and Smith, S. F. (2000) “Modeling GA performance for control parameter optimization”, in D., Whitley, D., Goldberg, E., Cant-Paz, L., Spector, I., Parmee, and H., Beyer (Eds.), GECCO-2000: Proceedings of the Genetic and Evolutionary Computation Conference, Las Vegas, NV, pp. 235–242.

De Jong, K. (1975) “The analysis of the behavior of a class of genetic adaptive systems”, Ph.D. dissertation, Department Computer Science, University of Michigan, Ann Arbor, MI.

DOE (1999) Classified Information Systems Security Manual. Retrieved February 1, 2010, from https://www.directives.doe.gov/directives/archive-directives/471.2-DManual-2/at_download/file.

Dracopoulos, D. C. (2008) “Evolutionary Learning”, in B. Wah (Ed.), Wiley Encyclopedia of Computer Science and Engineering. New York, John Wiley and Sons.

Eiben, A. E. , Hinterding, R., and Michalewicz, Z. (1999) “Parameter control in evolutionary algorithms”, IEEE Transactions on Evolutionary Computation, Vol 3, No. 2, pp. 124–141.

Eiben, A. E., and Smith, J. E. (2003) Introduction to Evolutionary Computing, Berlin, Springer.

Holland, J. H. (1975) Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, Cambridge, MA, MIT Press.

ISACA (2009) “An Introduction to the Business Model for Information Security,” ISACA.

Kirt, T., and Kivimaa, J. (2010) “Optimizing IT security costs by evolutionary algorithms”, in C. Czosseck, and K. Podins, (Eds.), Conference on Cyber Conflict Proceedings 2010, Tallinn, Estonia, Cooperative Cyber Defence Centre of Excellence Publications, pp. 145–160.

Kivimaa, J. (2009) “Applying a costs optimizing model for IT security”, in H. Santos (Ed.), Proceedings of the 8th European Conference on Information Warfare and Security, Reading, UK, Academic Publishing Limited, pp. 142–153.

Kivimaa, J. Ojamaa, A. and Tyugu, E. (2009) “Graded security expert system”, in CRITIS 2008: Third International Workshop on Critical Information Infrastructure Security, Rome, Springer.

Michalewicz, Z., and Fogel, D. B. (2004) How To Solve It: Modern Heuristics, Berlin, Springer.

NISPOM (2006) “National Industrial Security Program Operating Manual,” U.S. Department of Defense..

Ojamaa, A., Tyugu, E., and Kivimaa, J. (2008) “Pareto-optimal situation analysis for selection of security measures”, in Military Communications Conference MILCOM 2008: Unclassified Proceedings, Piscataway, NJ, IEEE, pp. 3224–3230.

Wolpert, D., and Macready, W. G. (1997) “No free lunch theorems for optimization”, IEEE Transactions on Evolutionary Computation, Vol 1, No. 1, pp. 67–82.

STUDY VII

QUANTITATIVE SYSTEM RELIABILITY APPROACH FOR OPTIMIZING IT SECURITY COSTS IN AN AI ENVIRONMENT

Alberghs, Geert; Grigorenko, Pavel; Kivimaa, Jyri

Alberghs, Geert; Grigorenko, Pavel; Kivimaa, Jyri (2011).
Quantitative system reliability approach for optimizing IT security costs in an AI environment.
12th Symposium on Programming Languages and Software Tools, SPLST'11: Tallinn, Estonia, 5-7 October 2011, Proceedings. Tallinn: TUT Press, 2011, 219 – 230.
ISBN: 978-9949-23-178-2
Classification: 3.2

Geert Alberghs[1], Pavel Grigorenko[2], and Jyri Kivimaa[3]

[1]Ecole Supérieure d'Informatique Electronique Automatique, Paris, France, geert.alberghs@mil.be
[2]Institute of Cybernetics, Tallinn University of Technology, Tallinn, Estonia, pavelg@cs.ioc.ee
[3]Cooperative Cyber Defence, Centre of Excellence, Tallinn, Estonia, jyri.kivimaa@ccdcoe.org

Abstract

The Graded Security Model (GSM) addresses the IT Security cost optimization, problem by trying to find an answer to the following question: ”For a certain budget level, in which IT security measures should be invested to achieve the highest possible overall security level?” This paper describes how reliability engineering can be applied to solve the GSM optimization problem. The organization’s IT security measures are represented in a reliability block diagram, which in turn can be translated to an undirected graph. The total reliability of the diagram can be calculated after the identification of Minimal Cut Sets (MCSs). Cellular Automata (CA) are combined with Monte Carlo (MC) sampling to allow the identification of all MCSs. This approach allows the replacement of every possible user provided diagram by a series structure of parallel components, for which the total reliability can always be calculated. Additionally, this new model allows the calculation of cut set criticalities and component Fussell-Vesely (FV) importance values. All implementations have been realized with the Artificial Intelligence (AI) platform CoCoViLa.

1 Introduction

1.1 IT Security Investment Optimization

Information security has turned out to be a critical business component. The success of an organization is closely related to its ability to appropriately manage risks. That is why Cost-effectiveness analysis[14] software for security investments is now becoming an absolutely indispensable decision support tool.

Over the past few decades several models and frameworks have been suggested to help management with the selection of appropriate security measures. These models can be categorized into three main research areas.

The first type of models, the *think like an attacker* models ([4]), is the most intuitive. Sequential or tree analysis techniques are used to identify possible hacker actions. Security measure selection is based on incident likelihoods, cost-benefit criteria, pruning of duplicate security measures in the attack tree, etc.

[14] Cost-Effectiveness analysis is distinct from cost-benefit analysis, which assigns a monetary value to the measure of effect.

The main drawback of these models is that selection of security measures is only considered during the production phase and is not embedded in the Software Development LifeCycle (SDLC).
The problem that arises with the second type of models known as SDLC models ([5]), is the definition of security goals like Confidentiality, Integrity and Availability as functional requirements. This is why in some SDLC models ([9]) only best practices are implemented. The drawback here is that the commonly identified best practices might not be the optimal solution for a particular organization.

This paper is situated in the third research area: *Economics of Investments in Information Security*. In this field metrics as Return On Security Investments, Cost-Benefit analysis, Net Present Value and Internal Rate of Return are used to select the correct security measures. There are two subgroups of economic models: general economic models ([6]) which describe information security investment trends and laws, and models that use economical measurements to identify, select and optimize security measures for a particular organization ([3]).

Our research is part of the second subgroup and uses Cost-Effectiveness analysis as a metric. Case studies have been performed based on Estonian SEB Bank and SwedBank expert data.

1.2 The Graded Security Model (GSM)

Selection of the right security measures appears to be a complex problem, because multiple objectives need to be achieved at the same time.
Organisations need to:

- 1. attain their security goals,
- 2. with maximum efficiency and
- 3. at minimum cost[15].

The security goals to reach can be confidentiality, integrity and availability. Other security goals can be added according to specific organizational needs. (e.g. non-repudiation, authentication)

A major obstacle for finding a conclusive answer for the cost-effectiveness optimization issue in IT

security is the lack of reliable metrics. In our Graded Security Model (GSM) the metrics of the NISPOM 2006 approach [17] are used to express the relations between security goals and security measure groups, where each security measure group *i* can be implemented at different levels *li*. As in [2] each level has

[15] Losses considerations have been omitted for reasons of clarity, but are definitely included in our cost-effectiveness analysis model and - tool

additionally been characterized by its maintenance cost mi,l_i, its investment cost ii,l_i and its efficiency ei,l_i. The efficiency levels are expressed as probabilities and indicate how confident[16] we are that our security measure group implemented at a certain level, will not be the underlying reason of any security incident.

1.3 The Graded Security Expert System (GSES)

Based on the GSM described in Section 1.2 a cost-effectiveness analysis tool for IT security investments, the Graded Security Expert System (GSES), has been developed with the Artificial Intelligence (AI) software CoCoViLa [1, 7]. The GSES aims to maximize the overall system effectiveness E while staying within the available budget b, and this for a certain range of budget levels.

Now let the protection profile $p = (l_1, \ldots, li, \ldots, ln)$ be the tuple representing a security level li for each security measure group between 1 and n.

The overall cost functions, Investment Cost $I(p)$, Maintenance Cost $M(p)$ and Total Cost $C(p)$ can then easily be written as follows:

$$I(p) = \sum_{i=1}^{n} i_{i,l_i} \tag{1}$$

$$M(p) = \sum_{i=1}^{n} m_{i,l_i} \tag{2}$$

$$C(p) = I(p) + M(p) \quad \text{where of course} \quad C(p) \leq b \tag{3}$$

1.3.1 The weighted average approach

The first versions of the GSES software [11, 13, 14, 15] used a weighted average for determining the overall efficiency $E(p)$. wi represents the weight of security measure group i.

$$E(p) = \sum_{i=1}^{n} w_i e_{i,l_i} \quad \text{with} \quad \sum_{i=1}^{n} w_i = 1 \tag{4}$$

This method has several drawbacks. Users have difficulties assigning correct weights to each security measure group and since weights are constants there is no possibility to define dependencies between effectiveness values of security measure groups or to include the influence of the security goals.

[16] The notion confidence can be considered as the exact opposite of the term likelihood used in risk management: Likelihood = 1 – Confidence

1.3.2 The measure group relationship diagram

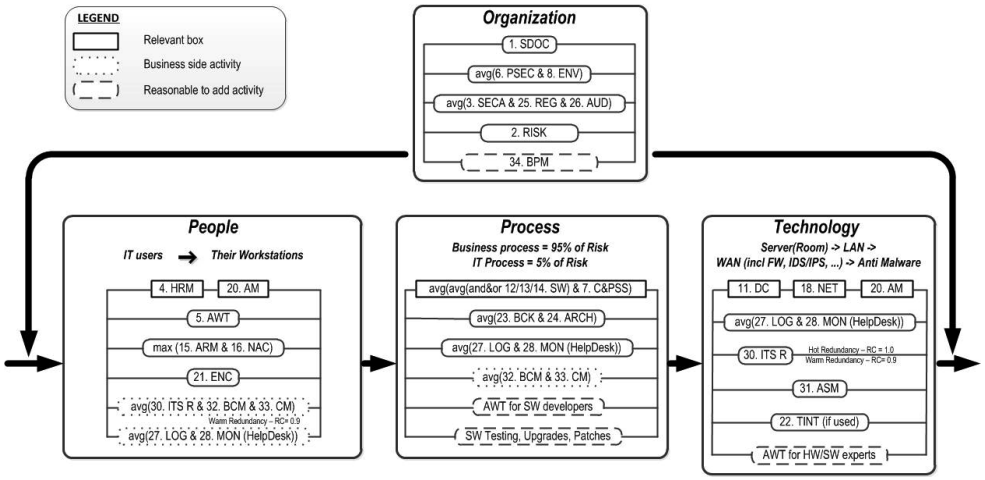


Figure 1: Example of a GSM measure group relationship diagram

To cope with the drawbacks mentioned in Section 1.3.1 the redundancy coefficient Rc has been introduced in [12]'s GSES version[17] to represent inter-component dependencies in a measure group relationship diagram. The values of this coefficient are generally easier to estimate than the weights of (4).

Rc represents the dependency between a so-called relevant measure group and his supporting measure group. When establishing the structure of a system, it seems reasonable to be able to reduce the system to the components that play a direct role for the functioning ability of the system. The components we are left with are called relevant components. To avoid the usage of the term "irrelevant", the components that are not relevant are called "supporting". Supporting measure groups are always drawn parallel to the relevant measure groups they influence, the latter being outlined in red in Figure 1.

Good examples of supporting measure groups are "logging" and "monitoring". They improve, for instance, an organization's capability to detect hardware errors. The efficiency of the security measure group "redundant hardware" would thus clearly be influenced by changes in the implementation level(s) of "logging" and "monitoring".

As for the diagram based calculations:

[17] Although the new approach hasn't been explicitly mentioned in this paper

- A series configuration is always less efficient than its weakest component:

$$E(p) = \prod_{1 \le i \le n} e_{i,l_i} \tag{5}$$

- A parallel configuration is always more efficient than its strongest component:

$$E(p) = 1 - \prod_{1 \le i \le n} (1 - e_{i,l_i}) \tag{6}$$

- Finally redundant connections are calculated as follows:

$$E(p) = 1 - (1 - E_r) \prod_{1 \le i \le n} (1 - R_{c_i} e_{i,l_i}) \tag{7}$$

In (7) r represents the relevant measure group, with measure groups 1 through n supporting it and $\forall R_{Ci} \in [0, 1], 1 \le i \le n$. *If* $R_{Ci} < 0.1$ the influence of measure group i is probably too small to invest in it, and if $Rc_i = 1$ measure group i is said to be fully redundant.

Now, using (5), (6) and (7) the serial and parallel subsystems of the diagram can be recursively replaced by their single equivalent components until the overall efficiency is found.

The idea behind the relevant measure groups in the measure group relationship diagram is that if one of them fails ($ei,l_i = 0$) the entire system should fail ($E(p) = 0$). This means that in this model relevant measure groups cannot be placed in parallel. Another problem is that relations between measure groups can only be serial or parallel: bridge-, star- and other topologies are not possible.

1.4 Information Security Models

ISACA mentions in [8] that until January 2009 there was no official holistic or dynamic model for security responsible to use as a guidance for managing IT security risks. There are many standards and frameworks to address specific needs, but no overarching model that could exist in any organization regardless of geographic location, industry size, regulation or existing protocol. In fact, the answer ISACA sought is exactly what is needed in the GSM to model security efficiency. Their solution is to represent an organization by using 4 elements and 6 dynamic interconnections as shown in Figure 2 and *by assigning all organization's security measure groups to the correct elements and interconnections*. Each security measure group can be present in more than one element and/or interconnection and depending on its location it can be more or less efficient.

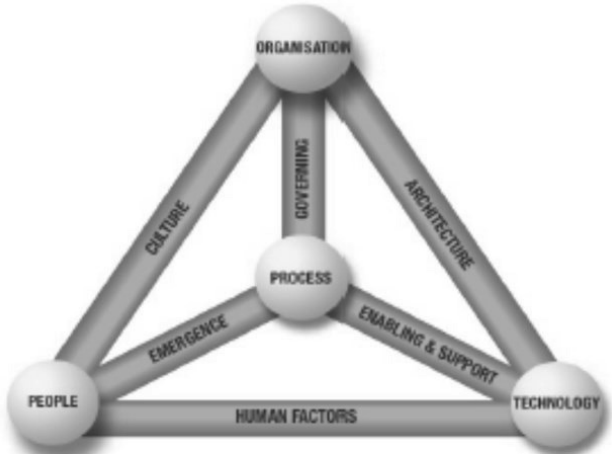


Figure 2: The Business Model for Information Security

ISACA's Business Model for Information Security (BMIS) and the measure group relationship diagram suggest the usefulness of a graph structure for representing the security posture of an organization.

1.5 Improving the model

The main idea behind this paper is that the GSM should become a holistic model as the BMIS, able to represent all types of organizations. It cannot be subject to the limitations mentioned in 1.3.2. A solid mathematical background will be added. The efficiency levels, previously expressed as roughly estimated weights and confidence values, will be made more quantifiable. It will also be possible to prioritize among relevant security measure groups by using Fussell-Vesely importance values. Finally the Minimal Cut Sets (MCSs) concept will allow us to look at an organization through the attacker's eyes: A MCS actually is the smallest set of IT Security controls which, when disabled, prohibits an organization to reach its security goals.

2 Graph structure

An undirected graph (Figure 3) is used for modeling the security efficiency. [5] The relevant measure groups are the edges of the graph connecting the circular nodes. The nodes are considered being fully reliable (efficiency of 1).

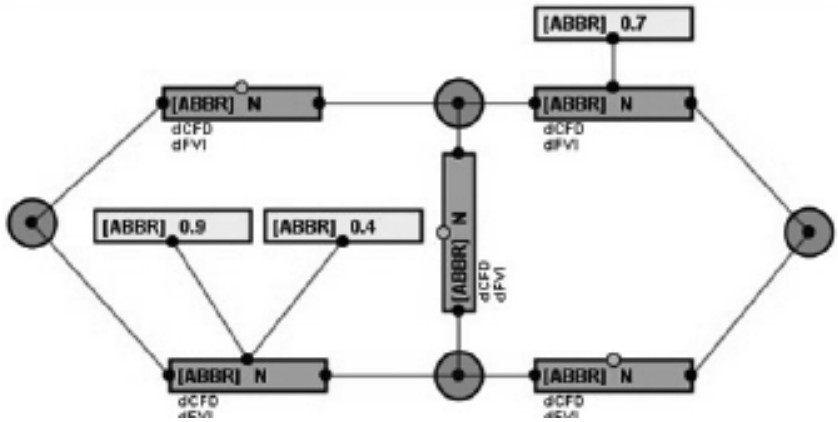
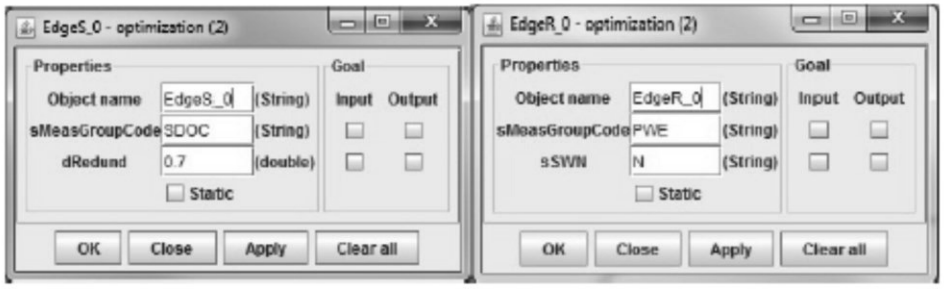


Figure 3: Graph structure used as a test-case in CoCoViLa

The boxes, connected to the relevant measuregroups which are not part of the graph structure as such, represent the supporting security measure groups. The properties of both types of security measure groups are shown in Figure 4 and explained in Table 2.



(a) Supporting Edge (b) Relevant Edge

Figure 4: Edge Properties in CoCoViLa

To reduce Figure 3 to a real graph structure, (7) is used between relevant and supporting security measure groups.

Variable	EdgeS	EdgeR	Description
String ObjectName String	X	X	Name of the object instance
sMeasGroupCode	X	X	Allows access to the cost and efficiency data of the referenced security measure group
String sSWN		X	Allows (e.g. ±10%) variations on efficiency values as suggested by the BMIS Strong, Weak, Neutral attributes
double dRedund	X		Redundancy coefficient

3 System Reliability Approach

3.1 Introduction

Threats exploit vulnerabilities and manifest themselves through a certain impact on the organization. Impacts can be measured rather easily, threats and vulnerabilities unfortunately not. All information about the measure group is described with the probability density function *f (t)* of its time to failure T. No explicit modeling of the threats and vulnerabilities is carried out. Reliability characteristics like *failure rate* and *Mean Time To Failure (MTTF)* are deduced directly from the probability density function *f(t)*. After several components (measure groups) are combined into a system (organization) a System Reliability Analysis can be performed.

By applying the ISO 8402 definition of reliability to our model, efficiency can be formulated as: *the ability of the security measure group to perform a required security function, under given threats and vulnerabilities and for a stated period of time*.

To verify if the measure group performs its required security function:

1. the security incidents need to be recorded in an incident management system
2. the causes of the incidents need to be identified. (i.e. find out which security measure group failed)
3. the *failure rate* of the involved measure group must be updated after each incident

The next section explains how the measure group efficiencies can be derived from these failure rates.[186]

3.2 Incident model

A well maintained and updated measure group can be considered as good as new during its entire useful lifetime, meaning that the failure rate λ is approximately a constant.[197] That is why the exponential distribution, the most commonly used life distribution in applied reliability analysis, can also be used in the GSES. Its main benefits are its mathematical simplicity and that it has often proved to lead to realistic lifetime models.

The definition of an exponential distribution is as follows:

$$f^i(t) = \begin{cases} \lambda_i e^{-\lambda_i t} & \text{for } t > 0 \quad \lambda_i > 0 \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where $f^i(t)$ represents the probability density function of the time to failure T for a certain measure group i. The cumulative distribution function then becomes:

$$F_T^i(t) = Prob(T \le t) = \begin{cases} 1 - e^{-\lambda_i t} & \text{for } t > 0 \quad \lambda_i > 0 \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

Its reliability or efficiency function can then be written as:

$$E_T^i(t) = Prob(T > t) = 1 - F_T^i(t) = e^{-\lambda_i t} \quad \text{for} \quad t > 0 \tag{10}$$

with $MTTF = 1/\lambda i$ and the failure rate function $z^i(t) = \lambda i$

So this means that:

- A measure group in us $\lambda_i = \frac{\text{Number of Incidents for security measure group } i}{\text{Observation Time}}$
- Only one parameter

needs to be collected (or estimated by experts) for each measure group

Pseudo random generators always follow a uniform distribution $U(0, 1)$, but random incidents against our measure groups respecting an exponential distribution can be simulated by applying the probability integral transform. The probability integral transform says that if a variable T has a continuous distribution for which the cumulative distribution function is $FT(t)$, then the random variable $Y = FT(t)$ has a uniform distribution.

Applied to our exponential distribution one can easily obtain the following equation:

$$T = \frac{-1}{\lambda_i} \ln(1 - U) \quad \text{with} \quad F_U(u) = \begin{cases} 1 & \text{for} \quad 0 \le u \le 1 \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

[18] For more in-depth explanations about reliability engineering please read [16], which has been used as the mathematical basis for this section.

[19] Burn-in and wear-out periods are not considered here. Extra caution is always needed during implementation and retirement phases of security measures. In IT security particular attention should also be devoted to the fast technological evolutions.

So now we are able to randomly generate incidents for the measure groups by:

1. setting the observation time *t*
2. randomly generating a sample *U* from a uniform distribution *FU (u)*
3. calculating *T* according to (11)
4. comparing *T* with *t*: if $T < t$ then an incident has occurred

Now the GSES is able to model the security efficiency and to generate random incidents for all measure groups separately. The next step is the definition of the overall efficiency.

3.3 Reliability Block Diagrams (RBDs)

A Reliability Block Diagram (RBD) is a success-oriented network describing the function[208] of the system. It shows the logical connections between components needed to fulfill this specified system function.

In the GSM matters have been simplified by assuming that the components are non-repairable and that the order in which the incidents occur does not matter. When the systems are repairable and/or the order in which failures occur is important, the more complex Markov methods should be used. In Markov methods the different states of the system need to be defined and the probabilities of transition between states should be estimated. The former is difficult but feasible, the latter however would be an almost impossible task in our case.

A system composed of n components will be denoted a system of order *n*. The set of components is denoted by: $C = (1, 2, \ldots, n)$

For both the components and the system itself a distinction between a functioning and a failed state is made. The state of component *i*, $i = 1, 2, 3, \ldots, n$, can then be described by the binary variable *xi*, where

$$x_i = \begin{cases} 1 & \text{if component i is functioning} \\ 0 & \text{if component i is in failed state} \end{cases}$$

, $x = (x_1, x_2, \ldots, x_n)$ is called the *state vector* (12)

Similarly the state of the system can be described by a binary function

where $\varphi(x) = \varphi(x1, x2, \ldots, xn)$, and

$$\phi(x) = \begin{cases} 1 & \text{if the system is functioning} \\ 0 & \text{if the system is in failed state} \end{cases} \quad (13)$$

[20] In our case the system function would be: provide security to the organization

φ(x) is called the structure function of the system and can be written as:

$$\phi(x) = \begin{cases} \prod_{i=1}^{n} x_i & \text{for serial components} \\ 1 - \prod_{i=1}^{n}(1-x_i) = \coprod_{i=1}^{n} x_i & \text{for parallel components} \end{cases} \quad (14)$$

3.4 Minimal Cut Sets (MCSs)

A cut set κ is a set of components in *C* which by failing causes the system to fail. A cut set is said to be minimal if it cannot be reduced without loosing its status as a cut set.

Now consider a saboteur who wants to bring the system in a failed state, with the least possible effort on his/her part. What the saboteur would need is a list of the MCSs of the system.

With the definition of MCSs in mind the structure function can be rewritten as:

$$\phi(x) = \prod_{j=1}^{k} \kappa(x) = \prod_{j=1}^{k} \coprod_{i \in \kappa_j} x_i \quad (15)$$

Until now our model was deterministic in nature, but the state variables x_i of the *n* components should be looked at as random statistical variables $X_i(t)$ representing the statistical events of security incidents occurring. The state vector 12 and system structure function (15) should be adapted accordingly.

$$X(t) = (X_1(t), X_2(t), \ldots, X_n(t))\phi(X(t)) = \prod_{j=1}^{k} \coprod_{i \in \kappa_j} X_i(t) \quad (16)$$

Because the distributions of the state variables $X_i(t)$ are known (9), the structure function of the complete system can be calculated by using the MCSs as shown in (16).

In our environment the saboteur would be called the attacker or the threat; the system is referred to as the organization and failures would be replaced by security incidents. The structure function represents the overall security efficiency of the organization.

The only remaining problem now is to find an algorithm which is able to find all cutsets in a given reliability diagram.

4 Minimal Cut Set search algorithm

4.1 Introduction

A methodology based on a combination of Cellular Automata (CA) and Monte Carlo (MC) sampling is used to identify the MCSs of our system reliability diagram. A ranking of the measure groups criticalities can be achieved through the calculation of their Fussell-Vesely importance values.[219]

A candidate cut set is generated by using the probability integral transform as explained in Section 3.2. CA will be used to decide if the generated set of failed edges is a cut set or not. And finally MC will allow us to determine the MCSs.

4.2 Cellular Automata (CA)

To verify if a connection between the source and the target still exists, after applying random failures to the edges of our graph, CA is used. Consider a graph containing n nodes with a source node S and a target node T. Each node i can be in 2 states: active ($s_i(t) = 1$) or passive ($s_i(t) = 0$) and each edge i j can be in 2 states: success ($e_i\,j(t) = 1$) or failure ($e_i\,j(t) = 0$). The transition rule which is used for our particular CA setup is very simple: a node may only be activated (1) if there is at least one active node in its neighborhood and (2) if the edge connecting it to this node has not failed. This can be formulated as follows:
$s_i(t) = (s_p(t) \wedge e_i\,p(t)) \vee (s_q(t) \wedge e_iq(t)) \vee \ldots \vee (s_r(t) \wedge e_{ir}(t))$ with $p, q, \ldots, r \in N_i$ (17)

The neighborhood Ni of each node can be determined by using the adjacency and incidence tables representing the graph.

The algorithm then goes as follows:

1. step t = 0
2. set all node states to passive: $\forall$i : si(0) = 0
3. activate the source node: sS (0) = 1
4. step t = t + 1
5. update the node states according to rule 17
6. if sT (t) = 1 stop (a path has been found)
7. else if t < n − 1 go to 4
8. else sT (t) = 0 and no path has been found

[21] A similar approach is proposed for the assessment of the unreliability of complex networks in [18]

4.3 Monte Carlo (MC)

To determine which cut sets are minimal the following algorithm is used:

1. the candidate is compared to MCS of lower order already present in the archive of MCSs. If one of these cut sets is included in the sampled one, the counter associated to this cut set is incremented by one. Otherwise,
2. the candidate is compared to the cut sets of the same order in the archive to check if it is already present. If so the associated counter is incremented by one. Otherwise,
3. the candidate is added to the archive with its counter set to 1 and it is compared with higher order cut sets to verify if it is included in any of them, in which case they are deleted and the associated counter is added to the counter of the newly found.

Of course one can never be sure that the algorithm has been exhaustive in finding the MCSs, but even with a relatively low number of trials the most probable MCSs will be found.

If a MCS is not found, it is highly probable that it contains measure groups with high efficiencies. [22]

Let *MA* be the MCS with the smallest probability to be found during the Monte Carlo sampling. This means that *MA* has the highest efficiency of all MCSs. (Its value will be the closest to "1") And since the overall efficiency *E(p)* is calculated as a series structure of all MCS we can say that *MA* is the MCS that influences *E (p)* the least.

So the MCS that have not been identified by the Monte Carlo algorithm are the ones with the smallest influence on the overall efficiency.

Additionally, for the optimization itself the exact calculation of *E(p)* may not be required. One only needs to be able to compare different candidate solutions to each other and select the best.

4.4 Fussell-Vesely Importance values

We are not only able to identify the cut sets. The criticality of each edge can also be computed using the Fussell-Vesely importance measure. It is computed as the ratio between the number of occurred cut sets containing edge ij and the number of Monte Carlo trials performed.

[22] Since each MCS is a set of parallel measure groups we can also say that its efficiency is always higher than the highest efficiency of its components.

The higher the Fussell-Vesely importance value, the more critical the edge is. This can be due to:

- its efficiency: A low efficiency value, implies higher probabilities of being selected as a failed edge in the graph;
- its location in the graph: Generally the closer the edges are to the Source and or Target node, the more important they get and;
- the number of downstream edges connected to it: Generally the more edges that are connected, the more critical the edges becomes.

A good understanding about these effects for each particular measure group within the organizational graph structure will be of great value to security managers, because it will enable them to correctly prioritize among IT security investments.

5 CoCoViLa implementation

5.1 Introduction

CoCoViLa is an Artificial Intelligence software development platform. It synthesizes algorithms based on inputs from attribute declarations, bindings between attributes, attribute dependencies and goals using its declarative specification language. The realization of the dependencies are pure Java methods. More information about the tool can be found in [1, 7].
The CoCoViLa platform contains:

- a Class Editor for creating the domain-specific language, defining class properties and their visual representations;
- a Scheme Editor which allows users to:
  - visually specify computational problems by drawing objects/instantiating classes on schemes,
  - set values of object properties
  - define relations between object attributes,
  - make use of expert tables
- a synthesizer built into Scheme Editor for generating Java programs from schemes

5.2 The scheme

The scheme created in CoCoViLa is shown in Figure 5. Its components are:

- The security measure groups (vertically aligned purple boxes), containing the investment costs, maintenance costs and efficiencies for each level of implementation.
- the graph structure as explained in Section 2.

- the superclass (blue box), collecting all attribute values through the specification language's alias mechanism and containing all references to the "hidden"[23] classes containing the actual reliability calculations.
- the optimizer (green box) collecting all the optimization results and containing the GSM cost and efficiency functions plus a reference to a "hidden" class with a slightly modified Evolutionary Algorithm. (the original algorithm is described in [10])
- the security class (red box) containing all information about the security goals together with the losses calculation. (not covered in this paper)
- a graph2D object allowing the representation of our optimization results in a 2D graph.

The bindings between the different components represent their equality. They allow the exchange of values between classes.

The superclass also allows the selection of different efficiency levels. One can choose between the usage of:

- the current efficiency levels of the measure groups. This way the overall efficiency and importance values reflect the situation of the organization as it is.
- the efficiency levels of the measure groups as required by the security goals. This way the overall efficiency and importance values reflect the situation of the organization as management wants it to be.
- an average efficiency value which is the same for all measure groups. This can be used when
  - no other data is available or
  - when only the influence of location and number of connections need to be investigated
  - when all MCSs need to be found, not only the most probable ones.

[23] hidden in this context means: not visible for a normal user. More details can be found in the next section.

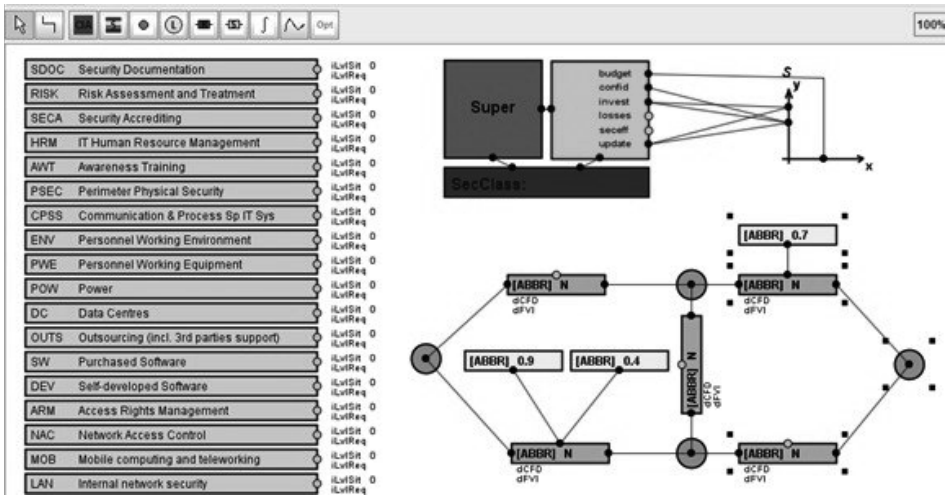


Figure 5: Representation of the optimization problem in a CoCoViLa scheme

6 Conclusion

A solid mathematical background has been added to the efficiency function $E(p)$ of the GSES. The definition of $E(p)$ evolved from a weighted average to a measure group relationship diagram and finally in this paper to a graph based reliability diagram with which the following benefits could be realized:

1. The efficiency of the measure groups is now quantifiable, unlike the previously defined weights and confidence levels. For this only the recording of the security incidents is needed.
2. CoCoViLa's user interface can be used to create the graph structure. No hardcoding of parameters is needed anymore.
3. The new approach is generic. A solution for all possible graphs can be found contrary to the measure group relationship diagram where parallel relevant measure groups, bridge- and star- topologies were not allowed.
4. The Fussell-Vesely importance values allow security managers to prioritize among IT security investments.
5. Threats and vulnerabilities don't need to be modeled, only incidents. Additionally MCSs reflect an attackers point of view with regard to the organization's security.
6. The influence of the security goals can be included by using the required measure group efficien- cies as an input
7. It is justified by a well-founded mathematical theory

Recording security incidents and finding out which measure groups have failed remain non-trivial tasks however. Also for each measure group needs to be defined

what would be considered as an incident. This definition will largely affect the measure group's failure rate. But if no statistical security incident information is available, the efficiencies can still be estimated by security experts as it was done before.

Important to know is that Monte Carlo sampling might not find all MCSs, but as stated in Section 4.3 this isn't absolutely necessary either.

The new version of the GSES should also be applied to an existing organization to verify if the predicted losses and efficiencies will correspond to the real losses and efficiency values.

References

[1] CoCoViLa: Model-based software development p latform.

[2] Cyberprotect, version 1.1, July 1999.

[3] J.L. Duffany. Optimal resource allocation for securing an enterprise information infrastructure. In Association for Computing Machinery, editor, LANC7, pages 35–42. Association for Computing Machinery, October 2007.

[4] J.L. Duffany. Exploring security countermeasures along the attack sequence. In ISA8, pages 427–432. IEEE, April 2008.

[5] I. Flechais, C. Mascolo, and M.A. Sasse. Integrating security and usability into the requirements and design proces. International Journal of Electronic Security and Digital Forensics, 1(1):12–26, 2007.

[6] P.L. Gordon and P.M. Loeb. The economics of information security investments. ACM Transactions on Information Security, 5(4):438–457, 2002.

[7] Pavel Grigorenko. Higher Order Attribute Semantics of Flat Languages. PhD thesis, Institute of Cybernetics at Tallinn University of Technology, Akadeemia Tee 21, 12618 Tallinn, November 2010.

[8] ISACA. The Business Model for Information Security. ISACA, September 2010.

[9] R.L. Jones and A. Rastogi. Secure coding: building security into the software development lifecycle. Infor- mation Systems Security, 13(5):29–39, 2004.

[10] Toomas Kirt and Jyri Kivimaa. Optimizing IT security costs by evolutionary algorithms. In C.Czosseck and K. Podins, editors, CCDCOE, pages 145–160. Cooperative Cyber Defence Centre Of Excellence, June 2010.

[11] Jyri Kivimaa. Applying a cost optimizing model for IT security. In Henrique Santos, editor, ECIW, pages 142–153. UK Reading, Academic Publishing Limited, July 2009.

[12] Jyri Kivimaa and Toomas Kirt. Evolutionary algorithms for optimal selection of security measures. In Rain Ottis Demergis, editor, ECIW,

pages 172–184. Readings, UK, Academic Publishing Limited, July 2011.

[13] Jyri Kivimaa, Andres Ojamaa, and Enn Tyugu. Graded security expert system. In Roberto Setola and Stefan Geretshuber, editors, CRITIS, volume 5508 of Lecture Notes in Computer Science, pages 279–286. Springer, October 2008.

[14] Jyri Kivimaa, Andres Ojamaa, and Enn Tyugu. Pareto-optimal situation analysis for selection of security measures. In MILCOM, pages 3224–3230, November 2008.

[15] Jyri Kivimaa, Andres Ojamaa, and Enn Tyugu. Managing evolving security situations. In MILCOM, pages 1–7. Institute of Electrical and Electronics Engineers(IEEE), October 2009.

[16] Marvin Rausand and Arnljot Hoyland. System Reliability Theory, Models, Statistical Methods and Applica- tions. John Wiley & Sons, 2 edition, 2004.

[17] U.S. Department of Defense, Defense Security Service. National Industrial Security Program Operating Manual, February 2006.

[18] E. Zio, M. Librizzi, and G. Sansavini. Determining the minimal cut sets and Fussell-Vesely importance measures in binary networks by simulation. In Guedes Soares & Zio, editor, ESREL, volume 1, pages 723–729. Taylor & Francis Group, London, October 2008.

CONCLUSIONS

The growing trend is to provide managers with ISs that can assist them in their most important task – making decisions. Unlike the deterministic decision-making process, in the IT security we act under uncertainty of the variables and the future progress that are often difficult to predict, measure (or just are not measured) and control. And Decision Support Systems (DSSs) precisely deal with problems that arise when managers in organizations are faced with decisions where some aspects of a task or procedure are not known (or there is no certainty that all aspects are known). [24]

DSSs are especially valuable in situations in which the amount of available information is prohibitive for the intuition of an unaided human decision maker and in which precision and optimality are of importance. DSSs do not replace humans but rather augment their limited capacity to deal with complex problems, whether the ultimate quality of decisions will be higher than that of an unaided decision maker.[25]

In conclusion, we can say that the graph-based GSM/GSES method, which has been described in this thesis, is a reasonably good solution to the problem of information security cost optimization. The decision to base the method on the IT view of information security and to use the People-Process-Technology approach has proven justified. Gb_GSM/GSES method is a particularly suitable tool/utility for multilevel IT security standards and models such as NISPOM 2006, NIST SP 800-53 r4 or ISKE v6.0.

The gb_GSM/GSES-method creates new opportunities to reduce the gap between IT Security experts and organization management. It is now easier for experts to explain and justify the necessary information security costs in a way that is understandable to management. At the same time, management can now require well-reasoned explanations from the experts.

The gb_GSM/GSES-method provide us the optimal IT security expenses value and the corresponding optimal security measures list, and it can be done with an acceptable workload (labor cost) for SMEs.

Our expectation is that more expert knowledge and statistical data will be collected when interactive analysis applications with a graphical user interface, such as the prototype presented in this thesis, become available. As more expert information is collected, the required work for follow-on optimization decreases.

---

[24] http://home.ubalt.edu/ntsbarsh/opre640a/partix.htm↑

[25] http://www.pitt.edu/~druzdzel/psfiles/dss.pdf

It would be very useful, if companies would publish concrete and comparable Cost and Effectiveness values of their IT and information security solutions. This would considerably simplify expert information collection, as well as its accuracy. Unfortunately, such good analyses are currently only available for sale – and they are very expensive.[26]

If an information security standard or Best Practice does not systematically cover (meaning, in more detail than the ROI of single information security activities or technical solutions) a topic as important as the rationality and optimality of information security costs, then this standard or Best Practice is inadequate and not up-to-date.

Finally, GSES is also suitable for any business process optimization, if:

- it is able to describe a corresponding graph-based business process model and
- the sub-processes can be described and implemented with grades, and if their costs and effectiveness values can be defined.

SUMMARY OF THE FINDINGS

Theoretical development

1. In II.1 we have described a developed graph-based GSM, which is suitable for information security cost optimization, based on two ideas:
   - widely used People-Process-Technology Business Modeland
   - relevant and Rc-redundant supporting IT and IT security activities.
     In order to describe IT ja IT Sec as a process, there are two very important basic ideas:
   - Information security is like a chain, where the weakest link determines the strength of the whole chain. This means that information security is at a good level, if all of its relevant activities are at a good level. According to expert assessments from the banking sector, Losses will start to decrease considerably, when the Effectiveness of information security (as a process) is ~0,9.
   - The principles of multilevel security and defense in depth – meaning parallel supporting activities to relevant activities.

2. We described a security metric that is suitable for information security cost optimization:
   - In II.2 the three stages to the final optimality and the information that is needed for it:

[26] For example, www.nsslabs.com – NSS Labs_NGFW_SVM_2012.pdf, Next Generation Firewall Security Value Map, Price: $3,500.00.

  - Do rational things – to achieve the required security level and not secure more (wasting money) and not less (too many security incidents – i.e. security losses will be too high) than needed. The required work is approximately 1-2 days.
  - Do things right – use resources optimally, i.e. maximal security effectiveness for the security system with our existing resources (time, experts, money). The required work is approximately 1-2 weeks.
  - Do right things – global optimum - find minimal Total Costs to IT Security, i.e. the sum of security investment/maintenance costs and security losses must be minimal. The required work is approximately 1-2 months.
- In II.3 Security Effectiveness calculations formula for parallel IT security specific situation: relevant and Rc-supporting IT and IT security activities.

3. In II.4 – II.6 we described the algorithms and methods for optimization (gb_GSM/GSES-method) for a CoCoViLa based expert system - the optimization process is computationally very laborious, so we had to describe an Evolutionary Algorithms and Pareto-frontier based optimization method, in order to implement our bi-dimensional optimization (maximal effectiveness with minimal costs).

In summary, we can say that the main goal of this work is completed:

- **we have developed an IT security costs optimization gb_GSM/GSES-method based on the graph-based Graded Security Model and on the Graded Security Expert System,**
- **and the gb_GSM/GSES-method has been successfully tested by the Banking Case Study.**

PROPOSALS FOR FURTHER RESEARCH

Proposals for developing the model:

1. Rc is a function, not a constant. The redundancy coefficient Rc is a function of its own (i.e. supported measure group) - and supported by it relevant measure group levels.
2. Security variations that differ from the (calculated) optimum by only 1-2% should be researched in more detail. Since the expert assessments have an error of approximately ± 20%, then there exists a possibility that the real optimum is actually one of the other security profiles that are close to the currently calculated optimum. Therefore, we refer to these near-optimal security profiles as possible alternatives to optimal security profiles. The calculation of an alternative optimal security profiles (meaning very close to the calculated optimum security profile, and therefore possible candidates for the real optimum too) should be included in the model.
3. The current model considers the institutions' ISs as a single Integrated IS.

A multi-ISs version of the model could be developed, which would consider the institutions' information security at the level of all individual information systems.

4. A quite interesting issue for further investigation (topic started in S_IV) is the efficiency of the security situation, which has been achieved by lowering the security level of some security activity for some reason. In general it is a waste of money, because the level of spending for achieving a security level is on average at least 2-3 times higher than the level of the corresponding maintenance expenditures. But some changes in the company or in the company's plans are always possible, and these changes may result in the need for changes as lowering the security level of some security activities in IT security also.

Future Directions for Research:

1. The probability of attacks should be included, but we do not have such information in reality. We just assume that unprotected valuable information will be attacked. However, the question about attack types and probabilities for concrete institution merits further research.
   A new element, such as Duffany's attack tree, could be included in the model, but this would likely make the model considerably more complicated and increase the required work load.
2. Researching the options for quantifying Losses (in monetary terms) in the public sector (including the military).
   Losses from security incidents should be included, but in reality, we do not have that information for the public (military) sector at this time. The problem is to find a suitable model and calculation method, such as a public interest as a value for public services or something like.
3. Enterprise level CD/CS cost optimization model. There are no fundamental problems, mainly is needed to add to model some additional security goals (such as non-repudiation, authenticity, resilience, mission criticality).
4. Enterprise level IT Cost Optimization model. Since the current graph-model describes all information processing as one business process in the enterprise, then it should be possible to take a step forward and adapt the model and the expert system for optimizing all IT costs. Therefore, the next step would be to research "IT cost optimization", since it is very difficult to differentiate between IT costs and IT security costs. Basically, a model like COBIT should be used as a good example (COBIT is model and Best Practice about IT management). And again – no fundamental problems, necessary is only to introduce some additional IT goals (such as IT Efficiency, IT Effectiveness, Compliance).

Developing solutions that will make implementing the model easier:

1. Achieve our models information integration with widely used business and accounting software.
2. Include the principles of amortization into the expert system. For example, every five years a new Investment may be required for an activity and that

year is likely different for different activities. This should be introduced to the expert system.

3. The current model assumes that IT is not the primary service or product of the company. IT as a product or service would likely introduce many additional economic variables.

NB! All of the proposals above would require corresponding changes in the GSES software.

REFERENCES

2.0 GOVERNANCE. http://oimt.hawaii.gov/wp-content/uploads/2012/09/Governance_2.0.pdf (Accessed: 1. March 2013).

Alberghs, Geert; Grigorenko, Pavel; Kivimaa, Jyri. 2011. *Quantitative system reliability approach for optimizing IT security costs in an AI environment.* In: 12th Symposium on Programming Languages and Software Tools, SPLST'11 : Tallinn, Estonia, 5-7 October 2011, Proceedings. Tallinn: TUT Press, 2011, 219 - 230.

AS SEB Pank. Aastaaruanne 2009. http://www.seb.ee/files/aruanded/SEB_Pank_Aastaaruanne2009.pdf (Accessed: 1. March 2013).

BSI IT-Grundschutz. 2011. (Bundesamt für Sicherheit in der Informationstechnik, German Federal Office for Information Security) - the most up-to-date-version in german is: IT-Grundschutz Catalogues - IT-Grundschutz-Kataloge,12. Ergänzungslieferung - September 2011, http://www.bsi.bund.de/grundschutz (Accessed: 1. March 2013).

BSI IT Baseline Protection Manual. 2005. (Bundesamt für Sicherheit in der Informationstechnik, German Federal Office for Information Security) - the english version of the IT-Grundschutz Catalogues is available in the pdf-format: IT-Grundschutz Catalogues 2005 (23,5 MB), https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Grundschutz/download /it-grundschutz-kataloge_2005_pdf_en_zip.zip?__blob=publicationFile (Accessed: 1. March 2013).

Buldas, A., Laud, P., Priisalu, J., Saarepera, M., and Villemson, J. 2006.

Rational Choice of Security Measures via Multi-Parameter Attack Trees.
CRITIS'06, August 30 – September 2, 2006, Samos Island, Greece.

http://uuslepo.it.da.ut.ee/~peeter_l/research/attacks11.pdf (Accessed: 1. March 2013).

Carr, **N**icholas G. May 2003. *IT doesn't matter.* Harvard Busines Review. http://nofieiman.com/wp-content/lectures/MIS-IT-doesnt-matter.pdf (Accessed: 1. March 2013).

CASE STUDIES: Research Methods. Centre for Excellence in Learning and Teaching, The University of Melbourne, Faculty of Business and Economics. http://fbe.unimelb.edu.au/__data/assets/pdf_file/0005/647609/Casestudy_Research.pdf (Accessed: 1. March 2013).

Common Criteria (CC) for IT Security Evaluation (ISO/IEC 15408). http://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R1.pdf (Accessed: 1. March 2013).

COBIT (IT Governance Institute). http://www.itgi.org/ (Accessed: 1. March 2013).
CoCoViLa: Model-based software development platform - a compiler compiler for visual languages. http://www.cs.ioc.ee/cocovila/ (Accessed: 1. March 2013).
CRAMM. http://www.cramm.com/ (Accessed: 1. March 2013).

CRAMM (CCTA Risk Analysis and Management Method). A Qualitative Risk Analysis and Management Tool. http://www.sans.org/reading_room/whitepapers/auditing/ qualitative- risk-analysis-management-tool-cramm_83 (Accessed: 1. March 2013).

Grigorenko, P., Saabas, A., Tyugu, E. Visual tool for generative programming. 2005. ACM SIGSOFT Software Engineering Notes, 2005.5, 249-252.

Dewri, R., Poolsappasit, N., Ray, I., and Whitley, D. 2007. *Optimal Security Hardening Using Multi-objective Optimization on Attack Tree Models of Networks*. Department of Computer Science Colorado State University. http://www.cs.colostate.edu/~iray/research/ccs07.pdf

Guth, Paul. August 2011. Reliability vs Availability and the Magic of MTTR. http://constructolution.wordpress.com/2011/08/07/reliability-availability/ (Accessed: 1. March 2013).

CyberProtect ver 2.0. *Information Assurance Training & Awareness*. March 2010, US DoD. Defense Information Systems Agency, Information Assurance Education, Training and Awareness. [Online]. http://iase.disa.mil/eta/product_description.pdf (Accessed: 1. March 2013).

DOE (1999) Classified Information Systems Security Manual. https://www.directives.doe.gov/directives/archive-directives/471.2-DManual-2/at_download/file (Retrieved February 1, 2010).

Duffany J.L. 2007. *Optimal resource allocation for securing an enterprise information infrastructure.* Proceedings of the 4th International Latin American Networking Conference, San Jose, Costa Rica, 2007, pp. 35-42. http://dl.acm.org/citation.cfm?id=1384117&picked= =prox&CFID=287470139&CFTOKEN=33958854 (Accessed: 1. March 2013).

EAR/Pilar Magerit. http://www.ar-tools.com (download EAR), http://www.ccn-cert.cni.es (download PILAR).

EBIOS. Expression des Besoins et Identification des Objectifs de Sécurité. http://www.club-ebios.org/site/.

Estonian Information Systems Three-Level Security Baseline System – ISKE ver. 6.0. http://www.ria.ee/iske (Accessed: 1. March 2013).

Gordon, Lawrence P. and Loeb, Martin P. *The Economics of Information Security Investment.* ACM Transactions on Information Systems Security, November 2002, ppg 438-457. http://ns1.geoip.clamav.net/~mfelegyhazi/courses/BMEVIHIAV15/readings/04_GordonL02economics_security_investment.pdf (Accessed: 1. March 2013).

P. Grigorenko, A. Saabas, E. Tyugu. Visual tool for generative programming. ACM SIGSOFT Software Engineering Notes, 2005, 30, 5, 249–252.

Grossklags, J., Christin, N., and Chuang, J. 2008.
Secure or Insure? A Game-Theoretic Analysis of Information Security Games.
http://www.andrew.cmu.edu/user/nicolasc/publications/GCC-WWW08.pdf
(Accessed: 1. March 2013).

Herzog, A., Shahmehri, N., Duma, C. 2007. *An Ontology of Information Security*

HP Enterprise Security. 2011. *Safeguarding Against a World of Threats.*
A SearchCompliance.com E-Book. http://www.bitpipe.com/data/demandEngage.action?resId=1336488652_824 (Accessed: 1. March 2013).

ISACA. 2010. *The Business Model for Information Security*. http://www.isaca.org/Knowledge-Center/BMIS/Documents/BMIS-22Sept2010-Research.pdf?id=2ef87cf9-8400-4b4f-872e-5b9e0f6269bc (Accessed: 1. March 2013).

ISF. Information Security Forum. (Online). https://www.securityforum.org/downloadresearch/publicdownload2011sogp/ (Accessed: 1. March 2013).

ISKE. Information Systems 3-level baseline information security v6.00. Estonia. https://ria.ee/iske/ (Accessed: 1. March 2013).

ISM (Information Security Management Maturity Model) v2. http://www.lean.org/FuseTalk/Forum/Attachments/ISM3_v2.00.pdf (Accessed: 1. March 2013).

ISO 13569. *INTERNATIONAL STANDARD ISO/TR 13569:2005 Financial services. Information security guidelines.* http://shop.bsigroup.com/ProductDetail/?pid=000000000030112577
(Accessed: 1. March 2013).

ISO 27000, *INTERNATIONAL STANDARD ISO/IEC 27000 Information technology — Security techniques — Information security management systems — Overview and vocabulary,* http://www.iso27001security.com/html/27000.html (Accessed: 1. March 2013).

ISO 27001. *INTERNATIONAL STANDARD ISO/IEC 27001:2005 Information technology — Security techniques — Information security management systems — Requirements*. http://www.iso27001security.com/html/27001.html (Accessed: 1. March 2013).

ISO 27002. *INTERNATIONAL STANDARD ISO/IEC 27002:2008 Information technology — Security techniques — Code of practice for information security management*. http://www.iso27001security.com/html/27002.html (Accessed: 1. March 2013).

ITIL®– IT Infrastructure Library® for IT Service Management (ITSM) - ITIL 2011. http://www.itgovernance.co.uk/itil.aspx (Accessed: 1. March 2013).

Jürgenson, A., and Willemson, J. *Processing Multi-Parameter Attack trees with Estimated Parameter Values.* IWSEC'07 Proceedings of the Security 2nd international conference on Advances in information and computer security. Pages 308-319.
http://dl.acm.org/citation.cfm?id=1778930

Kirt, T., and Kivimaa, J. (2010) "Optimizing IT security costs by evolutionary algorithms", in C.Czosseck, and K. Podins, (Eds.), Conference on Cyber Conflict Proceedings 2010, Tallinn, Estonia,Cooperative Cyber Defence Centre of Excellence Publications, pp. 145–160

Kivimaa J., Kirt T. (2011) "Evolutionary Algorithms for Optimal Selection of Security Measures," in *Proceedings of the 10th European Conference on Information Warfare and Security*, Tallinn, Estonia, 2011, pp. 172-184

Kivimaa, Jüri; Ojamaa, Andres; Tyugu, Enn (2009). Graded security expert system. In: Critical Information Infrastructures Security : Third International Workshop, CRITIS 2008, Rome, Italy, October 13-15, 2008, Revised Papers. Berlin: Springer, 2009, (Lecture Notes in Computer Science; 5508), 279 – 286

Kivimaa, J. (2009) "Applying a costs optimizing model for IT security", in H. Santos (Ed.), Proceedings of the 8th European Conference on Information Warfare and Security, Reading, UK, Academic Publishing Limited, pp. 142–153.

Kotkas, Vahur; Ojamaa, Andres; Grigorenko, Pavel; Maigre, Riina; Harf, Mait; Tyugu, Enn. 2011. *CoCoViLa as a multifunctional simulation platform.* In: Proceedings of the 4th International ICST Conference on Simulation Tools and

Techniques : 21-25 March 2011, Barcelona, Spain, SIMUTools 2011: Brussels: ICST, 2011, 195 - 205.

Librizzi M., Sansavini G. and Zio E. 2006. *Determining the Minimal Cut Sets and Fussell-Vesely importance measures in binary networks by simulation.* Safety and Reliability for Managing Risk, vol. 1, pp. 723-729.

Moore, David. 2005. Do Things Right Or Do The Rights Things? http://therulesofwealth.wordpress.com/2008/08/18/do-things-right-or-do-the-right-things/ (Accessed: 1. March 2013).

NISPOM 2006. *National Industrial Security Program Operating Manual.* DoD 5220.22-M. USA**.** http://www.dss.mil/isp/odaa/nispom06.html (Accessed: 1. March 2013).

NIST sp800-53. *Final Public Draft of NIST Special Publication 800-53 Revision 4.* http://csrc.nist.gov/publications/drafts/800-53-rev4/sp800_53_r4_draft_fpd.pdf (Accessed: 1. March 2013).

Ojamaa, A., Tyugu, E., & Kivimaa, J. (2008). Pareto-optimal situation analysis for selection of security measures. In Military Communications Conference MILCOM 2008, San Diego (USA): Unclassified Proceedings (pp. 3224–3230). Piscataway, NJ: IEEE.

Ojamaa A., Tyugu E. Kivimaa J., "Managing evolving security situations," in *MILCOM 2009 : Unclassified Proceedings*, Boston, Massachusetts, USA, 2009, pp. 1-7.

Olovsson T. 1992. *A structured Approach to Computer Security*. Gothenburg. Sweden.

PCI Security Standards Council. *Payment Card Industry Data Security Standard v2.0.* October 2010 [Online]. https://www.pcisecuritystandards.org/documents/pci_dss_v2.pdf (Accessed: 1. March 2013).

Schlicher, Bob G., and Abercrombie, Robert K.
Information Security Analysis Using Game Theory and Simulation. Oak Ridge National Laboratory, Oak Ridge, TN 37831 USA.
http://info.ornl.gov/sites/publications/Files/Pub37664.pdf (Accessed: 1. March 2013).

SSM CMM (Systems Security Engineering Capability Maturity Model v3) http://www.faa.gov/about/office_org/headquarters_offices/aio/library/media/SafetyandSecurityExt-FINAL-web.pdf (Accessed: 1. March 2013).

System Reliability and Availability, http://www.eventhelix.com/realtimemantra/faulthandling/system_reliability_availability.htm#.UU4fhLUijb1 (Accessed: 1. March 2013)

TISN for CIP (Trusted Information Sharing Network for Critical Infrastructure Protection). 2008. *Defense in Depth.* http://www.tisn.gov.au/Documents/SIFTD-I-D+-+Full+-++15+Oct+2008+-+1.pdf (Accessed: 1. March 2013).

US DoD. (2006, February) U.S. Department of Defense, Defense Security Service. [Online]. http://www.dss.mil/isp/odaa/nispom06.html

US DoD. (2010, March) Defense Information Systems Agency, Information Assurance Education, Training and Awareness. [Online]. http://iase.disa.mil/eta/product_description.pdf

Wikipedia Community. Wikipedia. [Online]. http://en.wikipedia.org/wiki/Main_Page.

Yin, Robert K. 1994. *Case Study Research: Design and Methods (*2nd ed.) http://www.communique.utwente.nl/wp-content/uploads/2010/05/intervenq-samenvatting-yin.doc (Accessed: 1. March 2013).

APPENDICIES

Appendix 1. The Case Study to test the GSM/GSES method

The tool can be used for all organizations where valuable information needs to be safeguarded by implementing a certain level of information security. In principle, this is a report from the front lines of information security. During peace time, the banks are one of the most interesting targets for hackers, since there is a direct access to money. The model has been applied in two major banks in Estonia: Swedbank Baltic and SEB Estonia.

The goal is to show that it is possible to collect the necessary and real information for the model with an acceptable work load. For example, in models that are based on detailed risk analysis, game theory or attack trees, this information collection process is much more problematic, and sometimes practically impossible.

We have used the *a posteriori* method, where we observe, calculate the possible outcomes and then decide. The main idea is that if we can develop a model that can describe the information security in an enterprise for the previous year, then it is probably usable for optimizing the information security costs of the following year(s). Obviously, the model will also have to take into account the changes that take place in the enterprise IT, as well as general in IT and information security.

The implementation of the Graded Security Expert System in a specific organization requires several tasks to be performed:

1. Collect expert data about Cost (Appendix 3, 4, and 5.) and Effectiveness values of (Appendix 6) security measure groups, and Losses (Appendix 7) from security incidents.
2. Cost optimization with GSES.
3. Analyze results and Compare with previous year's real IT security situation.

The model requires that updates in expert information and/or even changes in IT and IT security Business Process graph can be collected. Once new information is available, a new cycle begins: Expert Data_2 -> GSM and GSES_calculations 2 -> Analyze_2. And (if needed) so on.

Tasks performed in the Case Study:

1. Define the information assets (mainly business IS's) that need to be protected.
2. Describe the GSM for the institution:
    - Security goals – basic CIA? or more?
    - Security goals levels – 4? or more? or less?
    - Security activity areas – 40? or less? or more?
    - Identification and description of the security measure groups that are required to achieve a specific security level.

- Define the possible security levels for each security measure group, that are mandatory for achieving the organization's security goals.
- Dependency Matrix – based on NISPOM 2006 or Banking Matrix examples.
- Graph–based GSM based on Banking GSM-graph example.
- Collect Cost and Effectiveness values for all security areas and their levels.
- Specify Redundancy Coefficients for supporting areas.

3. Estimate the possible losses from possible security incidents.

We received the information on Losses from the business side questionnaires (see Appendix 7. Risk assessment for Business Infosystems for 2009.). However, there are also some official numbers:

The actual information security situation of SEB Estonia according to the Estonian Financial Supervision Authority report "*AS SEB Pank. Aastaaruanne 2009*" (see Table 11):

Data from end of 2008	Value (kEEK)	Value (k€)
Total assets. (Value of the bank)	74 400 000	4 770 000
IT costs	63 900*	4 100*
Operational risks. Total losses (gross)	95 500	6 120

•

- Physical Security, Environmental costs, Risk Assessment, Security Audit, Security Accrediting, and Business Continuity Management costs in SEB Estonia are not included (at least fully) in the IT Budget – i.e. clarification and adjustment is needed. This value is now significantly lower than the one used in the model (it is IT Budgeting specifics in SEB Estonia, quite important part of IT costs are covered from the Budget of Administrative Department).

Table 11. Data from SEB Estonia's report to the Financial Supervision Authority.

The max risk of the Bank = value of the Bank.

On the other hand, we are not aware of any cases of Banks going bankrupt due to IT security incidents. In reality, bankruptcies are practically always due to business mistakes (bankruptcies are caused by business risks). Therefore, we would divide business risks and IT risks with the ratio of 90/10 of maximum Risk.

For SEB Estonia: total Loss was 95,5 million EEK and the value of SEB Estonia (as reported to the Financial Supervision Authority) was 74,4 billion EEK. 10% of the value of SEB Estonia would be 7,44 billion and that would be equal to the

Annual Risk from IT.
I.e. that in reality IT security mitigation Rate for SEB Estonia in 2009 was:
mR=(0,1*74400000000)/95500000=78.

4. And GSES will do the rest – i.e. based on calculations on collected information we can:

- Calculate the Losses curve *Losses=f(Budget)* or *Losses=f(Effectiveness)* (see Figure 17):

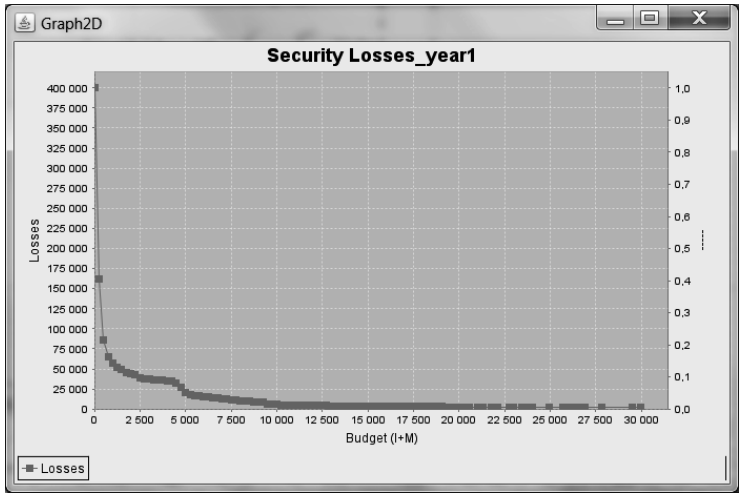


Figure 17. *The "function" Losses = f(Budget)*

- find the Global Optimum for IT security Costs – i.e. we can easily find minimal IT security Total Costs from *TotalCosts=f(Budget)*-curve (see Figures 18 – 20):

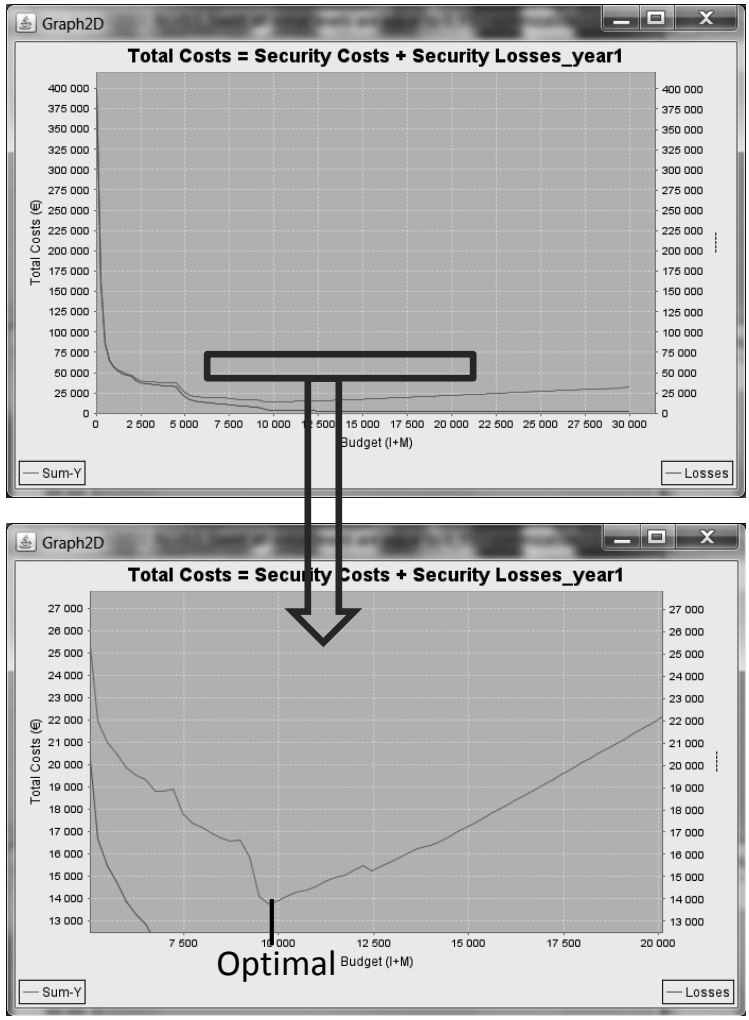


Figure 18. *The "function" Total Costs = f(Budget)* for first year.

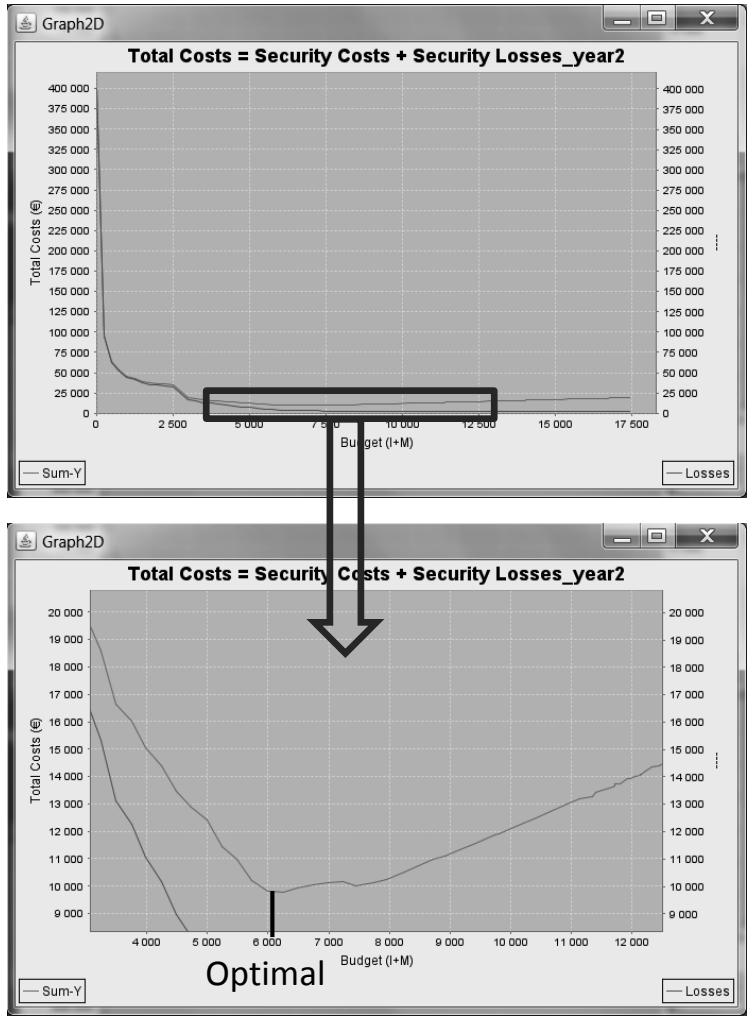


Figure 19. *The "function" Total Costs = f(Budget)* for second year.

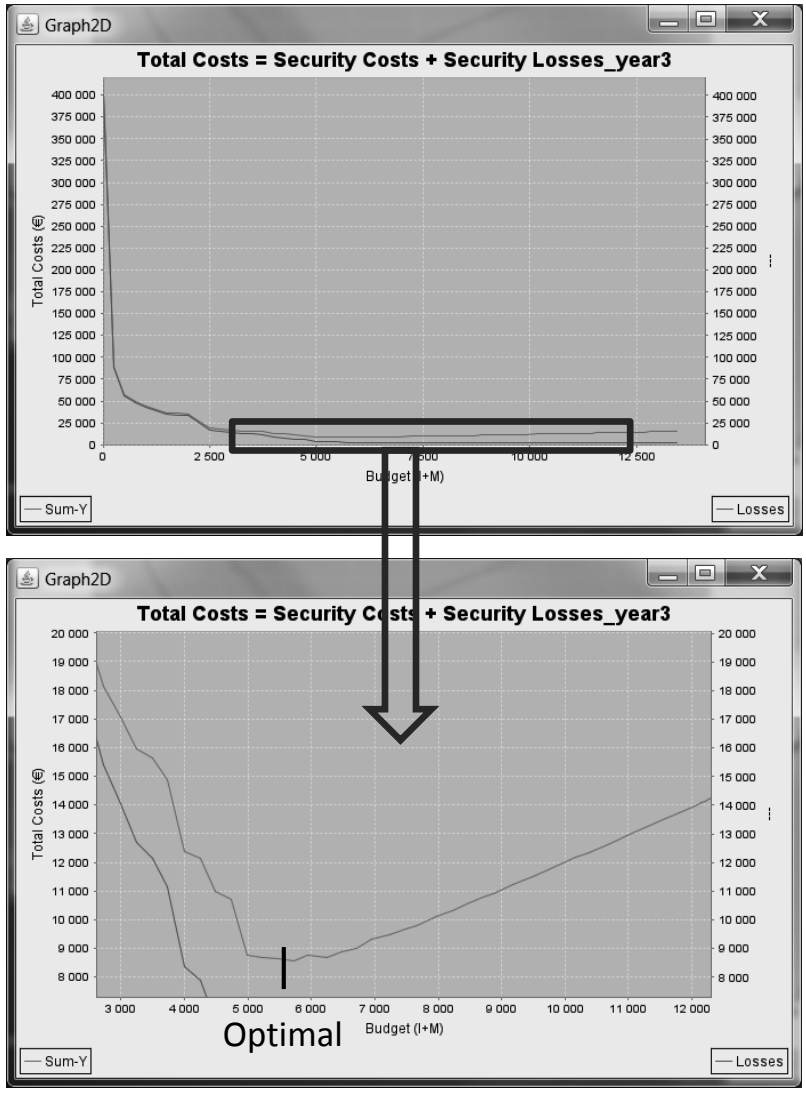


Figure 20. *The "function" Total Costs = f(Budget)* for third year.

	Optimal Budget	Total Cost = Losses + Opt Budget	Effective-ness	mitigation Rate
First year	9 750 000 €	13 750 000 €	0,9935	153
Second year	6 250 000 €	9 750 000 €	0,9943	174
Third year	5 750 000 €	8 750 000	0,9953	213
Fourth year	5 250 000 €	8 050 000 €	0,9954	217
Fifth year	5 200 000 €	7 950 000 €	0,9955	222

Table 12. The Global Optimums for the first five years (Figures 18-20).

We have found a very interesting result (Table 12) – the optimal costs for the second year are approximately 75% of the optimal costs of the first year, and approximately 50% of the first year for every following year. This is not totally new – for example, in CyberProtect the same principle is used, although in there the costs drop to 50% starting from the second year.

However, the amortization period is roughly 5 years in IT – meaning that every fiver years on has to basically start from the beginning.

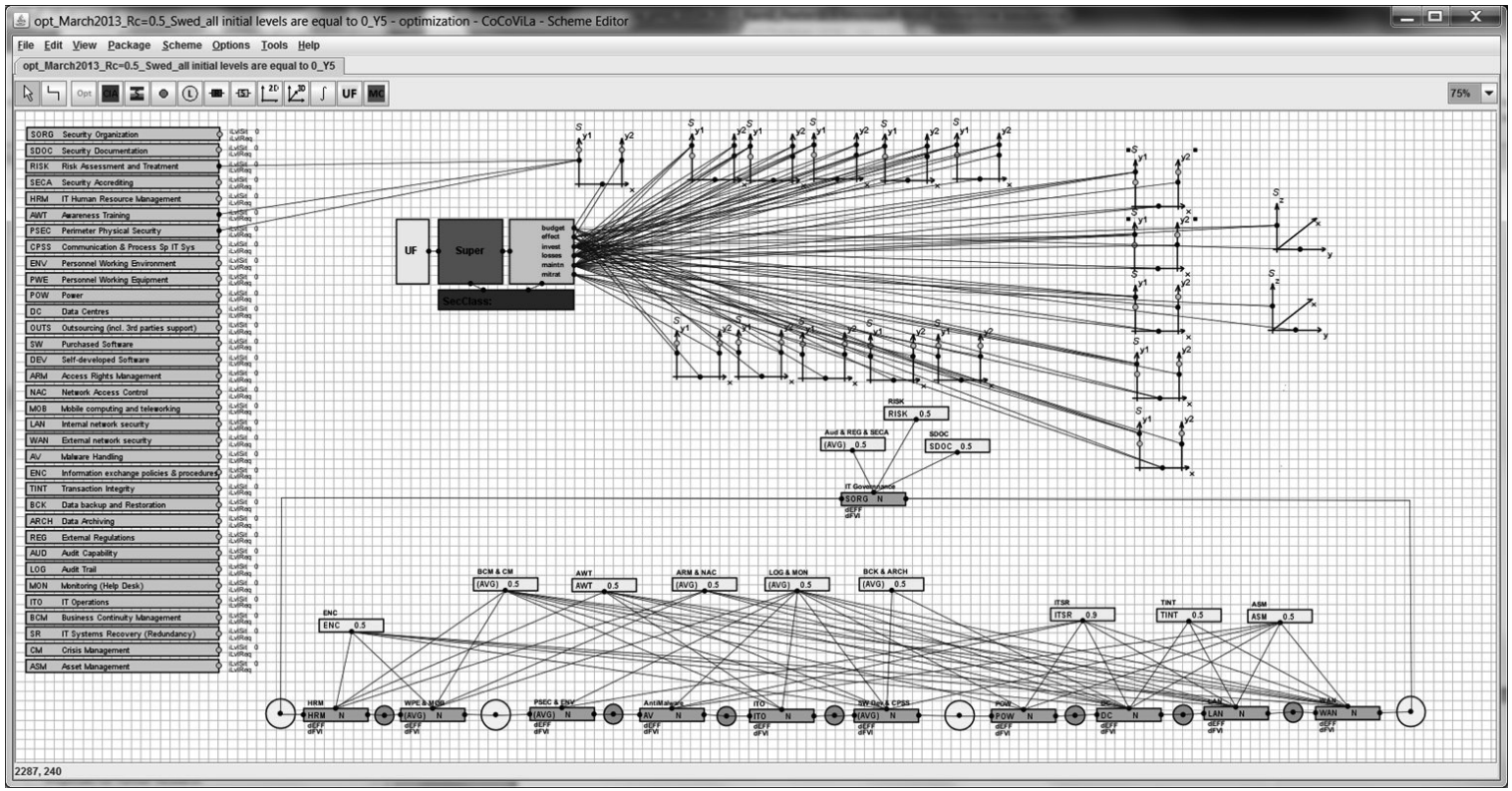


Figure 21. GSES - window for visual description of the optimization task (Case Study for Bank).

- from Java-console we specified corresponding optimal security profiles for first five years:

	Year 1.	Year 2.	Year 3.	Year 4.	Year 5.
G1: Organization of information security					
Security Documentation	4	4	4	4	4
Risk Assessment and Treatment	4	4	4	4	4
Security Accrediting	4	4	4	4	4
G2: Human resources security					
IT Human Resource Management	4	4	4	4	4
Awareness Training	5	5	5	5	5
G3: Physical and environmental security					
Perimeter Physical Security	4	4	4	4	4
Communication & Process Support IT Systems (Basic SW)	0	0	0	0	0
Personnel Working Environment(Physical Security)	5	4	4	5	5
Personnel Workplace Equipment	2	2	2	2	2
Power	2	2	1	2	2
Data Centres	1	1	1	1	1
G4: Information systems acquisition, development and maintenance					
Outsourcing (incl. 3rd parties support)	1	2	2	3	3
Purchased SoftWare	0	0	0	0	0
Self-developed SoftWare	0	0	0	0	1
G5: Access control					
Access Rights Management	3	3	3	3	3
Network Access Control	4	4	4	4	4
Mobile computing and teleworking	5	5	5	5	5
G6: Communications and operations management (ISO 17799)					
Internal network security	3	1	4	5	5
External network security (incl. PerimProt, IDS/IPS, ...)	2	0	0	2	2
Malware Handling	4	4	4	4	4
Encryption/Information exchange policies and procedures	5	5	5	5	5
Transaction Integrity	1	1	1	1	1
Data backup and Restoration	1	2	3	2	3
Data Archiving	4	4	4	4	4
G7: Compliance					
External Regulations	2	2	2	2	2
Audit Capability	4	4	4	4	4
G8: Information security incident management					
Audit Trail	4	4	4	4	4
Monitoring(Help Desk)	4	4	4	4	4
IT Operations	3	3	4	4	4
G9: Business continuity management					
Business Continuity Management(main input to ITS R)	4	4	4	4	4
IT Systems Recovery (Redundancy, etc)	4	5	5	5	5
Crisis Management	4	4	4	4	4
G10: Asset Management					
Asset Management	5	5	5	5	5

Table 13. Optimal security profile for years 1 ÷ 5.

- We can easily find optimal IT security Effectiveness from TotalCosts=f(Budget)-curve:

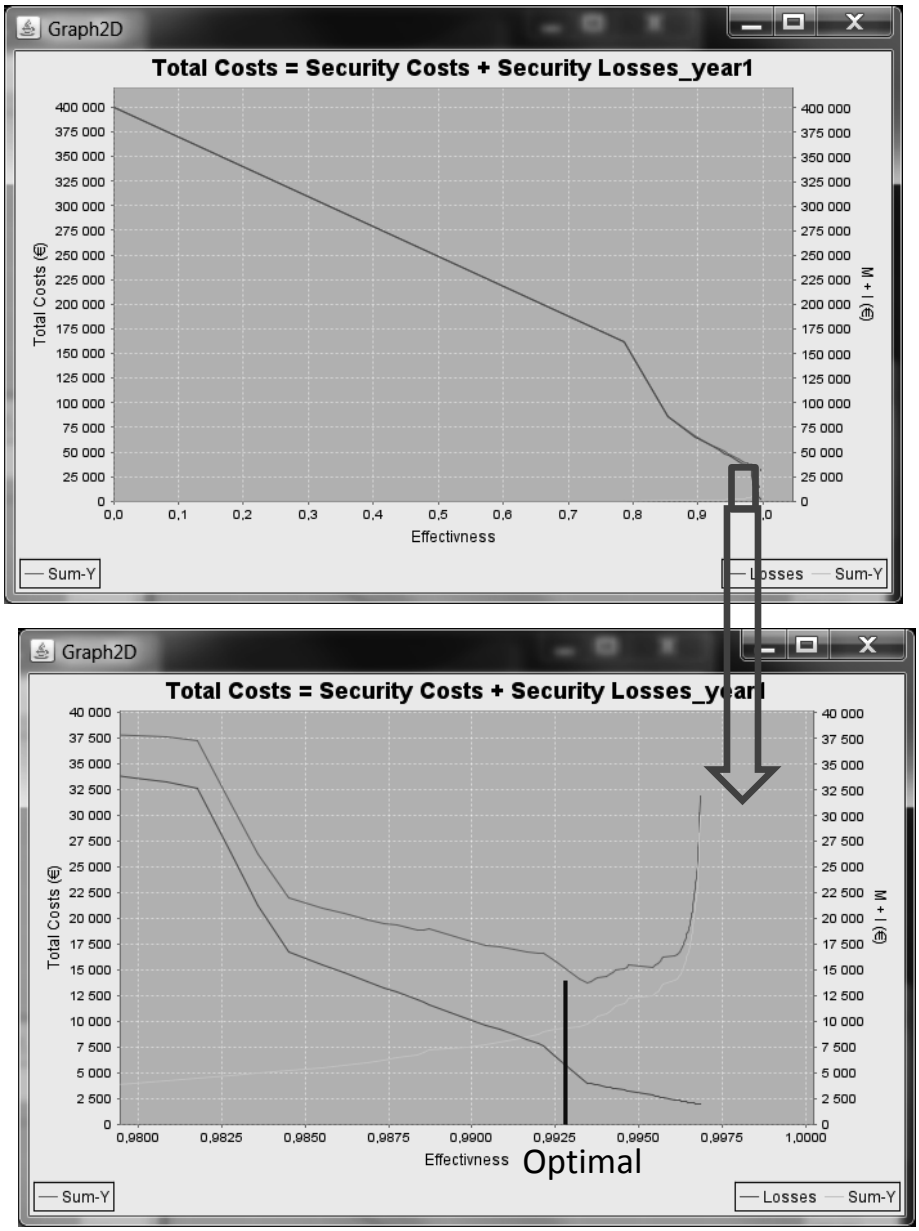


Figure 22. *The "function" Total Costs = f(Effectiveness)* for first year.

- We can also specify the optimal mitigation Rate from *mR=f(Budget)*-curve on Optimal Budget value (9750€ from Figure 18) X-axis point (Figure 23) or just calculate: $mR_{optimal}=1/(1-E_{optimal})$, where $E_{optimal}$, for example, we found from the curve *TotalCosts=f(Effectiveness)* on Figure 22.

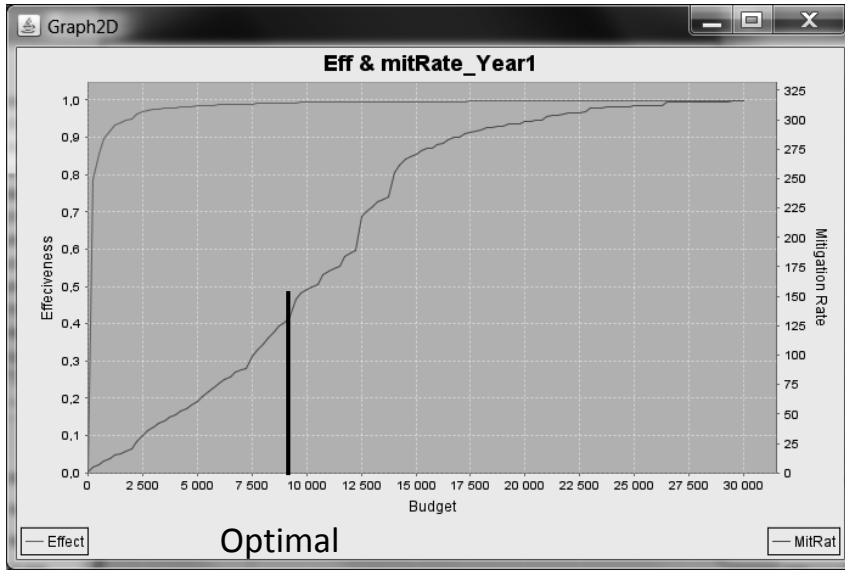


Figure 23. *The "function" Effectiveness = f(Budget) and mRate = f(Budget)* for first year.

Analysis of the Case Study results and Conclusions

The goal of our qualitative research is to collect basic information/expert knowledge needed and to test our GSM and GSES in a real organization. In this chapter we covered the qualitative research approach, presented the design and methodology, as well as collected and coded the data:

1. We were able to collect the cost/effectiveness information for the model with the accuracy of about ±20%, which leads to an "Do things right" optimization stage mitigation Rate exactness of +4% (±20% expert assessment error ensures accuracy for mitigation Rate in the range of ±5%, Table 10).

It is realistic that the expert assessment accuracy in the first case study tends to be more than ±20% (especially for Losses from security incidents). As expert assessment precision about Losses was only ±82%, it is understandable that it leads to a difference of real versus calculated in "Do things right" stage for Losses more than 100% and for Total Costs about 60% (Table 14).

	2009 real situation thousand €	**Calculated situation** for 6500 thousand €	**Calculated optimum** thousand €
max Annual Loss	477 000	400 000	400 000
IT Budget	6500*	6500	9 750
mR	**78**	**81**	**152**
Losses	6 120	14 260	5 700
Total Costs	12 620	20 760	15 450

* The sum of the IT costs of IT and Administrative Departments (a more detailed description on page 216, Table 11).

Table 14. SEB Estonia information security in 2009 – real vs calculated.

NB! We must keep in mind that the report to Financial Supervision Authority (in Table 14 "2009 real situation" data) is not the final truth. There is also possibility for potential inaccuracies.

2. We noticed that there was a strong resemblance between the results provided by our model and the data collected by business and IT security experts for the Annual official report to Financial Supervision Authority. The calculated mRate was quite close to the real value.

About the IT security costs optimality: the Bank IT security budget was approximately 2/3 the optimal and the real mitigation Rate achieved was approximately half of the potential optimum.

Therefore, if we are able to collect the information for the model with accuracy about ±20%, then the GSM/GSES method provides results that match reality and is usable.

3. The GSM/GSES method will provide much more accurate results, if IT security information (statistical data) would be systematically collected.
4. The reports generated by the GSES will support the senior management's decision making process. In principle, (at least in the first years) we do not have an automated decision system, but an information security decision support system. This is because the real situation (at least currently and in the near future) in IT changes too quickly and substantially.

Appendix 2. Security activities Dependency Matrix in Banking Case Study

Security Activities		Reference		Confidentiality				Integrity				Availability			
				C0	C1	C2	C3	I0	I1	I2	I3	A0	A1	A2	A3
Identified Security Goals				Public	Confidential	Secret	Top secret	Identification not needed	Non-authorized usage identified	Authorized usage identified	Identification usable in court	No requirements	Availability 90%	Availability 99%	Availability 99,9%
G1: Organization of information security															
1	Security Documentation	**SDOC**	CIA	1	2	3	4	1	2	3	4	1	2	3	4
2	Risk Assessment and Treatment	**RISK**	CIA	1	2	3	4	1	2	3	4	1	2	3	4
3	Security Accrediting	**SECA**	CIA			1	2			1	2			1	2
G2: Human resources security															
4	IT Human Resource Management	**HRM**	CA	0	2	3	4					0	2	3	4
5	Awareness Training	**AWT**	CA	1	2	3	4					1	2	3	4
G3: Physical and environmental security															
6	Perimeter Security(Physical Security)	**PSEC**	C		2	3	4								
7	Communication & Process Support IT Systems (DBserver, failserver, printserver, meilisüsteem, e-room, ...)	**C&PSS**	CIA	0	2	3	4	0	2	3	4	0	2	3	4
8	Personnel Working Environment(Physical Security)	**ENV**	C		2	3	4								
9	Personnel Workplace Equipment	**WPE**	A									0	1	2	3
10	Power	**POW**	A									0	2	3	4
11	Data Centres	**DC**	CIA	0	2	3	4	0	2	3	4	0	2	3	4
G4: Information systems acquisition, development and maintenance															
12	Outsourcing (incl. 3rd parties support)	**OUTS**	CIA	0	1	2	3	0	2	3	4				
13	Purchased SoftWare	**SW**	CIA	0	1	2	3	0	2	3	4				
14	Self-developed SoftWare	**DEV**	CIA	0	1	2	3	0	2	3	4				
G5: Access control															
15	Access Rights Management	**ARM**	CA	0	1	2	3					0	1	2	3
16	Network Access Control	**NAC**	CI	1	2	3	4	1	2	3	4				
17	Mobile computing and teleworking	**MOB**	CA	0	2	3	4					0	2	3	4
G6: Communications and operations management (ISO 17799)															
18	Internal network security	**LAN**	CA	0	1	2	3					0	2	3	4
19	External network security (incl. PerimProt, IDS/IPS, ...)	**WAN**	CA	1	2	3	4					1	2	3	4
20	Malware Handling	**AM**	CI	1	2	3	4	1	2	3	4	1	2	3	4
21	Information exchange policies and procedures	**ENC**	CIA		2	3	4			3	4				
22	Transaction Integrity	**TINT**	I							1	2				
23	Data backup and Restoration	**BCK**	IA					1	2	3	4	1	2	3	4
24	Data Archiving	**ARCH**	IA					1	2	3	4	1	2	3	4
G7: Compliance															
25	External Regulations	**REG**	CI		1	1	2		1	1	2				
26	Audit Capability	**AUD**	CI	1	2	3	4	1	2	3	4				
G8: Information security incident management															
27	Audit Trail	**LOG**	C	1	2	3	4								
28	Monitoring(Help Desk)	**MON**	A										2	3	4
29	IT Governance(IT quality, fiduciarity & security management)	**IT Gov**	CIA	0	1	2	3	1	2	3	4	0	1	2	3
G9: Business continuity management															
30	Business Continuity Management(main input to ITS R)	**BCM**	A										1	2	3
31	IT Systems Recovery(Redundancy jms, sisend BCM'ist)	**ITS R**	A										2	3	4
32	Crisis Management(tagab, et max IT riskid ~5% Panga kogu r	**CM**	A										1	2	3
G10: Asset Management															
33	Asset Management	**ASM**	A										2	3	4

Appendix 3. Investment Cost values for measure groups in Banking Case Study

			THE LEVEL INVESTMENT COSTS					
	Security Activities	**Ref**	**0 - None**	**1 - Low**	**2 - Medium**	**3 - High**	**Baseline L4 4 - Critical (aggregated costs)**	**5 - Mission Critical**
G1: Organization of information security								
	Security Organization	**SORG**	**0**	**10**	**50**	**75**	**100**	**150**
1	Security Documentation	SDOC	0	7,6	22,8	30,4	38,0	38,0
2	Risk Assessment and Treatment	RISK	0	156,0	312,0	468,0	624,0	624,0
3	Security Accrediting	SECA	0	36,3	145,0	145,0	145,0	145,0
G2: Human resources security								
4	IT Human Resource Management	HRM	0	240,0	720,0	960,0	1 200	1 200,0
5	Awareness Training	AWT	0	14,1	70,7	84,9	99	113,1
G3: Physical and environmental security								
6	Perimeter Security	PSEC	0	42,5	127,5	255,0	510	1 020,0
7	Communication & Process Support IT Systems	C&PSS	0	120,8	725,0	1 450,0	2 900	5 800,0
8	Personnel Working Environment	ENV	0	114,3	571,4	685,7	800	914,3
9	Personnel Workplace Equipment	WPE	0	150,0	300,0	900,0	900	900,0
10	Power	POW	0	20,4	81,6	244,7	734	734,0
11	Data Centres	DC	0	29,2	233,5	467,0	934	1 868,0
G4: Information systems acquisition, development and maintenance								
12	Outsourcing (incl. 3rd parties support)	OUTS	0	27,2	81,6	244,7	734	2 202,0
13	Purchased Software	SW	0	93,8	187,5	375,0	750	1 500,0
14	Self-developed Software	DEV	0	161,1	483,3	1 450,0	1 450	1 450,0
G5: Access control								
15	Access Rights Management	ARM	0	74,7	149,3	224,0	224	224,0
16	Network Access Control	NAC	0	67,6	202,7	337,9	473	473,0
17	Mobile computing and teleworking	MOB	0	10,0	40,0	60,0	80	160,0
G6: Communications and operations management (ISO 17799)								
18	Internal network security	LAN	0	14,9	44,6	74,3	104	133,7
19	External network security	WAN	0	26,4	79,2	237,7	713	713,0
20	Malware Handling	AV	0	64,8	259,3	324,2	389	389,0
21	Information exchange policies and procedures	ENC	0	15,4	30,8	61,5	123	246,0
22	Transaction Integrity	TINT	0	0,0	0,0	0,0	132	396,0
23	Data backup and Restoration	BCK	0	30,5	91,4	274,3	823	2 469,0
24	Data Archiving	ARCH	0	27,9	55,8	111,5	223	223,0
G7: Compliance								
25	External Regulations	REG	0	89,0	89,0	89,0	89	**198,0**
26	Audit Capability	AUD	0	21,3	64,0	128,0	256	256,0
G8: information security incident management								
27	Audit Trail	LOG	0	2,1	6,3	19,0	57	57,0
28	Monitoring (Help Desk)	MON	0	23,9	71,6	214,7	644	644,0
29	IT Governance	IT Gov.	0	111,6	557,8	1 115,5	2 231	2 231,0
G9: Business continuity management								
30	Business Continuity	BCM	0	23,1	69,3	208,0	624	624,0
31	IT Systems Recovery	SR	0	22,3	66,9	200,7	602	1 806,0
32	Crisis Management	CM	0	1,2	5,8	11,5	23	23,0
G10: Asset Management								
33	Asset Management	ASM	**0**	**8,0**	**24,0**	**48,0**	**72**	**144,0**
					6019,7		**19800**	30068,1

Appendix 4. Maintenance Cost values for measure groups in Banking Case Study

			MAINTAIN THE LEVEL (People, Process, Technology)					
	Security Activities	Ref	0 - None	1 - Low	2 - Medium	3 - High	Baseline L4 4 - Critical (aggregated costs)	5 - Mission Critical
G1: Organization of information security								
	Security Organization	**SORG**	**0**	**5**	**25**	**35**	**50**	**75**
1	Security Documentation	SDOC	0	2,3	4,6	3,0	3,8	3,8
2	Risk Assessment and Treatment	RISK	0	31,2	62,4	46,8	62,4	62,4
3	Security Accrediting	SECA	0	14,5	29,0	29,0	29,0	29,0
G2: Human resources security								
4	IT Human Resource Management	HRM	0	168,0	360,0	288,0	360,0	360,0
5	Awareness Training	AWT	0	11,3	35,4	25,5	19,8	11,3
G3: Physical and environmental security								
6	Perimeter Security	PSEC	0	12,8	38,3	51,0	102,0	102,0
7	Communication & Process Support IT Systems (DB-server, file server, print-server, mail-system, e-room, ...)	C&PSS	0	48,3	290,0	435,0	870,0	1740,0
8	Personnel Working Environment	ENV	0	34,3	57,1	68,6	80,0	91,4
9	Personnel Workplace Equipment	WPE	0	30,0	60,0	90,0	90,0	90,0
10	Power	POW	0	6,1	24,5	48,9	146,8	146,8
11	Data Center	DC	0	8,8	93,4	280,2	560,4	1307,6
G4: Information systems acquisition, development and maintenance								
12	Outsourcing (incl. 3rd parties support)	OUTS	0	13,6	40,8	73,4	220,2	440,4
13	Purchased Software	SW	0	46,9	75,0	112,5	150,0	450,0
14	Self-developed Software	DEV	0	48,3	193,3	725,0	725,0	725,0
G5: Access control								
15	Access Rights Management	ARM	0	37,3	59,7	67,2	67,2	67,2
16	Network Access Control	NAC	0	20,3	121,6	168,9	236,5	236,5
17	Mobile computing and teleworking	MOB	0	5,0	24,0	30,0	32,0	64,0
G6: Communications and operations management								
18	Internal network security	LAN	0	8,9	22,3	29,7	41,6	53,5
19	External network security	WAN	0	13,2	39,6	142,6	427,8	427,8
20	Malware Handling	AV	0	45,4	155,6	129,7	194,5	194,5
21	Information exchange policies and procedures	ENC	0	12,3	18,5	30,8	36,9	73,8
22	Transaction Integrity	TINT	0	0,0	0,0	0,0	33,0	118,8
23	Data backup and Restoration	BCK	0	21,3	36,6	219,5	658,4	1728,3
24	Data Archiving	ARCH	0	13,9	27,9	44,6	89,2	89,2
G7: Compliance								
25	External Regulations	REG	0	53,4	53,4	53,4	53,4	**104,0**
26	Audit Capability	AUD	0	12,8	25,6	51,2	76,8	76,8
G8: information security incident management								
27	Audit Trail	LOG	0	0,8	1,9	7,6	28,5	28,5
28	Monitoring (Help Desk)	MON	0	14,3	28,6	85,9	322,0	322,0
29	IT Governance	IT Gov.	0	78,1	390,4	446,2	892,4	892,4
G9: Business continuity management								
30	Business Continuity Management	BCM	0	16,2	34,7	104,0	374,4	374,4
31	IT Systems Recovery (Redundancy)	SR	0	13,4	33,4	80,3	361,2	1264,2
32	Crisis Management	CM	0	0,6	2,3	5,8	11,5	11,5
G10: Asset Management								
33	Asset Management	ASM	0	4,8	12,0	19,2	28,8	43,2
					2477		**7435,5**	**11805,3**

Appendix 5. Upgrade Cost values for measure groups in Banking Case Study

			UPGRADE THE LEVEL				
	Security Activities	**Ref**	**0 -> 1**	**1 -> 2**	**2 -> 3**	**3 -> 4**	**4 -> 5**
G1: Organization of information security							
	Security Organization	**SORG**	10	40	50	50	
1	Security Documentation	**SDOC**	7,6	15,2	7,6	7,6	0,0
2	Risk Assessment and Treatment	**RISK**	156,0	156,0	156,0	156,0	0,0
3	Security Accrediting	**SECA**	36,3	108,8	0,0	0,0	0,0
G2: Human resources security							
4	IT Human Resource Management	**HRM**	240,0	480,0	240,0	240,0	0,0
5	Awareness Training	**AWT**	14,1	56,6	14,1	14,1	14,1
G3: Physical and environmental security							
6	Perimeter Security	**PSEC**	42,5	85,0	127,5	255,0	510,0
7	Communication & Process Support IT Systems (DB server, fileserver, print-server, mail system, e-room, ...)	**C&PSS**	120,8	604,2	725,0	1 450,0	2 900,0
8	Personnel Working Environment	**ENV**	114,3	457,1	114,3	114,3	114,3
9	Personnel Workplace Equipment	**WPE**	150,0	150,0	600,0	0,0	0,0
10	Power	**POW**	20,4	61,2	163,1	489,3	0,0
11	Data Centers	**DC**	29,2	204,3	233,5	467,0	934,0
G4: Information systems acquisition, development and maintenance							
12	Outsourcing (incl. 3rd parties support)	**OUTS**	27,2	54,4	163,1	489,3	1 468,0
13	Purchased Software	**SW**	93,8	93,8	187,5	375,0	750,0
14	Self-developed Software	**DEV**	161,1	322,2	966,7	0,0	0,0
G5: Access control							
15	Access Rights Management	**ARM**	74,7	74,7	74,7	0,0	0,0
16	Network Access Control	**NAC**	67,6	135,1	135,1	135,1	0,0
17	Mobile computing and teleworking	**MOB**	10,0	30,0	20,0	20,0	80,0
G6: Communications and operations management (ISO 17799)							
18	Internal network security	**LAN**	14,9	29,7	29,7	29,7	29,7
19	External network security	**WAN**	26,4	52,8	158,4	475,3	0,0
20	Malware Handling	**AV**	64,8	194,5	64,8	64,8	0,0
21	Information exchange policies and procedures	**ENC**	15,4	15,4	30,8	61,5	123,0
22	Transaction Integrity	**TINT**	0,0	0,0	0,0	132,0	264,0
23	Data backup and Restoration	**BCK**	30,5	61,0	182,9	548,7	1 646,0
24	Data Archiving	**ARCH**	27,9	27,9	55,8	111,5	0,0
G7: Compliance							
25	External Regulations	**REG**	89,0	0,0	0,0	0,0	0,0
26	Audit Capability	**AUD**	21,3	42,7	64,0	128,0	0,0
G8: information security incident management							
27	Audit Trail	**LOG**	2,1	4,2	12,7	38,0	0,0
28	Monitoring (Help Desk)	**MON**	23,9	47,7	143,1	429,3	0,0
29	IT Governance	**IT Gov.**	111,6	446,2	557,8	1 115,5	0,0
G9: Business continuity management							
30	Business Continuity Management	**BCM**	23,1	46,2	138,7	416,0	0,0
31	IT Systems Recovery (Redundancy)	**SR**	22,3	44,6	133,8	401,3	1 204,0
32	Crisis Management	**CM**	1,2	4,6	5,8	11,5	0,0
G10: Asset Management							
33	Asset Management	**ASM**	8,0	16,0	24,0	24,0	72,0

Appendix 6. Effectiveness values for measure groups in Banking Case Study

	Security Activities	Ref	Graded Security Levels					
			0 - None	1 - Low	2 - Medium	3 - High	4 - Critical	5 - Mission Critical
G1: Organization of information security								
	Security Organization	**SORG**	0	25	50		75	90
1	Security Documentation	**SDOC**	0	10	30	50	80	
2	Risk Assessment and Treatment	**RISK**	0	15	20	40	70	
3	Security Accrediting	**SECA**	0	30	80		80	
G2: Human resources security								
4	IT Human Resource Management	**HRM**	10	30	50	70	80	
5	Awareness Training	**AWT**	0	30	50	70	80	95
G3: Physical and environmental security								
6	Perimeter Security (Physical Security)	**PSEC**	5	15	30	50	80	95
7	Communication & Process Support IT Systems (DB-server, file server, print-server, mail system, e-room, ...)	**C&PSS**	0	20	40	60	80	95
8	Personnel Working Environment (Physical Security)	**ENV**	10	15	30	60	85	95
9	Personnel Workplace Equipment	**WPE**	5	15	40	40	60	
10	Power	**POW**	5	10	40	60	90	
11	Data Centers	**DC**	0	10	30	60	85	99
G4: Information systems acquisition, development and maintenance								
12	Outsourcing (incl. 3rd parties support)	**OUTS**	0	15	30	50	80	90
13	Purchased Software	**SW**	15	20	30	40	60	70
14	Self-developed Software	**DEV**	10	30	60	60	80	
G5: Access control								
15	Access Rights Management	**ARM**	10	15	20	40	40	
16	Network Access Control	**NAC**	5	20	40	60	90	
17	Mobile computing and teleworking	**MOB**	5	20	40	50	70	90
G6: Communications and operations management (ISO 17799)								
18	Internal network security	**LAN**	10	20	30	60	80	90
19	External network security (incl. PerimProt, IDS/IPS, ...)	**WAN**	0	20	50	60	95	
20	Malware Handling	**AM**	0	15	30	60	80	
21	Information exchange policies and procedures	**ENC**	0	5	10	30	60	70
22	Transaction Integrity	**TINT**	0	0	0	0	80	90
23	Data backup and Restoration	**BCK**	5	40	50	70	85	90
24	Data Archiving	**ARCH**	0	30	50	55	80	
G7: Compliance								
25	External Regulations	**REG**	5	40			40	
26	Audit Capability	**AUD**	0	10	30	50	70	
G8: Information security incident management								
27	Audit Trail	**LOG**	5	10	40	60	80	
28	Monitoring (Help Desk)	**MON**	0	10	30	50	75	
29	IT Governance (IT quality, fiduciarity & security management)	**IT Gov.**	0	20	25	60	80	
G9: Business continuity management								
30	Business Continuity Management (main input to ITS R)	**BCM**	5	15	30	50	85	
31	IT Systems Recovery	**ITS R**	5	10	20	50	80	99
32	Crisis Management	**CM**	5	20	30	60	80	
G10: Asset Management								
33	Asset Management	**ASM**	0	30	50	60	80	90

Appendix 7. Risk assessment for SEB Business Inform Systems for 2009.

Infosüsteem	Infoturbe klassifikaator	Salastatuse väärtus (miljonites EEKides)				Käideldavuse väärtus (miljonites EEKides)				Tervikluse väärtus (miljonites EEKides)				Potentsiaalne CIA Kahju Aastas
		Riskid salastatuse st kui tagatud S0 - st nõutud Sx (S3 või S2 või S1) -> S0	Riskid salastatuse st kui tagatud S1 - st nõutud Sx (S3 või S2) -> S1	Riskid salastatusest kui tagatud S2 - st nõutud Sx (S3) -> S2	Riskid salastatusest kui tagatud S3	Riskid käideldavusest kui tagatud K0 - st nõutud Kx (K3 või K2 või K1) -> K0	Riskid käideldavuse st kui tagatud K1 - st nõutud Kx (K3 või K2) -> K1	Riskid käideldavusest kui tagatud K2 - st nõutud Kx (K3) -> K2	Riskid käideldavusest kui tagatud K3	Riskid tervikluse st kui tagatud T0 - st nõutud Tx (T3 või T2 või T1) -> T0	Riskid tervikluse st kui tagatud T1 - st nõutud Tx (T3 või T2) -> T1	Riskid tervikluse st kui tagatud T2 - st nõutud Tx (T3) -> T2	Riskid tervikluse st kui tagatud T3	2009
CRM	S2K3J3Õ2T2	0,55	0,10	0,00	0,00	55,55	1,10	0,10	0,00	0,65	0,20	0,00	0,00	58,25
Data Warehouse	S2K2J2Õ2T2	0,10	0,10	0,00	0,00	0,20	0,65	0,00	0,00	0,20	0,65	0,00	0,00	1,90
e-agent	S2K3J2Õ3T2	0,10	0,10	0,00	0,00	0,20	0,20	0,00	0,00	0,20	0,20	0,00	0,00	1,00
Elukas	S3K3J2Õ3T2	55,00	0,55	0,10	0,00	110,00	0,65	0,00	0,00	110,00	55,10	0,00	0,00	331,40
Kaardisüsteem	S3K3J3Õ3T3	55,00	0,55	0,55	0,00	110,00	110,00	5,50	0,00	110,00	110,00	55,00	0,00	556,60
Kõnesalvestussüsteem Partcom	S2K3J3Õ2T3	0,55	0,10	0,00	0,00	110,00	110,00	55,00	0,00	55,55	55,10	55,00	0,00	441,30
LIISU	S2K3J3Õ3T2	55,00	0,10	0,00	0,00	55,55	0,65	0,10	0,00	55,55	55,10	0,00	0,00	222,05
NBSX	S2K3J3Õ3T2	55,00	0,10	0,00	0,00	110,00	55,55	0,10	0,00	110,00	56,00	0,00	0,00	386,75
Pangaliides	S2K3J3Õ3T3	0,10	0,10	0,00	0,00	110,00	55,55	0,10	0,00	110,00	110,00	55,00	0,00	440,85
SWIFT	S2K2J2Õ3T3	55,00	0,10	0,00	0,00	55,55	0,20	0,00	0,00	55,10	0,20	0,10	0,00	166,25
Teleteenus	S3K3J3Õ3T3	55,00	0,10	0,10	0,00	110,00	55,55	0,10	0,00	110,00	110,00	55,00	0,00	495,85
U-Liising+	S2K3J2Õ3T2	0,55	0,10	0,00	0,00	0,20	0,20	0,00	0,00	1,55	0,65	0,00	0,00	3,25
U-Net	S3K3J3Õ3T3	55,00	55,00	0,10	0,00	110,00	55,55	0,10	0,00	110,00	110,00	55,00	0,00	550,75
U-Net Business	S3K3J3Õ3T3	55,00	55,00	0,10	0,00	110,00	55,55	0,10	0,00	110,00	110,00	55,00	0,00	550,75
TREMA	S2K3J2Õ3T2	55,00	0,10	0,00	0,00	60,50	0,20	0,00	0,00	110,00	1,10	0,00	0,00	226,90
Arhiivi IS	S2K2J2Õ1T1	0,10	0,10	0,00	0,00	0,65	0,20	0,00	0,00	55,10	0,00	0,00	0,00	56,15
Cognos	S2K2J2Õ1T1	0,10	0,10	0,00	0,00	0,20	0,20	0,00	0,00	0,20	0,00	0,00	0,00	0,80
Corona	S2K1J2Õ1T2	55,00	0,10	0,00	0,00	0,20	0,10	0,00	0,00	110,00	0,10	0,00	0,00	165,50
eRoom	S2K2J2Õ1T1	0,55	0,10	0,00	0,00	1,10	0,20	0,00	0,00	0,20	0,00	0,00	0,00	2,15
ORACLE FINANCIALS	S2K2J2Õ3T2	0,10	0,10	0,00	0,00	0,20	0,65	0,00	0,00	0,20	0,65	0,00	0,00	1,90
Fraud Guard	S2K2J2Õ1T1	55,00	0,10	0,00	0,00	0,20	0,20	0,00	0,00	0,20	0,00	0,00	0,00	55,70
Intranet	S2K2J2Õ1T1	0,10	0,10	0,00	0,00	0,20	0,20	0,00	0,00	0,20	0,00	0,00	0,00	0,80
Kodulehekülg	S0K2J2Õ1T1	0,10	0,00	0,00	0,00	0,20	0,20	0,00	0,00	0,20	0,00	0,00	0,00	0,70
Mobiilipank	S3K2J2Õ3T3	55,00	0,55	0,55	0,00	1,10	1,10	0,00	0,00	55,55	55,55	0,55	0,00	169,95
Personalisüsteem	S3K2J2Õ1T2	55,00	0,55	0,10	0,00	1,10	0,20	0,00	0,00	0,65	0,10	0,00	0,00	57,70
Postipank	S2K2J2Õ3T3	0,10	0,10	0,00	0,00	1,10	0,65	0,00	0,00	1,10	1,10	0,55	0,00	4,70
Solidus	S2K2J2Õ1T1	0,55	1,00	0,00	0,00	1,10	1,10	0,00	0,00	0,65	0,00	0,00	0,00	4,40
Postipoiss	S3K2J2Õ2T1	55,00	0,10	0,10	0,00	0,20	0,20	0,00	0,00	0,65	0,10	0,00	0,00	56,35
Taavi Palk	S2K2J1Õ2T2	0,55	0,10	0,00	0,00	0,65	0,10	0,00	0,00	0,65	0,20	0,00	0,00	2,25
Tel-U-Net	S2K2J2Õ3T3	55,00	0,10	0,00	0,00	1,10	0,20	0,00	0,00	55,55	55,55	55,00	0,00	222,50
ULJA	S2K2J2Õ1T2	55,00	0,55	0,00	0,00	0,65	0,20	0,00	0,00	0,65	0,10	0,00	0,00	57,15
Workplace Service	S2K0J1Õ0T1	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Infra Base Service	S2K3J3Õ1T1	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
e-post	S2K3J3Õ1T2	0,10	0,10	0,00	0,00	6,50	0,65	0,10	0,00	0,20	0,10	0,00	0,00	7,75
Kõnesalvestussüsteem DIREC NT	S2K3J3Õ2T3	0,55	0,10	0,00	0,00	110,00	110,00	55,00	0,00	55,55	55,10	55,00	0,00	441,30
SafeWach	S2K2J2Õ3T2	55,00	0,10	0,00	0,00	55,55	0,20	0,00	0,00	55,10	0,20	0,00	0,00	166,15
Info Väärtus = Keskmine Tõenäoline Kahju Aastas = Intsidendi Potentsiaalne Keskmine Kahju x Tõenäoline Intsidendi Esinemissagedus Aastas	KOKKU	939,85	116,25	1,70	0,00	1289,75	618,15	116,30	0,00	1441,40	943,15	441,20	0,00	5907,75

Appendix 8. ΔIT Risk/ΔIT Budget≥1 for SEB in 2009.

Calculated Loss values (maxALE/mRate) and the Loss values estimations from the business experts:

Information Value = 400 000

Confidence	Budget	mR	Calculated Losses = MaxALE/mR	Losses based on business experts opinions
0	0	1	400000	400000
0,787234043	250	4,7	85106	172100
0,86013986	500	7,15	55944	161500
0,898477157	750	9,85	40609	112500
0,923664122	1000	13,1	30534	79500
0,952380952	1500	21	19048	56300
0,982142857	2000	56	7143	41800
0,989690722	2500	97	4124	33500
0,995798319	3000	238	1681	20100
0,997635934	**3500**	**423**	**946**	7240
0,998360656	**4000**	610	656	**4351**
0,99907919	5000	1086	368	3805
0,999454148	7500	1832	218	2985
0,999602228	10000	2514	159	2632
0,999713056	12500	3485	115	2370
0,999765423	15000	4263	94	2255
0,999789341	17500	4747	84	2194
0,999803227	20000	5082	79	2168
0,999812488	22500	5333	75	2144
0,999816682	25000	5455	73	2135

Table 15. Loss values - calculated (maxALE/mRate) versus business experts estimations.

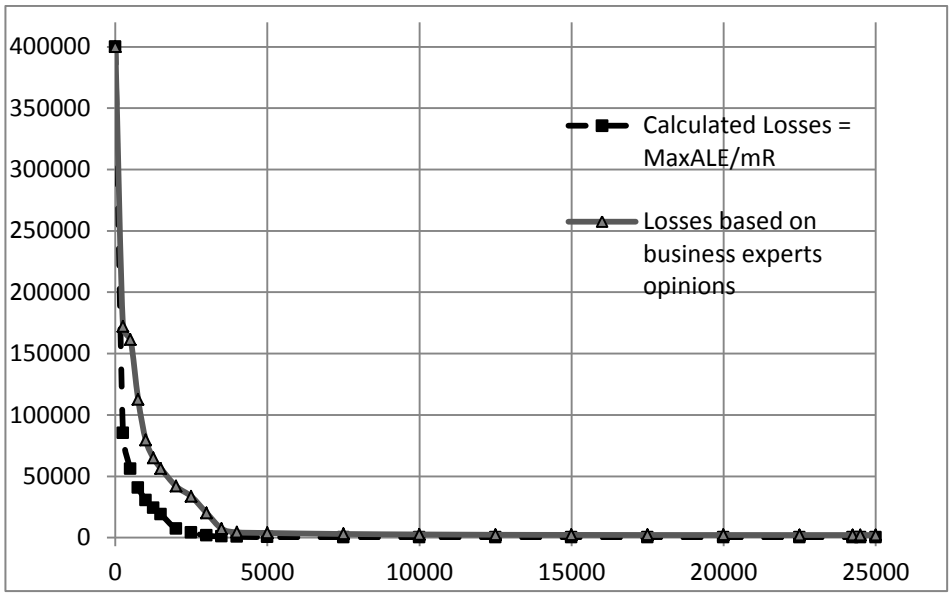


Figure 24. Loss curves - calculated (maxALE/mRate) versus business experts estimations.

ΔLoss/ΔBudget values for Budget points i and i+1:

- based on calculations ‘$maxALE/mRate_i - maxALE/mRate_{i+1}) \geq (Budget_i - Budget_{i+1})$’, and
- based on expert assessments about Losses.

Budget	mR	ΔLoss/ΔBudget_Loss Calculated	ΔLoss/ΔBudget_Loss from experts
0	1		
250	4,7	1259,574468	911,6
500	7,15	116,6493081	42,4
750	9,85	61,33967555	196
1000	13,1	40,29914364	132
1500	21	20,77922078	34,8
2000	56	23,80952381	29
2500	97	6,038291605	16,6
3000	238	4,886078143	26,8
3500	**423**	**1,470091583**	25,72
4000	610	0,579777545	**5,778**
5000	1086	0,28741358	0,546
7500	1832	0,059993406	0,328
10000	2514	0,023692649	0,1412
12500	3485	0,017732549	0,1048
15000	4263	0,008378793	0,046
17500	4747	0,003826756	0,0244
20000	5082	0,00222183	0,0104
22500	5333	0,001481793	0,0096
25000	5455	0,000974268	0,006

Table 16. ΔLoss/ΔBudget values – calculated versus based on experts estimations.

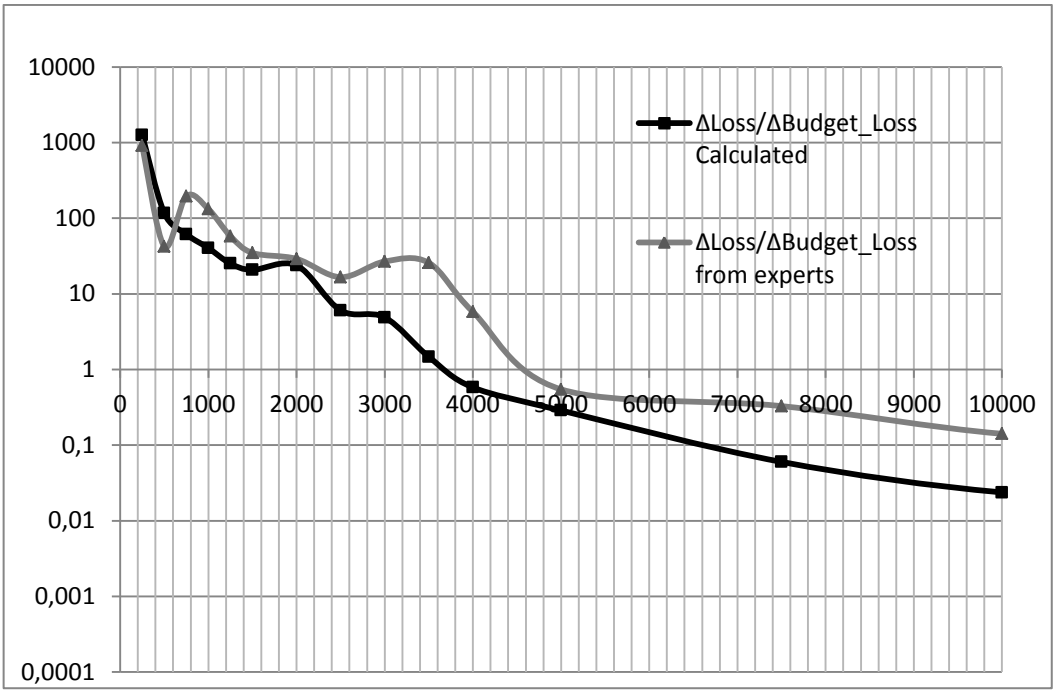


Figure 25. ΔLoss/ΔBudget curves – calculated versus based on experts estimations.

Appendix 9. First practical and very interesting results from gb_GSM/GSES method.

1. We have found a very interesting result (Table 12): the optimal cost of information security for the second year is approximately 75% of the optimum for the first year, and in all the following years the optimal cost is only about 50% of the optimum for the first year.
2. With the help of GSES we have proved that the currently widely used "Do rational things" approach ("no more and no less than required" approach, used in ISKE and NISPOM, for example) is very likely quite far from optimal.

   As we can see from Table 15:
   - To achieve rational optimality for the ISKE security level Low (C1I1A1), 352 thousand € (5 500 000 kr) is needed and mRate=19. However, it is possible to reach almost 8 times better security (mRate=147) for the same money, if the security levels for security activities are not artificially limited to "1".
   - To achieve rational optimality for the ISKE security level Medium (C2I2A2), 519 thousand € (8 100 000 kr) is needed and mRate=75. However, it is possible to reach almost 4 times better security (mRate=315) for the same money, if the security levels are not artificially limited to "2".

Rational level	Security Class	Budget (thousands kr)	Security Effectiveness		Risks Mitigation		Optimal versus rational
			Rational (ISKE)	Optimal	Rational (ISKE)	Optimal	
Low	C1I1A1	5 500	0,947	0,993	19	147	~ 8 times
Medium	C2I2A2	8 100	0,987	0,997	75	315	~ 4 times

Corresponding Security Profiles:

L - 5500 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1

Opt -5500 3, 3, 3, 3, 3, 3, 3, 3, 1, 2, 0, 3, 1, 2, 0, 3, 2, 3, 1, 1, 0, 2, 3, 3, 0, 3, 3, 1, 1, 0, 0, 0, 3, 1, 0, 3, 3, 2, 3

M - 8100 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2

Opt -8100 3, 3, 3, 3, 3, 3, 3, 3, 2, 3, 1, 3, 3, 3, 0, 3, 3, 3, 1, 3, 3, 3, 3, 3, 0, 3, 3, 3, 3, 0, 0, 2, 3, 1, 0, 3, 3, 2, 3

Table 17. Rational versus optimal. [27]

Rational Low and Medium security levels from ISKE remain very far from the optimum. Subject certainly needs further investigation. Estonian public sector tens of millions of euros range of information security spending is likely to be sufficient cause and justification.

At least for Estonian ISKE would make sense to rename "rational optimality" as a "seemingly rational optimality" or "quasi-rational optimality".

[27] Calculations for ISKE are based on gb_GSM and cost/effectiveness data from master thesis from Lauri Palkmets (supervised by me) - „Astmelise infoturbe mudeli rakendatavus Eesti Kaitseväes" (A Graded Security Model applicability in Estonian Defense Forces).

SUMMARY IN ESTONIAN
EESTIKEELNE RESÜMEE

Mudel infoturbe kulutuste optimeerimiseks

Käesolev infoturbe kulutuste optimaalsust käsitlev teadustöö koondab seitse akadeemilist artiklit ning hõlmab infoturbe ja ärijuhtimise uurimisvaldkondi.

Töö aktuaalsus

Käesoleval ajal on IT-valdkonnas lõpule jõudmas, arenenud riikides juba ka lõpule jõudnud, mitmed väga olulised muutused. Kõige olulisem neist on ilmselt fakt, et IT- ja infosüsteemid pole üldjuhul enam tegevusvaldkond, millega oleks võimalik tagada konkurentsieelist võrreldes teiste analoogsete organisatsioonidega. IT on muutunud iseenesestmõistetavaks tegevusvaldkonnaks, mis on otseselt vajalik põhitegevuse tagamiseks. Tänapäeval halvaks IT mittetoimimine praktiliselt kogu organisatsiooni töö: teenuste osutamise, tootmisprotsessid, toormaterjali ja valmistoodangu tarnete teostamise, infovahetuse klientide ja koostööpartneritega jms.

Samas on väga oluliselt tõusnud IT-teenuste mittetoimimisega seotud riskid. Võib öelda, et olukord on analoogne näiteks elektrienergia tagatusega seotud riskidega – ka organisatsiooni infosüsteemide mittetoimimisel praktiliselt peatub kogu organisatsiooni põhitegevus.

Eelnev on oluliselt muutnud suhtumist infotehnoloogiasse ja infoturbesse. IT ja infoturbega seotud kulud on muutunud küllaltki oluliseks osaks organisatsioonide põhitegevuslikest kulutustest ning nende minimaalsus on just see, mis tagab konkurentidega võrreldes teatud ärilise eelise.

Doktoritöö käigus välja töötatud astmeline infoturbe ekspertsüsteem on põhimõtteliselt otsuste langetamise tugisüsteem organisatsiooni IT-juhtidele. Paremad juhtimisotsused tagavad organisatsioonile kindlasti küllaltki olulise konkurentsieelise.

Infotehnoloogia ja seega ka infoturve arenevad praegu vägagi tormiliselt ning on praktiliselt võimatu kõiki arenguid ja muutusi eelnevalt hinnata ja arvesse võtta – st hetkel ei ole reaalne välja töötada täisautomaatset (ilma inimese osaluseta) edukalt toimivat infoturbe juhtimissüsteemi.

Samas on infoturbe kulutusi optimeeriv ekspertsüsteem kasutatav otsuste tugisüsteemina suvalise äriprotsessi kulutuste optimeerimiseks, juhul kui:

- suudame kirjeldada selle protsessi graafipõhise mudeli;
- allprotsessidonkirjeldatavad(stkapraktiliseltteostatavad)mitmeastmelistena (st teostatavad mitmetel võimalikel alternatiivsetel tasemetel, vägagi levinud on astmelisus madal/keskmine/kõrge) ning suudame mõõta ja/või määratleda nende astmete maksumust ja efektiivsust (vmt).

Uurimistöö eesmärk ning hüpotees

Viimasel kümnendil on toimunud arengud IT ja infoturbe kulutuste optimeerimise valdkonnas (vt *Part* I), kuid tänaseni puudub terviklik ja süsteemne käsitlus infoturbest. Käesoleva uurimistöö eesmärk on välja töötada infoturvet kirjeldav mudel ja otsustuste tugisüsteem, et aidata langetada põhjendatud otsuseid organisatsiooni infoturbe optimaalse efektiivsuse tagamiseks. Tugisüsteemi abil saaks ekspert juhile põhjendada infoturbeks vajalikke kulutusi ning juht oskaks omakorda eksperdilt küsida talle arusaadavaid põhjendusi.

Käesoleva töö peamine idee seisneb selles, et põhinedes lihtsal, aga samas arusaadaval graaf-mudelil, mis lähtub IT-vaatest infoturbele, võimaldab seda teostav ekspertsüsteem ka väikestele ja keskmistele organisatsioonidele vastuvõetava töömahukusega kätte saada just selle, mida organisatsiooni juhtkond infoturbe valdkonnas otsustamiseks kõige enam vajab: visuaalne, ühe pilguga mõistetav väljund turbe efektiivsuse ja turbele kulutatud ressursside vahelisest sõltuvusest – *SecurityEffectiveness=f(SecurityCosts)*.

Käesoleva uurimistöö põhiküsimus on järgmine:
kuidas määratleda ka väikestele ja keskmistele organisatsioonidele vastuvõetava tööde mahuga (~ 1–2 inimkuud) organisatsioonile nõutav ja/või optimaalne infoturbe tase, st optimaalsed kulutused infoturbele ja optimaalne teostatavate turvameetmete loetelu?

Veidi lahtiseletatult – vaja on välja töötada infoturbe mudeli, mis võimaldaks:
1. infoturbe kulutuste optimeerimist,
2. leitud optimaalsete infoturbe kulutuste põhjal määratleda optimaalne teostamist vajavate turvameetmete loetelu,
3. eelnevad tööd teostada 1–2 inimkuuga, mis oleks sobiv ka väikestele ja keskmistele organisatsioonidele.

Põhihüpotees:
- kirjeldatud astmeline infoturbe mudel (AIM, *Graded Security Model*) on piisavalt detailne infoturbe kulude täpseks optimeerimiseks;
- kirjeldatud astmeline infoturbe mudel (AIM, *Graded Security Model*) on samas piisavalt lihtne, et tagada alusinfo kogumine, optimeerimine ja analüüs suurusjärgu võrra väiksema töömahukusega, kui seni vajalike turvameetmete määratlemiseks üldkasutataval detailsel riskianalüüsil

(GSM-i rakendamisel tööde maht on 1–2 inimkuud, samaväärse töö teostamisel detailse riskianalüüsiga on tööde mahuks 1–2 inimaastat).

Infoturvet kui protsessi kirjeldava *Graded Security Model*'i loomisel on tuginetud võimalikult lihtsatele ja laia kasutamist võimaldavatele lähenemisviisidele, sh:

- infoturbe käsitlemine kitsalt IT-süsteemide turvameetmeid määratlevate paremate praktikate ja standardite raames (mis on tunduvalt lihtsam võrreldes näiteks äririskide põhise vaatega);
- IT ja infoturbe protsessi kirjeldamine *People-Process-Technology* (vt *Figure* 2) käsitlusviisi abil, mis on tunduvalt lihtsam võrreldes näiteks ISACA BMIS-iga (vt *Figure* 3).

Põhiline, et ligikaudu suurusjärk (st kümme korda) vähem detailne mudel loob IT-töötajatele võimaluse ka ligikaudu suurusjärgu võrra vähema tööga anda juhtkonnale otsuste tegemiseks vajalik info.

Töö peamise eesmärgi saavutamiseks tuli leida lahendus mitmele alamprobleemile:

1. Infoturbe spetsiifikat toetav ja infoturvet kui protsessi kirjeldav mudel.
2. Juhtimisotsuste tegemiseks sobiv infoturbe meetrika.
3. Astmelise infoturbe graaf-mudelil põhinevad sobivad algoritmid infoturbe efektiivsuse arvutamiseks ja optimeerimiseks.
4. Nõuded tarkvaralisele lahendusele, sobiva tarkvaraplatvormi valik.

Neid probleeme ongi detailsemalt käsitletud käesoleva töö teoreetilises osas. Lõpptulemusena on kirjeldatud Astmelise infoturbe mudel (*Graded Security Model* ehk GSM) ning seda realiseeriv Astmelise infoturbe ekspertsüsteem (*Graded Security Expert System* ehk GSES).

Hetkel enamkasutatavates infoturbe standardites ja *Best Practice* jms mudelites on just infoturbe optimaalsuse probleemistik jäänud vajaliku tähelepanuta. Töö käigus on selgunud, et nendes kirjeldatud vajalike turvameetmete valikud on vägagi subjektiivsed ja saavutatavad riskileevendused on kordades väiksemad, kui on sama raha eest maksimaalselt võimalik.

Infoturve on vägagi kulukas tegevusvaldkond ja üldlevinud on ressursside puudus selle vajalikul tasemel teostamiseks. Seda enam peame kasutama olemasolevaid vahendeid optimaalselt.

Uurimisstrateegia ning metoodika

Uurimismeetod organisatsiooni infoturvet kirjeldava mudeli väljatöötamiseks
Astmelise infoturbe mudeli väljatöötamiseks on kasutatud teoreetilist uurimismeetodit – on võrreldud ja analüüsitud seni kasutatud mudeleid ning sünteesitud uudne graafi-põhine organisatsiooni üldine IT ja infoturbe mudel ning sellele vastavad optimeerimise algoritmid.

Meetod astmelise mudeli testimiseks ja konkreetseks rakendamiseks
Infoturve on organisatsioonile väga eriomane, umbes nagu näpujälg inimesele. Suurte organisatsioonide heal tasemel infoturbe puhul (näiteks pankades) on reaalne 4^{40} ehk $\approx 10^{26}$ erinevat teostamise variatsiooni, samas väikeste ja keskmiste organisatsioonide tasemel (st väga piiratud ressursside korral) 10^{10}. Seega põhineb mudeli reaalne rakendamine paratamatult konkreetsete organisatsioonide juhtumiuuringutel: optimeerida saab ikkagi ainult konkreetse organisatsiooni konkreetset infoturvet.

Astmeline infoturbe mudel on kavandatud prototüüp-mudelina, mis tuleb juhtumipõhiselt sobitada, st konkreetsele organisatsioonile konkreetne optimeerimine.

Samas juba teostatud juhtumiuuringud on vägagi kasulikeks eeskujudeks järgmistele, vähendades nende töömahtu oluliselt (isegi kuni suurusjärgu ulatuses, kui eeskujuks on suhteliselt sarnase organisatsiooni juhtumiuuring).

Seega on konkreetsetes organisatsioonides teostatud juhtumiuuringud ühtlasi ka eelnevalt teaduslikult väljatöötatud astmelise mudeli testimisteks päriselus. Niiviisi on teoreetilised ideed osutunud praktikas kontrollitavateks ja kontrollituteks.

Nõnda oleme saanud vastuse kahele olulisele küsimusele:
1. Kas suudame koguda vajaliku alusinfo?
2. Kas saame tegelikkusele vastava tulemuse?

Töötamaks välja konkreetse organisatsiooni juhtkonnale põhjendatud otsustuste tegemiseks tugisüsteemi ning samas kindlustamaks, et see kirjeldab reaalset olukorda piisavalt täpselt, tuleb mudel:
1. kirjeldada,
2. testida ning rakendada.

Esimes(t)eks juhtumiuuringu(te)ks on kasulik valida organisatsioon(id), kus infotöötlus ja infoturve on äärmiselt olulised. Kui mudel kirjeldab piisavalt hästi olukorda kriitilises organisatsioonis, siis on tõenäoline, et see kirjeldab piisavalt hästi infoturvet kõigis (või vähemalt enamikes) organisatsioonides. Käesolevas uurimuses toodud juhtumiuuring pärineb otse infoturbe rindejoonelt, milleks rahuaja tingimustes on pangandus (vt *APPENDICIES*).

Kasutatud on kogemuse-järgset meetodit (*a posteriori method*). Kirjeldatakse Panga eelmise X-aasta infoturvet. Kui saame X-aasta infoturbe tegelikkusele vastavad tulemused ka arvutuslikult mudelist, st tegelikud ja arvutuslikud turvaintsidentidest tingitud kahjud on ligikaudu võrdsed, siis eeldame, et mudel on piisavalt hea ka järgmise X+1-aasta infoturbe investeeringute optimeerimiseks. Muidugi tuleb järgmise aasta mudelis arvesse võtta aasta jooksul nii IT-s kui ka infoturbes toimunud muutusi, nii **ülemaailmsel** kui ka organisatsiooni tasemel.

Juhtumiuuring kahes Eesti suurimas pangas tõendas:

1. Mudeli jaoks vajalik ekspertteave on ka reaalselt kogutav.
2. Mudel annab tegelikkusele piisavalt hästi vastava tulemuse.

Mudeli efektiivseks rakendamiseks tuleb see sobitada organisatsiooni raamatupidamise ja riskianalüüsi tööpõhimõtetega ja süsteemidega. Sellisel juhul saaksime põhilise alusinfo infoturbe kulutuste optimeerimiseks juba ühe hiireklikiga. Näiteks andmed infoturbe kulutuste kohta raamatupidamise infosüsteemist ning turvaintsidentidest ja neist tingitud kahjudest riskihalduse infosüsteemist. Kogu optimeerimise töömaht väheneks oluliselt veelgi.

Infoturbe kirjeldav mudel (GSM)

Algne kava oli lihtsalt ühendada kaks head ideed: USA-s (DoE) 1999. a välja töötatud ja 2006. a ajakohastatud maatriks-mudelit NISPOM 2006 (vt *Table* 3) ja *CyberProtect*'i infoturbe tegevusvaldkondade tasemete hinna ja efektiivsuste kirjeldamine (vt *Figure* 1). Plaanitud oli ameeriklaste mudelit veidi ajakohastada (suurendada infoturbe eesmärkide ja tasemete arvu) ning kirjeldada kas 1024- või 4096-astmeline mudel. Mitme tuhande võimaliku infoturbe variatsiooni hulgast on ju suurem tõenäosus leida maatriks-mudelist organisatsiooni tegelikule vajalikule infoturbe tasemele lähedane võimalik aste.

Juhul kui organisatsioonil ei ole piisavalt ressursse vajaliku taseme saavutamiseks, määratleksime infoturbe tegevusvaldkondade võimaliku maksimaalse keskmise efektiivsuse, st parima võimaliku turbeprofiili.

Esmased probleemid maatriks-mudeli osas tekkisid 2009. a, vt *Study* III (S_III). Teostatud juhtumiuuring panganduses tõi selgelt välja selle mudeli mitu olulist puudust. Kui maatriksmudeli üheks peamiseks eelduseks on maatriksi ridade vaheline sõltumatus, siis reaalselt infoturbes:

1. Tegevusvaldkondade vahel on olulised seosed ja sõltuvused. Näiteks organisatsiooni infosüsteemi perimeetri kaitsel mõjutab tulemüüri allsüsteemi efektiivsust tihti isegi rohkem tulemüüri administraatori parem koolitus – st koolituse taseme tõstmine on otstarbekam, kui parema ja kallima tulemüüri hankimine.
2. Kõik infoturbe tegevusvaldkonnad ei ole võrdväärsed, osa neist on olulised ja osa on tugiteenused.

Põhiline sisuline erinevus infoturbe tegevusvaldkondade vahel on järgmine:

- Kui oluline teenus ei toimi, siis ei toimi kogu infoturve kui teenus.
  Näiteks kui infosüsteemi riistvara kui tegevusvaldkond ei tööta (st selle efektiivsus = 0), siis pole üldse oluline, kui hästi on lahendatud ülejäänud infoturbe tegevusvaldkonnad – tarkvara, toide, õelvara vastane kaitse jms, reaalselt ei toimi kogu infosüsteem ja seega selle koondefektiivsus on null.

- Kui tugiteenus ei toimi või toimib ebapiisaval tasemel, siis see mõjutab teatud ulatuses ainult neid teenuseid, mille toeks tugiteenus on mõeldud, kuid ei mõjuta üldse paljude teiste teenuste taset.
  Näiteks lõppkasutaja koolitus ei mõjuta riistvara, tarkvara, toite jms töö efektiivsust ega kindlust, kuid mõistagi mõjutab lõppkasutaja enda efektiivsust.

Seega analüüsi tulemusena sai selgeks, et infoturbe kirjeldamiseks on maatriks-mudel ilmselt liigne lihtsustus. Maatriks-mudeli baasil infoturbe kui süsteemi integraalse efektiivsusena määratletav infoturbe tegevusvaldkondade keskmine või kaalutud keskmine efektiivsus (kindlus) ei anna adekvaatset pilti infoturbe tegelikust seisukorrast. Muidugi on võimalik maatriksisse ridade vahelisi sõltuvusi funktsioonidena sisse kirjutada, kuid sellega kaob mudeli ülevaatlikkus ja arusaadavus.

Kõiki eelnevaid maatriks-mudeli puudusi arvestades oli ilmne vajadus leida või kirjeldada eelloetletud puudustest vaba infoturbe mudel, mis kirjeldaks infoturvet kui protsessi.

Ei ole võimalik vaadelda lahus infotehnoloogia ja infoturbe tegevusvaldkondi. St peame kirjeldama tegelikult IT ja infoturbe ärimudeli. Praktiliselt kõigis infotehnoloogia tegevusvaldkondades sisaldub ka infoturbe komponent.

Näiteks:
Riistvara (HW) – server on ~10x kallim kui tavaline arvuti (PC), st 90% kulutustest läheb infoturbele;
Tarkvara (SW) – inimlike tegevuste ja vigade jms fikseerimise ja välistamise funktsionaalsuse loomine ja testimine on üldjuhul ~10x suurem töö kui põhifunktsionaalsuse programmeerimine, st ~90% tööst (kulutustest) läheb taas infoturbele.

Probleem ilmnes väga selgelt Panga juhtumiuuringus, kus kulutuste aluseks võtsimegi Panga kogu IT eelarve (st IT kulutused).

IT ja infoturbe protsessi mudeli kirjeldamist alustasin nagu suvalise äriprotsessi mudelit: Personal – Protsess – Tehnoloogia ja sellega paralleelne Organisatsioon (vt *Figure* 2) ehk rohkem lihtsustatult ja IT-spetsiifiliselt Personal – Tarkvara – Riistvara // Organisatsioon.

Tasemel infoturve koosneb 30–40 tegevusvaldkonnast, mis on ärimudelis omavahel ühendatud kas järjestikku või paralleelselt, vastavalt kahele infoturbespetsiifilisele põhimõttele:

- Järjestikku on IT-ärimudelis ühendatud nn olulised IT ja infoturbe tegevusvaldkonnad, millest igaühe rike tingib kogu IT süsteemi mittetoimimise (st kehtib keti põhimõte).

- Paralleelselt on IT-ärimudelis olulistele valdkondadele ühendatud nn tugivaldkonnad, mis on mõeldud oluliste valdkondade teenuste parendamiseks, st mitmetasemeline infoturve (*Multilevel Security, Defence in Depth*).

Detailsemalt on infoturbe tegevusvaldkonnad esitatud IT ja infoturbe ärimudelis (panganduses) *Figure* 4-l suhtediagrammina ja *Figure* 5-l graaf-mudelina.

Olulised tegevusvaldkonnad (järjestikühenduses):

- Personal: IT-süsteemide kasutajad ja nende tööarvutid.
- Protsess: hooned ehk infrastruktuur, füüsiline turve, antiviirus, IT-haldus, IT-süsteemide tarkvara.
- Tehnoloogia: toide, serveriruum (koos arvutite ja muu tehnikaga), LAN, WAN.
- IT organisatsioon: turvadokumentatsioon (sh infoturbe strateegia, -poliitika jne), turvaaudit, vastavus seadustele ja lepingutele, riski-juhtimine.

Tugivaldkonnad (paralleelsed olulistele tegevusvaldkondadele):

- Kasutajate koolitus.
- Pääsu- ja võrguõiguste haldus.
- Äri talitluspidevus, kriisijuhtimine, IT-teenuste taaste.
- Logide haldus, monitooring, abiliin.
- Varukoopiad, arhiivindus.
- Tarkvara testimine, infoturbe testimine.
- IT-varade juhtimine.

Uus mudel on nn "graafipõhine astmeline infoturbe mudel" (g_AIM, *gb_GSM*). Infoturbesüsteemi efektiivsus on arvutatav Süsteemiteoorial põhinevate järjestik- ja paralleelühendustes ühendatud osade käideldavusarvutuste abil.

Graafi-põhises astmelises infoturbe mudelis kasutatav turbe mõõt (kui eelneva kaalutud keskmise edasiarendus) on käsitlus järjestik-paralleelühenduses IT ja infoturbe protsesside (tegevusvaldkondade) turbe efektiivsusest.

Sobivamaks infoturbe meetrikaks on käideldavus (sest üldjuhul on teada vaid intsidentide arv aastas), kuid erinevalt tavalisest süsteemide käideldavuse käsitlusest, võib turbesüsteem mitteadekvaatselt toimida kahel põhjusel:

1. Turvasüsteemi enda rike – st süsteemi tehniline mittetoimimine.
2. On ilmnenud uued senitundmatud ründevariatsioonid ja turvasüsteem vajab ajakohastamist – st süsteemi kaitseomaduste mitteajakohasus.

Eelneva iseärasuse väljatoomiseks on ilmselt mõttekas kasutada "käideldavuse" asemel terminit "efektiivsus" – st infoturbe efektiivsus tähendab sisuliselt, et organisatsiooni IT ja infoturbe süsteemi kõik turvaeesmärgid on tagatud (põhiliselt info salastatus, terviklikkus ja käideldavus).

Analoogne lähenemine on ka ISO 27000-seerias: *Effectiveness* (ISO 27000 : 2008) - 4.13, mille tõlge eesti keelde võiks olla "efektiivsus on tase, kuivõrd kavandatud tegevused on ellu viidud ja kavandatud tulemused saavutatud".

Astmelise infoturbe mudelis kasutatav ühe konkreetse infoturbe tegevusvaldkonna ja ka kogu turvasüsteemi efektiivsus on:

infoturbemeetmete grupi või ka kogu infoturbesüsteemi võimekus osutada nõutud turbeteenust olemasolevatel tingimustel (konkreetsete nõrkuste ja ohtude, st konkreetse kaitsevõimekuse ja ründe aktiivsuse juures) etteantud perioodi jooksul (meie juhtumil aastas). Turbe efektiivsus on väljendatud kui tõenäosus vahemikus 0 kuni 1.

Tõrgete arv aastas (*annual failure rate*, AFR, või *annual rate of occurrence*, ARO) on tavaliselt ainuke olemasolev statistiline näitaja, mis annab meile keskmise tõrgete vahelise aja MTTF=1/AFR (aastat), kus MTTF – *Mean time to failure*.

Infoturbe spetsiifiliselt keskmine (*average*, av) Efektiivsus stabiilsusperioodil:
E_{av} = MTTSI/(MTTSI + MTTR + MTTU)
Sealjuures:
MTTSI – *Mean time to security incident* (keskmine turvaintsidentide vaheline aeg)
MTTR – *Mean time to repair* (keskmine rikke kõrvaldamise aeg)
MTTU – *Mean time to upgrade/update* (keskmine ajakohastamise aeg)
Kuna oleme suures osas põhinenud infoturbe tegevusvaldkondade efektiivsuste määratlemisel eksperthinnangutel, siis on varasemates artiklites (S_I – S_V) kasutatud terminit "kindlus" (*Confidence*) – st kui kindel on ekspert, et mingi konkreetne turvalahendus täidab oma turbefunktsiooni (tõenäosus 0 ÷ 1).

Algselt kasutasime kogu infoturbe süsteemi efektiivsuse näitajana komponentide efektiivsuste kaalutud keskmist (S_I – S_V). Kuid seoses graaf-mudeli kasutuselevõtuga, kus IT ja infoturbe süsteemi komponendid on ühendatud ühte järjestik- ja paralleelühenduste mudelisse, on organisatsiooni IT ja infoturbe üldine efektiivsus (*alias* käideldavus) arvutatav süsteemide käideldavuse kvantitatiivse analüüsi tehnikaid kasutades (S_VI, S_VII).

Järjestik(*serial*)-ühenduses elementidega süsteemi efektiivsusarvutused
$E_{Serial} = E_1 \times E_2 \times ... \times E_n$ (kui komponentide efektiivsused ($E_{1 \ldots n}$) on erinevad).
Seega graafi-põhises astmelise infoturbe mudelis on meil nn olulised järjestikused komponendid (IT ja infoturbe tegevusvaldkonnad), mille kohta kehtib nn keti põhimõte – kui üks komponent ei tööta, siis ei tööta terve süsteem.
NB! Kui $E_{\text{mistahes oluline}} = 0$, siis $E_{\text{kogu süsteem}} = 0$.

Rc-paralleelühenduses elementidega süsteemi efektiivsusarvutused
Infoturbes on olukord erinev komponentide tavalisest paralleelühendusest. Kui tugivaldkond on identne temaga paralleelse olulise turbe tegevusvaldkonnaga

(näiteks oluline tegevusvaldkond "Riistvara" ja tugivaldkond "Dubleeriv riistvara"), siis neile kehtib põhimõte „süsteem töötab, kuni töötab vähemalt üks komponentidest". Samas on üldjuhul olulistele tegevusvaldkondadele rakendatud ka paralleelsed tugivaldkonnad, mis aga pole täiesti identsed ja dubleerivad. Tavaliselt paralleelne tugivaldkond parendab tema poolt toetatava olulise valdkonna mingit ühte konkreetset funktsionaalsust (näiteks oluline valdkond "Riistvara", mida toetav "Logide haldus/monitooring" parendab ainult vea leidmise kiirust, kuid ei mõjuta vea parandamise kiirust) ja tegemist pole identselt dubleeriva olukorraga. Sellise olukorra kirjeldamiseks oleme sisse toonud dubleerimiskoefitsiendi (*Redundancy Coefficient*, R_C**)**. Seda teemat on esimest korda käsitletud töös S-VI.

Kui tavalisel paralleelühendusel $E_{1//2} = 1 - (1 - E_1)(1 - E_2)$,
siis olulise (*relevant*) IT või infoturbe teenust mitteidentselt dubleerival paralleelühendusel tugiteenusega (*supporting*) dubleerimiskoefitsiendiga R_C**:**
$E_{relevant//supporting} = 1 - (1 - E_{relevant})(1 - R_C * E_{supporting})$.

Reaalselt $R_C = 0{,}1 \div 1$. Täielikul dubleerimisel $R_C = 1$. Paralleelne tugiteenus dubleerimisega vähem kui 0,1 on enamasti lihtsalt mõttetu raha raiskamine.

Muidugi saaks osalise paralleelsuse kohe eksperthinnanguna tugitegevusvaldkonna efektiivsuses arvesse võtta, kuid see ei näita probleemi tegelikku kohta. Rc kasutamine tagab, et saame lahus vaadata oluliste ja tugivaldkondade meetmegruppide teostatuse taset (st tasemete efektiivsuste väärtusi) ja olulistele valdkondadele tugivaldkondade poolt osutatava toe taset.

Kogu IT ja infoturbe süsteemi efektiivsuse (*alias* käideldavuse) arvutused
Asendades rekursiivselt järjestik- ja paralleelühenduses süsteemi komponente ühe ekvivalentse komponendiga, määratleme lõpuks kogu IT ja infoturbe süsteemi efektiivsuse (*alias* käideldavuse). Uus graafi-põhine mudel avas meile võimalused sellisteks arvutusteks.

Meetrika kõige olulisemale optimeerimisjuhule „Tee asju õigesti" (*Do things right*)
Sisendiks on tegelikult võimalik ressurss (eelarve) infoturbeks ja väljundiks infoturbe kui süsteemi maksimaalne efektiivsus.

Seega on optimeerimise aluseks olev *"funktsioon": SE = f(SC)*,
kus SE – turbe efektiivsus (*Security Effectiveness*) ja SC – turbe kulutused ehk eelarve (*Security Costs*).

Funktsioon on jutumärkides ja kursiivis (analoogselt Olovsson 1992, *Figure* 11) rõhutamaks, et kasutame funktsiooni mõistet oluliselt lihtsustatuna – st *„funktsiooni"* all mõistame sõltuvust ainult meile peamist huvi pakkuvast näitajast

– põhiliselt ainult kulutustest infoturbele *f(SecurityCosts)* või *f(SecurityBudget)* (detailsem selgitus lk 16).

Lihtsaim võimalus kogu infoturbe süsteemi efektiivsuse arvutamiseks on näiteks teostatav tegevusvaldkondade efektiivsuste aritmeetiline keskmine, kuid käesolevas töös on kirjeldatud ja kasutatud paremini reaalset olukorda kirjeldav infoturbe kui äriprotsessi graaf-mudel ning selle järjestikuste ja paralleelsete protsesside käideldavus.

Vägagi levinud on ka sisuliselt sama meetrika väljendamine riski leevendamise määrana või siis aastas lubatava jääkriskina:

- riski leevendamise määr: mR (*mitigation Rate*) = 1 / (1-SE) = ALE/AL,
- potentsiaalne jääkrisk aastas (*Annual Loss*): AL = ALE /mR,
  kus ALE (*Annual Loss Expectency*) on potentsiaalne risk aastas ilma rakendatud turvameetmeteta ning AL (*Annual Loss)* on kas arvutuslik potentsiaalne jääkrisk või siis eelneva aasta tegelik infoturbe intsidentidest tekkinud kahju.

Juhtkonnale kulutuste optimeerimisel aluseks olev info oleks sellisel juhul ekspertsüsteemi poolt arvutatav väljund-*"funktsioon"*:
mRate = f(SC) või *AL = ALE /mR = f(SC)*.

Nende alusel on määratletav infoturbe konkreetsele eelarvele (konkreetsele SC väärtusele) vastav maksimaalne turbe efektiivsus ning selle vastav optimaalne turbe profiil.

Optimeerimisel on arvutuste maht ülisuur ja meie kahemõõtmelise optimeerimise (maksimaalne turbe efektiivsus minimaalsete kulutustega) reaalseks teostamiseks on välja töötatud evolutsioonilistel algoritmidel ja Pareto kõveral põhinev optimeerimismeetod. Samuti on välja töötatud eelnevat teostav CoCoViLa-põhine ekspertsüsteem: *Graded Security Expert System*.

Graafipõhine astmeline infoturbe mudel (g_AIM, *gb_GSM*) kirjeldab organisatsiooni IT ja infoturbe süsteemi piisavalt täpselt, et võimaldada infoturbe kulutuste optimeerimist. Põhinedes lihtsamal ja samas ka arusaadavamal (n-ö ühe pilguga haaratavaal) graaf-mudelil, mis lähtub IT-vaatest infoturbele, võimaldab seda teostav ekspertsüsteem suurusjärgu võrra väiksema tööga kätte saada just selle, mida organisatsiooni juhtkond infoturbe valdkonnas otsustamiseks kõige enam vajab: hästi visualiseeritud väljund (Pareto kõver) turbe efektiivsuse ja turbele kulutatud ressursside vahelisest sõltuvusest – *SecurityEffectiveness = f(SecurityCosts)*.

Ekspertsüsteem infoturbe kulutuste optimeerimiseks (GSES)

Tuleb rõhutada, et optimeerimine on väga arvutusmahukas. Et saada ettekujutust mudeli ja kogutud alusinfo põhistest arvutuslikest lõpptulemustest ja hinnata nende tegelikkusele vastavust, peame olema suutelised optimeerimisarvutusi ka reaalselt teostama – st vajalik on optimeerimist teostav tarkvara.

Graafi-põhise astmelise infoturbe mudeli jaoks tarkvara prototüübi, Astmelise infoturbe ekspertsüsteemi (GSES), väljatöötamisel on tarkvara platvormiks valitud mudelipõhine tarkvara arendusplatvorm CoCoViLa, mis on sobiv platvorm ekspertsüsteemide väljatöötamiseks. Optimeerimise tööriista, st astmelise infoturbe ekspertsüsteemi (GSES), prototüüp on CoCoViLa pakett, mis sisaldab astmelise infoturbe mudeli kirjeldamiseks ja edasiseks optimeerimiseks loodud spetsiaalset visuaalset keelt. Konkreetse mudeli või süsteemi kirjeldamiseks kasutab ekspert juba visuaalset programmeerimist. Visuaalne programmeerimine on sisuliselt väga kasutajasõbralik graafiline liides, mis ei nõua spetsiaalseid programmeerimisoskusi – konkreetse organisatsiooni infoturvet kirjeldava mudeli, optimeerimisülesande ja soovitud graafiliste väljundite kirjeldamine on lihtne nagu legoklotsidest mingi asja kokkupanek. Teemat on käsitletud põhjalikumalt S_I.

CoCoViLa platvormi kasutamise kasuks on mitmeid argumente:

- CoCoViLa töötab Windowsi, Linuxi ja Maci platvormidel, tehtud rakendused ei vaja mingeid muutmisi tööks neil platvormidel.
- CoCoViLa on vaba tarkvara, mille lähtekood on jaotatav vastavalt GNU Üldise Avaliku Litsentsi (GNU *General Public License*, GNU GPL) põhimõtetele.
- Programmeerimiskeel rakenduste tegemiseks on programmeerijate hulgas laialt kasutatav Java.
- Konkreetse mudeli või süsteemi kirjeldamiseks kasutab ekspert juba visuaalset programmeerimist.
- Graafiline kasutajaliides ja visuaalne programmeerimine tagavad, et vajalike muutuste sisseviimine mudelisse ja ekspertsüsteemi on lihtne ja operatiivselt teostatav.
- Visuaalne väljund on ühe pilguga haaratav ning sobib suurepäraselt ekspertidele oma turbelahenduste otstarbekuse ja optimaalsuse selgitamiseks juhtkonnale.

Peamised tulemused

1. Kirjeldatud on infoturbe kulutuste optimeerimiseks sobiv graafi-põhine Astmelise infoturbe mudel, mis põhineb kahel kesksel lähenemisel (II.1.):
   - Äriprotsesside kirjeldamiseks üldkasutataval Personal-Protsess-Tehnoloogia käsitlusel ja
   - IT ja infoturbe tegevusvaldkondade jagunemine olulisteks tegevusvaldkondadeks ja Rc-dubleerivateks tugitegevusvaldkondadeks.

IT ja infoturbe kui protsessi tegevusvaldkondade jaotamisel olulisteks ja tugiteenusteks on kaks väga olulist aluspõhimõtet:

a) Infoturve on nagu kett, mille tugevuse määrab selle nõrgim lüli – st infoturve on heal tasemel, kui selle kõik olulised (*relevant*) tegevusvaldkonnad on heal tasemel. Kui oluline teenus ei toimi, siis ei toimi kogu IT ja infoturve kui teenus. St olulised teenused on n-ö järjestikühenduses.

b) Infoturve on mitmetasemeline (*Multilevel Security, Defence in Depth*) – st tugiteenused on mõeldud oluliste teenuste parendamiseks ning on mudelis olulistele teenustele n-ö paralleelsed.

2. Kirjeldatud on infoturbe kulutuste optimeerimiseks sobiv meetrika ja kolm võimalikku infoturbe optimaalsuse astet (II.2):
   - Tee mõttekaid asju (*Do rational things*) – st tagada nõutud turvaklass ja mitte turvata rohkem kui nõutud (see oleks raha raiskamine) ja mitte ka vähem kui nõutud (kahjud turvaintsidentidest läheksid liiga suureks), töömaht ligikaudu 1–2 päeva.
   - Tee asju õigesti (*Do things right*) – st kasutada ressursse optimaalselt, et tagada maksimaalne turbe efektiivsus reaalselt olemasolevate ressurssidega (raha, aeg, persinal); töömaht ligikaudu 1–2 nädalat.
   - Tee õigeid asju (*Do right things*) – st määratleda optimaalne turvaprofiil, et oleksid tagatud minimaalsed kulud infoturbele (st minimaalne turvameetmetele ja personalile kulutuste ning turvaintsidentidest tingitud kahjude summa); globaalne optimum; töömaht ligikaudu 1–2 kuud.

3. Välja on töötatud on algoritmid infoturbe efektiivsuse arvutamiseks, valitud optimeerimiseks sobiv meetod (gb_GSM/GSES-method; II.3. – II.7.) ning on loodud eelnevat teostav CoCoViLa-põhine visuaalse programmeerimisega ja visuaalse graafilise väljundiga ekspertsüsteem – GSES.

Võib öelda, et käesoleva töö peaeesmärk on täidetud: välja on töötatud graafi-põhisel astmelise infoturbe mudelil ja astmelise infoturbe ekspertsüsteemil põhinev organisatsiooni infoturbe kulutuste optimeerimise meetod.

Mudel on eriti sobiv kui abivahend astmelistele infoturbe standarditele nagu näiteks NISPOM 2006, NIST SP 800-53 r4 või ISKE v6.0. Need infoturbekesksed mitmetasemelised (astmelised) standardid lihtsustavad oluliselt leitud optimaalse

turbe profiili teisendamist vastavaks konkreetsete turvameetmete loeteluks. Muidugi on võimalik vastav standard või mudel ka ise kirjeldada, kuid selleks vajalik tööde maht kujuneb juba vägagi suureks.

Võin öelda, et väljatöötatud mudel on just see, mille järele praktikuna puudust tundsin – st mudel võimaldab määratleda ka väikestele ja keskmistele organisatsioonidele vastuvõetava töömahuga (1–2 inimkuud) optimaalsed infoturbe kulutused ja nendele otseselt vastavad vajalikud turvameetmed.

Samas GSES sobib ka suvalise äriprotsessi optimeerimiseks, kui

- oskame sellele konkreetsele äriprotsessile kirjeldada graaf-mudeli;
- allprotsessid on astmeliselt kirjeldatavad ja teostatavad ning kulutuste ja efektiivsuste väärtused määratletavad.

Infoturbe kulutuste optimeerimisest Eesti avaliku sektori organisatsioonides

Infoturbe valdkonnas on esimene ja kõige suurem risk ressursipuudus – st rahapuudus ja/või heade spetsialistide puudus (mis sisuliselt taandub ikkagi rahale, millega personali palgata). Infoturbeks vajalike ressursside piiratus on välja öeldud isegi NATO tasemel. Kuid just väikeste ja keskmiste organisatsioonide jaoks on ressursside puudus väga üldlevinud. Ülemaailmses mastaabis on aga kõik asutused Eestis väikesed või keskmised. Esmane lahendus ressursipuuduse korral on infoturbe kulutuste optimeerimine ja infoturbega seotud kulude minimeerimine.

Vabariigi Valitsuse ISKE määrusega «Infosüsteemide turvameetmete süsteem" kehtestatakse riigi ja kohaliku omavalitsuse andmekogudes sisalduvate andmekoosseisude töötlemiseks kasutatavate infosüsteemide ning nendega seotud infovarade turvameetmete süsteem.

Turvameetmete süsteemi ISKE rakendamine seisneb:

- infoturbe eesmärkidele vastavate turvaklasside määramises;
- nendele vastavate turvameetmete valimises vastavalt infosüsteemide kolmeastmelise etalonturbe süsteemi (ISKE) rakendamisjuhendile;
- nende rakendamises ning rakendamise auditeerimises.

ISKE kohustuslikkus jõustus 01.01.2008 ja auditeerimiskohustus 25.01.2009.
Samas kui avalik organisatsioon vastavalt vabariigi valitsuse määrusele rakendab andmekogude info turbeks ISKEt, siis on väga otstarbekas tehtut kasutada organisatsiooni kogu info turbeks. Veidi lihtsustatult võib öelda, et kõik Eesti avalikud organisatsioonid (välja arvatud riigisaladust töötlevad) on kohustatud infoturbeks kasutama ISKEt.

Avalikele organisatsioonidele kehtib põhimõte – mis pole seadusega lubatud, on keelatud.

Seega erinevused infoturbe mudeli valikuks era- ja avalikus erasektoris on järgmised:

1. Iga erasektori organisatsioon otsustab ise, millist infoturbe standardit ja/või mudelit ja/või meetodit kasutada, väga soovitud on optimaalne infoturve ja minimaalsed infoturbe kogukulud.
2. Avalikus sektoris on infoturbe mudel ja eelarve üldjuhul valitsuse poolt määratletud ning teostus kontrollitav/auditeeritav, st Eestis kohustuslik ISKE ja selle ratsionaalse optimaalsuse tase.

ISKE ratsionaalse optimumi puudused:

1. ISKE turvameetmete loetelu aluseks on võetud üks-üheselt turvameetmed Saksa Liitvabariigi infoturbe mudelist BSI (~1000 turvameedet) ning juurde lisatud kõrge taseme meetmed (~500). St Eesti riigiasutus oma ~100 korda väiksemate ressurssidega peaks olema suuteline teostama oluliselt enam turvameetmeid kui Saksa riigiasutus? Ilmselgelt küllaltki lootusetu olukord – suure mehe ülikond päris kindlasti ei sobi väikesele mehele. Praktikas jätavad Eesti riigiasutused ressursside puudusest tingitult sadu BSI/ISKE turvameetmeid lihtsalt realiseerimata (info ISKE audititest).
2. Ratsionaalne optimum on vägagi kaugel tegelikust optimaalsest. Näiteks esimese juhtumiuuringu tulemused (Lauri Palkmetsa magistritöö „Astmelise infoturbe mudeli rakendatavus Eesti Kaitseväes“): tase L (*Low* – Madal) andis võimalikust maksimaalsest riskide leevenduskoefitsiendist ~8 korda madalama tulemuse ja tase M (*Medium* – Keskmine) ~4 korda viletsama tulemuse (vt *Table* 15). Teema vajab kindlasti täiendavat uurimist. Eesti avaliku sektori kümnetesse miljonitesse eurodesse ulatuvad infoturbekulutused on selleks ilmselt piisavaks põhjuseks ja põhjenduseks.

Olulised erinevused era- ja avalikus sektoris potentsiaalsete infoturbe intsidentidest tingitud kahjude määratlemises:

- Erasektor – kahjud on määratletud riskide hinnangutega riskianalüüsis, mida eraorganisatsioonid valdavalt nagunii teostavad. Riskianalüüsi väike kohendamine infoturbe kulutuste optimeerimise meetodile vastavaks pole üldjuhul probleemiks.
- Avalik sektor – kahjud määratleb avalik huvi. Avalikku huvi on üldjuhul võimalik ka rahas määratleda, kuid selleks peab avalikul organisatsioonil olema vastav seaduslik kohustatus (ja ressurss teostamiseks). Seda kohustust Eesti avalikel organisatsioonidel pole ja praktiliselt pole Eestis ka tehtud ühtegi avaliku huvi rahalise väärtuse määratlemist.

ISKE parendamise võimalused:

1. Iga infoturbe standard, sealhulgas ka ISKE, vajaks abivahendit, mis võimaldaks arvutada reaalselt saavutatud turvataset ja optimeerida kulutusi infoturbele.
2. Oleks väga otstarbekas läbi viia mõned konkreetsed kasutusuuringud konkreetsetele riigiasutustele (nt suur ministeerium, väike ministeerium,

kohalik omavalitsus) – leida ISKE L/M/H-tasemete turbe maksumused ja efektiivsused ning võrrelda neid sama ressursi/raha eest saavutatavate optimaalsete efektiivsustega. Sisuliselt üritaks ISKE viia optimeerimise tasemele „tee õigeid asju“ – st minimaalsed kulutused infoturbele.

3. Avaliku sektori asutustele on vägagi probleemne senini olnud oma teenuste väärtuste rahaline hinnang. Seetõttu on avalik sektor sunnitud piirduma optimeerimise tasemega „tee asju õigesti“. Väga vajalik oleks riiklik tellimus avalike organisatsioonide teenuste jaoks avaliku huvi rahalise väärtuse määratlemiseks, mis sisuliselt võimaldaks ISKE viia optimeerimise tasemele „tee õigeid asju“ – st määratleda optimaalne infoturbe profiil, et oleksid tagatud minimaalsed kulud infoturbele (st minimaalne infoturbe kulutuste ning turvaintsidentidest tingitud kahjude summa).
4. ISKE auditites lepitakse hetkel küllaltki subjektiivselt kokku mingite turvameetmete mitteotstarbekuses. Astmelise mudeliga saab arvutada nõutud ja auditi käigus kokkulepitud turbe profiilide turbe efektiivsused, st määratleda kui palju turbes tegelikult rahaliselt kaotame – kokkulepete subjektiivsus kaoks.

Ehk tekitab käesolev töö Riigi infosüsteemi ametis (RIA), kelle üheks otseseks kohustuseks on Eesti avaliku sektorile infosüsteemide turvameetmete süsteemi arendus, huvi ja teoreetilist kindlustunnet ISKEga (st Eesti avaliku sektori infoturbega) seotud kulutuste optimeerimise teema käsitlemiseks.

Teemad edasisteks uuringuteks ja arendusteks

Ettepanekud mudeli arendusteks:

1. Paralleelsuskoefitsient (Rc) on tõenäoliselt pigem funktsioon kui konstant, sõltub ilmselt tugitegevusvaldkonna enda turvatasemest ja toetatava olulise tegevusvaldkonna turvatasemest. Vajab täpsemat uurimist.
2. Edasist uurimist vääriksid optimumile väga lähedased turbe variatsioonid, mille erinevus n-ö *tõelisest optimumist* on vaid kuni 1–2%. Kuna ekspertefektiivsused on nagunii määratletud võimaliku veaga ligikaudu ± 20%, siis on olemas võimalus, et mõni optimumile lähedane turbeprofiil on hoopis tegelik reaalne optimum. Seetõttu oleme neid optimumilähedasi turbeprofiile nimetanud ka alternatiivseteks optimaalseteks turbeprofiilideks. Alternatiivsete optimaalsete turvaprofiilide (st optimaalsetele väga lähedaste ja seega ka tegelikult võimalike optimaalsete) käsitlus tuleks mudelisse sisse tuua.
3. Praegune mudel vaatleb organisatsiooni infosüsteemi kui ühte integreeritud infosüsteemi, kuid võiks teha ekspertsüsteemi multiinfosüsteemide-versiooni, mis käsitleks organisatsiooni infoturvet juba infosüsteemide tasemel.
4. Välja tuleks töötada organisatsiooni tasemel küberkaitse kulutuste optimeerimise mudel – siin põhimõttelisi probleeme pole, tuleb ainult sisse

tuua mõned täiendavad turbe-eesmärgid (näiteks salgamise vääramine, autentsus, missioonikriitilisus).

5. Küllaltki huvitav ja täiendavat uurimist vajav teema (alustatud S_IV) on turbe efektiivsus olukorras, kus on tegemist mingis valdkonnas saavutatud taseme mingil põhjusel langetamisega. Üldjuhul on tegemist raha raiskamisega, sest taseme saavutamise kulutused on keskmiselt vähemalt 2–3 korda suuremad kui taseme hoidmise kulutused. Kuid muutused firma plaanides on alati võimalikud ning need võivad tingida vajaduse muutusteks ka infoturbes.
6. Infoturbe intsidentidest tingitud kahjude rahas määratlemine avalikule sektorile (sh ka militaarsektorile) on jäänud täiendavat uurimist vajavaks probleemiks.
7. Organisatsiooni IT-kulude optimeerimine: kuna välja töötatud graaf-mudel kirjeldab sisuliselt organisatsiooni kogu infotöötlust kui üht organisatsiooni äriprotsessi, siis on vägagi võimalik käesoleva organisatsiooni infoturbe kulutuste optimeerimiselt astuda samm edasi ning välja töötada analoogne mudel ja ekspertsüsteem organisatsiooni kõigi IT-kulutuste optimeerimiseks. Seega võiks järgmiseks uurida "IT-kulude optimeerimist", sest IT ja infoturbe kuludel on vägagi raske vahet teha. Praktiliselt iga IT-kulutus sisaldab endas ka olulist infoturbe komponenti. Optimeerimiskriteerium sel juhul mõistagi muutub, selleks võiks olla näiteks "minimaalsete kulutustega vajalik IT funktsionaalsus" vmt. Esimeseks eeskujuks ja ideede allikaks sobib COBIT, mis ongi just organisatsiooni IT-juhtimise mudel.

Vajalikud arendused mudelile parema alusinfo saamiseks:

1. Peaks arvestama ka ründe tõenäosusi, kuid reaalselt meil seda infot pole – lihtsalt on eeldatud, et kaitsmata või halvastikaitstud väärtuslikku infot kindlasti rünnatakse.
   Võiks kaasata midagi *à la* Duffany ründepuu (Duffany 2007), kuid samas muudab see mudeli ilmselt oluliselt keerukamaks, seega ka töömaht kasvab oluliselt.
2. Uurida turvaintsidentidest tingitud kahjude rahas määratlemise võimalusi avalikus sfääris, sealhulgas spetsiaalselt ka militaarsfääris.
   Peaks arvestama ka turvaintsidentidest tekkinud kahjusid (rahas mõõdetuna), kuid reaalselt meil seda infot praegu pole. Probleemiks on sobiva mudeli ja arvutusmeetodite määratlemine.

Arendused mudeli organisatsioonides juurutamise lihtsustamiseks:

1. Ühildada ja integreerida mudeli ja organisatsioonis kasutatava raamatupidamis-, majandus- ja riskihaldustarkvarade põhilised infokäsitlused – st mudelisse info kuludest otse raamatupidamisrakendusest, kahjudest riskihaldusrakendusest jne.
2. Amortisatsiooni põhimõtted ekspertsüsteemi sisse tuua, st ligikaudu iga viie aasta järel on nõutav uus investeering. Kuna eri tegevusvaldkondades on tõenäoliselt erinev nn jooksev aasta, siis on vaja see ka ekspertsüsteemi sisse tuua.

3. Praegune mudel vaatleb olukorda, kus IT pole firma toode või teenus. IT kui toote või teenuse korral tuleks ilmselt sisse tuua mitmeid täiendavaid majanduslikke näitajaid, mis on vajalikud osutatava teenuse või toote majanduslikuks juhtimiseks.

NB! Kõik need eelpool välja pakutud arendused nõuavad asjakohaseid muudatusi ka GSES-i tarkvaras.

CURRICULUM VITAE

Personal Information

Name:	Jüri Kivimaa
Date and place of birth:	10.09.1948, Otepää, Estonia
Citizenship:	Estonia
Permanent address	Järve, 48-11, Tallinn, 11314, Estonia
Phone (mobile)	+372 51 89531
Email:	jyri.kivimaa@gmail.com

Education

2010 – now	Estonian Business School (EBS), Estonia, PhD studies
1966 – 1972	Diploma Engineer in Electronics (equal to the Master) at Tallinn University of Technology (*Tallinna Pol*ütehniline *Instituut*)

Professional Employment

2010 – now	Lecturer in Tallinn University of Technology
2007 – 2012	Scientist at the NATO Cooperative Cyber Defence Centre of Excellence
1998 – 2007	*Eesti Ühispank*/SEB Estonia, IT Department, system analyst, IT security expert
1997 – 1998	Estonian Savings Bank, IT Division, project manager of money and capital markets
1995 – 1997	Estonian Social Insurance Agency, Tallinn Bureau, deputy manager
1994 – 1995	Estonian Social Insurance Agency, leading IT specialist
1994 – 1994	Estonian Health Insurance Fund, chief IT specialist
1992 – 1994	LanProfit Limited, director
1989 – 1992	ESSR Academy of Sciences, The Institute of Cybernetics, The Development Company "Kompakt", leading engineer-programmer
1986 – 1988	The Tallinna New Port, Data Center, leading engineer-programmer
1981 – 1986	ESSR Ministry of Health, Computing Center, chief engineer
1979 – 1981	Tallinn Scientific-Production Center „Algoritm", head of laboratory
1975 – 1979	ESSR Ministry of Finance, Information and Computing Center, electronics engineer, group leader

1972 – 1975	ESSR National Planning Committee, Computing Center, electronics engineer

Special Courses and Conferences

2011 (7–8 July)	10th European Conference on Information Warfare and Security, The Institute of Cybernetics at the Tallinn University of Technology, Tallinn, Estonia
2010 (1–2 July)	9th European Conference on Information Warfare and Security, University of Macedonia, Thessaloniki, Greece
2010 (28–30 September)	NATO Information Assurance Symposium (NIAS) 2010, Mons, Belgium
2009 (21–25 September)	NATO Information Assurance Symposium 2009, Mons, Belgium
2009 (6–7 July)	8th European Conference on Information Warfare and Security; Military Academy Lisbon and the University of Minho Braga, Lisbon, Portugal
2009 (12–15 May)	11th NATO CYBER DEFENCE WORKSHOP, Athens, Greece
2008 (17–19 November)	MILCOM:2008, San Diego, USA
2008 (6–10 October)	NATO Information Assurance Symposium 2008, Mons, Belgia
2008 (30 June–1 July)	7th European Conference on Information Warfare and Security; University of Plymouth, UK
2007 (12–16 November)	NATO Information Assurance Symposium 2007, Mons, Belgia
2007 (5–8 November)	Concept Development & Experimentation Conference, Istanbul, Turkey
2007 (29–31 October)	8th NATO CyberDefence Workshop, Rooma, Italy
2007 (2–3 July)	6th European Conference on Information Warfare and Security; Defence College, Shrivenham, UK
2007 (2– 5 April)	Computer Network Attacks and Defence; Petersburg Institute of Informatics and automation of the Russian Academy of Sciences
2006 (19–20 December)	Wireless Hacking, PIT Consulting Estonia
2006 (13–17 November)	Countering cyber terrorism course, NATO CoE DAT, Ankara, Turkey
2005 (21–23 March)	ITIL Foundation for IT Service Management; HP Education
2005 (07– 08 March)	Hands On Hacking II; Domina Security

2004 (23– 24 March)	Hands On Hacking; Domina Security
2003 (18–19 June)	Hands On Hacking; Domina Security
2002 (16–17 September)	Essential Security II; Domina Security
2002 (23–24 May)	Hacking and Internet Based Attacks; Domina Security
2002 (14–15 May)	Essential Security; Domina Security
2002 (16–17 April)	ISO 17799 Information security; Domina Security
1996 (29.01–02.02)	Developer 2000 : Application Design, Forms and Reports Generator; AboBase Systems
1995 (11–15 December)	CASE Method overview and Detailed System Analysis; AboBase Systems
1995 (20–23 November)	Administering an ORACLE7 Database; AboBase Systems
1995 (6–10 November)	Designer 2000 : System modelling and Data Design; AboBase Systems
1995 (20–21 April)	Project work methodology, Governmental Data Center
1995 (01–04 September)	ORACLE 7 : SQL and PL/SQL; AboBase Systems

Scientific Work

Academic Publications:

Alberghs, Geert; Grigorenko, Pavel; Kivimaa, Jyri (2011). Quantitative system reliability approach for optimizing IT security costs in an AI environment. *In: 12th Symposium on Programming Languages and Software Tools, SPLST'11 : Tallinn, Estonia, 5–7 October 2011.* Tallinn: TUT Press, 2011, 219–230.

Kivimaa, Jüri; Kirt, Toomas (2011). Evolutionary Algorithms for Optimal Selection of Security Measures . *In: Proceedings of the 10th European Conferences on Information Warfare and Security: 10th European Conference on Information Warfare and Security ECIW-2011; Tallinn, Estonia; 7–8 July 2011.* Readings, UK: Academic Publishers, 2011, 172–184.

Kirt, Toomas; Kivimaa, Jüri (2010). Optimizing IT security costs by evolutionary algorithms. *In: Conference on Cyber Conflict Proceedings 2010: Conference on Cyber Conflict; Tallinn, Estonia; 15–18 June, 2010.* Tallinn: Cooperative Cyber Defence Centre of Excellence Publications, 2010, 145–160.

Kivimaa, Jüri (2009). Applying a Cost Optimizing Model for IT Security. *In: Proceedings of the 8th European Conference on Information Warfare and Security: Lisbon, Portugal, 6–7 July, 2009.* Reading, UK: Academic Conferences Limited, 2009, 142–153.

Kivimaa, Jyri; Ojamaa, Andres; Tyugu, Enn (2009). Managing evolving security situations. *In: MILCOM 2009 : Unclassified Proceedings, 18–21 October, 2009, Boston, MA.* Piscataway, NJ: IEEE, 2009, 1–7.

Kivimaa, Jüri; Ojamaa, Andres; Tyugu, Enn (2009). Graded security expert system. *In: Critical Information Infrastructures Security : Third International Workshop, CRITIS 2008, Rome, Italy, 13–15 October, 2008, Revised Papers.* Berlin: Springer, 2009, (Lecture Notes in Computer Science; 5508), 279 –286.

Ojamaa, Andres; Tyugu, Enn; Kivimaa, Jyri (2008). Pareto-optimal situation analysis for selection of security measures. *In: MILCOM 08 : Assuring Mission Success : Unclassified Proceedings, 17–19 November, 2008.* San Diego: 2008, 3224–3230.

Kivimaa, Jüri; Ojamaa, Andres; Tyugu, Enn (2008). Graded security expert system. *In: CRITIS 2008 : Third International Workshop on Critical Information Infrastructure Security, Villa Mondragone, Monte Porzio Catone, Rome, 13–15October, 2008, (Pre-Proceedings).* AIIC, ENEA, 2008, 333–339.

Ojamaa, Andres; Tyugu, Enn; Kivimaa, Jüri (2008). Modeling and optimizing graded security measures. *In: Info- ja kommunikatsioonitehnoloogia doktorikooli IKTDK kolmanda aastakonverentsi artiklite kogumik : 25.–26. aprill 2008, Voore külalistemaja.* Tallinn: Tallinna Tehnikaülikool, 2008, 123–125.

Program Committee Member of the Following Conferences:

9th European Conference on Information Warfare and Security (ECIW 2010), University of Macedonia, Thessaloniki, Greece.

10th European Conference on Information Warfare and Security (ECIW 2011), The Institute of Cybernetics at the Tallinn University of Technology, Tallinn, Estonia

11th European Conference on Information Warfare and Security (ECIW 2012), The Institute Ecole Supérieure en Informatique, Electronique et Automatique, Laval, France

12th European Conference on Information Warfare and Security (ECIW 2013), University of Jyväskylä, Jyväskylä, Finland

International Conference on Cloud Security Management **(**ICCSM 2013), University of Washington, Seattle, USA

Academic teaching:

2009 – now	IT Security and Cyber Security assurance in organization, Master's Cource (Master of Cyber Security) in Tallinn University of Technology

Supervizing master theses:

2013	Andres Järv and Davit Agniashvili, Tallinn University of Technology, Cyber Security
2012	Ben Othman and Predrag Tasevski, Tallinn University of Technology, Cyber Security
2011	Geert Alberghs, ESIEA, Paris-Laval
2010	Lauri Palkmets and Heigo Tark, Tallinn University of Technology, Cyber Security
2008	Mehis Ottis, Tallinn University of Technology, Cyber Security
2007	Ahto Saks, Estonian Business School, IT management


9 789949 944736