IT Architecture-Based Confidentiality Risk Assessment in Networks of Organizations

Ayşe Moralı

IT Architecture-Based Confidentiality Risk Assessment in Networks of Organizations

Ayşe Moralı

Composition of the PhD dissertation committee:

Prof. dr. A.J. Mouthaan	Universiteit Twente, NL (chairman and secretary)
Prof. dr. R.J. Wieringa	Universiteit Twente, NL (promotor)
Prof. dr. S. Etalle	Universiteit Twente, NL (promotor)
Prof. dr. P.H. Hartel	Universiteit Twente, NL (intern lid)
Prof. dr. W. Jonker	Universiteit Twente, NL (intern lid)
Prof. dr. F. Massacci	Universita di Trento, IT (extern lid)
Prof. dr. E.R. Verheul	Radboud University Nijmegen, NL (hoogleraar/UHD)
Dr. S.H. Houmb	SecureNOK Ltd., NO (referent)

This Research is conducted within the VRIEND Project supported by STW and University of Twente.

UNIVERSITY OF TWENTE.

Distributed and Embedded Security Group
P.O. Box 217, 7500 AE, Enschede, The Netherlands.

This research is supported by the research program Sentinels of STW (http://www.sentinels.nl), under the project number 07635.

CTIT PhD Thesis Series Number 11-197
Centre for Telematics and Information Technology (CTIT)
P.O. Box 217-7500 AE Enschede-The Netherlands.

IPA: 2011-06
The work in this thesis has been carried out under the auspices of the research school IPA (Institute for Programming research and Algorithms).

ISBN: 978-90-365-3165-8
ISSN: 1381-3617
DOI: 10.3990/1.9789036531658
URL: http://dx.doi.org/10.3990/1.9789036531658
Cover design: Michaël Sterckx and Ayşe Moralı
Printed by Wöhrmann Print Service.

IT ARCHITECTURE-BASED CONFIDENTIALITY RISK ASSESSMENT IN NETWORKS OF ORGANIZATIONS

DISSERTATION

to obtain
the doctor's degree at the University of Twente
on the authority of the rector magnificus,
prof. dr. H. Brinksma,
on account of the decision of the graduation committee,
to be publicly defended
on Thursday, 21st of April 2011 at 16.45.

by

Ayşe Moralı

born on 07th of January 1978,
in Istanbul, Turkey

The dissertation is approved by:

Prof. dr. R.J. Wieringa	Universiteit Twente (promotor)
Prof. dr. S. Etalle	Universiteit Twente (promotor)

Bu tezi
canım anneciğme and babacığıma
ithaf ediyorum.

Abstract

Today almost every organization benefits from business opportunities created by digitalization. Digitalization allows, among others, to develop software products on shared platforms, to remotely access and alter patient records or remotely control power generators. This change in the technical environment has triggered changes in the legal environment, and introduced new compliance requirements. Consequently, protecting the confidentiality of digital information assets has become a major concern for many organizations. This concern is even bigger for organizations that connect their IT system with other organizations to reduce costs.

Risk assessment methodologies provide stakeholders with sound knowledge on security risks that threaten the business. A risk assessment method should satisfy three conflicting requirements: accuracy, cost-efficiency, and inter-subjectivity. These three requirements form the dilemma of confidentiality risk assessment methods. Accuracy has to do with the level of granularity that a method allows when assessing the risk. Cost-efficiency is the crucial real limitation of all risk assessment methods. In practice, even risk assessments of large and information-intensive company sections rarely last longer than two weeks. The third requirement we look at in this dissertation is inter-subjectivity. Nowadays, despite the large use of standardized methods, the very result of a risk assessment is largely subjective, in the sense that other assessors may assess risks differently. This lack of inter-subjectivity means that risk assessments are difficult to replicate and risk assessment results are not comparable.

Based on the dilemmas of confidentiality risk assessment methods, in this dissertation we propose five IT confidentiality risk assessment and evaluation methods, each of which extends the previous one. More specifically we present:

Extended eTVRA extends the eEurope secure and trusted architecture threat, vulnerability, and risk assessment (eTVRA) method with an information elicitation and structuring step. eTVRA is a model-based method specifically developed for telecom systems. This extension aims at assessing security risks of complex IT systems more accurately than checklist-based approaches.

DCRA is a model-based confidentiality method that is automated with a computational tool. It models the information system based on the IT architecture the system

relies on, so that one can analyze how confidentiality breaches can propagate through the IT components of the system. DCRA aims at assessing confidentiality risks of complex IT systems more accurately than checklist-based approaches.

CRAC is a model-based confidentiality risk assessment method that sorts and compares two alternative technical solutions according to their risks. It analyzes risks according to where in the IT architecture information is accessible (information flow) and how difficult it is for different attackers to access it (attack paths). CRAC aims at increasing the inter-subjectivity of assessment results while reducing the assessment costs.

CRAC++ extends CRAC by gaining control over the confidentiality requirements in a network of organizations. Thus, it delivers a set of confidentiality control requirements that can be used for extending SLAs. CRAC++ aims at adapting IT architecture-based confidentiality RA methods to control confidentiality risks.

RiskREP is a risk-based security requirement elicitation and prioritization method, which is meant to be used for systems that are under development. It links business goals to IT risks based on the IT architecture. RiskREP aims at eliciting assessment-relevant information cost-efficiently.

We validate and evaluate these methods in seven real world case studies at multinational companies from telecommunications, electronics and chemical industries. The results indicate that multinational organizations that are connected to other organizations by means of digitalization can benefit from IT architecture-based confidentiality risk assessment. The methods we propose show that assessing risks based on IT architecture (1) helps to reduce the assessment costs, (2) allows one to adjust the accuracy according to the business-criticality of a system and (3) increases the inter-subjectivity of qualitative risk assessment results.

Samenvatting

Tegenwoordig haalt bijna elke organisatie voordeel uit de bedrijfsmogelijkheden van digitalisering. Digitalisering laat o.a. toe om softwareproducten op gedeelde platformen te ontwikkelen, om vanop afstand medische dossiers te raadplegen en wijzigen of elektriciteitsgeneratoren te controleren. Deze verandering in technische omgeving heeft geleid tot veranderingen in de juridische omgeving, en introduceerde nieuwe conformiteitseisen. Bijgevolg is de bescherming van de vertrouwelijkheid van digitale informatiebronnen een grote zorg geworden voor vele organisaties. Dit probleem is zelfs nog groter voor organisaties die hun IT systeem met andere organisaties verbinden om de kosten te verlagen.

Methoden voor *risk assessment* voorzien stakeholders van grondige kennis over de veiligheidsrisico's die het bedrijf bedreigen. Een methode voor risk assessment moet aan drie conflicterende vereisten voldoen: nauwkeurigheid, kostenefficiëntie, en intersubjectiviteit. Deze drie vereisten vormen het dilemma van de methoden voor risk assessment van vertrouwelijkheid. Nauwkeurigheid heeft te maken met de mate van verfijning die een methode toelaat bij het beoordelen van het risico. Kostenefficiëntie is de cruciale echte beperking van alle risk assessment methoden. In de praktijk duren risk assessments zelden langer dan twee weken, zelfs voor grote en informatie-intensieve bedrijfsafdelingen. De derde vereiste waarover dit proefschrift handelt is intersubjectiviteit. Tegenwoordig is het resultaat van een risk assessment grotendeels subjectief, ondanks het overwegend gebruik van gestandaardiseerde methoden, in die zin dat verschillende assessoren de risico's anders kunnen bepalen. Dit gebrek aan intersubjectiviteit betekent dat risk assessments moeilijk te herhalen zijn en hun resultaten niet vergelijkbaar.

Gebaseerd op de dilemma's van de methoden voor risk assessment van vertrouwelijkheid, stellen we in dit proefschrift vijf methoden voor de assessment en evaluatie van IT vertrouwelijkheidsrisico's voor, waarbij elke methode een uitbreiding is van de vorige. Meer specifiek stellen we voor:

Extended eTVRA breidt de eTVRA-methode (eEurope secure and trusted architecture threat, vulnerability, and risk assessment) uit met een stap voor informatie-elicitatie en -structurering. eTVRA is een modelgebaseerde methode speciaal on-

twikkeld voor telecomsystemen. Deze uitbreiding heeft als doel het nauwkeuriger beoordelen van veiligheidsrisico's in complexe IT systemen dan op basis van checklistgebaseerde benaderingen.

DCRA is een modelgebaseerde vertrouwelijkheidsmethode geautomatiseerd met een computationele tool. Ze modeleert het informatiesysteem op basis van de IT architectuur waarop het systeem gebaseerd is, zodat men kan analyseren hoe inbreuken op de vertrouwelijkheid zich voortplanten doorheen de IT componenten van het systeem. DCRA heeft als doel het nauwkeuriger beoordelen van veiligheidsrisico's voor complexe IT systemen dan op basis van checklistgebaseerde benaderingen.

CRAC is een modelgebaseerde methode voor risk assessment van vertrouwelijkheid die twee alternatieve technische oplossingen rangschikt en vergelijkt op basis van hun risico's. Ze analyseert de risico's op basis van waar in de IT architectuur informatie toegankelijk is (*information flow*) en hoe moeilijk het is voor verschillende aanvallers om toegang te krijgen (*attack paths*). CRAC heeft als doel het vergroten van de intersubjectiviteit van de assessment-resultaten, en tegelijkertijd het verminderen van de kosten ervan.

CRAC++ breidt CRAC uit door het verkrijgen van controle over de vertrouwelijkheidsvereisten in een netwerk van organisaties. Hiervoor levert het een set van controle-eisen voor vertrouwelijkheid die kunnen gebruikt worden voor de uitbreiding van SLA's. CRAC++ heeft als doel het aanpassen van vertrouwelijkheids-RA methoden gebaseerd op de IT architectuur voor het controleren van vertrouwelijkheidsrisico's.

RiskREP is een risicogebaseerde methode voor de elicitatie en het prioriteren van veiligheidsvereisten, die is bedoeld om gebruikt te worden voor systemen in ontwikkeling. Ze koppelt bedrijfsdoelstellingen aan IT risico's op basis van de IT architectuur. RiskREP heeft als doel het kostenefficiënt eliciteren van assessment-relevante informatie.

We valideren en evalueren deze methoden in zeven echte casestudy's in multinationale ondernemingen uit de telecommunicatie, elektronica en chemische industrie. De resultaten geven aan dat multinationale organisaties die verbonden zijn met andere organisaties door middel van digitalisering voordeel kunnen halen uit de assessment van vertrouwelijkheidsrisico's op basis van de IT architectuur. De methoden die we voorstellen tonen aan dat risk assesment op basis van IT architectuur (1) de kosten helpt te verlagen, (2) toelaat de nauwkeurigheid aan te passen volgens het bedrijfskritische karakter van een systeem en (3) de intersubjectiviteit verhoogt van kwalitatieve risk assessment resultaten.

Acknowledgements

Four years ago when I decided to move to the Netherlands for doing a PhD I knew that I chose a hard path. However, I also knew that my family would support me in all possible ways and I would have an enthusiastic supervisor, who would lead me through this path and provide me the motivation to hard working. What I did not know was that I would have great friends, colleagues and Michaël who would fill my four years with fun and ease my pace. Here I would like to thank everyone who turned my last four years into a great experience.

Sandro, thank you for leading me and motivating me throughout my PhD period. You promised me in our first meeting that you would make sure that I would successfully get my PhD in four years. And it happened, despite the fact that you were awfully busy with forming the security group at Eindhoven and founding your own start-up company. It was a great comfort for me to know that you were there as my mentor, supervisor and promoter, and would make sure that not only my research is in good shape but also my personal life. You took me as a naive first-year PhD candidate and shaped me up into a PhD. Thank you!

Roel, my promoter, thank you for holding my hand just at the right moment. You helped me to see the big picture and draw the path when I was lost in details. You taught me how to conduct practical research. This thesis would not be possible without you.

Besides my supervisors, I would like to thank all committee members for reading my dissertation and providing useful feedback. Particularly, I would like to thank Pieter for providing me a warm and comfortable working environment during my PhD.

Emmanuele, my research partner and travel friend, it is your research steps that lead me to this dissertation. Thank you for being a secret supervisor to me.

Karin, Wim, Jeroen, Coen, Luc, and Andrea, the official and unofficial industrial partners of the VRIEND project, thank you for giving me insight to security practices and providing me case studies for validating and evaluating the treatments that I present in this dissertation.

Dina, my secret paranimph, you brought smiles to my office life. Thank you for the short gossipy breaks accompanied with nuts and snacks at hard working days, and pizzas at PuntoPASTA.

Christoph, Stephan, Arjen, Saeed, Luan, Trajce, Damiano, Begül, Qiang, Andre, Virginia, Wolter, Bertine and Suse, my fellow colleagues, thank you for turning the coffee breaks into warm and cosy gatherings at the DIES coffee room, dull dinners into relaxing and fun BBQs or cooking workshops at my place.

Svetla, my travel friend. Thank you for sharing the long drives to and from Leuven, the nice chats and all the useful tips on research, cooking and traveling. Despite my great motivation, without you, the 300 km would be very boring and hard. I hope we will continue sharing many more times together not to and from but in Leuven.

Nienke, my dear friend with the big smile and optimism. Thank you for bringing sunshine to my workdays, listening to me when I was pissed off or needed advise, and providing me the solution whenever I felt stuck.

Ayşegül, Kamil, Özlem, Mustafa, Cem, Seçkin and Michél, it is hard to find words to show my gratitude to you. You are the real friends I found in Enschede and will take with me to where ever I go. Thank you for accepting and supporting me through the complicated times of my life. Without you I would not be where I am now.

Yonca, my dear roomy. Unfortunately, you joined my life during the last year of my PhD, when it was time to wrap things up. With your positive and relaxed approach you turned our house into a cosy home. Thank you for being more than just a roommate.

The turkish community, former KisaBirMola current Demgah group and Turkish Student Association at Twente (TUSAT) members, especially Pınar, Berk, Fehmi, Neşe, Buket, Hasan, Selim, Janet, Feridun, Suzan, and Erhan thank you for bringing joy to my everyday life and distracting me from research. Special thanks to Pınar for proofreading certain parts of my dissertation. Semih and Didem, what a coincidence but you were the reason I ended up in this university. Thank you for giving me a warm welcome when I first came to this very unknown place and supporting me during my baby steps.

Chris and Patrick, thank you for providing me the warmth of a family and making me feel at home although I was far away from my parents.

Michaël, despite the 300 km between us, you were always there whenever I needed you. Without you on my side I would not be able to write this dissertation. You encouraged me when I needed motivation, criticized me when I was wrong, and comforted me when I was stressed. If now I can look back and say I have an easy PhD life, that is mainly because of you. Thank you aşkım :* I know that I can trust you and I will always be there for you.

Biricik anneciğim and sevili babacığım, aramızdaki kilometrelere rağmen sıcak sevginizi ve koruyucu desteğinizi üzerimden bir an bile çekmediğiniz için size çok çok teşekkür ediyorum. Bu tez benim olduğu kadar sizin de emeğinizin eseri. Benim başarılarımı ve mutluluğumu her zaman kendi özlemlerinizden önde tuttunuz, ben de hayatım boyunca sizlere layık bir evlat olmaya çalıştım ve çalişacağım. Sizin benimle gurur duyduğunuzu görmek en büyük mutluluk kaynağım olacak.

24 March 2011
Enschede, The Netherlands

Contents

Chapter 1

Introduction

Not so long ago information was stored as hard paper copies, which were relatively hard to duplicate and transfer. Today, almost every organization benefits from business opportunities created by digitalization. Digitalization allows for instance to manage customer records with enterprise resource planning (ERP) applications, to develop software products on a shared platform, or to remotely control power generators. Organizations that were operating without computers before digitalization are now heavily dependent on confidentiality, integrity and availability (CIA) of Information Technology (IT) to carry out their business.

Due to digitalization, information assets have become fluid and the confidentiality concerns increased. The Internet Crime Complaint Center (IC3), a partnership of the FBI and the National White Collar Crime Center, showed in its 2009 annual report that the loss due to data theft and cybercrime doubled from 2008 to 2009. According to MCafee [102], businesses around the world lost more than $1 trillion worth of intellectual property in 2009. Digitalization of the technical environment triggered changes in the legal environment. For instance, the Sarbanes-Oxley Act of 2002 [98] requires the companies that are in the US stock exchange market to assess security risks and the implications of countermeasures. Protecting the confidentiality of digital information assets has become a major concern for organizations. However, the today's IT systems are usually complex in nature, and distributing the available security budget is not a trivial task. Thus, organizations need techniques and tools to manage security risks.

Risk Management The primary aim of IT security risk management (RM) is to provide stakeholders with sound knowledge on security risks that threaten the business. Managing the security risks is also required by legislation, such as the Sarbanes-Oxley Act of 2002 [98], Basel II [80] (International Convergence of Capital Measurement and Capital Standards), and HIPAA [92] (Health Insurance Portability and Accountability Act). There are numerous standards, such as ISO/IEC 27005:2008 [95], ISO 31000:2009 [82], NIST800-30 [99], that provide system owners with guidelines for

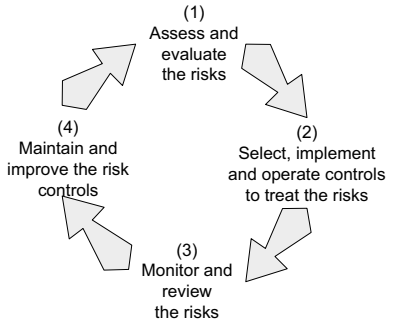

Figure 1.1: Risk management process model of ISO 27005 [95]

managing security risks. Despite differences among these standards, in essence they all suggest to follow the risk management steps that we present in Figure 1.1. In this dissertation we refer to "IT security risks" as risks.

RM is a continuous activity that aims to improve efficiency and effectiveness of results perpetually. It consists of four steps: (1) assess and evaluate the risks (plan); (2) select, implement and operate controls to treat the risks (do); (3) monitor and review the risks (check); and (4) maintain and improve the risk controls (act). To assess the risks one has to first identify the threats and vulnerabilities and determine the risks. Then, based on a predefined (usually business-dependent) risk scale, risk assessors evaluate these risks and finally suggest risk reduction controls. The next step is treating the risk. The main activities of this step are to prioritize the controls, and select and implement the most effective and cost-efficient ones. In the last two steps (3 and 4) a security team monitors and reviews the implemented controls. The aim of these two steps is to ensure that the changes in the environment or the IT do not affect the correct functioning of the controls. Corrective actions are taken in case of deterioration of controls and the RM cycle starts from the beginning.

Information security consists of intrinsically different goals, namely: confidentiality, integrity and availability. These security goals often conflict with each other. For instance, adding confidentiality-preserving measures could often affect the availability of data. In fact, there are different sets of threats and vulnerabilities for each security goal. Furthermore, business-criticality of each security goal is usually different.

In practice, RA methods assess risks either by checking whether an IT system satisfies a predefined list of security-related properties or by building a model of the system and analyzing interrelations among the system components. We refer to the first as *checklist-based methods* and the latter as *model-based methods*. Checklist-based methods rely on security patterns extracted from previous risk assessments that are conducted with model-based methods. Thus, model-based methods are more general than checklist-based methods.

In the last decade protecting the confidentiality of business-critical and private infor-

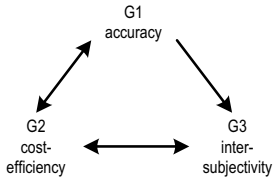

Figure 1.2: Dilemmas of confidentiality risk assessment methods

mation has become a major privacy concern for organizations. Thus, in this dissertation we focus on assessment and evaluation (first step in Figure 1.1) of confidentiality risks. *Confidentiality* is defined in ISO/IEC 27001 [93] as "the property that information is not made available or disclosed to unauthorized individuals, entities, or processes".

1.1 Problem identification

Our experiences in the risk assessment field show that a risk assessment method should satisfy three conflicting requirements: accuracy, cost-efficiency and inter-subjectivity (see Figure 1.2).

Accuracy has to do with the level of granularity that a method allows when assessing the risk (the confidentiality risks, as far as this dissertation is concerned). Accuracy is particularly important when for instance the stakeholders need to compare the risks presented by two alternative IT solutions.

Cost-efficiency is the crucial real limitation of all risk assessment methods. In practice, even risk assessments of large and information-intensive company sections rarely last longer than two weeks. There is neither enough monetary nor legal incentive in making it longer than this. In this time span, the risk assessor needs to first collect information about business goals, stakeholder requirements, technical information (such as IT architecture and measures in place), and information about threats and vulnerabilities; then formalize and analyze this information as well as the inter-relations among them; and finally determine and evaluate the risks, and discuss them with the stakeholders.

The third requirement we will look at in this dissertation is *inter-subjectivity*. Nowadays, despite the large use of standardized methods, risk assessment methods still rely heavily on the expertise and the experience of the risk assessor; consequently, the very result of a risk assessment is largely subjective in the sense that other assessors may have assessed risks differently. We will elaborate on the causes of this later in the dissertation. This subjectivity means that risk assessments are difficult to replicate; this is a problem in particular when one needs to repeat the RA after some time or when one needs to compare the (confidentiality) risks presented by two different IT system architectures.

These requirements are in conflict with each other (indicated by arrows pointing

at opposite directions). In particular, increasing accuracy usually means that the RA methods will become more expensive to implement. One should not underestimate how stringent the cost-efficiency requirement actually is: no company would allow any increase in the cost of the RA, unless it would be forced by (for instance due to changes in the law and regulations). For this reason, we say that these three requirements form the *dilemma* of confidentiality risk assessment methods.

1.2 Research questions

Based on the dilemmas of confidentiality risk assessment methods, this dissertation focuses on the following practical research aim:

> *To develop an IT confidentiality risks assessment and evaluation method that is:*
>
> - *accurate enough for business applications;*
> - *cost-efficient for business-criticality of the IT system; and*
> - *more inter-subjective than present methods.*

Based on this aim we formulated three more concrete research goals (G1-G3). Figure 1.2 illustrates the research goals and dilemmas between them. In the following we describe each goal.

G1. How can we assess confidentiality risks of complex IT systems accurately? Risks of IT systems can be assessed either by less costly and less accurate checklist-based methods, or by more costly but also more accurate model-based methods. For accuracy it is vital to assess each security goal with a specific method due to the intrinsic differences among the security goals. We claim that model-based approaches can deliver more accurate results than checklist-based ones (Thesis 1).

G2. How can we assess confidentiality risks of IT systems cost-efficiently? Most RA methods assume that the information the method requires is available in explicit form. However, in practice it is usually either incompletely documented or available only in undocumented tacit form. This is especially the case for confidentiality risks. Eliciting this information is a time and resource consuming activity. We claim that this information can be elicited cost-efficiently by (1) analyzing the IT architecture documents and system specifications for information flow paths and attack propagation paths, and (2) linking business goals to IT components (Thesis 2).

G3. How can we assess and control confidentiality risks of outsourced IT systems inter-subjectively? Due to changes in the technology, organizations often need to choose between alternative IT systems. However, the available methods do not allow comparison of confidentiality risks. This is mainly due to the subjectivity of the results and the way the assessment results are presented. We claim that one can increase the inter-subjectivity of an assessment by using IT architecture and functional documents and formalizing the risk assessment activities (Thesis 3).

Furthermore, outsourcers need to control confidentiality risks of IT systems that they outsource. However, available tools either do not work for confidentiality or are not accepted by outsourcees. We claim that by adapting IT architecture-based confidentiality RA methods to identify requirements that form service level agreements (SLA), one can control confidentiality-related risks (Thesis 4).

1.3 Research Method

The research method for this dissertation follows a nested problem-solving approach proposed by Wieringa [74]. This approach comprises of three levels: (1) *practical problem investigation*, (2) *problem treatment design*, and (3) *treatment validation and evaluation*.

In this dissertation we investigate the practical problems that industrial partners of the VRIEND research project (http://vriend.eemcs.utwente.nl/) are confronted with when assessing IT security (confidentiality) risks. We formulate these problems as research goals in Section 1.2.

Based on the problem investigation and *literature study* we design possible treatments, which are risk assessment methods and tools.

We validate the useability of these treatments and evaluate how well they satisfy both the research goals and the stakeholder goals by means of seven *case studies*. Here we adopt two types of case studies: *action research* and *lab demo* [72]. Action research requires the technique to be used by the designer in the field. Field is a realistic and uncontrolled context, such as an authentication and authorization system that is used by the employees of the case study provider. Mainly the industrial partners of the VRIEND project provide us with the context. Lab demo requires the technique to be used by the designer on a realistic example in an artificial environment. To discuss how well some of our treatments perform compared to alternative treatments we applied the alternative method to the context of the action research in our research lab.

For evaluation purposes we use both quantitative measures (e.g. man hours spent on executing the method) and qualitative measures (e.g. subjective opinions of the stakeholders).

1.4 Dissertation outline and contributions

So far our industrial partners were using checklist-based approaches. In this dissertation we propose 5 methods, each of which extends the previous one. Table 1.1 lists these solutions, their main contribution, as well as how each method relates to the research goals and thesis.

Table 1.1: Research outline

Chapter	Research Goals	Thesis	Solution	Contribution
Chapter 2	G1	Thesis 1	extended eTVRA	information elicitation and structuring
Chapter 3	G1	Thesis 1	DCRA	IT architecture-based confidentiality risk assessment
Chapter 4	G2, G3	Thesis 2, Thesis 3	CRAC	alternative comparison
Chapter 5	G3	Thesis 3, Thesis 4	CRAC++	outsourcing SLA
Chapter 6	G2	Thesis 2	RiskREP	linking business goals

Chapter 2: Model-based Methods vs. Checklist-based Methods: a Case Study

In this chapter we present the extended eEurope secure and trusted architecture threat, vulnerability, and risk assessment (eTVRA) method and compare it to a checklist-based method to motivate our research direction. eTVRA is a model-based method, specifically developed for telecom systems. We extend eTVRA with an information elicitation and structuring step. With this extension we address G1 (How can we assess confidentiality risks of complex IT systems accurately?). We validate the feasibility of extended eTVRA by applying it in an action research and lab demo in the telco domain. We evaluate our claim that model-based approaches can deliver more accurate results than checklist-based approaches (Thesis 1).

This work appeared in a refereed conference paper [3], which is joint work with E. Zambon, S.H. Houmb, K. Sallhammar and S. Etalle.

Chapter 3: Confidentiality Risk Assessment: an Architecture-Based Approach

In this chapter we present the Dynamic Confidentiality Risk Assessment (DCRA) method, which addresses G1 (How can we assess confidentiality risks of complex IT systems accurately?). DCRA is a model-based confidentiality method that is automated with a computational tool. It models the information system based on the IT architecture the system relies on, so that one can analyze how confidentiality breaches can propagate through the IT components of the system. We validate the feasibility of DCRA by applying it in a lab demo and executing an action research. We evaluate our claim that model-based approaches can deliver more accurate results than checklist-based ones (also with respect to confidentiality risks) (Thesis 1).

This work appeared in a refereed workshop paper [4], which is joint work with E. Zambon, S. Etalle and P. Overbeek.

Chapter 4: CRAC: a Method for Confidentiality Risk Analysis and Comparison

In this chapter we present the Confidentiality Risk Assessment and Comparison (CRAC) method, which addresses G2 (How can we assess confidentiality risks of IT systems?) and G3 (How can we assess and control confidentiality risks of outsourced IT systems inter-subjectively?). CRAC is a model-based confidentiality risk assessment method that sorts and compares two alternative technical solutions according to their risks. It analyzes risks according to where in the IT architecture information is accessible (information flow) and how difficult it is for different attackers to access it (attack paths). We validate the feasibility of CRAC by using it in two independent action research cases at two large multinational companies and one lab demo. We evaluated our claim that (a) one can increase the inter-subjectivity of an assessment by using IT architecture and functional documents and formalizing the risk assessment activities (Thesis 3), and (b) cost-efficiency can be achieved by eliciting information based on semi-structured interviews with stakeholders and analysis of IT-architectural documents, as well as using ordinal scale values instead of ratio scale values (Thesis 2). We compare CRAC with respect to these claims with the checklist-based method used by our industrial partners and CRAMM.

This work appeared in a refereed conference paper [2], which is joint work with E. Zambon, S. Etalle and R. Wieringa.

Chapter 5: CRAC++: a Method for Risk-Based Confidentiality Requirements Specification for Outsourced IT Systems

In this chapter we present CRAC++, which addresses G3 (How can we assess and control confidentiality risks of outsourced IT systems inter-subjectively?). CRAC++ extends CRAC by gaining control over the confidentiality requirements. Thus, it delivers a set of confidentiality control requirements that can be used for extending SLAs. It specifies these requirements by comparing confidentiality-related risks of outsourced IT systems based on the requirements of outsourcers and outsourcees. Outsourcers and outsourcees use this set as discussion basis for extending the SLA and gaining control over the confidentiality risks of the outsourced IT system. We validate the feasibility of CRAC++ by applying it in an action research case at a large multinational company. Finally, we evaluate our claims that (a) one can increase the inter-subjectivity of an assessment by using IT-architectural and functional documents and formalizing the risk assessment activities (Thesis 3), and (b) one can control confidentiality risks by adapting IT architecture-based confidentiality RA methods to identify requirements that form SLA (Thesis 4).

This work appeared in a refereed conference paper [1], which is joint work with R. Wieringa.

Chapter 6: RiskREP: a Method for Risk-Based Requirements Elicitation and Prioritization

In this chapter we present RiskREP, which addresses G2 (How can we assess confidentiality risks of IT systems cost-efficiently?). RiskREP is a risk-based security requirement elicitation and prioritization method, which is meant to be used for systems that are under development. It describes stepwise how to identify quality (security) goals from business goals and how to link them to IT risks. It finally delivers the best set of requirements (security controls) based on the risks they encounter, costs they introduce and business goals they contribute to. We validated Thesis 2 and the feasibility of RiskREP with an action research at a German university. We evaluate our claim that assessment-relevant information can be elicited cost-efficiently by linking business goals to IT components (Thesis 2).

This work is submitted to 19th IEEE International Requirements Engineering Conference (RE'11). It is joint work with A. Herrmann, S. Etalle and R. Wieringa.

Chapter 2

Model-based Methods vs. Checklist-based Methods: a Case Study [1]

We start here with the first research goal:

G1: How can we assess confidentiality risks of complex IT systems accurately?

This chapter first presents extended eTVRA, a mode-based risk assessment method. Then, to motivate our research direction, compare it with a more pragmatic checklists-based method. By extending eTVRA we aim at guiding the risk assessors at accurately and cost-efficiently eliciting the input information that the method needs and structuring it.

2.1 Introduction

Nowadays, many organizations evaluate the security level of their IT products according to the ISO/IEC 15408 (Common Criteria) [89–91]. The Common Criteria are tailored to industrial applications and are the result of the experience and recommendations of researchers and experienced developers both within the military sector and from industry. The Common Criteria uses a hierarchy of predefined evaluation classes called Evaluation Assurance Levels (EAL). There are seven such EALs, where EAL 7 provides the highest assurance. The EALs and associated guidelines take an evaluator through a well-formulated and structured process of assessing the security of specific parts of an (or the complete) IT product.

The Common Criteria evaluation is considered a healthy approach for tackling the security issues of an IT product, as it gives detailed guidelines about the procedure

[1] This chapter is a minor revision of the paper titled "Extended eTVRA vs. Security Checklist: Experiences in a Value-Web" published in the Companion Volume of the Proceedings of the 31st International Conference on Software Engineering (ICSE'09), pages 130 - 140, IEEE Computer Society 2009.

to follow and it describes the activities that developers and security experts should undertake to ensure that all relevant security aspects have been considered. However, a Common Criteria evaluation is costly and not many companies have enough resources to take their IT products through such a formal evaluation process. This is especially the case if the necessary input for the Common Criteria evaluation cannot be elicited from readily available resources, such as a risk assessment. For this reason, the European Telecommunications Standards Institute (ETSI) developed a threat, vulnerability, risk analysis (eTVRA) method to support telecommunication companies in the Common Criteria (CC) evaluation. eTVRA builds on the risk management component of the CORAS framework [14] and is structured to provide output that can be directly used for the security evaluation. Thus, it aims a more cost-efficient CC evaluation process. Our experience from earlier assessments at ETSI has shown that eTVRA does not include context identification activities. Context identification is critical to produce accurate risk assessment results. Thus, we also extended eTVRA with a sub-process of the CORAS methodology.

In this chapter we first describe how we extend eTVRA with SWOT analysis and semi-structured interviews, as well as two techniques from the safety domain: HAZOP to structure the interviews with the stakeholders and TVA to structure the gathered information. Second, we validate the feasibility of extended eTVRA by applying it in an action research in the telecommunication domain. Here we have identified and analyzed the risks of a new SIM card that during the case study was being developed in collaboration between a small hardware company and a large telecommunication provider. Such a context, in which a set of profit and loss responsible actors cooperate to realize a common goal, is called a value-web [83]. Third, in the same value-web context, we evaluate the cost (time and resource) efficiency and accuracy of extended eTVRA by comparing it with a more pragmatic approach based on Protection Profile checklists (from here on we refer to this approach as checklist-based method). Finally, we report on lessons learned from applying extended eTVRA in an action research.

The rest of this chapter is structured as follows: in Section 2.2 we provide background information on CORAS, eTVRA and value-webs; in Section 2.3 we give the industrial context; in Section 2.4 we describe the extended eTVRA method; in Section 2.5 we present the checklist-based method as alternative to extended eTVRA; in Section 2.6 we compare the two risk assessment methods; in Section 2.7 we draw the lessons learned by using eTVRA in a value-web context; and in Section 2.8 present some concluding remarks.

2.2 Background information

2.2.1 CORAS

CORAS [14] is a framework for model-based risk assessment of security-critical systems. It is based on ISO 15408:2007 [89–91] and consists of four main components:

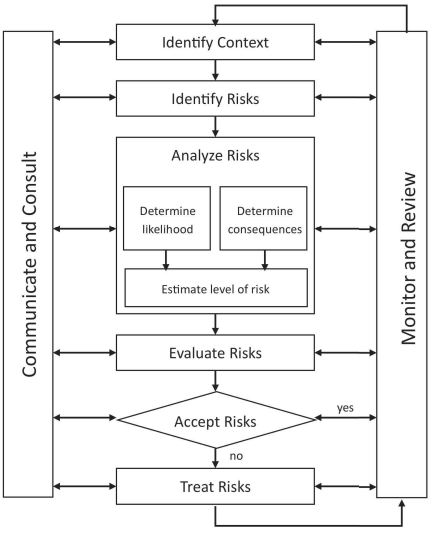

Figure 2.1: Sub-processes of CORAS risk management process [14]

(1) a risk documentation framework based on RM-ODP [69]; (2) a risk management process based on the AS/NZS 4360:2004 [97]; (3) an integrated risk management and system development process based on the Unified Process [28] and (4) a platform for tool inclusion based on data-integration using XML.

The CORAS framework is model-based in the sense that it gives detailed recommendations for modeling the system, the risk, and the security controls identified during the risk assessment using UML. Furthermore, CORAS is asset-driven, which means that the identification of assets is the driving task of the risk assessment process.

We show the sub-processes of the CORAS risk management process (main component 2) in Figure 2.1. Risk assessment related sub-processes of CORAS are shown in the middle of the figure and consist of context identification, risk identification, risk analysis, risk evaluation and risk treatment. Risk management related sub-processes of CORAS are parallel processes to these assessment sub-processes. These are risk communication and consultation, as well as monitoring and review.

2.2.2 eTVRA

eTVRA [58] refines the risk management process of CORAS (main component 2) for the telecommunication projects. It aims at analyzing the threats, identifying the best set of countermeasures and reducing the overall risk.

The process of eTVRA consists of 7 steps [59]:

Step 1 identification of security objectives;

Step 2 identification of security requirements;

Step 3 inventory of assets;

Step 4 identification and classification of vulnerabilities, threats and unwanted incidents;

Step 5 quantification of the occurrence likelihood and impact of threats;

Step 6 establishment of risk; and

Step 7 identification of countermeasures framework.

The process starts with the identification of the security objectives of a system or a system component, from which security requirements are extracted. Later, an inventory of the assets in the system is drafted. Purposes of eTVRA is to allow one to identify vulnerabilities that exist in the system. Therefore, after identifying assets and their vulnerabilities, the threats that exploit those vulnerabilities and cause incidents are determined. The security requirements are then extended according to threats and vulnerabilities. Then, the occurrence likelihood of the threats and their impact is analyzed and quantified. This is used in the following step to calculate the risk. Finally, the countermeasures for treating the risk are identified. This process is applied iteratively, until the risk of unwanted incidents is reduced to an acceptable level, or whenever there are changes in the environment.

eTVRA encapsulates the risk management-related parts of the Common Criteria and aims at producing accurate results that can be used for a Common Criteria evaluation. To note that, eTVRA is developed mainly for security standardization. Therefore, it considers only the technical vulnerabilities and countermeasures: the business impact of security breaches is - as usual - outside the scope of the standards.

2.2.3 Value-Webs

A value-web [83] consists of a set of profit and loss responsible actors that cooperate to realize a common goal. The actors can be independent companies or even business units of the same company. A value-web produces either a product or a service of some value. Some of the most common value-webs are marriages, outsourcing, insurance and contractor relationships.

The main challenge in protecting value-webs is that the actions taken should be profitable for each of the actors.

To evaluate the effects of value-webs on a risk assessment, the following criteria should be considered: (1) the goal(s) of each actor, (2) available resources, (3) confidentiality of business-critical information, (4) communication of confidential information, and (5) coordination of the responsibilities of the actors.

2.3 Industrial context

The industrial context of our study consists of two European companies, which collaborate as a value-web in the telecommunication domain. Together, they are developing the world's first GSM SIM card with embedded radio capabilities (802.11b). One of these companies is a small hardware producer, which is new to the telecommunication market, and the other one is a large European telecommunication provider that is already a major player in the field. The distribution of responsibility within the development project is that the hardware producer designs and produces the (Integrated Circuit) IC technology and its firmware, whereas the telecommunication company implements the software layer between the firmware and the operating system (OS) as well as the value-added service running on top of the OS.

One of the possible application areas for this new SIM card is automatic meter reading (AMR). AMR refers to the technology used for automatically collecting data from metering devices, e.g. water, gas, and electricity, and transferring readings to a central database for billing and analysis. In this context, a SIM card with wireless capabilities will reduce the number of terminals necessary to report the readings, saving a substantial amount of money. To limit the scope of the assessment and to make it feasible to do an evaluation between eTVRA and a checklist-based method, we focused on the security of the new SIM technology in the context of AMR.

2.4 Extended eTVRA

We evaluated the cost-efficiency and accuracy of two risk assessment methods; (1) extended eTVRA and (2) checklists-based method, as input to the Common Criteria evaluation. Here, we describe how we extended the eTVRA method.

Figure 2.2 gives an overview of extended eTVRA. The main changes we implemented were adding the context identification step taken from CORAS and adding concrete guidelines for methodologies to use for risk identification and risk analysis. The figure illustrates, besides the process flow, the information we used as input to the different steps involved, the information delivered as output of the steps and the methodologies that we used as support in producing the outputs.

2.4.1 Step 0: Context identification

Earlier case studies of eTVRA at ETSI have shown that "context identification" is critical for producing accurate results. As eTVRA does not include any specific context identification activities, we extended eTVRA with the context identification sub-process of CORAS. The aim of this sub-process is to describe the IT product to be assessed and its environment.

We used a Strengths Weaknesses Opportunities and Threats (SWOT) analysis [77] as

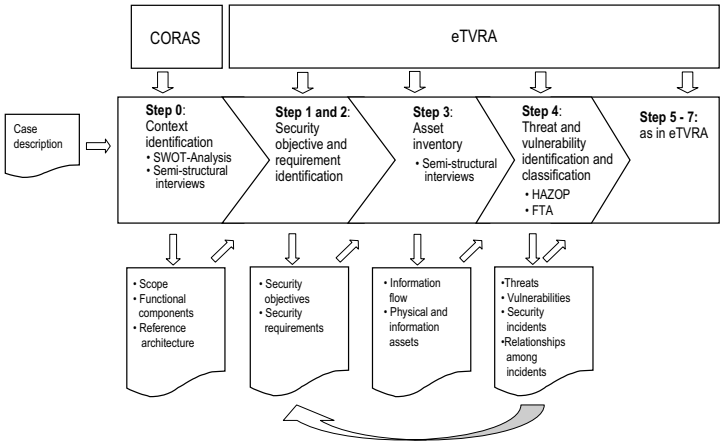

Figure 2.2: Extended eTVRA and the supporting methodologies

information gathering tool to identify the scope of the risk assessment and to ensure that the two stakeholders involved agreed on the goal and the objective of the assessment.

To prepare for and to carry out an effective SWOT session we referred to the case scenario documentation. Then, we (the risk analysts), together with the product owner (the two stakeholders in the value-web), went through the current case scenario document and made sure that we had a common understanding of the assessment context and of the role of the SIM card in an AMR setting.

The SWOT analysis helped us to determine the scope of the assessment and to focus only on those assets that are related with the scope. In addition to SWOT, we carried out semi-structured interviews with the two stakeholders. During the semi-structured interviews we agreed with the stakeholders on the functional components of the AMR deployment scenario which we previously extracted from the case scenario documentation.

The result of this step is documented in a context identification document, which consisted of the *case description* (including the deployment scenario), the *functional components*, the *reference architecture* and the *scope* of the assessment.

2.4.2 Step 1 & 2: Security objective and requirement identification

From this step on, we follow the eTVRA process as described in [59] and as a contribution specify how some activity can be performed by using tools and techniques from safety domain.

The first two steps of eTVRA are the specification of security objectives and the identification of security requirements. To establish the security objectives we based

ourselves on the output of the previous step; namely the SWOT-analysis and the semi-structured interviews, as reported in the context identification document.

We divided the *security objectives* of the new SIM technology into security objectives of the assets and security objectives of the environment. We then combined them and defined new security objectives for the desired level of confidentiality, integrity, availability, authentication and authorization for the assets involved.

These security objectives are high-level, e.g. "The new SIM technology should ensure continuous and correct operation of its core functionality and availability to authorized use upon request.". For operability reasons they had to be refined into *security requirements*. Security requirements describe the details of how the security objective will be achieved.

We listed the security objectives in a Target of Evaluation (ToE) document. At that time we did not have enough information to detail security requirements, so we postponed this activity to a later step. This document was then extended with the context identification descriptions from the previous step and given to the two stakeholders for approval.

2.4.3 Step 3: Asset inventory

In this step we used the information gathered in Step 1 and 2 as input. First, we had to complete the draft-list of assets that came out of the semi-structured interviews with the two stakeholders as described in Section 2.4.1.

For the interview with the large telecommunication company we used the reference architecture as input and we obtained a list of assets relevant for the information flow in the AMR case. These were assets at a high-level of abstraction, e.g. the concentrator functionality on the SIM card.

The interview with the hardware developer was carried out as a functional architecture walkthrough. This resulted in assets on the physical and logical layer. We then compared these assets with the information flow assets and modelled their internal relations, e.g. dependency and containment relationships. The result of this activity was given as output of Step 3.

2.4.4 Step 4: Threat and vulnerability identification

eTVRA includes activities to identify threats and vulnerabilities but does not provide how-to guidelines (i.e. it does not provide any method/tool to systematically extract threats and vulnerabilities). We therefore used the guidelines provided in CORAS to assist us in Step 4. In particular, we used Security-HazOp [75] and (Fault Tree Analysis) FTA [57].

A Hazard and Operability (HazOp) study [17] is a systematic analysis of how deviations from intended use of system components can arise, and whether these deviations can result in hazards. A *hazard* is defined in FAA Order 8040.4 [101] as

a "Condition, event, or circumstance that could lead to or contribute to an unplanned or undesirable event."

Although HazOp has been developed for safety rather than security, i.e. for industrial processes, notably the chemical, petrochemical and nuclear industries, years of experience have shown that the basic principle is applicable to different contexts, such as systems containing programmable electronics [19]. Security-HazOp [75] is a security-specific refinement of HazOp.

In general, HazOp is performed by defining a set of guide-words (e.g. intentional) and attributes (e.g. disclosure) as well as combining them with each other. The result can be used to describe so called generic deviations, which are anomalies with respect to the standard working of the system. These generic deviations help identifying specific safety-related deviations. Security-HazOp differs from HazOp in the chosen constructs, i.e. guide-words and attributes.

Srivatanakul et al. [66] criticize Security-HazOp and claim that the recommended guide-words are not flexible enough to bring out the analysts' creativity. They propose to apply guide-words to elements of a case, e.g. private data.

Furthermore, as recommended by CORAS, we use high-level threats and vulnerabilities discovered during the SWOT-Analysis and taken from relevant Smart Card Protection Profile [86] by defining the set of guide-words and attributes to perform Security-HazOp.

To determine and associate the guide-words we used the following method. First, we listed the actors, associations and elements of the AMR case. Second, we constructed a list of guide-words for the attributes of each of these main elements, as recommended by Srivatanakul. Third, considering that more than one guide-word may apply to an asset at one time, we grouped the guide-words as *pre guide-words* and *post guide-words* as recommenced in Security-HazOp. Last, we used the following notation to represent possible security incidents: <*pre guide-word*> <*attribute*> of <*component*> due to <*post guide-word*>. In this notation, *pre guide-words* are the possible causes of inadequate security attributes, e.g. *deliberate*, *unintentional*. *Attributes* are obtained by negating the security objectives, e.g. manipulation, denial and disclosure. *Components* are physical assets and information assets; and *post guide-words* are the possible threats, e.g. technical failure or outsider.

In this way, we obtained a list of 5400 possible incidents, e.g. *"Deliberate disclosure of meter readings due to technical failure"*. As it is not time- and resource-efficient to cover all of these incidents in one HazOp-session, we pre-processed and eliminated impossible incidents using the security objectives identified in Step 2 as filter. The incidents sub-set derived from this consisted of 88 possible incidents. This reduction indicates the ease of adjusting the accuracy level of an RA method that uses HazOp based on desired cost-efficiency.

We organized two structured brainstorming sessions: (i) the first session involved the telecommunication company alone and (ii) the second session involved both companies. During these HazOp sessions, to motivate the attendees to structured thinking, the risk

Table 2.1: Relation between HazOp and FTA methodologies [14]

To/From	HazOp	FTA
HazOp	HazOp identifies incidents at different levels of abstraction.	The incidents identified by HazOp are inserted in fault trees based on abstraction level and the relationship between the incidents.
FTA	A basic event (a leaf node in the fault tree representing an incident) may correspond to a sub-system/service on which HazOp may be applied.	A fault tree may be part of another fault tree, i.e. the top incident of one fault tree may be a causing incident in another fault tree.

assessor moderated the debate by using a set of "fault-statements", that are derived from the incident sub-set, e.g. *"How is it possible to deliberately disclose meter readings due to technical failure?"*. In all cases where potential hazards were detected, the risk assessor followed up by asking questions directed towards gathering information on its likelihood and its potential business impact. Furthermore, to increase the creativity of the attendees but remain objective in generating threats, we also used a light weight role-play.

After applying HazOp we achieved unstructured lists of vulnerabilities, threats and potential security incidents. We structured these lists in terms of cause-consequence relationships.

FTA is a system engineering method, which is mainly used in the safety domain. It represents, from the system point of view, the logical combination of various system states, faults, and possible causes which can contribute to a top event (specified event). A "Fault" is defined in ISO 31000:2009 [82] as an abnormal condition that may cause a reduction in, or loss of, the capability of a functional unit to perform a required function.

HazOp and FTA produce threats to a system or a component of the system at different levels of abstraction and from different view-points. When applied together they can be applied as input/output to each other, as documented by Table 2.1 [14].

We used FTA to illustrate at high-level threat/vulnerability pairs (security techniques, such as attack trees [61], originate from FTA [57]). Furthermore, we linked the incidents to each other with respect to their dependencies, e.g. if an incident a is a precondition for an incident b then we inserted incident a below incident b and indicated the relation with an arrow. Moreover, we differentiated between AND and OR causal relations. A small part of the resulting fault tree is shown in Figure 2.3.

Finally, we communicated the fault tree and the derived incident scenarios to the asset owners. The goal of this activity was to communicate and consolidate our findings and to gather additional information on the likelihood and consequence evaluation.

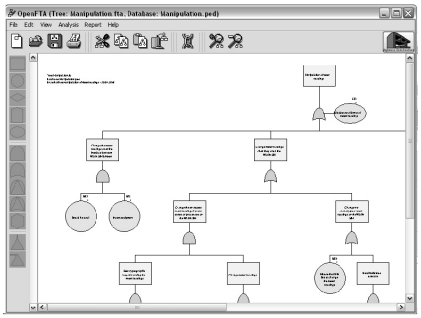

Figure 2.3: Part of the FTA that resulted from the HazOp brainstorming session[2]

2.5 Checklist-based method

In parallel to analyzing risks according to extended eTVRA, we employed a more pragmatic (i.e. less time consuming) approach, the checklist-based method. This approach requires almost no interaction with the main stakeholders for threat identification as the possible threats are extracted from an existing Common Criteria Protection Profile for Smart Cards [86]. The method consists of four steps:

Step 1: description of the risk assessment object and its security environment;

Step 2: specification of the security functional requirements;

Step 3: identification of the threat/vulnerability pairs and their impact; and

Step 4: risk analysis, prioritization and documentation.

Steps 1 and 2

The first step of this method is similar to the first step of extended eTVRA described in the previous section.

The security environment of the new SIM card for the AMR scenario includes (1) the assets to be protected and (2) the threat agents with their abilities to reach and exploit the object of the assessment and/or its environment during the expected product life-time (which is from product release to major update). To describe the security environment, we used the documentation provided in the first step of extended eTVRA. According to the results of the semi-structured interviews, we classified the components of the new SIM card in the context of the AMR scenario into physical and logical components. We further classified physical components according to how they interact with the external environment, e.g. wireless connection, serial connection. This classification is useful to clarify the main attack points of each component, e.g. a certain component may be attacked only through the wireless interface.

Step 3

The third step in this method is performed off-line, i.e. without interacting with the stakeholders.

We made a selection of the threats enumerated in the relevant Common Criteria Protection Profile [86]. The selection criteria we adopted were based on: (i) whether the threat agent fits in the usage scope of the new SIM card, e.g. terrorism is not a credible threat agent for the AMR scenario, and (ii) whether the threat can be perpetrated by means of the components of the new SIM card, i.e. if there is a component in the new SIM card which can be targeted by the threat. As the new SIM card also contains several components which are not part of a standard Smart Card, e.g. a wireless interface, the threat list provided in [86] covers only partly the range of possible threats. To fill this gap we included additional threats collected during a literature search [9, 12, 24, 63, 85].

Following the Protection Profile [86] threats are characterized by a threat agent, a threat scenario, a set of vulnerabilities enabling the threat and one or more assets targeted by the threat. The threat list can be summarized as follows:

- threats associated with physical attacks;
- threats associated with logical attacks;
- threats associated with access control;
- threats associated with unanticipated interactions;
- threats regarding cryptographic functions;
- threats of information monitoring;
- threats addressed by the operating environment; and
- miscellaneous threats.

For building a hierarchy among the threats, which in turn is needed to prioritize threats in the fourth step of this method, we additionally grouped threats according to the relevant security properties confidentiality, integrity and availability. The three resulting threat categories are:

- unauthorized disclosure of assets;
- theft or unauthorized use of assets; and
- unauthorized modification of assets.

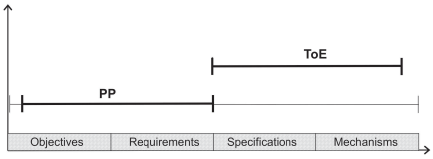

Figure 2.4: ST/ToE and ST/PP activity chart

Step 4

Step 4 is concerned with calculating the risk level of the threats and thereby prioritizing risks. The list of prioritized risks was submitted to the main stakeholders as an addition to the ToE document described earlier (See Section 2.4.2).

2.6 Comparison of the two methods

The main goal of the risk assessment for both stakeholders in the value-web was to produce information that could be used, preferably directly, as input to a Common Criteria evaluation. This goal put some constraints on the expected outcome of the risk assessment, and influenced how we carried out some of the steps of extended eTVRA. This is also the reason why we decided to compare eTVRA with a more pragmatic approach of security checklists derived from existing Protection Profiles (PP).

In this section we first briefly describe the Common Criteria evaluation process and then identify the key performance indicators (KPI)s based on which we evaluate the two methods.

2.6.1 Common Criteria evaluation

The Common Criteria recognize two types of evaluations: (1) ST/ToE (Security Target / Target of Evaluation) evaluation and (2) ST/PP (Security Target / Protection Profile) evaluation. In case of an ST/ToE evaluation, specific parts of the concrete IT product are defined into a Target of Evaluation (ToE). On the other hand, Protection Profile is an implementation-independent version of a particular IT product type, such as Smart Cards. This means that a Protection Profile can be looked upon as a template for a type of IT products. Figure 2.4 shows the different activities involved when carrying out ST/ToE and ST/PP evaluations. Since the output of ST/PP can serve as input for ST/ToE, the two types of evaluations are not orthogonal.

To enable reuse and thus increase cost-efficiency of Common Criteria evaluation, the Common Criteria offer a registry where IT product owners can choose to store documents from successful Protection Profile or ST/ToE evaluation. It is from the Protection Profile registry that we found the Smart Card PPs that we used for the checklist-based method.

In our case, the goal is to assess the ST/ToE to reach EAL 4 or 4+. Ideally, if the Smart Card PP [86] covered all aspects of our IT product, it could have been used as a template to produce the ST/ToE documents. However, as one always has to produce the ST part and as the ST is ToE-dependent, there is always at least some adaptation work needed, also in our case. To investigate the amount of adaptation work and the quality of the output produced, we performed a structured evaluation of the distance between the results produced and the needed input for an ST/ToE evaluation. This evaluation was done for both methodologies. Before we discuss the result of this evaluation, we list the ST/ToE requirements, which we use as evaluation criteria.

According to the Common Criteria Part 1 [91], the mandatory elements of an ST/ToE are:

ST introduction, containing three narrative descriptions of the ToE on different levels of abstraction.

Conformance claim, showing whether the ST claims conformance to any PPs and/or packages, e.g. threat lists, and if so, to which.

Security problem definition, showing the threats, the security policies and the assumptions that must be countered, enforced and upheld by the ToE and its operational environment (also referred to as security environment).

Security objective, which includes the security objectives for the ToE and the security objectives for the operational environment of the ToE.

Extended components definition, where new components (i.e. not included in the Common Criteria Part 2 [90] or the Common Criteria Part 3 [89]) may be defined. These new components are needed to define extended functional and extended assurance requirements.

Security requirements, where a translation of the security objectives for the ToE into a standardized language is provided. That is, standardized according to the recommendations in Common Criteria: security requirements should clearly specify the security functions, to a level where it is possible to directly check that these security functions are actually implemented as specified and to argue that they satisfy the security objective they address.

ToE summary specification, showing how the security functions specified are implemented in the ToE.

2.6.2 Key performance indicators

To make a comparison we identified four Key Performance Indicators (KPIs):

KPI1: number of relevant threats identified during the risk assessment;

Table 2.2: Comparison of the two methodologies

	extended eTVRA method	checklist-based method
KPI1: number of threats	77	48
KPI2: number of abstraction layers	6	2
KPI3: man-hours employed	310	68
KPI4: reformulated chapters of CC certification document	6/7	2/7

KPI2: number of abstraction layers in the threat hierarchy built during the risk assessment,

KPI3: number of man-hours employed to carry out the risk assessment; and

KPI4: number of reformulated chapters of CC certification document.

KPI1 and KPI2 express the accuracy of the results in terms of *result accuracy* and *result presentation accuracy*. KPI3 and KPI4 measure the cost-efficiency of the underlying process of each method in terms of *invested resources* during the assessment and the necessity effort in terms of *number of reformulated chapters* before using the output of the assessment for a CC evaluation.

By calculating the KPI3 we assumed a working day of 8 hours. We do not consider the costs of maintaining a protection profile up-to-date as the IT product itself or its treats environment changes. Furthermore, the chapters of a CC certification document are respective to the mandatory content of an ST/ToE.

2.6.3 Implementation evaluation and comparison

Here, we elaborate on which of the two methods is more cost-efficient and which produces the most accurate result in terms of coverage and match to the ST/ToE evaluation information requirements. Table 2.2 provides an overview of how each method performs with respect to the KPIs.

The result of the comparison indicates that the risk assessment following extended eTVRA delivered $\sim$37% better results than the checklist-based method with respect to KPI1. That is, it produced richer and more product-specific threat list. The main reason for this is that Protection Profiles are implementation-independent, so they aim rather to give an idea about the security level of the IT product than allowing detailed and product-specific security analysis. In contrast to Protection Profile-based checklist, extended eTVRA allows the risk analyst and the stakeholders to analyze security specifications of a system without and boundaries (and thus creative) during the risk identification process. This often means that extended eTVRA approaches the risk identification from several viewpoints.

Moreover, with respect to KPI2, in the presentation of the results produced from the checklist-based method we only used two levels of abstraction. This is in contrast with the six-layer incident hierarchy resulting from extended eTVRA. In general, having more layers is not always beneficial. However, for the critical components of an IT

product, having more layers eases the job of the Common Criteria evaluator: the six layers in the fault tree gives a deeper knowledge into how incidents may arise and thus how incidents can be prevented. On the other hand, such detailed results may not be necessary for less critical components and are both time and resource demanding.

If one considers the time spent on identifying threats (KPI3), the checklist-based method is five times more efficient than the extended eTVRA method. This makes the former more favorable than the latter in cases where time, resources and budget are limited or when the time-to-market window is relatively short (in time). Moreover, if there was no PP available for the ToE than the analyst would have to use a more generally applicable and longer checklist. Thus, in the absence of a PP the checklist-based method would perform even worse compared to extended eTVRA. Additional situations in which the checklist-based method works better is, when a limited ST/ToE is sufficient (only small parts of the IT product are evaluated), when targeting a low EAL (EAL 2 or 3) or when the PP is not used to support an ST/ToE evaluation but is merely domain knowledge.

A PP document has the same basic structure as an ST/ToE document. However, the PP introduction is narrative and does not provide the information necessary for an ST/ToE introduction. Therefore, we had to re-write this part. For the remaining parts, we had to add information for the wireless interface and to tailor the contents of the PP document to fit our IT product. We did so by adding new parts and by reformulating the text for the conformance claim, the security problem definition, the security objectives, the extended components definition, and the security requirements. As a PP document does not include a ToE summary specification, we had to write this part from scratch. We could reuse a substantial amount of the existing PP text (about 40%) and we also got help from the stakeholders in putting together the security controls necessary for the new SIM card.

The extended eTVRA method produced most of the underlying information needed for the ST/ToE document. However, the output had to be reformulated to fit the ST/ToE document requirements. The first step of extended eTVRA produced the goal and scope statements, which could easily be reused in an ST/ToE evaluation. Furthermore, it also identified which EAL to target and the ToE boundaries, i.e. which parts of the IT product were in the scope. The SWOT and the semi-structured interviews in the first step also brought to light cross-organizational challenges due to the value-web configuration. Finally, the ToE document in Section 2.4.1, produced as output of the first step, is at a level that made it easy to formulate the necessary ToE abstraction levels required for the ST/ToE introduction.

To summarize, with respect to KPI4, the effort spent to adapt the methodologies to cover the mandatory contents of CC, extended eTVRA allocates only the “ST introduction” and “Security Problem Definition” chapters, the checklist-based method requires re-writing only the “ST introduction” and “ToE Summary Specification” chapters. Hence, the latter requires three time less re-writing, and is therefore, more result-oriented.

We believe that these findings can be generalized for the cases where there is a close

match between an existing PP and the IT product. Therefore, in the AMR case, it is more time- and resource-efficient to follow the alternative method described in Section 2.5. Otherwise the checklist-based method is not more efficient than extended eTVRA. We based this last consideration on the experience we gained from the AMR case study, without a formal evaluation. In addition, the checklist-based method did not identify any of the added security challenges (due to the presence of wireless properties) which needed extra attention from the management perspective, as the roles of the two actors were not clearly defined.

2.7 Lessons learned from applying the eTVRA in the AMR case

After analyzing the accuracy of the results and the cost-efficiency of the process as discussed in the previous section, in this section we discuss the lessons learned from applying the eTVRA in the AMR case. We mainly focus on how extended eTVRA enables the communication needed in each step and whether it produced the information required as input for the next step. This is particularly challenging in a value-web context.

2.7.1 Communication

The industrial context with two relatively different companies collaborating in a value web affected the communication throughout the assessment. One of the companies was a relatively small hardware producer new on the telecommunication market. Its goals for the development project were thus naturally rather different than those of the second stakeholder: the large telecommunication company. A small company usually has limited monetary and human resources and when such a company is new to a market, the essence is to produce a good quality product and to get penetration in the new domain. A big international player with many years in the market could care less about time and market penetration issues, as it does not depend on a single product for market visibility and cash flow. However, the two stakeholders have a common goal in the development project and that is to produce a high quality product.

We experienced some communication difficulties that we believe are due to the configuration of the value-web. First, it seems that there were no clear agreements, neither internal to each stakeholders or across the two organizations regarding which information was company-internal, company-confidential or open to everybody involved in the value-web. This made it somewhat challenging to carry out assessment sessions in which both stakeholders were involved. Besides, the distribution of assignments within the development project due to technical difficulties seemed to have been shifted a bit since the start-up of the project.

We also experienced that it was much easier to get the communication flowing when interacting with each stakeholder separately, than it was in sessions where both were

present. This could be due to the tight deadline phase that the project was in at the time of the assessment, but it could also be a general observation that is valid outside of the AMR case.

What did work well were the semi-structured interviews in the first step of extended eTVRA and the separately executed risk identification sessions in Step 3. The common brainstorming sessions were less successful. We have identified two main reasons for this: (i) unspoken communication restrictions and (ii) possible unsuitability of Security-HazOp for risk identification in a value-web context.

Unspoken communication restrictions refer to the first evaluation criteria listed in Section 2.2.3. Both stakeholders had unspoken goals and expectations, that due to strategic reasons were kept hidden even though they would help clarifying some of the security challenges that were being discussed.

Additional communication difficulties arose from poor management of tacit knowledge, and poor alignment between own vision of role and expectations of others [62]. This is further explored in Section 2.7.2.

When it comes to Security-HazOp and whether the method is efficient for risk identification in a value-web context, we made some observations that we believe deserve further investigation. In particular, brainstorming sessions with all involved stakeholders were not effective due to the reasons mentioned above: hidden goals, assumptions and expectations. However, we believe it should be possible to adapt Security-HazOp to allow tacit information to be revealed in a non-threatening manner so that stakeholders do not feel unconformable. Furthermore, confidential information should always remain secret, even if its disclosure is in the best interest of the project.

2.7.2 Information

Accurate information is crucial to produce good risk assessment results. If information is missing or if there are problems in interpreting it, the results produced will be poor.

As always in development projects, not much information is available in the early stages of the development. That was also true for the AMR case. In particular, information was often not made explicit and people were often not aware of the knowledge they possessed or of how it could be valuable to others. Tacit knowledge is considered more valuable because it provides context for people, places, ideas, and experiences. Effective transfer of tacit knowledge generally requires extensive personal contact and trust. For risk management it is necessary to gain some understanding of the deployment scenarios to make security decisions, so it is important to extract the hidden knowledge.

In the AMR case, we extracted tacit knowledge through the semi-structured interviews. We first made guesses based on the scarce information available and then asked the stakeholders their opinion on our guesses as a kick-off for the semi-structured interview. Then we tried to use the feedback from stakeholders to structure our own

thoughts and to arrive at a preliminary understanding of the intended behavior and deployment of the new SIM card for the AMR scenario.

The reference architecture and the functional components of the new SIM card that were given to us during the first step of the extended eTVRA method is an example of implicit information. The diagram in itself did not give the risk analysts much information, as it did not have the required domain knowledge. The added information given in the semi-structured interview ensured that the diagram made sense, and could be used to articulate the security objectives. In a similar manner, we used at Step 4 of the extended eTVRA method the list of assets combined with our knowledge about the information flow to transfer the implicit knowledge on the threats and vulnerabilities into explicit knowledge.

2.8 Concluding remarks

This chapter presents an extension of eTVRA and compares it with a more pragmatic checklist-based method on a real world case. It furthermore evaluates the two methods in terms of accuracy and cost-efficiency. The result of the evaluation is that as extended eTVRA produced more accurate information in terms of threats and incident propagations, if a suitable PP exists and if the ToE has a rather limited scope, then the checklist-based method is more time-effective than extended eTVRA. However, the existence of a suitable PP depends on the presence of past Common Criteria evaluations and thus past model-based assessments. Considering that the difficulty of transferring a Protection Profile-based checklist from one context to another and high cost of maintaining a Protection Profile, we claim that model-based methods are more generally applicable and more cost-efficient than checklist-based methods.

We have extended eTVRA with a context identification step. The decision on this extension was based on our past experience with eTVRA which indicates that without context definition it is hard to keep threat identification sessions, in particular brainstorming sessions, targeted and focused.

We also extended eTVRA with methodology recommendations for threat identification and incident documentation borrowed from Security-HazOp and FTA. Security-HazOp has been in use in the safety domain for several decades. HazOp is well tested and well structured and, when adequate guide-words are selected, proved to be an effective threat identification brainstorming tool. The same can be said for FTA, which showed to produce an adequate set of abstraction levels.

Chapter 3

Confidentiality Risk Assessment: an IT Architecture-Based Approach[1]

Compliance to IT regulations require awareness of confidentiality risks and a good understanding of vulnerabilities and their exploitations. Confidential information is often distributed across the system. Besides, IT architecture enables an incident to propagate from one system component to another. Thus, to carry out an accurate risk analysis one has to consider where in the global architecture of the system information assets are stored and how incidents may propagate.

In the previous chapter we present the extended eTVRA method. It is a model-based IT security risk assessment method that is meant to be used for cost-efficiently eliciting and structuring assessment-relevant information. Then, we compared extended eTVRA to a more pragmatic checklists-based method with respect to cost-efficiency (time and resource) and accuracy. We reached the conclusion that model-based methods are better than the checklist-based methods, because they deliver more precise results and are more generally applicable.

In this chapter, we introduce the Dynamic Confidentiality Risk Assessment (DCRA) method. DCRA is a model-based method that systematically analyzes and assesses *confidentiality* risks based on the IT architecture.

3.1 Introduction

The World-Wide Web [13] has fuelled the deployment of a plethora of electronic services of increasing complexity, e.g. online banking, cross organizational interconnections to support supply chains, and digital patient IDs. To exploit these possibilities, organizations have to store valuable confidential information (e.g. patient records, bank

[1]This chapter is a minor revision of the paper titled "IT Confidentiality Risk Assessment for an Architecture-Based Approach" published in the Proceedings of the Third IEEE/IFIP International Workshop on Business-Driven IT Management (BDIM 08), pages 31-40, IEEE Computer Society, 2008

account information, and credit card details) in IT architectures that are usually exposed to malicious activities such as hacker attacks via the Internet and misuse by insiders.

The consequences of confidentiality breaches for an organization range from financial loss, to loss of market shares in the private sector and to compromise of national security in the public sector.

To deal with possible losses of confidential data, companies follow (largely standardized) risk management methods, i.e. ISO 31000:2009 [82], ISO/IEC 27002 [94], NIST 800-30 [99], OCTAVE [16] and COBIT [87].

When it comes to the management of IT confidentiality risks, we argue that one of the main limitations of present mainstream risk assessment and risk mitigation methods is that they do not take into consideration the IT architecture of the system under examination. To give an intentionally oversimplified example of how the IT architecture can greatly affect the resilience of the system with respect to confidentiality breaches, consider the IT system of a hospital: if its web-server is on the same sub-network of the patient database, then a hacker could reach the patient database via the web-server, whereas if the two systems were not directly interconnected, then this would be much harder. Indeed, the IT architecture determines to a great extent how robust an IT system is to confidentiality breaches and in case of breaches how much information will probably be disclosed (the damage is of a different magnitude whether a breach leads to the disclosure of only a few of the stored credit card numbers or all of them).

Since present risk management methods do not take the IT architecture directly into account, they completely delegate the issue of distinguishing a solid architecture from a less solid one to the specialist carrying out the assessment. The problem of distinguishing between solid architectures and less solid ones arises also during the engineering of a new system that has to deal with confidential information; also in this case there are no tools to be able to assess how good an architecture is, given the fact that it should preserve the confidentiality of the data stored in one or more of its subsystems.

In this chapter, we introduce the DCRA method. By modeling how confidentiality breaches can propagate through an organization, the DCRA method can be used as a tool for quantitatively measuring their actual impact (if the necessary information is available, also in monetary terms). The DCRA method can also be used to compare different architectures and identify the "best" one to cope with confidentiality risks, given the value of the data stored in it. Furthermore, by including in the DCRA an estimate of the risks the IT architecture is exposed to and of their likelihood, we can use it to calculate the global operational risks related to confidentiality an organization is exposed to. DCRA can be integrated with other RA methods to allow them to consider the underlying IT architecture.

We argue that the DCRA method assesses the IT confidentiality risk intrinsically better than other RA methods and allows one to measure how robust the system is to confidentiality risks.

The rest of this chapter is structured as follows: in Section 3.2 we provide a brief overview of risk management methods; in Section 3.3 we introduce the framework for

modelling an IT architectures; in Section 3.4 we introduce the framework for modelling incidents and the formalization of their propagation; in Section 3.5 we demonstrate how to apply the DCRA method in a real world case; in Section 3.6 we validate the feasibility for required information in building the models; in Section 3.7 we give an overview of the related research in the field of IT risk management; and in Section 3.8 we present some concluding remarks.

3.2 Present methods for risk management

There are a number of standards and methods for risk management, among which COBIT [87] and NIST SP800-30 [99] are of particular relevance to our work. COBIT is the *de facto* standard for information control and IT risk management, addressing IT Governance and control practices. It provides a reference framework for managers, users and security auditors. COBIT is mostly based on the concept of *control* (be it technical or organizational) which is used to assess, monitor and verify the current state of a certain process (that may refer to procedures, human resources, etc.) involved in the information system. To implement COBIT, the organization must benchmark its own processes against the control objectives suggested by the framework, using so-called *maturity models* (derived from the Software Engineering Institute's Capability Maturity Model [53]). Maturity models basically provide: (1) a measure expressing the present state of an organization, (2) an efficient way to decide on the goal to achieve and, finally, (3) a tool to evaluate progress toward this goal. Maturity modelling enables gaps in capabilities to be identified and demonstrated to management. Key Goal Indicators and Key Performance Indicators are then used to measure, respectively, when a process has achieved the goal set by management and when a goal is likely to be reached or not. Since COBIT does not suggest any technical solution but only organizational solutions, organizations combine COBIT and ISO 17799, applying the controls suggested in the part *Code of Practice for Information Security Management* of the standard.

As we mentioned before, current methods are not sufficiently taking into account how information assets are linked together and the way a single confidentiality breach could propagate and affect other related assets. The fact that COBIT and ISO 17799 do not consider dependencies between IAs has even greater impact in the mitigation phase of confidentiality risks: it is standard practice to protect the information assets whose confidentiality has a greater *direct impact* on the organization goals, whereas a more accurate analysis in many cases reveals that it is more cost effective to protect some of the information assets that have an *indirect impact* as well.

3.3 Modelling IT architecture

In this section we present the framework for modelling an IT architecture. We follow notable architecture frameworks, such as TOGAF [29] and Zachman [100] as well as

IT Governance solutions (IBM [21] and ISACA [87]), to determine the elements, which may directly or indirectly be involved in leakage of confidential information.

The DCRA method consists of two steps:

Step 1: A representation of the IT architecture of an organization, consisting of a set of information assets, the IT Assets that they depend on, and a set of relationships between them.

Step 2: A representation of estimated values assigned to the information assets. This can be integrated with the set of possible incidents affecting the confidentiality of information assets, annotated with the expected frequency estimation, measured in times per year (see Section 3.4).

Different from approaches that automate attack discovery based on IT architecture, such as UMLsec [40], the DCRA method models an IT system in 3 layers: business layer, IT layer, and physical layer. The *business layer* consists of business-related events and communications. This is the layer where the value of information assets is defined. In this sequel we follow [68] in calling *information assets* the semantic components of an information system that "an organization must have to conduct its mission or business". The *IT layer* is the layer where the interconnections between IT assets are defined. This layer consists of the applications, the middleware and the operating systems. The *physical layer* contains the hardware, on which the components of the IT layer run.

3.3.1 IT&I model

The IT&I model is the core of the DCRA method. It represents the IT architecture of an IT system using a graph. Here nodes represent information assets and IT assets, and labelled edges between nodes represent their relationships. There are two types of edges: Those indicating that a given information asset is contained in some IT asset and those indicating the propagation of an incident. The presence of an edge from node a at the business layer to a node b at the IT layer indicates that the information asset modelled with node a is contained in node b. We annotate these edges with the percentage of information asset stored on each IT asset. The presence of an edge at the IT layer from node a to node b indicates that the information stored in b depends on a in a way that the disclosure of confidential information in a may propagate to the connected assets (in this case b), and cause the confidential information stored in b to be disclosed as well. Similarly, the presence of an edge from node a at the physical layer to node b at the IT layer indicates that an incident at a may cause the confidential information stored in b to be disclosed. To model these propagations correctly, we refer to a measure (likelihood) of this propagation occurring: we annotate each edge modeling a propagation with the "propagation likelihood", i.e. the estimated likelihood that an attacker who has intruded in a is able to use the outcome of this attack for attacking b.

Table 3.1: Behavior of the • operator

•	high	m.-high	medium	m.-low	low	null
high	high	m.-high	medium	m.-low	low	high
m.-high	m.-high	m.-high	medium	m.-low	low	m.-high
medium	medium	medium	medium	m.-low	low	medium
m.-low	m.-low	m.-low	m.-low	m.-low	low	m.-low
low	low	low	low	low	low	low
null	high	m.-high	medium	m.-low	low	null

We model this probability in a qualitative way, as it is commonly done in many RA methods, such as NIST 800-30 [99], as well as in academic works, such as [44]. Particularly in this Chapter we refer to the following set of partially ordered likelihood values $\mathbb{L}$ = {*high, medium-high, medium, medium-low, low, null*}, and to the binary operator • on $\mathbb{L}$ whose behavior is defined in Table 3.1 (though the method would work with a different set of likelihood values as well).

Then, assuming that $\mathbb{R}^+$ indicates the set of non-negative real numbers, $\mathbb{V}$ is the domain of asset values such that $v : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ and $\mathbb{L}$ is defined above, the IT&I model is defined as follows.

Definition 3.1. ***(IT&I model)*** *An IT&I model is a tuple $\langle \boldsymbol{P}, \boldsymbol{I}, \xrightarrow{l}, v \rangle$, where $\boldsymbol{P}$ is a set of IT assets, $\boldsymbol{I}$ is a set of information assets, $\xrightarrow{l}$ is a mapping $\boldsymbol{P} \times \boldsymbol{P} \rightarrow \mathbb{L}$, and v is a mapping $\boldsymbol{P} \bigcup \boldsymbol{I} \rightarrow \mathbb{V} \in \mathbb{R}^+$.*

We write $p_n \xrightarrow{l} p_m$ as shorthand for $(p_n, p_m) \rightarrow l$ where $p_n, p_m \in \mathbf{P}$ and $l \in \mathbb{L}$. $p_n \xrightarrow{l} p_m$ indicates that an attacker who has access to the information assets stored on p_n may be able to disclose the confidential information stored on asset p_m with likelihood l. Furthermore, $v(p)$ indicates the operational value of the confidential information stored on asset p. We should mention that dependency relationships are typically OR relationships: an asset depending on two or more other assets may be hacked even if just one of them is affected by an incident. For the sake of simplicity, in this work we do not consider AND relationships, even though it would be straight forward to include them in our model.

From now on, we support the exposition of the model by means of a running example.

Running example - Part 3.1. *Here, we present a simple example of the IT architecture of a clinic, whose IT and information assets are listed in Tables 3.2 and 3.3. The IT&I model is reported in Figure 3.1. The edges connecting IT assets on the IT layer and the physical layer indicate the dependencies, and are annotated with the likelihood of propagation of incidents between assets. The edges connecting information assets of the business layer to the IT assets on the IT layer indicate that a given information asset is contained in some IT assets, and are annotated with the percentage of information stored on each IT asset. For instance, all instances of the information asset patient data is contained in the IT asset patient database, whereas up to only 5% of them may be contained in the patient management application that a doctor is running at his home.*

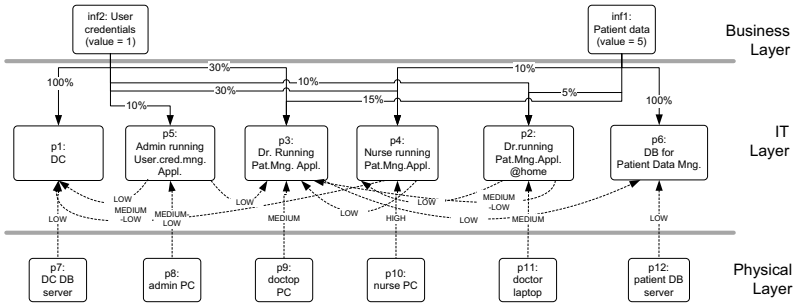

Figure 3.1: IT&I model that reports the architecture of the clinic example

Table 3.2: IT assets of the IT architecture of the clinic example

IT asset ID	Description
p_1	domain controller
p_2	doctor PC at home
p_3	doctor PC
p_4	nurse PC
p_5	admin PC
p_6	patient database
p_7	DC DB server
p_8	admin PC
p_9	doctor PC
p_{10}	nurse PC
p_{11}	doctor laptop
p_{12}	patient DB server

Table 3.3: Information assets of the clinic example and the and IT assets on which they are stored

Information asset ID	Description / Value	IT asset ID	→	Percentage
inf_1	patient data / 5	p_2	→	5%
		p_3	→	15%
		p_4	→	10%
		p_6	→	100%
inf_2	user credentials / 1	p_1	→	100%
		p_2	→	10%
		p_3	→	30%
		p_4	→	30%
		p_5	→	10%

Assuming that Alice, who is logged on to the nurse PC without authorization, scans the temporary files and finds a doctor's credentials. With low *probability she is then able to use this information to log onto the doctor PC. Furthermore, once she has penetrated to the doctor PC, she has* low *probabilities to disclose the confidential patient information stored in the patient database.*

3.3.2 The impact of information assets disclosure

The DCRA method requires a value to be assigned to each information asset which should be kept confidential. This value should help calculating the damage the company suffers in case the information asset gets disclosed to unauthorized parties. There are organizations that are able to indicate this value in terms of money, e.g. banks, insurance companies; for other organizations this can be harder. In such cases the value can be specified in a more qualitative way, i.e. using partially ordered values. The important thing to bear in mind when using partially ordered values is that these figures should reflect the relative values of the information.

Finally, we include the percentage of each information asset that is stored on each IT component (this is necessary to establish the *local impact* of the disclosure of an information asset). The percentage of each information asset $inf \in \mathbf{I}$ stored in each IT asset $p \in \mathbf{P}$ is modelled with a $M \times N$ matrix $\mathcal{P}$, where $N =\mid \mathbf{I} \mid$ and $M =\mid \mathbf{P} \mid$. For instance, according to Table 3.3, 15% of the patient data is stored in the doctor PC.

Assuming that the vector ψ of length N consists of the values of information assets, the local impact vector *v* defines the value of each asset, such that

$$v = \mathcal{P} \cdot \psi \tag{3.1}$$

Running example - Part 3.2. *According to Table 3.3 the value of information asset user credentials is set equal to 1, and the value of the patient data is set equal to 5. Table 3.3 also reports the percentage of confidential information stored in each asset. According to formula 3.1, the local impact of the disclosure of the assets in the clinic example are:* v_{p_1} *= 1,* v_{p_2} *= 0.35,* v_{p_3} *= 1.05,* v_{p_4} *= 0.80,* v_{p_5} *= 0.10,* v_{p_6} *= 5,* v_{p_7} *= 0,* v_{p_8} *= 0,* v_{p_9} *= 0,* $v_{p_{10}}$ *= 0,* $v_{p_{11}}$ *= 0 and* $v_{p_{12}}$ *= 0.*

Using the IT&I model in isolation The IT&I model is meant to be used within a risk assessment (as it is shown in the next section). However, it can also be used in isolation, to do the following:

1. Evaluating, for each component of the IT architecture, the global impact resulting from a confidentiality violation. As a consequence, it is also possible to find the most critical ones among the IT components, i.e. the components with the highest associated global impact.

2. Comparing how robust two different IT architectures are with respect to confidentiality of information stored in it.

We now indicate how we can achieve both points.

3.3.3 Global impact

First we need to define the global impact of an asset p, which is the cumulative loss caused by disclosure of confidential information stored in p, and the disclosure of confidential information stored in assets depending on p.

The sequence of IT assets that an attacker exploit to disclose an information asset that is available on IT asset p_{s_1} is $Seq(p_{s_1}) = p_{s_1}, p_{s_2}, ..., p_{s_S}$, where $p_s \in \mathbf{P}$ is the IT asset visited at step s.

Definition 3.2. ***(global impact)*** *Let v_p be the local impact of asset $p \in \mathbf{P}$. We define the global impact of p, gImp(p) as*

$$gImp(p) = v_p + \sum_{i=1, i\in\mathbb{N}}^{k} [l_i \cdot gImp(p_i)] \tag{3.2}$$

where $\{p_1, ..., p_k\}$ are the assets of $\mathbf{P}$ directly depending on p (i.e. for which $p \xrightarrow{l_i} p_i$), l_i is the likelihood associated to the edge $p \xrightarrow{l_i} p_i$, and $\cdot$ is a binary operator on $\mathbb{L} \bigcup \mathbb{V}$. Which behaves like the multiplication operation, if likelihood values are real numbers. So, gImp() is a function $\mathbf{P} \rightarrow \mathbb{L} \times \mathbb{V}$.

Since the IT&I model is acyclic the concept of global impact is well defined. Furthermore, formula 3.2 assumes that all the local impacts are independent since the method assumes that all dependencies among the nodes of the IT&I model are covered by the propagation relations.

Running example - Part 3.3. *In case the nurse PC is cracked, the confidential information stored on it gets disclosed. Accordingly, the local impact of this confidentiality violation on the nurse PC is 0.8 and on the admin PC is 0.1. According to this, the nurse PC a more critical component than the admin PC.*

Looking at the global impact, compromising the nurse PC can lead to compromising p_1 and/or p_3 (and – iteratively – p_6). This can be done by following two sequences of attacks. We denote these sequences as $Seq(p_4)_1 = p_4, p_1$ and $Seq(p_4)_2 = p_4, p_3, p_6$. The global impact of (exploiting) the nurse PC is then: $gImp(p_4) = 0.8 + medium\text{-}low \cdot 1 + low \cdot 6.05$.

Architecture comparison To compare the robustness of different architectures, we calculate the average and standard deviations of global impact values of disclosing the confidential information stored on each asset of the two architectures. The standard deviation tells us how spread the global impacts are. If the standard deviation is small, then the potential impact is almost equally distributed on many assets. Otherwise, there are few critical components in the system with high potential impact.

3.4 Modelling risk

In this section we introduce the concept of *incident* and we show how to integrate it in the DCRA method to carry out a complete risk assessment.

Incidents are security-related events affecting one or more assets on which some confidential information is stored. Incidents can happen several times a year, and risk

assessment methods always require an inventory of possible incidents, together with their expected frequencies. This information (type and expected frequency of incidents) is thus available after carrying out a standard RA, though it is usually indicated in partially ordered qualitative values (e.g. likely, moderate-likely, unlikely).

Definition 3.3. ***(incident)*** *Let* ***P*** *be a set of IT assets, an* incident *is a mapping* $inci$: $\boldsymbol{P} \rightarrow \mathbb{R}^+$.

In particular, $inci(p)$ indicates how often (per year) the incident $inci$ is expected to affect the IT asset p. If $inci(p) = 0$ then the incident $inci$ does not affect p. On the other hand, by setting $inci(p) \neq 0$ we model the situation in which an occurrence of $inci$ would cause the disclosure of all the confidential data on p; in this case we say that $inci$ directly affects p. Of course, an incident can cause an indirect damage by propagation, as described in the previous section. To measure the *global impact* of an incident we have to refer to the $gImp()$ function (Definition 3.2). Together with it, we can compute the *risk level* of an IT system.

Definition 3.4. ***(risk level)*** *Let* ***INCI*** *be a set of incidents and* ***P*** *be the set of IT assets in the system. The risk level of an incident is a mapping* $risk$: $\boldsymbol{P} \bigcup \boldsymbol{INCI} \rightarrow \mathbb{R}^+ \bigcup \mathbb{L}$, *which is indicated by the following formula:*

$$risk(i) = \sum_{p \in \boldsymbol{P}, inci \in \boldsymbol{INCI}} [inci(p) \otimes gImp(p)] \tag{3.3}$$

We now apply this definition to calculate the level of risk of the clinic example.

Running example - Part 3.4. *Let us assume that we have two incidents affecting the nurse PC directly; an attacker could break directly into the employee mail* ($inci_1$) *or get the nurse's authentication information by masquerading herself as system administrator* ($inci_2$). *The expected frequency of these incidents are respectively* moderately likely *(which corresponds to an expected frequency of twice a year) and* unlikely *(which corresponds to an expected frequency of once every three-four years). The global impact for the nurse PC is presented in Running Example - Part 2:* $gImp$(*nurse PC*) $= 0.8 + medium\text{-}low \cdot 1 + low \cdot 6.05$. *Furthermore, the nurse PC is affected by two incidents, and according to Definition 3.2 the risk level of incident* $inci_1$ *is* $moderately\text{-}likely \otimes (0.8 + medium\text{-}low \cdot 1 + low \cdot 6.05)$, *whereas the risk level of incident* $inci_2$ *is* $unlikely \otimes (0.8 + medium\text{-}low \cdot 1 + low \cdot 6.05)$.

3.4.1 Integrating the IT&I model in RA methods

RA methods require assessing the impact of incidents. For instance, [83, 99] recommends to use FIPS 199 [8] to categorize the impact level as *low - moderate - high*, according to a standard description of the effects of the incident itself.

The IT&I model is designed to be used together with standard RA methods to provide a more confidentiality-specific and architecture-dependent approach to evaluate

the impact of incidents. It can be easily integrated in other methods by using the incident information as input and providing the global impact of these incidents as output. To make a full integration possible we need to translate the output of our system (which is given in terms of a "sum" of likelihood-values) in terms of the usual *low - moderate - high* notation. It is simple to flatten our global impact into a single value (even though by doing so one looses a great deal of information the IT&I model is able to provide, we do not recommend to use IT&I solely for the purpose of providing impact values). In this case both the $\cdot$ and $\otimes$ operators behave like the multiplication operation.

Running example - Part 3.5. *Furthermore, the IT&I model delivers a further simplified version of the partially-quantitative risk value by quantifying, i.e. assigning quantitative values to, the qualitative likelihood values. Assuming that moderately-likely and unlikely are quantified in the running example respectively as 0.1 and 0.05; and medium-likely and low are quantified respectively as 0.1 and 0.05. Respectively, the quantitative risk related to the nurse PC is* $(0.1+0.05)\cdot(0.8+0.1\cdot 1+0.05\cdot 6.05) = 0.18$.

For the purpose of our running example we adopt this mapping: if the impact value is higher than the 10% of the total value of all the information assets, then it is mapped as high*, if it is higher than 0.1% then it is considered as* moderate*, otherwise* low*. Since the total value of the information assets in the clinic example is 6, and 0.18 is between 0.1% and 10% of the total value, the risk level of incident* $inci_1$ *is* moderate.

3.5 Application of the DCRA method

We apply our approach in a lab demo to a segment of the IT architecture of a real world telecommunication company for demonstrating how to use the DCRA method.

Part of the core business of a telecommunication company consists of generating proper invoices for the customers of the company by counting the calls they did. The invoicing process is composed of a number of steps, which we summarize in Figure 3.2: at first, the raw call records are provided by the physical network architecture. The record does not contain any information about the customer, but only a reference to the physical telephone line. These records are then enriched by the Post Processing application with the customer information provided by the Customer Relationship Manager (CRM) application. Since the data format used by the CRM application is too complex for the Post Processing application, the customer information is first normalized by the CRM exchanger application. After the post-processing phase, the enriched call records are stored in the Operational Traffic Database, where they are readily accessible for inspection by means of the traffic viewer application. Finally, the invoicing application uses the complete call records, together with the pricing information from the CRM, to calculate the exact amount of each customer invoice. Furthermore, the architecture includes other components, e.g. a complete test environment for post processing, operational traffic, traffic viewer and CRM exchanger applications, file and email servers used by the developers, as well as the laptops used by employees of the company and external consultants.

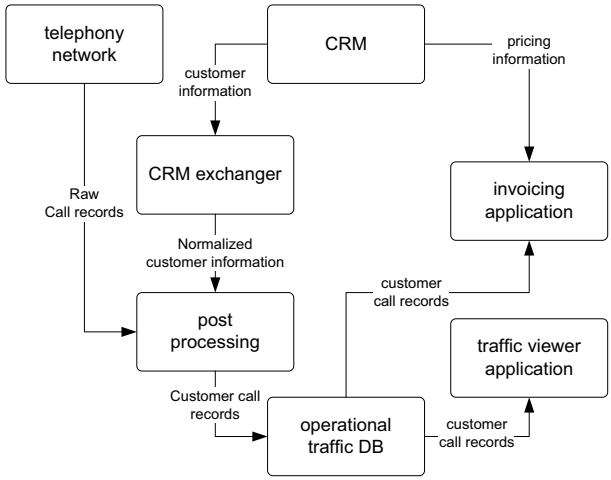

Figure 3.2: Telecommunication company invoicing process

Since applications run on different hardware components, the data is transferred from one to the other by means of encrypted flat files. Part of the information, such as the source and destination phone numbers and the customer ID are kept partially encrypted inside the operational traffic database. Access to this database is also controlled by strong authentication mechanisms and logs are generated for each read operation. Encryption keys are kept inside a key repository, and applications can access the repository to retrieve the keys and use the encrypted flat files.

3.5.1 Forming the model

We start forming the model from the business layer: Table 3.4 reports the information assets that we identified, together with the estimated (monetary) loss in the event of their disclosure. In this case the most important information assets are customer call records, raw call records, phone contact information and phone line information. These information assets have to be kept confidential because of laws and liability issues. The disclosure of the employee mail has a lower but still significant impact, whereas the disclosure of the other assets is judged to have no direct impact.

The application layer is composed of the custom applications used in the invoicing process. Table 3.5 reports the applications supporting the invoicing process, together with the information assets they contain and their percentage. The call records, which are among the most valuable pieces of information, are contained, in different percentages, in the following applications: post processing, traffic viewer and invoicing. Moreover, we observe that the CRM exchanger test application contains part of the production phone line information. This is due to the fact that generating fake data sets

Table 3.4: Information assets and related potential loss

Asset	Loss (Eur)
raw call records	10.000.000
user call records	100.000.000
phone contract info	20.000.000
phone line info	500.000
test data sets	0
application design specification	0
software test documentation	0
encryption keys	0
employee mail	70.000

Table 3.5: Information assets contained in each application

Application	Information asset	Percent
telephony network database	raw call records	100%
post processing	raw call records	5%
	user call records	5%
	phone line info	100%
operational traffic processing	-	-
traffic viewer	user call records	100%
CRM	phone contract info	100%
	phone line info	100%
CRM exchanger	phone line info	100%
invoicing	user call records	20%
	phone contract info	100%
post processing test	test data sets	100%
operational traffic test	test data sets	100%
traffic viewer	test data sets	100%
CRM exchanger	test data sets	100%
mail client	application design specifications	4%
	software test documentation	3%
	employee mail	1%

to test the CRM exchanger application is too time consuming, and some real phone lines are used for testing purposes. Finally, the mail client application contains both employee mail and application specification and test documentation, because employees share documents by means of the email service.

The architecture layer is composed of general purpose software components providing services to the users or to other software components. Table 3.6 reports the architecture components supporting the invoicing process, together with the information assets they contain and their percentage. As expected, the Oracle server that is used to implement the operational traffic database contains the whole user call records; moreover, since some employees need to regularly control the formal quality of the call

Table 3.6: Information assets contained in each IT asset

Component	Information asset	Percent
FTP service	-	-
operational traffic Oracle	user call records	100%
traffic viewer application server	-	-
employee FTP client	user call records	0.5%
SAMBA server	application design specifications	100%
	software test documentation	100%
MS Exchange server	employee mail	100%
	application design specification	70%
	software test documentation	60%
encryption key server	encryption keys	100%

Table 3.7: Hardware components that form the architecture

Hardware Component
telephony network
post processing server
operational traffic server
traffic viewer server
CRM exchanger server
CRM server
invoicing server
employee laptop
file server
network segment
test server
mail server

records shared between the various applications, some call record files are also stored on the FTP cache of the employee laptops.

The physical layer is composed of the hardware components on which the software runs; Table 3.7 reports those components for the invoicing process.

Figure 3.3 gives a complete overview of the DCRA method for this telecommunication company example.

To complete the DCRA method, we also need to assess how the disclosure of information can propagate within the organization. Some propagations are quite intuitive: compromising a physical asset such as a machine implies that with high probability the information contained in it will be disclosed. Table 3.8 reports the other, non-trivial cases we have found in this scenario, together with their estimated probability. The first propagation scenario assumes someone has control over the traffic viewer application server. Since the configuration of the application server also includes the credentials to access the Oracle traffic database, with a high degree of probability it will also be

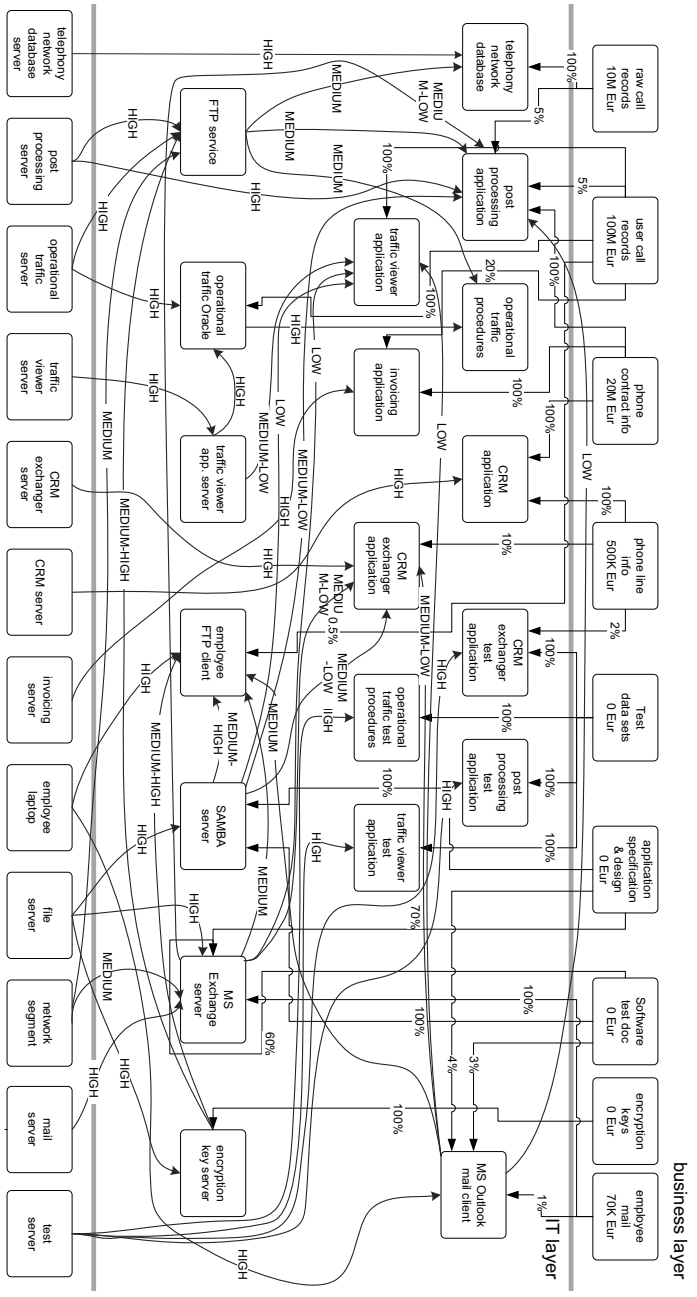

Figure 3.3: The model of the telecommunication company invoicing process

Table 3.8: Incident propagation

Source	Destination	Probability
traffic viewer server	traffic DB	high
key server	FTP service	medium-high
MS outlook mail client	FTP client	medium
MS outlook mail client	post processing application	low
MS outlook mail client	traffic viewer application	low
MS outlook mail client	CRM exchanger application	medium-low
MS Exchange server	FTP client	medium
MS Exchange server	post processing application	medium-low
MS Exchange server	traffic viewer application	low
MS Exchange server	CRM exchanger application	medium-low
SAMBA server	FTP client	medium-high
SAMBA server	post processing application	medium-low
SAMBA server	traffic viewer application	low
SAMBA server	CRM exchanger application	medium-low

possible for the person controlling the asset to obtain the user call records stored in the database. The second scenario assumes someone has broken into the key server and got hold of some of the keys stored on it: with this information, one can access the user call records by sniffing the FTP traffic transiting on the network and then trying to decrypt it; the probability of this event (*medium-high*) is evaluated by considering both the skill level needed to perform this operation and the number of tries necessary to use the right key to decrypt the sniffed file. The subsequent scenarios assume someone gets access to the test software documentation, this can be achieved by breaking into either the SAMBA server or the employee mail. In this case the attacker can use the information stored in those documents, such as the test credentials, the application behavior (and bugs), for different purposes. He or she could break into the FTP service to retrieve the call record flat files, or use a back-door on the post processing, traffic view and CRM exchanger applications to get sensitive information. The remaining two scenarios are similar, and assume someone has access to the specifications documentation of some applications and can exploit this information to bypass the security controls on the post processing and traffic view applications to obtain the user call records.

3.5.2 Using the model

After the presentation of the DCRA method we are ready to use it to assess the robustness of the telecommunication company IT architecture with respect to confidentiality of information. The first step towards the assessment of the architecture is to derive the local impact of each component. To do this we build the matrix $\mathcal{P}$ containing the percentage of each information asset contained in each IT component with the values from Table 3.5 and Table 3.6. We also build the value vector ψ containing the value of each information asset as reported in Table 3.4. Table 3.9 reports

Table 3.9: Local impact of the IT components

Component	Impact (Eur)
telephony network database	10.000.000
post processing	25.000.000
operational traffic procs	0
traffic viewer	100.000.000
CRM	20.500.000
CRM exchanger	50.000
invoicing	40.000.000
post processing test	0
operational traffic test	0
traffic viewer test	0
CRM exchanger test	10.000
mail client	700
FTP service	0
operational traffic Oracle	100.000.000
traffic viewer application server	0
employee FTP client	500.000
SAMBA server	0
MS Exchange server	70.000
encryption key server	0

the resulting v vector, corresponding to the total direct impact due to the disclosure of information contained in each IT component with respect to all the information assets it contains. Although it contains many different information assets, the Post Processing application is not the IT component with the highest associated amount, since it contains only small percentages of the most valuable assets (the call records) at one time. On the other hand, as expected, the traffic viewer application and the Oracle database containing the all user call records are the two most valuable components of the entire IT architecture. One unexpected outcome from this first analysis is that the CRM exchanger test application, which should be expected to have no importance, is "worth" 50.000 Euro. This is due to the choice of using production data to test the application, as we discussed in the previous section.

The second step to complete the assessment of the architecture is to evaluate the global impact to the disclosure of the information contained in each component of the IT architecture. This way we can find the most critical components of the architecture, evaluate the global impact distribution of the architecture, and subsequently check if the IT components are protected appropriately, i.e. according to their real importance. To evaluate the global impact we apply the *gImp()* function, which also takes into account how incidents can propagate from one asset to another. Table 3.10 reports the results; when applying the *gImp()* function we use the following rule: if two components of the resulting impact vector refer to the same information asset and have comparable values, then we only include the one with the highest likelihood. If both the values and the likelihood are different we keep both. As expected, some of the IT components,

such as the mail client and the SAMBA server, which at a first sight may seem to be of secondary importance, are more critical due to the possible propagation of information disclosure.

The last step of our assessment is now to calculate the average level of the global impact and its standard deviation. To be able to calculate such values from a semi-qualitative notation, we apply the following translation of the probability values into numerical ones: *high* = 0.9, *medium-high* = 0.5, *medium* = 0.3, *medium-low* = 0.1, *low* = 0.05. This way we are able to flatten the impact vectors and obtain a single value.

Concluding, the result of using the IT&I model in isolation shows that the IT architecture of the telecommunication company is quite heterogeneous: some components are at high risk, whereas others are almost safe. On the other hand, the amount of critical components in this architecture is very high with respect to the amount of non-critical ones.

3.6 Feasibility of the DCRA method

In this section, we argue that the DCRA method is feasible in practice. Due to the changes in the legal environment many organizations are obliged to document the majority of the input data we need, in the form of IT-architectural documents. For instance, the GRAAL framework [70] has been designed for architecture alignment of business requirements on IT systems and is structured in a form that is similar to our layered model. The GRAAL framework has been successfully used in case studies in many organizations. This shows that the layered structure of GRAAL is understood inside organizations.

IT-architectural documents provide us with the information about where information assets are located, allowing us to compile the matrix $\mathcal{P}$, which reports the percentage of the information asset stored in each physical asset.

Finally, risk assessment methods already require to make an inventory of possible incidents together with their estimated frequency. We can find this data in the deliverables of RAs that are carried out following standard methods. One possible obstacle is that the likelihood estimation is done in a subjective qualitative way, whereas our model requires a quantitative approach. However, it is possible to solve this (problem) by assigning standard values for each qualitative category (e.g. *high* = 0.9, *medium* = 0.5, *low* = 0.1).

3.7 Related work

ISO 31000:2009 [82] states that risk management (RM) can be applied at many levels within an organization and recommends embedding RM into operational and strategic planning. IT-related risks are classified as: strategic and operational [83].

Operational risk is defined in BASEL-II as “the risk of loss resulting from inadequate

Table 3.10: Global impact of the IT components

Component	Global Impact
telephony network database	10.000.000
post processing	25.000.000
operational traffic process	0
traffic viewer	100.000.000
CRM	20.500.000
CRM exchanger	50.000
invoicing	40.000.000
post processing test	0
operational traffic test	0
traffic viewer test	0
CRM exchanger test	10.000
mail client	700 + low · 100.000.000 + medium · 500.000
FTP service	medium · 25.000.000
operational traffic Oracle	100.000.000
traffic viewer application server	high · 100.000.000
employee FTP client	500.000
SAMBA server	medium-high · 500.000 + medium-low · 25.000.000 + medium-low · 50.000 + low · 100.000.000
MS Exchange server	70.000 + medium · 500.000 + medium-low · 50.000
encryption key server	medium · 10.000.000 + medium · 25.000.000
telephony network database server	high · 10.000.000
post processing server	high · 25.000.000
operational traffic server	high · 100.000.000
traffic viewer server	high · 100.000.000
CRM exchanger server	high · 50.000
CRM server	high · 20.500.000
invoicing server	high · 40.000.000
employee laptop	high · 700 + low · 25.000.000 + low · 100.000.000 + medium-low · 50.000.000
file server	medium-high · 500.000 + medium-low · 50.000 + low · 100.000.000 + high · 70.000 + medium · 25.000.000
network segment	medium · 70.000 + medium · 500.000 + medium-low · 50.000 + medium · 25.000.000
mail server	high · 70.000 + medium · 500.000 + medium-low · 50.000
test server	high · 10.000

or failed internal processes, people and systems or from external events". Compliance with BASEL requires banks to quantify IT-related operational risks [44], including legal risks and risks related to business processes of the organization.

Strategic risks are related to the high-level goals of an organization. They may be quantified by setting them equal to loss of market share, which depends on the monetary volume of the market and potential loss of market share in case of a confidentiality breach. Strategic risks are especially important when calculating the impact of confidentiality incidents.

There are various academic frameworks for carrying out RAs, but they all differ from our proposal in that they do not model the propagation of incidents across an organization as precisely as we do. Furthermore, they do not differentiate between methods for analyzing different security goals. We believe that differentiating between security goals allow us to determine risks more accurately. From this perspective we limit ourselves in this chapter to confidentiality-related risks.

For instance, Lenstra and Voss [44] present a quantitative RA approach to determine the optimal RA strategy given a limited budget. Their approach requires performing a risk assessment on all the applications supporting business processes and identifying the (monetary) loss due to each threat on the business process they support, thus the risk is evaluated in terms of likelihood and loss. Since this approach is designed to deal with threats to all three aspects of information security (CIA), to keep it feasible it lacks a complete representation of the constituents of an IT architecture (machines, applications, etc.) and it does not consider the functional dependencies between them. Functional dependencies are essential for accurately modelling the confidentiality risks. Our model, on the other hand, being specifically tailored for confidentiality risks, considers the IT architectures on which the confidential information relies and the analyzes the interdependencies among the components of the IT architecture.

Another proposal is that of Braber et al. [14], who developed the CORAS framework to produce an improved method for precise, unambiguous, and efficient risk analysis of security-critical systems. CORAS focuses on the tight integration of viewpoint-oriented visual modelling in the RA process, using a UML-based approach in the context of security and RA. Although, both our approach and CORAS are asset-oriented, our approach distinguishes itself by considering the IT architecture in modelling the risk propagation.

Furthermore, our model is designed to be used with standard RA methods. Ciechanowicz [20] states a number of requirements for risk analysis methods. These requirements are grouped in 6 categories: common sense requirements, business requirements, functional requirements, security, audit and control requirements. Our model is compatible with Ciechanowicz's requirements.

In our approach we model the relations among the system components using so-called layers. The motivating idea is that layers enable concentrating on different attributes of assets, studying the interrelations between assets on different abstraction layers, meanwhile remaining expressive. Eck et al. [70] present GRAAL to provide a conceptual framework to describe an ICT architecture in a business. It differentiates

between business layer (events, communication channels and stimuli), software layer (system transactions, software library) and physical layer (network topology, machines) layers. Our approach is orthogonal to GRAAL, since we use the layered architecture of GRAAL for modelling IT-related confidentiality risks.

Another layered approach to RA is introduced by Innerhofer-Oberperfler and Breu [38]. Differently from our approach, they consider the enterprise architecture to model the interrelations between stake-holders, business processes and information assets. They use the model to derive security requirements that are linked to the threats and integrated in the RM process. Our approach instead is based on the IT architecture and on the propagation of confidentiality breaches. Therefore the two approaches may be used in a complementary way.

Finally, our model is designed for supporting the dynamic RA process, as the authors did in [76] in the field of availability RA and business continuity. As for the availability model presented in [76], this one is meant to be implemented by a tool and used to assess the risks in a continuously changing environment. This approach is especially suitable for organizations where it is important that the level of risk is constantly kept under control.

3.8 Concluding remarks

In this chapter we present a model-based confidentiality RA method, which takes into consideration the interdependencies between information assets and the IT architecture that they rely on.

Although the necessity of considering the interrelations between information assets and components of an IT architecture, as well as the protection of seemingly uncritical data, is indicated in present methods (e.g. NIST SP 800-30 [99]), it is not specified how this should be realized. Furthermore, the research in this field does not analyze the interrelations among IT assets and consequently does not systematically analyze the propagation of risk. This leads to risk analyses which are not as accurate as they should be and which cannot easily deal with changes in the architecture.

The DCRA method is a proposal to solve these problems. It represents a first step towards more accurate, more precise and dynamic assessment of confidentiality risks.

Chapter 4

Confidentiality Risk Assessment and IT Architecture Comparison[1]

Chapter 2 and 3 presents two methods which can be used for assessing IT security (confidentiality) risks accurately yet cost-efficiently. In this Chapter we address the second research goal:

G2: How can we assess confidentiality risks of IT systems cost-efficiently?

We do this by introducing the Confidentiality Risk Assessment and Comparison (CRAC) method. Like the DCRA method, CRAC is based on the IT architecture, but in addition it also elicits necessary input information, such as the volume of information that leaks and attack paths. It allows one to assess and *compare* risks of *distributed* IT systems with ordinal scale values.

4.1 Introduction

Nowadays, most data exchange within and across boundaries of organizations takes place electronically. Exchanged data often contains confidential information, the loss of which could result in economic damage. We call "confidential data" exactly the data that by business policy should not be disclosed to unauthorized users, e.g. business information, patient or client data, and passwords. Verizon 2010 Data Breach Investigations Report [79] shows that in 2009 141 breach cases were confirmed in which more than 143 million records were disclosed. Nine out of ten of those records were breached due to victims' lack of knowledge about the existence of digital information assets on system components.

[1]This chapter is a minor revision of the paper titled "CRAC: Confidentiality Risk Assessment and IT-Infrastructure Comparison" published in the Proceedings of the Sixth International Conference on Network and Service Management (CNSM'10), IEEE Computer Society, 2010

Good security, on the other hand, is also costly and even the best and most expensive countermeasures cannot mitigate all possible confidentiality incidents. Therefore, one of the goals of security officers is to realize and maintain an IT system which strikes the right balance between security, cost, and practicality. To achieve this, they typically refer to well-established standards and best practices such as ISO 27002 [94] and NIST 800-30 [99].

Assessing IT (confidentiality) risks becomes particularly challenging in the presence of cross-organizational cooperations, e.g. IT outsourcing. As part of the cooperation, organizations typically connect their IT architectures together and they grant access rights to each other's (confidential) information. This process establishes a so-called *network of organizations* which increases complexity of the confidentiality risk assessment because one has to deal with a more complex IT architecture and with an extended set of potential threats.

One of the crucial factors influencing the confidentiality risks a company faces is the *IT architecture* the company (and its outsourcees) employs to store its confidential data. This is intuitively obvious: an encrypted database behind firewalls accessible only via VPN is less vulnerable than a similar unencrypted database without firewalls protecting it. In general, how and where confidential data is stored has a big impact on its overall security. In spite of the obviousness of this statement, mainstream risk assessment methods take into little consideration the IT architecture where confidential data is stored. There are exceptions, for instance *attack trees* [61] provide users with a method to assess how vulnerable an architecture is. We discuss attack trees in the related work section.

Summarizing, to make informed decisions on the (security) design of its IT architecture, an organization needs to fully understand the confidentiality risks that derive from choosing a given architecture, particularly when there is a collaboration with other organizations. Therefore, decision makers need to be able to assess and compare different IT architectures also according to the confidentiality risks. This can only be achieved if risks are assessed consistently for each considered solution. However, the results of typical risk analysis methods cannot be consistently compared with each other if they were carried out by different people. This happens because they are mostly based either on the subjective opinion of the different risk assessor(s) or on event histories. Confidentiality risks are even harder to assess in an inter-subjective (independent of personal judgment) way, because of their non-functional nature and the (typical) lack of logs about past incidents.

To address these problems, we present the CRAC method. With the CRAC method one can determine the confidentiality risks by taking into account the effects of the leakage of confidential information (e.g. industrial secrets and user credentials) in the underlying IT architecture, and the paths that may be followed by different attackers (e.g. insider, outsider and outsourcee). We use *information flow* [52] to analyze where and "how much" critical information is located in the various parts of the system, and a customized version of *attack paths* [61] to analyze how attackers with different profiles may reach this information.

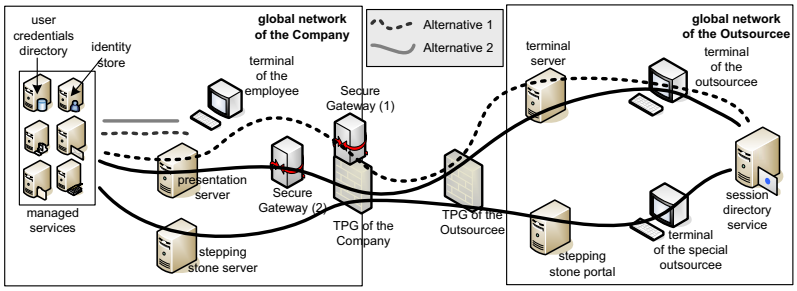

Figure 4.1: Simplified version of the two architectures and access paths

The main contribution of CRAC to confidentiality risk assessment is that it supports decision makers by allowing them to compare the confidentiality risks of alternative IT architecture design solutions.

CRAC is meant to support IT-enabled networks of organizations, in reducing the subjectivity of the (confidentiality) risk assessment results. We show the feasibility of the CRAC method by applying it in a real world case and by evaluating its subjectivity, practicality, and precision based on the success criteria that we discussed with the case study stakeholders. CRAC improves and extends the DCRA method that we present in Chapter 3 for confidentiality RA. We refer to the related work section for a description of the extensions and improvements over DCRA.

The underlying idea of the CRAC method is that of linking the likelihood of loss of confidentiality of a given data asset with its "reachability" within the IT architecture. This is at the same time the strength and the limitation of the method: on one hand it allows to reference the IT architecture as a crucial element for assessing confidentiality risks; on the other hand it is a limitation as it does not consider factors and risks such as insiders trading companies stock information to third parties. This is a design choice rather than a bug: CRAC can be easily integrated with other risk assessment methods and can be used as a specialized "plug in" when they need to assess the intrinsic confidentiality risks of an IT architecture.

4.2 The industrial case

In this section we present a real yet simple application of our method [2].

A large multinational electronics manufacturing company (from now on: *the Company*) is outsourcing the management of its authentication and authorization system (*the System*) to a multinational IT service provider (*the Outsourcee*). The System is used by the Company's employees to access the Company's data and services, and by the employees of the Outsourcee for configuring, monitoring, and maintaining the system

[2] Appendix A contains the application of the CRAC method to the case we present in Chapter 3.

(*managed services*). The Outsourcee proposes to replace the IT architecture on which the System is built. The company needs to know if the new IT architecture is at least as secure as the one that is currently in use.

The first architecture (Alternative 1 in Figure 4.1) is the one currently in use. All access attempts from the Outsourcee's network to the managed services are monitored by the *session directory services*. The *terminal server* makes applications available to the Outsourcee's employees in a terminal session. The *secure gateway (1)*, which is installed on the third party gateway (TPG) of the Company, is responsible for authenticating the Outsourcee's employees to the managed services. The *presentation server* is used by the Outsourcee's employees as an interface to manage applications. The second architecture (Alternative 2 in Figure 4.1) contains a second access path which allows a special group of the Outsourcee's employees to access the managed services in emergencies. Unlike the first path that uses two-factor authentication (i.e. an RSA token plus a password), the second path uses IP-based authentication.

The Company classifies information (for managing the digital access rights) in three confidentiality levels: private, highly confidential and company-confidential. Privacy-related information, e.g. user credentials containing social security numbers, are *private*; product patents are *highly confidential*; and IT-architectural documents (may cause a loss only if 3rd parties access them) are *company-confidential*.

The stakeholders involved are two independent business units of the company: the *Global Infrastructure Board* (GIB) and the *Risk, Performance Monitoring & Compliance Unit* (RMC). GIB owns the System and wants to know which of the two IT architectures is more robust with respect to confidentiality breaches. GIB also determines the business impact of confidentiality breaches, which in this case depends on (a) the criticality of an information asset, (b) the volume of information that gets disclosed, and (c) to whom the information is disclosed to. RMC customarily uses a checklist-based risk assessment method to assess the risks and compliance requirements of the Company's IT systems. From here on, we will call this method the *RA method*. The RA method is based on ISO 27001 [93] and NIST 800-30 [99] and customized for the needs of the Company. It consists of two main parts: (1) Business Impact Analysis; and (2) Threat and Vulnerability Analysis. According to RMC, the RA method does not allow linking threats to the component of the IT architecture under assessment. For these reasons, its results cannot be used for comparing alternative IT architectures.

4.3 The CRAC method

The CRAC method extends the IT-security-related concepts of ISO/IEC 27001 [93], by trying to assess how *difficult* it is for unauthorized users to access confidential information. The CRAC method is based on two ideas. (1) Information is a logical asset, so it can flow from one component to the other, e.g. it could flow to a component because a user copies it there. (2) An attacker may penetrate into a system through

different components and follow different attack paths. CRAC analysis consists of four steps:

Step 0: collecting the basic information;

Step 1: analyzing information flow and determining impact;

Step 2: analyzing attack propagation and determining reachability; and

Step 3: presenting and comparing risk.

We now illustrate these steps.

4.3.1 Step 0: Collecting the basic information

In this step we collect:

- the list of information assets present on the system, their confidentiality level and homogeneity property;
- the list of components that form the IT architecture of the system in scope and of the links among them;
- the list of relevant vulnerabilities; and
- the list of possible threat agents.

We adopt the following notation: **L** is the set of confidentiality levels (e.g. {top secret, confidential, public}); **N** is the set of number of instances of an information asset that can be retrieved from a component at once (e.g. {all, single, none}); **I** is the set of single impact values (e.g. {high, medium, low, null}); **TI** is the set of total impact values (e.g. {very-high, high, medium, low, null}); **P** is the set of qualitative likelihood values (e.g. {very-likely, likely, unlikely}); and **H** is the set of homogeneity values (**H** = {homogeneous, nonhomogeneous}). We call *information assets* the "semantic components of an information system that are required for an organization to conduct its mission or business" [43], e.g. customer information and user credentials. **A** is the set of information assets we consider. To each information asset $a \in \mathbf{A}$ we associate a confidentiality level $l : \mathbf{A} \to \mathbf{L}$. **C** is the set of components (i.e. hardware, software or network segment) which may contain one or more instances of a given information asset a. An information asset a is *homogeneous* if the damage due to its disclosure can be considered proportional to the number of its instances that are disclosed. For instance, "social security numbers" are homogeneous, since the damage due to the loss of one hundred social security numbers is larger than the damage due to the loss of a single social security number. Conversely, an information asset is *nonhomogeneous* if the damage due to the disclosure of one instance is as big as the damage of the disclosure of all instances. For instance, if the login credentials of one user are disclosed, the damage to the company is basically the same as if the credentials of 100 users with equal access rights would be disclosed. We model this with the mapping $h : \mathbf{A} \to \mathbf{H}$.

Running example - Part 4.1. *Following the information classification scheme that is adopted by the Company, we use the following sets:* $\boldsymbol{L} = \{\text{high, medium, low}\}$ *and* $\boldsymbol{N} = \{\text{none, single, all}\}$. *We furthermore limit our assessment to the information assets user credentials and business information. Employees of the outsourcee use the instances of user credentials to access the managed services.* $l(\text{user credentials}) = \text{high}$ *and* $h(\text{user credentials}) = \text{nonhomogeneous}$. *Instances of the information asset user credentials are available (among others) on the IT component user credentials directory. All instances of the user credentials in the user credentials directory can be retrieved at once. However, not all information assets can be retrieved from all components. The business information is data related to the business of the Company, for which we have* $l(\text{business information}) = \text{medium}$ *and* $h(\text{business information}) = \text{homogeneous}$.

4.3.2 Step 1: Analyzing the information flow

In this step we first analyze the logical and physical connections among components (in other words, we analyze the IT architecture). Then, we determine the impact of each component by considering the information assets that may flow to it. If there is a possibility for an information asset to flow to a component then we proceed as if that information asset was actually present on that component. We furthermore assume that information flows in a predictable way. Thus the information flow analysis can recognize all possible paths according to policies and documented properties of the components. More specifically, to model information flow we build for each information asset a a set of *flow paths*. A flow path is a path in the architecture graph which starts at a component where a is stored.

Definition 4.1. ***(architecture graph)*** *An architecture graph* $arch = \langle \boldsymbol{C}, \boldsymbol{E} \rangle$ *is a directed graph in which* $\boldsymbol{C}$ *is a set of vertices representing components and* $\boldsymbol{E}$ *is a set of edges* $\boldsymbol{E} \subseteq \boldsymbol{C} \times \boldsymbol{C}$ *and* $(c_1, c_2) \in \boldsymbol{E}$ *if and only if there exists a direct connection between* c_1 *and* c_2 *such that information can flow from* c_1 *to* c_2 *or an attacker who has access to* c_1 *can disclose the information available on* c_2.

The nodes of a flow path represent the components in which information asset a can be accessed by an attacker. We represent a flow path by an ordered list (with no repetitions) $fp = [c_1, c_2 \ldots, c_n]$ where $c_1 \ldots c_n \in C$ and $fp_a \in \mathbf{C}^*$. We call $\mathbf{FP_a} = \{fp_1, fp_2, \ldots, fp_m\}$ the set of flow paths of a, such that $\mathbf{FP_a} \in \wp(\mathbf{C}^*)$ (the power set of $\mathbf{C}^*$). We use (if desired) the maximum number of instances that can flow to a component c from its connected components to determine the number of instances an attacker can disclose by gaining access to c.

Running example - Part 4.2. *Figure 4.2 illustrates two information flow paths from* $\boldsymbol{FP_{\text{user credentials}}}$ *in Alternative 2. The user credential directory is the component where user credentials reside. For the sake of presentation we included in the paths only the components which are also listed in Figure 4.1. The remaining ones are represented by dots "…". In the leftmost path instances of the user credentials flow from the user*

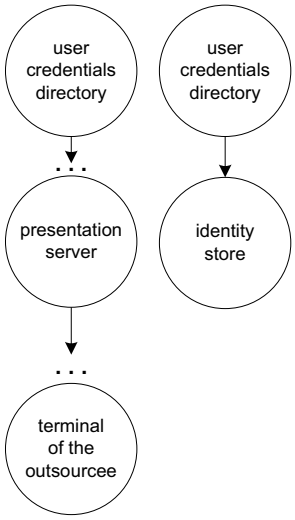

Figure 4.2: Two flow paths of the instances of user credentials

credentials directory to the terminal of the outsourcee, which is the terminal used by the employees of the Outsourcee. In the second path, credentials are synchronized between the user credential directory and the identity store.

After constructing the flow paths we determine for each information asset a, for each component c and for each flow path $fp \in \mathbf{FP_a}$, the number of instances of a that are retrieved from component c at once according to fp using the function n such that $n : \mathbf{A} \times \mathbf{C} \times \mathbf{FP_a} \rightarrow \mathbf{N}$. If we call $index(c, fp)$ the index of c inside fp, then $n(a, c, fp) = min_{i \leq index(c,fp)} n(a, c_i)$, where $c_i \in fp$.

Summarizing, fp allows us to determine how many instances of a are present on c. Now, we can determine for each a, c and fp, the impact of the disclosure of the instances of a which are present in c using the function $fp{-}imp : \mathbf{A} \times \mathbf{C} \times \mathbf{FP_a} \rightarrow \mathbf{I}$. This function considers the number of instances of an information asset which can be extracted from a component at once, its confidentiality level and homogeneity. Recall that $l(a)$ and $h(a)$ are respectively the confidentiality level and the homogeneity property of information asset a and that $n(a, c)$ is the maximum number of instances of a that can be extracted from c.

$$\textit{fp-imp}(a, c, \textit{fp}) = \begin{cases} l(a) \odot n(a, c, \textit{fp}) & \text{, if } h(a) = \textit{homogeneous;} \\ l(a) \odot \textit{all} & \text{, if } n(a, c, \textit{fp}) \neq \textit{none;} \\ \textit{null} & \text{, else.} \end{cases} \tag{4.1}$$

Here $\odot : \mathbf{L} \times \mathbf{N} \rightarrow \mathbf{I}$ is a monotonic composition operator for the values in **L** and **N**. $\odot$ should be agreed on with the stakeholders to guarantee that everybody understands

Table 4.1: Behavior of the $\odot$ operator

$\odot$	all	single	none
high	very-high	high	null
medium	high	medium	null
low	medium	low	null

how values are composed. As discussed before, quantitative values for the impact are difficult to obtain in practice. Consequently, the CRAC method determines the impact with partially ordered qualitative values, as it is commonly done in many risk assessment methods. However, if quantitative values are available, then the $\odot$ operator behaves as a multiplication. Now, we are able to compute the *impact* of the disclosure information asset a via component c, $imp : \mathbf{A} \times \mathbf{C} \rightarrow \mathbf{I}$, as

$$imp(c, a) = max_{fp \in \mathbf{FP_a}} fp - imp(a, c, fp) \tag{4.2}$$

Running example - Part 4.3. *We agreed with the Company on the binary merge operator $\odot$ on* **L** *and* **N** *as reported in Table 4.1. Let us now determine the impact of the disclosure of instances of user credentials and business information on the components of the architecture. Because the information asset user credentials is* nonhomogenous*, its confidentiality level is* high *and if in all the flow paths* n(user credentials, c, fp) $\neq$ null*, then the impact on all components in the architecture to which instances of user credentials flow is* high*. The business information is* homogeneous *and its confidentiality level is* medium*. The "number" of instances of user credentials flowing to the Secure Gateway is* all*. Accordingly, the impact of the Secure Gateway is* high*. On the other hand, the number of instances flowing from the Secure Gateway to its children is* single*. Consequently, the impact on the children of the Secure Gateway (e.g. the terminal of the special outsourcee) is* medium*.*

Summarizing, in this step we have built a set of information flow paths: one for each architecture graph, information asset, component the assets resides on and graph path. Then, we determined the impact of the leakage of the information asset stored on each component in the architecture.

We call *total impact* for component c the impact of the disclosure of all confidential information assets available on c. If c contains only one information asset a, then $imp(c) = imp(c, a)$. On the other hand, if c contains two or more assets (say a_1 and a_2) then we "add" $imp(c, a_1)$ and $imp(c, a_2)$. To this end we use the monotone operator $\oplus : \mathbf{TI} \times \mathbf{I} \rightarrow \mathbf{TI}$. As for $\odot$, $\oplus$ shall be agreed on with the stakeholders. More formally, the total impact of c is:

$$imp(c) = \oplus_{a \in A} imp(c, a) \tag{4.3}$$

Running example - Part 4.4. *We assume that on the component terminal of the special outsourcee both information assets, user credentials and business information, are*

available. Now, recall from Running Example - Part 4.3 that imp(terminal of the special outsourcee, user credentials) = high *and* imp(terminal of the special outsourcee, business information) = medium. *By applying the* $\oplus$ *operator we obtain the total impact:* high.

4.3.3 Step 2: Analyzing attack propagation

In the second step of the CRAC method we build the Attack Propagation Paths (APPs) which describe how different threat agents can penetrate into the IT architecture. Then, we determine the likelihood that a threat agent gets unauthorized access to the information available on each component. We call T the set of threat agents.

Running example - Part 4.5. *To assess the risks of the System we distinguish three threat agents:* $\boldsymbol{T} = \{employee, outsider, outsourcee\}$.

We call *vulnerabilities* weaknesses of the components which make attack propagation possible. We call **V** the set of all vulnerabilities. We represent the fact that v is a weakness of c with a mapping $w : \mathbf{V} \times \mathbf{C} \rightarrow \{true, false\}$, and the likelihood that a threat agent t exploits a vulnerability v to compromise a component c with the mapping $p : \mathbf{T} \times \mathbf{V} \times \mathbf{C} \rightarrow \mathbf{P}$.

To model confidentiality breaches we build for each threat agent t a set of APPs. The nodes of an APP represent the components that an attacker can compromise during an attack. We build each APP in two steps. We first add a node (c_1) to the APP for each component that can be directly reached by a threat agent (for external threat agents we can add a special component "the internet"). Second, we iteratively add new nodes and edges as follows: if node c_1 is in the architecture graph and is connected to the component c_2, then we add c_2 to the APP. Similarly to information flow paths, we represent an APP by an ordered list $app = [c_1, \ldots, c_n]$ where $c_1 \ldots c_n \in C$ with no repeated occurrences of c_i. We call $\mathbf{APP_t} = \{app_1, \ldots, app_n\}$ the set of APPs a threat agent t can follow.

Running example - Part 4.6. *Figure 4.3 illustrates two APPs that the outsourcee may follow to access the user credentials on the user credential directory, in the scenario of IT architecture 2. The outsourcee may access user credential directory via the terminal of the special outsourcee or via the terminal of the outsourcee. We iteratively included components that are physically or logically connected to the terminal of the special outsourcee and the terminal of the outsourcee until all connected components of the System are present in the APP.*

After constructing the set of APPs we can make an estimate of the likelihood that a threat agent t compromises each component c by following an attack propagation path in $\mathbf{APP_t}$. In doing so we need to take into account two properties: (1) each component may have more than one vulnerability that t can exploit, in this case we assume that the threat agent will exploit the vulnerability with the highest associated likelihood; and (2)

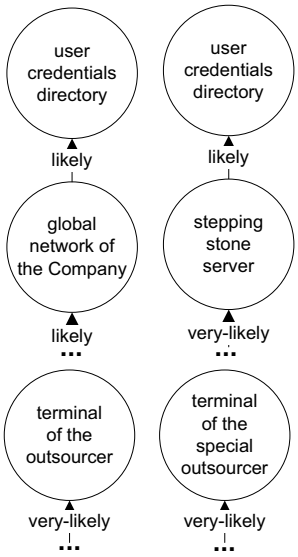

Figure 4.3: Two APPs followed by the Outsourcee

if the threat agent needs to compromise other components in order to compromise c, then we assume that the likelihood of compromising c is the lowest likelihood of the list (i.e. the hardest step).

Given a component c, a threat agent t, a set of vulnerabilities $\mathbf{V}$, an attack path $app \in \mathbf{APP_t}$, and $index(c, app)$ the index of c in the ordered list app, we call $p : \mathbf{T} \times \mathbf{C} \times \mathbf{APP_t} \rightarrow \mathbf{P}$ the likelihood of t compromising c by following app where

$$p(t, c, app) = min_{i<index(c,app)} max_{v \in \{v | v \in \mathbf{V}, w(v,c)=true\}} p(t, v, c_i) \tag{4.4}$$

Finally, by merging the likelihood of exploiting a component with respect to the alternative APPs, we determine the component's *reachability level*.

Definition 4.2. ***(Reachability)*** *Given a component c, the reachability level of c* reach : $\boldsymbol{C \rightarrow P}$ *equals the likelihood of the APP that leads to c and is the easiest (i.e. highest likelihood) among alternative APPs that can be followed by a threat agent t. Accordingly,*

$$reach(c) = max_{t \in \boldsymbol{T}}(max_{app \in \boldsymbol{APP_t}}(p(t, c, app))) \tag{4.5}$$

Running example - Part 4.7. *To enumerate vulnerabilities we refer to the threat and vulnerability list the Company uses in their RA method. Furthermore, to determine the likelihood of each attacker exploiting vulnerabilities of each component we cross-check the attacker profiles [31, 37, 50] with necessary conditions to exploit vulnerabilities. For instance, the only attacker profile outsourcee has is system knowledge. However*

physical access is the necessary condition for stealing a hard drive. Consequently, the likelihood of outsourcee stealing a hard drive is unlikely. *Following this method, we estimate that the likelihood that outsourcee will compromise the user credentials directory by following the path starting with the terminal of the special outsourcee is* likely. *That is because the likelihood of the hardest step in the path is* likely.

Finally, assuming that the user credentials directory is only on the APP of the outsourcee and the outsider, and the likelihood that the outsider will compromise it is very-likely. *Thus, we say* $reach(usercredentialsdirectory)$ *is* very-likely.

4.3.4 Step 3: Risk calculation and comparison

In this step we combine the output of steps 1 and 2 to identify the weak spots in the system and eventually compare the security of alternative IT architectures. We identify the weak spots based on their confidentiality *risk*.

Definition 4.3. ***(Risk)*** *Given a component* c *with total impact* $imp(c)$ *and reachability level* $reach(c)$*, we call the risk of* c *the pair* $risk(c) = \langle imp(c), reach(c) \rangle$.

After determining the risk of all components of each IT architecture we sort them. We identify the most critical components as those components with the highest total impact and reachability level. Then, we determine which architecture is more robust with respect to confidentiality risks by comparing the risk of assets on the different architectures.

Running example - Part 4.8. *In Step 1 we computed the total impact of the terminal of the special outsourcee,* high*. In Step 2 we computed its reachability level,* very-likely*. Accordingly, the risk of the terminal of the special outsourcee for Alternative 2 is* ⟨high,very-likely⟩*. Comparing the risk of the components of Alternative 1 and 2 we see that 1 is more robust than 2. In particular, Alternative 2 contains 2 additional components with risks* ⟨very-high,very-likely⟩*, and 3 components common with Alternative 1 that have higher reachability levels.*

For more complex systems presenting risk in a table may be unsuitable. For those cases we can calculate the percentage of components with the same total impact and reachability level, and present the results in a (smaller) matrix.

4.4 Evaluation

In this section we discuss how effective the CRAC method has been in our case study in terms of bringing the stakeholders closer to their goals.

4.4.1 Solution criteria

According to the stakeholders a successful confidentiality risk method should satisfy the following criteria:

(C1) the method should allow a detailed representation of risk;

(C2) the method should be practical to implement; and

(C3) the method should deliver less subjective results than the RA method.

We measure how well our solution scores with respect to these criteria based on the following measures:

(M1) the number of risk-related concepts the method is able to represent;

(M2) the percentage of optional risk-related concepts;

(M3) the percentage of adjustable risk-related concepts; and

(M4) the percentage of inter-subjective concepts.

(C1) indicates that a good risk assessment method should allow the risk assessor to represent the complexity of the system to be assessed in a detailed manner and is justified by the goal of RMC. We measure (C1) with (M1) and (M3). (M1) expresses the number of confidentiality-related concepts a method is able to model (e.g. attacker profiles, attack propagation and the amount of instances that can be disclosed). From here on we call them *concepts*. For this comparison we assume that all concepts are equal weighted. We implicitly assume that the more concepts the method considers the more precisely it can assess risks (we realize this is debatable, but we believe the figure obtained gives an indication that is valuable to assess (C1)). (M3) indicates the possibility of using the method with risk-related concepts at different detail levels. For instance, when considering threat agents, we look at whether the method considers only one type of threat agent (e.g. attacker) or many types of threat agents (e.g. insider, outsider, and outsourcee).

The accuracy of a risk assessment method (C1) often has a negative impact on the ease of its implementation (C2). The implementation effort of a method should ideally be adjustable to the sensitivity of the system to be assessed for confidentiality breaches (the RMC needs to assess the risks of less sensitive systems with lower effort than for highly sensitive systems). We measure (C2) with (M2) and (M3), which reflect the flexibility of the method. Flexibility is a desirable feature in case acquiring complete and detailed information is not possible because of limited resources. Here, flexibility can be described as (1) how well a risk assessment method can be adjusted to work at different detail levels and (2) how easy it is to refine or abstract the method at technical level.

The goal of the Company is to compare the confidentiality risks of two alternative IT architectures. This requires assessing the risks of these two IT architectures separately and then comparing the assessment results. Different risk assessors must be able to work on the two assessments. Therefore, the method they use must be inter-subjective (C3). Since the subjectivity of assessment depends on the subjectivity of

Table 4.2: Comparison of risk assessment methods

Measures	CRAC	CRAMM	Checklist
M1: number of concepts	17	25	16
M2: percentage of optional concepts	18%	4%	44%
M3: percentage of adjustable concepts	47%	36%	25%
M4: percentage of inter-subjective concepts	82%	72%	63%

the concepts used for determining the incident likelihood and impact, we measure it by the percentage of inter-subjective concepts (M4). The concepts that we consider as inter-subjective are: (1) documented facts (e.g. the components determined based on IT-architectural drawings), (2) the knowledge shared among all stakeholders (e.g. the list of vulnerabilities determined based on a publicly available vulnerability database) and (3) any combination of the first two.

4.4.2 Comparison

We now compare CRAC with two risk assessment methods with respect to the success criteria presented. The methods we consider are: the checklist-based RA method that the Company is currently using and the CRAMM method [84] a commercial product recommended by the British Standards Institution. For this comparison we disregard the governance-related concepts of CRAMM and the checklist-based method, which are outside the scope of this chapter. Table 4.2 reports a summary of this comparison (see Appendix B for details).

Regarding M1, using the CRAMM method one is able to take into account almost 50% more concepts than the checklist-based method and our CRAC method. Some of the concepts that the CRAMM method takes into account (and CRAC does not) are the number of persons using the assets, threat level and potential impact scenarios. However, there are also concepts that CRAC considers and CRAMM does not. They are homogeneity of information and volume of information flow.

Regarding M2, the checklist-based method allows to ignore 26% more concepts than the CRAC-method and 40% more concepts than the CRAMM method. [3]

Regarding M3, the CRAC method considers 22% more concepts with adjustable granularity than the checklist-based method and 11% more concepts with adjustable granularity than the CRAMM method. For instance, if the assessor wants to have a more detailed RA, then the CRAC method allows (but does not force) the assessor to consider the number of instances of an information asset on a given component. Consequently, with the CRAC method the risk assessor can adjust the granularity of the impact determination depending on the desired detail level of an assessed risk. Accordingly, we argue that the CRAC method and the checklist-based method are more practical to implement compared to the CRAMM method.

[3] We believe the optional concepts in CRAC are more evenly distributed between impact and likelihood than in the RA method and the CRAMM method.

When we consider M4, among the three methods, CRAC uses the highest percentage of inter-subjective concepts. It is followed by the CRAMM method, which uses 10% less inter-subjective concepts than CRAC. This happens because most of the information CRAC uses is either generally well-documented or it must previously be agreed on by all stakeholders. Although the CRAC method considers almost the same number of concepts as the checklist-based method, the checklist-based method has 19% more subjective concepts than CRAC. Accordingly, we argue that both the CRAC method and the CRAMM method represent confidentiality risks less subjectively than the checklist-based method.

Repeatability of CRAC depends on satisfying two assumptions: (A1) IT-architectural drawings on the system to be risk-assessed are available; and (A2) for risk assessment purposes staff with good security understanding can be interviewed. Providers of outsourcing services usually have to deliver a high-level IT-architectural document describing the system to be outsourced. Furthermore, large outsourcers usually employ security staff and a chief security officer. Therefore, if applied to a case where the outsourcee and outsourcer are big organizations, then both assumptions are satisfied.

We interviewed our industrial partners after applying CRAC in their system. We believe that one of the reasons why we achieved good results is that the CRAC method is specifically designed for assessing "confidentiality" risks, whereas the other methods aim to assess confidentiality, integrity, and availability risks at once. Furthermore, we developed the CRAC method with the success criteria defined by the stakeholders in our minds, whereas the CRAMM method is not developed to serve the goals of the stakeholders in this case.

4.5 Related work

Some well-known risk assessment methods, e.g. CORAS [14], CRAMM [84] and OCTAVE [16], give detailed recommendations about which modeling techniques are more suitable for which step of a risk assessment. The CRAC method can be used to extend these risk assessment methods by modeling confidentiality risk at IT architecture level: it links vulnerabilities to components and determines the reachability of these components taking into account the profile of a threat agent. Differentiating between threat agents and considering the effects of IT architectures on risk is essential for assessing IT risk.

In [4] we introduced the DCRA model. CRAC improves and extends DCRA for outsourced IT systems. In these scenarios detailed information on confidentiality concepts is not explicitly available, e.g. volume of information stored on each component. Therefore, the CRAC method presents a more practical approach that systematically elicits information on confidentiality concepts of not very confidentiality-critical systems. Here we consider the volume of information flowing and information flow paths. Furthermore, the DCRA method does not consider attacker profiles and to whom the information gets disclosed. These concepts have become especially critical

at cross-organizational cooperations. CRAC addresses these concepts by extending the DCRA method with the concept of threat agents for finer analysis of attack paths and impact determination.

For confidentiality it is essential to model how information flows. In the literature we find a number of approaches for modeling security with information flow graphs, e.g. [18, 45, 52]. Among them, only Chivers [18] uses information flow trees and forms attack paths for analyzing risk. Nodes in these graphs represent information carriers (e.g. data, messages, and events) whereas the edges represent system behavior (e.g. system functions and services). Therefore, the diagrams they propose cannot be used for comparing risks of two IT architectures.

Attack paths and attack trees are introduced by Schneier [61] and are widely used in the security literature (e.g. [15, 37]) to model different ways of compromising a system. In most cases, the nodes of an attack graph represent threats or vulnerabilities, as attack trees do. Our approach resembles attack trees because we model how an attack propagates. However, we carry out the propagation analysis at the IT architecture level.

4.6 Concluding remarks

In this chapter we present the CRAC method and how it can be used (1) as a supplement to the existing risk management approaches for practically assessing confidentiality risks of an IT system, and (2) as a stand-alone tool for comparing the security of IT architectures with respect to confidentiality. The CRAC method extends the concept of IT architecture-based confidentiality risk assessment in the absence of explicit information on confidentiality concepts by (a) eliciting impact-related information by modeling the information flow and (b) eliciting the reachability of information on critical information assets by modeling "attack paths".

We validate CRAC by applying it in two real world case, and evaluate with respect to the success criteria defined by the stakeholders. Accordingly, the CRAC method represents the confidentiality risks in a more detailed manner than the currently employed checklist-based method. We also show that CRAC can be gracefully adjusted to work at different detail levels and with both ordinal and ratio scale values, depending on the criticality of the system to be risk-assessed.

Chapter 5

Risk-Based Confidentiality Requirements Specification for Outsourced IT Systems[1]

Chapters 2, 3 and 4 presents methods that can be used for assessing IT security (confidentiality) risks of distributed IT systems accurately and cost-efficiently. In this Chapter we address the third research goal:

> *G3: How can we assess and control confidentiality risks of outsourced IT systems inter-subjectively?*

Specifically, we introduce the CRAC++ method. CRAC++ adapts the CRAC method to confidentiality requirements specification and control. It allows comparing the confidentiality requirements of an outsourcer with the confidentiality requirements of an outsourcee, and delivers a set of requirements that form the basis of the confidentiality service level agreement (SLA).

5.1 Introduction

As mentioned in Chapter 3 current regulations, e.g. Basel II [80], SOX [98], ISO/IEC 27002 [94], and BDSG § 42a [81], require companies to be in control of the security (confidentiality, integrity and availability) of their IT assets and to provide proof of this in the form of audit reports. We call this the *control requirement*; by implication the more detailed IT requirements derived from control requirements are also control requirements. Satisfying control requirements is usually perceived as not contributing to the company's products or services, so companies always aim at satisfying control requirements in the most cost-effective way.

[1]This chapter is a minor revision of the paper titled "Risk-Based Confidentiality Requirements Specification for Outsourced IT Systems" published in the Proceedings of the Proceedings of the 18th IEEE International Requirements Engineering Conference (RE 2010), IEEE Computer Society, 2010

Satisfying control requirements is further complicated because organizations outsource tasks which are not part of their core business, such as IT management, by doing so some of their IT architecture is now actually under the control of *other* organizations. In this chapter, we introduce and evaluate a method for identifying and specifying a particularly important control requirement in outsourcing, namely confidentiality of information.

Assessing the confidentiality risks of networks of organizations requires knowledge of the IT architecture of all organizations in the network, and also mitigation often requires actions in all organizations [35, 49]. However, this is challenging because, for instance in case of outsourcing, outsourcees are commonly large organizations that provide IT services to several outsourcers and are usually unwilling to modify their own architecture for the needs of a single client. Furthermore, to maintain confidentiality and protect business secrets, and to satisfy their *own* control requirements, outsourcees do not want to reveal more about their IT architecture than what is strictly necessary.

Outsourcees usually demonstrate their trustworthiness by showing their compliance to regulations, e.g. SOX [98], and additionally by independent audits, by means of reports under Statement of Auditing Standards No. 70 Service Organizations (SAS 70 [78]). An outsourcer who thinks that these compliance reports alone are not enough, must additionally specify the content of the audits in the form of SLAs that define service-specific requirements.

An SLA is a mid- to long-term contract that specifies service quality levels for the outsourcee and fines for failing to deliver them. SLAs usually specify quality levels for availability and response time, but so far in practice they are not used to specify quality levels for confidentiality. Yet today, outsourcers have to show that they satisfy the control requirement of treating their data confidentially, and so they now need to specify their confidentiality requirements in SLAs.

The problem with specifying confidentiality requirements in an SLA is that outsourcers do not want to specify a quantified confidentiality level, e.g. on average no more than 1% of the data will be lost per month. Furthermore, even if they would be willing to specify a confidentiality level like this, this attribute could not be monitored because, typically, confidentiality incidents are not observable by the outsourcer, both because attackers keep their actions secret and because the outsourcee would not allow any outsourcer to monitor the outsourcee's IT architecture. Thus another approach to confidentiality level specification must be chosen, that satisfies at least three criteria that we have identified so far:

C1 it does not specify confidentiality levels as percentages of data loss;

C2 it is not based on monitoring incidents; and

C3 it does not require an outsourcee to disclose confidential information.

Companies currently use checklist-based methods to assess the risks of an outsourcing architecture. These checklists neither explicitly consider confidentiality, nor provide

sufficient insights in confidentiality risks to support negotiation with the outsourcee. Discussion with several companies taught us that the method should satisfy several criteria additional to the ones mentioned above:

C4 The method should be usable with acceptable effort for the outsourcer. In particular, experienced risk assessors should be able to use it without following a course and it should not increase the time allowed for RA. We call this criterion *ease of use*.

C5 The method should deliver results (confidentiality control requirements to be included in an SLA) that are independent of personal judgment by making less use of subjective estimates than the checklist-based method. We call this criterion *repeatability*.

C6 The method should increase the outsourcer's understanding of confidentiality risks in this outsourcing relationship.

In this chapter we propose and evaluate a method that meets these requirements in the cases that we investigated. The method is based on specifying confidentiality requirements according to risk assessment results.

5.2 Research method

In this chapter we follow a nested problem-solving approach (Figure 5.1) described in the Section 1.3. At the top level we have the practical problem of specifying confidentiality control requirements of an outsourcer in an SLA. A *practical problem* is a difference between the real world and the way stakeholders would like it to be. To resolve it, some change must be applied to the real world. In this chapter we call this change a *treatment*.[2] In a rational problem solving cycle, the treatment is designed after an investigation of the problem and validated before implementation; and it is evaluated after implementation. In this chapter, we describe our treatment, the *CRAC++* method, in Section 5.3, and describe its validation from Sections 5.4 to 5.7.

The question whether a treatment is valid requires determining whether it will have the desired effects. This is a research question. To answer a research question, we have to *do* something, and this is a new practical problem at a lower level of nesting (Figure 5.1, middle column). Standard treatment validation questions are:

RQ1 what the effects of a treatment will be, and whether this will satisfy the stakeholders' criteria;

RQ2 how this compares to alternative treatments; and

[2]Earlier we called it a solution [73] but this hides the fact that a treatment may not solve the problem completely but only bring the stakeholders closer to their goals, or may even make the problem worse, as when a doctor prescribes a wrong medicine.

Figure 5.1: Structure of this research (The top-level problem is shown on the left.)

RQ3 whether this will work in other problem contexts too.

The middle column of Figure 5.1 shows a rational problem solving cycle in which the researcher investigates the research problem, designs research to answer the research questions, validates the research design, executes it and analyzes the execution.

To validate a method, we eventually need a realistic context in which the method is applied. Applying it in a toy problem is fine for illustration, and testing in an experiment is good for improving our understanding of the method. However, to know whether the method will work in practice, it has to be used in practice. This could be done by a field experiment, in which practitioners use the method to solve an experimental problem [65]. This is extremely expensive but not impossible. In our case, we opted for the more realistic option, given our budget, of using the method ourselves for a real world problem. In other words, we took an action research approach to validation [11].

We have acquired a case organization that needed to specify confidentiality requirements in an outsourcing relation (Section 5.4), and have used CRAC++ to specify confidentiality requirements that could be included in an outsourcing SLA (Section 5.5). We have evaluated whether CRAC++ brought the stakeholders closer to their goals (Section 5.6). This is the right column of Figure 5.1. Returning to the middle column, this case allows us to find a first, approximate answer to RQ1 (Section 5.7.1). Analyzing the mechanisms at work during our application of CRAC++ allows us also to assess generalizability (RQ3, (Section 5.7.3)), and the comparison with what would happen when using other methods. This allows to assess trade-offs with other treatments (RQ2, (Section 5.7.2)). We discuss the validity of our action research approach in Section 5.8.

5.3 CRAC++

The CRAC method compares confidentiality risks of two alternative networked IT architectures by analyzing how information can flow through a network, and how unauthorized persons can get access to nodes in the network. The information flow allows us to determine the information that can be present in a node of the network, and therefore allows us to assess the impact of a confidentiality breach (information disclosure) at that node. Combining this with an analysis of the possible access paths that an unauthorized person can exploit allows us to assess the risk of a confidentiality breach per node.

In CRAC++, we adapt this method to identify confidentiality requirements of the outsourcer that are *not* implied by the known confidentiality requirements of the outsourcee, and which therefore are candidates for inclusion in an SLA with that outsourcee. CRAC++ analysis consists of four steps:

Step 0: Collecting the basic information;

Step 1: Analyzing information flow;

Step 2: Analyzing attack propagations; and

Step 3: Determining candidate confidentiality requirements.

We now illustrate these steps.

5.3.1 Step 0: Collecting the basic information

Relevant documents to consider are IT architecture specifications, existing SLAs, best practices, relevant recommendations, standards and laws that contain confidentiality control requirements, e.g. the NIST vulnerability list [96]. Relevant stakeholders may include the company's security officer, system architect, and security architect.

At the end of this step the risk assessor has the following data, which is used in the following steps of the method:

Information assets We call information assets the data stored on system components. It may be both functional and organizational data that is of value to the organization, such as user credentials, client data and functional specifications of the system. We classify these information assets based on their confidentiality-relevant properties, such as the cost to the organization if disclosed. Information assets may have instances. For instance, if client data is an information asset, then each client record is an instance of this asset.

Threat agents These are potential attackers, e.g. hackers, or people who intentionally or accidentally access information assets they are not suppose to. We classify threat agents based on their estimated capabilities, such as system knowledge and hacking skills.

IT-architectural components These can be hardware (servers, terminals, routers, USB-sticks, a physical location (e.g. buildings), software (e.g. applications, operating systems, firewalls), or a network location (e.g. a network segment).

Relevant vulnerabilities A vulnerability is a condition of the IT architecture or its organization that facilitates confidentiality attacks on architectural components. For instance, if "reuse of storage media without proper erasure" is a vulnerability, then a threat agent may exploit this to access information. *Relevant vulnerabilities* are vulnerabilities that need to be mitigated according to the confidentiality requirements of the outsourcer.

Confidentiality requirements We make lists of both the confidentiality requirements of the outsourcer and those of the outsourcee.

5.3.2 Step 1: Analyzing information flow

First, for each information asset and each component on which the asset can reside, we build a set of *flow paths* (FP) that shows how this information can flow through the network. Recall from Chapter 4, that an FP is a path in the IT architecture graph that represents the flow of instances of an information asset from one information source, such as a database. Each node represents an architectural component. The combination of FPs tells us which information can be present in an architectural component.

The confidentiality expert together with the system architect form the FPs by analyzing desired or undesired retrieval of instances of each information asset over physical and logical connections between components. For instance, if the FP represents the flow of client data, a client record could flow from the client database over the network to the terminal PC. Note that there are components, such as a router, that have no confidentiality value because no information asset instance can be stored on them.

Next, for each component, a confidentiality expert together with the security officer assess the total value of information on the component. We call this value *total impact*, because it indicates the impact of disclosing information on a component.

At the end of this step we identify the components for which unauthorized access would create a total impact higher than a certain value that is determined by system owners (criticality threshold). We call these components *confidentiality-critical components*.

5.3.3 Step 2: Analyzing attack propagations

Having assessed the total impact of information loss per component, we now assess the likelihood that a component will be accessed by an unauthorized agent. Frequencies of access by unauthorized agents are not available, so we cannot assess this likelihood numerically. Instead, a confidentiality expert will assess, with the security officer and architect, the reachability of each component for each class of threat agent and will use this to estimate the risk of information disclosure per component.

Ease of exploiting vulnerabilities For this the confidentiality expert together with the security officer and security architect, assess for each threat agent the *ease* of exploiting vulnerabilities, based on the agent's capabilities. This assessment only depends on an assessment of the agent's capabilities and on the relevant vulnerabilities, and does not require knowledge of the IT architecture. We express ease of exploiting vulnerabilities in an ordered scale of fractions between 0 and 1, such as 2/3. The absolute numerical value of these fractions has no meaning, but their relative ordering expresses the expert's opinion about which exploit is usually more difficult for a threat agent. We assume here that these opinions can be totally ordered.

Ease of accessing one component Together with the security architect the confidentiality expert assesses the vulnerabilities of each component, and the effectiveness of

any preventive measures taken for these vulnerabilities per component. This is then combined with the previous analysis of ease of exploiting vulnerabilities by a threat agent. This provides us with an assessment of the ease of accessing a component for each threat agent. Again, the ease is qualitatively expressed in terms of a totally ordered set of values.

Reachability of components in the network If there were only one component in the network we would be done after assessing the ease of accessing this component. However, each component is part of the architecture and an attacker can take many paths through the network. To analyze the effect of possible paths an attacker can take to reach a certain component, we build a set of *Attack Propagation Paths* (APP)s that lead to each component and for each threat agent. Recall that - as defined in Chapter 4, an APP represents a path that an attacker can take to a valuable node. The nodes of an APP represent components of the system and the edges represent attack steps. The edges are annotated with the ease of this step for the attacker.

We construct an APP by first drawing nodes representing the entry points of the system and then gradually connecting further components by considering all possible propagations, until we reach a component that contains all instances of an information asset. These are the terminal nodes, because we assume that the threat agent will be satisfied when he reaches these nodes, either because this was his goal or because he is pleasantly surprised by what he finds there.

For each path from an entry node to a terminal node, we define the *bottleneck* of the path as the node that is hardest to access for the threat agent. The bottleneck may cause the threat agent to stop pursuing this path. For each terminal node c, we then select the path with the easiest bottleneck. The ease of access of this bottleneck is then by definition the reachability of c for this threat agent. Finally, for all APPs in which c is a terminal node, we define the easiest bottleneck as the *reachability* of c. Components with low reachabilities need attention. In particular, the outsourcer may want to require the outsourcee to increase the reachability of this component.

5.3.4 Step 3: Determining candidate confidentiality requirements

Confidentiality requirements of the outsourcer that are not implied by known confidentiality requirements of the outsourcee (and that affect the ease of exploiting vulnerabilities of confidentiality-critical components) are candidate requirements to be included in an SLA. First, we identify vulnerabilities against which the outsourcer wants to defend himself. For this we identify the outsourcer's confidentiality requirements that are not implied by known confidentiality requirements of the outsourcee. We assume that the related vulnerabilities are not mitigated by measures of the outsourcee and call these *unmitigated vulnerabilities*.

In Step 2 we have identified reachabilities under the assumption that the outsourcers confidentiality requirements were satisfied. Now, we have two scenarios: either the outsourcer asks the outsourcee to satisfy his own confidentiality requirements so that

all of the unmitigated vulnerabilities will be mitigated sufficiently (*best case*), or we do nothing (*worst case*). The best case has been dealt with in Step 2, so now we do the worst case.

Finally, the confidentiality expert compares the reachability levels of critical components in the best and worst cases and identifies the confidentiality requirements that the outsourcee must satisfy. These could be all of the requirements needed to realize the best case. More realistically, the security officer has to deal with finite budgets. Each additional requirement in the SLAs will increase the cost of outsourcing, and from this point on, confidentiality requirements' specification will be a negotiation between outsourcer and outsourcee. CRAC++ has provided the information security officer of the outsourcer with sufficient architectural information to conduct these negotiations, namely by allowing him to reason about what would happen if a requirement is included or dropped from the SLA. CRAC++ is therefore a method to support decisions about confidentiality requirements.

5.4 The case: Problem investigation

5.4.1 Stakeholders

The case we are about to describe regards a large multinational industrial company, which we refer to as X, with a total of 23,500 employees and divisions in 49 countries (Figure 5.2).

CIT is a competency center of X responsible for providing information and communication services to the system units. The confidentiality requirements of these services are defined in a Corporate Master Agreement (CMA) and if necessary detailed by Service Agreements (SA). The CMA contains control objectives that are extracted from the corporate rules. A *control objective* is a measure that indicates fulfillment of a control requirement. For instance, the control requirement

> "The organization's approach to managing information confidentiality and its implementation shall be reviewed independently at planned intervals ..."

is operationalized in the SA by the control objective

> "CIT shall provide yearly a compliance statement ... declaring compliancy to corporate regulations on confidentiality of service providing as contracted. ..."

The Managing Board of X is responsible for managing and protecting the benefits of all competency centers. Competency center managers yearly report to the Managing Board on the fulfillment of the requirements in the CMA.

One set of services provided by CIT to users in X is Enterprise Resource Planning (ERP). Furthermore, CIT has outsourced the ERP data center hosting services to an outsourcee. An Outsourcing Master Agreement (OMA) describes the quality attributes of the services that the outsourcee delivers to CIT and an SLA details the case-specific

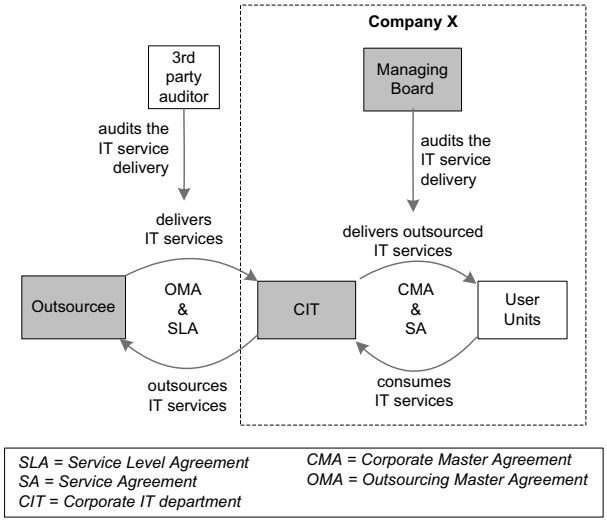

Figure 5.2: Stakeholders and their inter-relations with respect to the action case (Boxes represent stakeholders, arrows represent business relations, the dashed box indicates the boundary of X. Gray boxes are the stakeholders whose confidentiality goals affect the content of the OMA and SLA.)

requirements. There is only one rule in the OMA that describes the confidentiality-related quality attributes:

> "In protecting Confidential Information, [Outsourcee] will take all necessary precautions and the confidential information will be treated in the same manner and with the same degree of care as [Outsourcee] applies with respect to its own confidential information. [Outsourcee] shall keep all Confidential Information disclosed to it by X and [further clients of the Outsourcee] in secure places, under strict access and use restrictions."

The current SLA between the outsourcee and outsourcer contains no confidentiality requirements.

5.4.2 IT architecture

The ERP system and the hardware it runs on are owned by CIT but most of it is located in data centers owned by the outsourcee. The ERP database contains four business-confidential information assets:

- *Application information* is the business-related information of X, e.g. customer records and product prices.

- *Functional information* is monitoring-related information, e.g. access logs and IDS (Intrusion Detection System) rule sets.

- *User information* is information on the system users, e.g. roles and credentials.

- *Technical information* is the IT architecture-related information of the service, e.g. tunnelling data, of the ERP environment.

Figure 5.3 illustrates the IT architecture that supports the services in the scope of the case study. In the figure gray rectangles represent physical buildings owned by the outsourcee or by X and components with the same functionality that are located in the same network segment, e.g. User PCs, are represented by a single symbol. Firewalls between network segments provide IP-based access control.

5.4.3 Stakeholder goals

Table 5.1 summarizes the stakeholder goals and obstacles to goal achievement (our use of goals and obstacles is similar to KAOS [22]). Our aim is to find the confidentiality requirements that will help X to mitigate the effect of these obstacles.

As a response to recent changes in governance requirements, the Managing Board aims to be compliant with the Corporate Governance Code (G1). One consequence of this is that corporate units, such as CIT, have become responsible for the quality of the services that they deliver to users in the company. To audit this, the Managing Board requires the corporate unit managers to periodically present reports on the quality of the services they deliver to the System Users. However, the outsourcee does not allow CIT to directly analyze the confidentiality properties of the ERP system. The outsourcee periodically delivers third-party audit reports based on their own confidentiality requirements to CIT (O1). This is an obstacle to G1 because these audit reports do not reflect the confidentiality requirements of X but those of the outsourcee.

CIT aims to deliver the system users ERP services that are compliant with the corporate requirements of X (G2). However, the OMA and the SLA that specifies the ERP Data Center Hosting Service originate from a period before the new Governance Code, and therefore they do not cover the confidentiality requirements that follow from this code (O2).

The outsourcee aims to deliver ERP Data Center Hosting Services as specify in the OMA and SLA, and to convince X that his confidentiality baselines satisfy the confidentiality requirements of X (G3); but also to remain compliant with SOX [98] and SAS70 [78] (G4).

The outsourcee has difficulty keeping track of the technical changes related to delivered services and changing confidentiality requirements of its customers (O3). The outsourcee says that this is because the outsourcers are not communicating their confidentiality requirements clearly (O4).

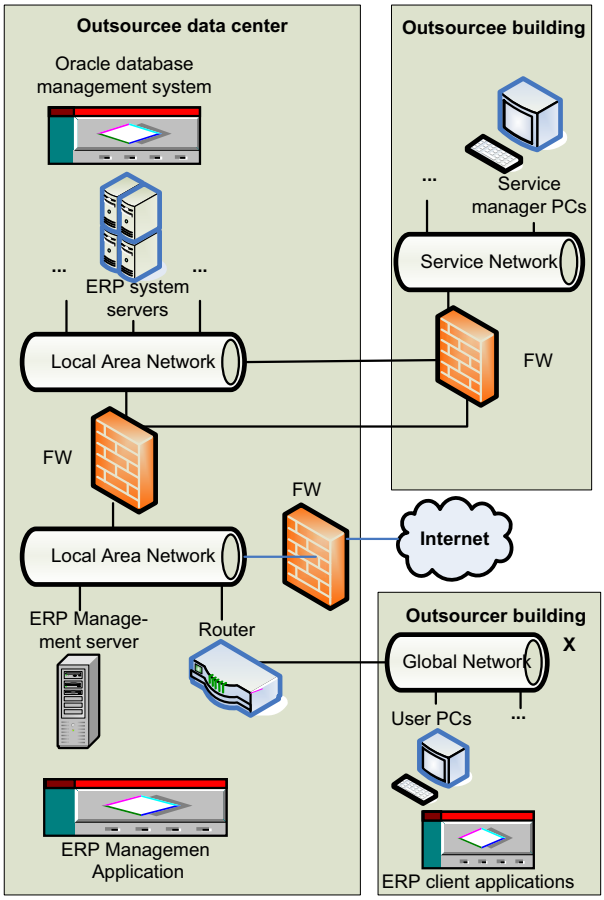

Figure 5.3: The IT architecture that supports the services in the scope of the case study (Gray rectangles represent physical buildings owned by the outsourcee or by X.)

5.5 The case: Applying CRAC++

Together with company X, we made a plan for applying CRAC++ and validated the plan with the decision makers to check whether this would help them reach their goals (treatment design and validation in the lowest-level cycle of Figure 5.1). After obtaining approval, we executed the plan. Here, we briefly report on the results.

5.5.1 Step 0: Collecting the basic information

At the end of this step we obtained the following information:

Table 5.1: Confidentiality goals and problems

Stakeholders	Goals	Obstacles
Managing board	**(G1)** To be compliant with Corporate Governance Code.	**(O1)** The outsourcee does not give direct insight into confidentiality of its systems to X.
CIT	**(G2)** To deliver user units of X CIT services that are compliant with CIT confidentiality requirements.	**(O2)** The SLAs and the OMAs do not contain confidentiality indicators, therefore it cannot be measured how well systems of the outsourcee meet the confidentiality requirements of X.
Outsourcee	**(G3)** To deliver ERP Data Center Hosting Services as specified in the OMA and SLA and convince X that the confidentiality level of the services they deliver is enough for the requirements of X. **(G4)** To remain compliant with SOX [98] and SAS70 [78].	**(O3)** Confidentiality requirements are changing dynamically. **(O4)** Outsourcers are not communicating their confidentiality requirements clearly

- a list of information assets (Application Information, Functional Information, User Information, and Technical Information) and their confidentiality values in a range of low to high;
- a list of components of the IT architecture of the system (basically, Figure 5.3);
- a list of confidentiality requirements and control objectives of the outsourcer and a list of control objectives of the outsourcee;
- a list of known relevant vulnerabilities (Some of these vulnerabilities are Unprotected Communication Lines, Possibility To Access The Applications Remotely, Weak Authentication and Inadequate Patch Management Process); and
- a classification of possible threat agents (Insider, Malicious Insider, Outsourcee, Subcontractor, and Outsider) and a classification of their competencies (Physical Access, System Knowledge, Technical Knowledge, Social Knowledge, Social Hacking Skills, Hacking Skills, and Motivation To Damage).

5.5.2 Step 1: Analyzing information flow

We produced 56 FPs and the total information disclosure impacts of components composing them. For instance, the ORACLE Server component is in the FPs modeling the flow of Application Information (with confidentiality value high), Functional Information (with confidentiality value low), User Information (with confidentiality value *high*) and Technical Information (with confidentiality value *low*). We determine the total impact of the ORACLE Server as *very high* by aggregating the confidentiality values of instances of the information assets that are stored on the ORACLE Server.

All in all, we identify that 27% of the components have *very high* total impact, 20% of the components have *high* total impact, 7% of the components have *low* total impact and the rest of the components have *null* impact. The CIT sets the criticality threshold as medium. Accordingly we say that 47% of the components are confidentiality-critical.

5.5.3 Step 2: Analyzing attack propagations

We constructed 72 APPs and assessed the reachability levels of components that comprise them. For instance, the terminal node ORACLE Server is in the APP of Insider, Malicious Insider, Outsourcee, and Subcontractor with the respective reachability levels $1/6, 1/2, 4/9$ and $1/15$. Consequently we define the reachability level of the ORACLE server as $1/2$, which indicates the easiest exploit.

5.5.4 Step 3: Determining candidate confidentiality requirements

In our case we did not have access to the list of confidentiality requirements of the outsourcee but we did have access to his control objectives, which operationalize the outsourcee's confidentiality requirements. We therefore first specified the control objectives of the outsourcer that are related with his confidentiality requirements. Then, we checked which of these were *not* implied by control objectives of the outsourcee. There were nine of those. Then, we assessed the reachability levels for the critical components mentioned in these objectives in the worst case and found that 20% of the critical components are affected by at least one of those nine objectives. Vulnerabilities of these critical components (unmitigated vulnerabilities) can be mitigated by adding to the SLA control objectives that mitigate these vulnerabilities.

For instance, the requirement of X "Removal of Property" is operationalized by the control objective

> "All items of equipment containing storage media shall be checked to ensure that any sensitive data and licensed software has been removed or securely overwritten prior to disposal."

This is not implied by any control objective of the outsourcee and so "Use of removable media is allowed" is one of the unmitigated vulnerabilities. It can be exploited by a threat agent to access the ORACLE Server. According to the worst case scenario we determined that the reachability level of the ORACLE Server is $9/18$. In the best case the reachability level of the ORACLE Server is $8/18$. The outsourcer may now use this information as an argument to include "Removal of Property" in the SLA.

5.6 Evaluation of stakeholder goal achievement

Evaluation of the achievement of stakeholder goals with the security officer of X and a representative of the outsourcee led to the following conclusions.

G1 By applying CRAC++, the necessary control requirements can be included in the SLAs. Consequently the audit reports of CIT to Management can improve their compliance to Corporate Governance Code.

G2 Including in the SLA confidentiality requirements that are currently not satisfied by the outsourcee allows CIT to provide services to units of X that comply with CIT confidentiality requirements.

G3 The outsourcee cannot be held accountable for requirements not stated in the OMA and SLA, which takes away O3. Furthermore, since the necessary control requirements are a part of the new SLA, the outsourcee is able to convince X that the confidentiality level of the services he delivers satisfies the requirements of X, which also takes away O4.

G4 CRAC++ does not require the outsourcee to disclose any confidential information to the risk assessor or to X, that he is not currently sharing. In return for this, the outsourcee must implement further confidentiality controls as specified in the new SLA; these do not negatively affect the outsourcee's compliance to SOX [98] or SAS70 [78].

5.7 Answering the research questions

5.7.1 RQ1

RQ1 analyzes whether CRAC++ satisfies the success criteria.

C1: Confidentiality level specified as percentage of data disclosure In Step 1, CRAC++ uses an estimation of the relative value of disclosure of instances of information assets to determine the impact and total impact of components. Furthermore, in Step 2, we estimate the relative reachability level of terminal nodes to determine the ease of disclosing instances of information assets. We use these in Step 3 to determine confidentiality levels, not to define levels of "acceptable" percentages of information disclosure. Therefore, CRAC++ satisfies C1.

C2: Incident monitoring CRAC++ does not depend on monitoring incidents but on domain-specific knowledge of the security officer about the capabilities of threat agents and the presence of vulnerabilities in components.

C3: Not disclosing confidential system information To identify the vulnerabilities of components and compare reachability levels in best and worst cases, we used only shared knowledge about the IT architecture of the outsourcee.

C4: Ease of use In the field study understanding X's problem (lowest level problem investigation in Figure 5.1) and conducting Step 0 of CRAC++ took one week. One additional week was necessary for developing the spreadsheets and executing the other steps. X has so far no experience with confidentiality RA, but our practical experiences show that assessing risks of a similar size system with a checklist-based method usually takes one to two weeks. Also, the security officer of X said that the steps and results were easily understood and if tool support is provided then they would be able to use it.

C5: Repeatability We compared the repeatability of CRAC++ with that of the checklist-based method of the company by counting the number of concepts that require subjective interpretation used in each. We have excluded the concept "risk" from the checklist-based method, because CRAC++ does not aim to present risk, and the concept "unmitigated vulnerabilities" from CRAC++, because the checklist-based method does not aim to elicit requirements. The result is that 3 out of the 20 (15%) CRAC++ concepts are subjective and 5 out of the 15 (33%) checklist concepts are subjective (see Appendix B for details). We consider this an indication that the method is less subjective (more inter-subjective) than the checklist-based method.

In Chapter 4 we have shown that the CRAC method is more repeatable than CRAMM [84] and the checklist-based method. For instance, if the assessment would be conducted with the CRAMM method, then in total 72% of the variables would be inter-subjective (see Table 4.2). Therefore, we conclude that CRAC++ is more repeatable than assessing confidentiality risks with CRAMM and specifying control objectives as we described in Step 3 of CRAC++.

C6: Increased understanding After applying CRAC++ to the case, CIT reported increased understanding of the effects of confidentiality requirements on the confidentiality levels of the components and was able to prioritize them according to the impact of incidents. So for this particular case, CRAC++ increased understanding.

5.7.2 RQ2

RQ2 compares CRAC++ with the alternative treatments. As an alternative treatment to achieving G2, X suggested to monitor the outsourced IT systems with a Security Incident and Event Monitoring (SIEM) tool. However, SIEM tools generate logs with confidential data and possibly increase the criticality of components, so they increase the confidentiality risks for X. They also require the outsourcee to disclose confidential information, which violates C2.

As another alternative treatment to achieve G2, X executed a third party audit based on the control objectives of CIT. However, this treatment did not succeed either. Although the audit report indicated some noncompliance, X did not have a mechanism to enforce the outsourcee to implement measures. Furthermore, the control objectives

of X were not linked to risks. So, X had no identification of how to mitigate the risk by applying measures to the part of the system that he has control over.

5.7.3 RQ3

RQ3 analyzes in which contexts CRAC++ is usable. CRAC++ makes a number of assumptions about its context of use. These assumptions govern its reusability in different contexts. We assume (A1) that the outsourcee does have confidentiality control objects and that he satisfies these. The CRAC++ method does however not contain a step to check this. Furthermore, CRAC++ does not assume that the outsourcee discloses confidential information or that the outsourcer has quantified the value of information assets or the likelihood of unauthorized access per component. By implication, (A2) we do assume that there are security officers who have informed opinions about this, and the method then helps in drawing conclusions from these opinions.

Large outsourcees are subject to control requirements and will satisfy A1. Large outsourcers with a security staff and chief security officer will satisfy A2.

So far, we have applied CRAC++ twice with similar results, both in multinational industrial companies where confidentiality was not a critical requirement until external regulators enforced it. Operating in highly competitive markets, these companies are very cost-sensitive and they will therefore not aim at maximum confidentiality. This might be different in privacy-sensitive organizations such as health care or insurance companies, or in high-confidentiality organizations such as the military. We do point out though that the qualitative assessments in CRAC++ could be replaced by more quantitatively informed techniques without changing the overall logic of the method. Nevertheless, as a third assumption for use we hypothesize that (A3) in the context of use, confidentiality is not the highest-priority requirement.

All of this supports reusability to any context that satisfies the three assumptions, with similar answers to the research questions for those contexts

5.8 Threats to validity

In the previous section we proposed answers to three questions relevant to the validation of CRAC++. Now we analyze the validity of these answers themselves: what is the risk that we answered the questions incorrectly? The higher this risk, the lower the validity of our answers.

Answering RQ1, we found that CRAC++ satisfies C1, C2, C3, and C6; analyzing the reasons for these answers we find no reasoning errors or observational mistakes so we claim these answers are valid. C4 could not really be checked, since the user of the method is also the inventor of the method. More systematic usability studies would require tool support.

Repeatability (C5) has been checked indirectly by counting the number of subjective

concepts. This is not a sure indicator of repeatability, but it does provide the basis for a rational argument about repeatability.

CIT reported increased understanding (C6), but we did not apply a formal test (e.g. an exam) to get prove of this.

The comparison with other approaches (RQ2) does not introduce new threats to validity that we can think of.

We answered the reusability question (RQ3) by identifying the conditions under which CRAC++ can be used, and actually showing that it could be used in another case satisfying these assumptions. Like all inductive conclusions, our conclusion that CRAC++ can be used in other cases is uncertain, but because we used analytic reasoning rather than statistical reasoning, we cannot quantify this uncertainty.

5.9 Related work

Several methods have been proposed for managing security when outsourcing IT management [36, 46, 55, 60]. Data Protection Agreement (DPA) [56] specifies what an outsourcee may and may not do with the outsourcer's data. CRAC++ can be used to identify relevant confidentiality requirements for a DPA. Insurance Contracts (IC) [30] define security requirements based on past incidents, which, for confidentiality, is not realistic. Protection Level Agreement (PLA) [41] specifies metrics to define protection levels. This can be used in combination with CRAC++.

Haley et al. [33] describe a method for defining security requirements as constraints on functional requirements. This differs from CRAC++ because we focus on confidentiality, which is independent of functional requirements of the system, and serve the control objectives that are imposed by regulators. We do make explicit trust assumptions as Haley et al. [32] do, because we assume that the outsourcee can be trusted to satisfy his own control requirements.

Common Criteria [89–91] evaluation is a further tool that organizations use to present that they are in control of security for the products they deliver. However, these evaluations consist of merely comparing two sets of requirements and neither enforce verification nor assure effectiveness and correct implementation of requirements. Due to its IT architecture-centered character, CRAC++ provides traceability between the requirements and the security features of the system components. Thus it assures effectiveness and correctness of requirements. Furthermore, it allows updating the evaluation results easily in case of changes in the system components. Mellado et al. [48] introduce a security requirements engineering method that is based on reusing the results of previous evaluations. However, they express the accuracy and veracity of requirements in terms of incident propagations.

5.10 Concluding remarks

This chapter is based on two ideas: (1) confidentiality requirements specification cannot be based on incidents, but must be based on an assessment of the risk of disclosure of confidential information; and (2) requirements specification in an outsourcing relation is budget-constrained. Our case studies and analyses indicate that CRAC++ can satisfy our criteria (C1-C6). We have also shown that IT architecture-based assessment of confidentiality risk provides the necessary information to form a confidentiality SLA. Such an SLA provides control over the confidentiality risks that arise due to outsourcing IT systems.

Chapter 6

Risk-Based Security Requirements Elicitation and Prioritization[1]

Chapter 4 addresses G2 (How can we assess confidentiality risks of IT systems cost-efficiently?) by analyzing the IT architecture for information flow paths and attack propagation paths. In this chapter we go back to G2, and address the goal of eliciting necessary input information cost-efficiently by linking business goals to IT components.

We do this by introducing the Risk-Based Requirements Elicitation and Prioritization (RiskREP) method. RiskREP provides necessary input information, e.g. business value of assets, and helps analyzing the effects of security requirements on business goals. It does this by deriving security requirements from business goals and prioritizing these requirements according to the risks they mitigate.

6.1 Introduction

As we have seen in the preceding chapters, protecting the security of IT assets is difficult, and becomes even more challenging when the IT architecture changes in time. This is often the case especially for systems that are under development. For instance, as the physical or logical location of a server or the authentication method changes. Thus, to protect the security of IT assets cost-efficiently one needs to keep track of the dynamics of the IT systems starting from an early development phase. To keep track of the dynamics of the IT systems, it is necessary to have a light weight requirements elicitation method which can easily and reliably be repeated. Such a process should document requirements and the reason why these requirements are necessary, so that when there is a change in the system its effects can be easily analyzed.

The elicitation of IT security requirements is a challenging process. During it, tacit assumptions need to be made explicit and combined with expert knowledge of

[1]This chapter is a minor revision of the paper titled "RiskREP: Risk-Based Security Requirements Elicitation and Prioritization" which is submitted for publication to RE'11.

stakeholders from different domains. These domains are the business domain, the IT domain and the security domain.

Most of the present requirements engineering (RE) frameworks, e.g. [26, 51, 71], allow one to gather the system requirements from system owners without explicitly differentiating which stakeholder knows the most about which domain. This leads to long (costly) and inefficient meetings which all stakeholders (or their representatives) have to attend. To increase the efficiency of information elicitation and requirements management we need a systematic and stakeholder-specific method.

As Dubois et al. [23] state, security risk models are largely unable to address cost-effectiveness concerns in a satisfactory manner. A cost-effective security evaluation requires prioritizing possible countermeasures according to their costs and effectiveness with respect to business goals.

The importance of integrating security in RE is indicated by many researchers, e.g. [26, 50, 67, 71]. These researchers usually extend currently available RE methods and supporting diagrams so that they can model and analyze the effects of security-related concepts. In theory it is usually possible to integrate these methods with each other. However, in practice integration is challenging due to overlapping activities and different granularities at which each method analyzes non-functional requirements. Therefore, we argue that an *integrated method* is necessary. Such a method should combine the security requirements elicitation and management features of both academically and commercially acknowledged methods. By comparing the features of these method (see also Table 6.1 and 6.2) we put together a list of features that the integrated method should have. These features are:

- provide a systematic process for eliciting requirements;
- consider different stakeholders' perspectives when eliciting input information, which means to differentiate between business goals and quality goals of the IT system;
- consider both intentional use and misuse;
- consider both impact and likelihood;
- systematically draw diagrams and form tables;
- consider how effectively requirements mitigate incidents;
- provide trade-offs among conflicting requirements; and
- consider the implementation cost of requirements.

In this chapter, we present a systematic and cost-efficient requirements elicitation model supported by a method that prioritizes security requirements according to the risks they counteract. We developed this solution by integrating concepts of the CRAC++ method (see Chapter 5) with the concepts of MOQARE [34] (a method for

systematic requirements elicitation) with the aim of providing an integrated method. The new method describes stepwise how to identify the quality goals with the top-down approach of MOQARE and link them to security risk of IT assets that are made explicit with the bottom-up approach of CRAC++. The objective of this solution is to identify the most effective set of requirements at an early development phase in a repeatable way.

We claim that the main strengths of RiskREP are: (1) Business-IT-alignment by linking business goals traceably to IT requirements and vulnerability analysis. The latter means that the priority of each IT requirement can be traced back to the business goals which it supports based on documented rationales which answer the question "Why is this requirement important in this specific system?". (2) It combines a graphic overview presentation (used on the business perspective of the analysis) with better scalable table presentations on the user perspective. This is important for structuring the information elicitation process. Furthermore, (3) it provides a clear process with phases which demand well-defined knowledge and therefore specific stakeholders. It is clear which phase demands the contribution of management or of a security officer. This is important for gathering input information cost-efficiently.

6.2 Background: MOQARE in a nutshell

In this section, we introduce the main elements of the MOQARE [34] method. We presented the CRAC++ [1] method in Chapter 5. These methods constitute the basis for RiskREP.

MOQARE (Misuse-Oriented Quality Requirements Engineering) is a method for the top-down elicitation of quality requirements. Its fundamental principle is to combine the elicitation of wanted elements and unwanted elements. The MOQARE analysis starts with the business perspective where business goals are identified, i.e. the reasons for developing the system. Business damages (which are defined in Section 6.4) are also identified. The business damages are partly caused by quality deficiencies of the system, and these are further analyzed by defining quality goals for the system. Threats to these quality goals are then elicited in the form of misuse cases [64]. Misuse cases are similar to use cases, but they describe in scenario form what must not happen, such as intentional attacks or user errors. Then, one seeks for countermeasures, which are requirements for a system or its development that can detect, prevent or mitigate the misuse case. MOQARE has shown its merits of systematically supporting the creative process of deriving realizable non-functional requirements from abstract quality goals (such as data security) and documenting rationale of the requirements. This process is supported by checklists for threat agent types, potential assets and their vulnerabilities and threats which endanger them. However, the prioritization of the misuse cases and requirements is left to experts and not supported by the MOQARE process.

Table 6.1: Comparison of widely known RE methods with respect to requirements elicitation and prioritization

	Elahi and Yu [25]	Misuse Cases [64]	extended KAOS [71]	ATAM [42]	NFR framework [51]
Requirements elicitation					
Systematic process	drives soft-goals from goals	no	no	no	drives soft-goals from goals
Differentiation between business and quality goals	goals and soft-goals	no	functional goals and non-functional goals	yes	technical objective and business objective
Considering both intentional use and misuse	intentional use and misuse	use cases and misuse cases	goal and anti-goal	no	no
Considering different Stakeholder views	yes	yes	no	yes	no
Requirements prioritization					
Systematic estimation of impact	no	no	no	no	no
Systematic estimation of incident likelihood	level of evidence	no	for determining the granularity	no	no
Prioritization based on monetary costs of requirements	yes	real cost	no	volume of change	no
Considering effectiveness levels of requirements	3 level differentiation	no	no	no	no
Considering combined effects of requirements	between soft-goals	no	no	trade-of points	between soft-goals

6.3 Related work

In this section, we compare widely known requirements engineering and risk assessment methods based on their requirements elicitation and prioritization properties. Table 6.1 and 6.2 present an overview of this comparison. Please note that here we present only those methods that satisfy most of the properties. We consider these properties as success criteria by developing the RiskREP method.

To systematically elicit security requirements Elahi and Yu [25], Stamatis [67] and Mayer et al. [47] propose to derive requirements from high-level goals. This is especially important for the completeness of the requirements and applicability of the method. On the other hand – as we argued in the introduction – we believe that a *security* requirements elicitation method should also differentiate between business goals and quality (security) goals. Despite the fact that most of the approaches that we compare, e.g. [26, 42, 51, 71] differentiate between functional and non-functional goals, none of them differentiate between business and quality goals.

To address the security concerns of system owners, recently developed requirements engineering methods, e.g. [25, 39, 64, 71], model not only intentional uses but also

Table 6.2: Comparison of widely known RA methods with respect to requirements elicitation and prioritization features

	FMEA [67]	Attack Graphs [54]	CORAS [14]	Goal-Risk Model [10]	GSRM [39]
Requirements elicitation					
Systematic process	yes	no	no	no	yes
Differentiation between business and quality goals	no	no	no	strategic layer, event layer and treatment layer	project goals and sub-goals
Considering both intentional use and misuse	no	no	no	no	risk events and tasks
Considering different Stakeholder views	no	no	yes	no	yes: user representative, business analyst, requirements engineer, risk manager
Requirements prioritization					
Systematic estimation of impact	failure effect	no	depends on selected model	no	risk impact
Systematic estimation of incident likelihood	occurrence of failure	probability, average time or cost/effort	depends on the model	level of evidence	risk likelihood
Prioritization based on monetary costs of requirements	no	financial loss or loss of system	no	yes	no
Considering effectiveness levels of requirements	detection rate	no	no	contribution relation	effectiveness
Considering combined effects of requirements	no	no	no	between goals	no

misuses of system components. However, eliciting information on intentional uses and misuses requires expertise of stakeholders with different backgrounds. Only a few of the approaches that we consider in this comparison ([14, 25, 39, 42, 64]) express how different stakeholder views can be considered by eliciting information.

Once the security requirements are identified, one has to check whether they are implementable within the available budget. Usually, this is not the case, and one has to decide which set of requirements should be implemented and which requirements can be disregarded. Making such a decision requires a fine-grained estimation of the security risks the system is exposed to, considering the trade-off among the different requirements, as well as their costs and effectiveness. However, only some requirements engineering methods (such as FMEA [67], Tropos-based approaches [10, 25], GSRM IH10, Attack Graphs [54], extended KAOS [71], and the approach propozed by Mayer et al. [47]) take into consideration the risk the system is exposed to.

The methods that take into consideration effectiveness levels of requirements refer to different attributes of the IT system that is analyzed. Elahi et al [26] differentiate among three levels according to whether the countermeasure alleviates the effects of

vulnerabilities, patches them or prevents malicious activities. Goal-Risk Model [10] differentiates between four levels based on contribution relations between security events and goals. Finally, FMEA [67] differentiates according to incident detection rate.

When applied together, requirements may contradict with each other or support each other. Elahi and Yu [25], NFR framework [51], Mayer et al. [47], and Asnar and Giorgini [10] consider these combined effects and prioritize the system requirements accordingly. ATAM [42] also considers how countermeasures affect each other and refer to it as "trade-off points".

6.4 Meta model

In this section, we briefly present the concepts of RiskREP. To note that we do not intend to present a conceptual framework for security requirements engineering as Fabian et al. [27] do, but to illustrate how RiskREP concepts are connected to each other at meta level. The meta model consists of concepts belonging to three perspectives: the business perspective, the user perspective and the technical perspective. Figure 6.1 illustrates these concepts and their relations.

Business goals are desired properties of the business. They are stakeholder-specific and might be supported by the IT system. Business goals finally justify system requirements. An example of a business goal is "efficient business processes".

A *business damage* is a state or activity of the business that *violates* a business goal. The business damage completes the business view by stating what should not happen.

Quality goals are desired qualities of the IT system, i.e. a desired state of the system. They are non-functional system goals that *support* business goals. These goals are expressed as high-level quality requirements that consist of a quality attribute and an asset, like "confidentiality of password". They help to focus the analysis of the quality of an IT system on the most important quality attributes.

Quality attributes are attributes of the system to be protected. They describe aspects or characteristics of quality, e.g. confidentiality. We use the quality attributes of ISO 9126 [88] and assume that these completely categorize all relevant aspects of an IT system's quality.

Assets are parts of the system that are valuable for the organization, e.g. information, software, and hardware. They need to be protected from malicious activities in order to achieve business goals.

Value quantifies the criticality of each quality goal with respect to the business. The value is used to prioritize the quality goals with respect to each other. It is determined by the *impact* that the compromise of an asset would cause to the business.

A *quality deficiency* is a lack of quality attribute for an asset that *violates* quality goals and *causes* a business damage.

A *threat agent* is a person, i.e. an insider, an outsourcee or an outsider, who

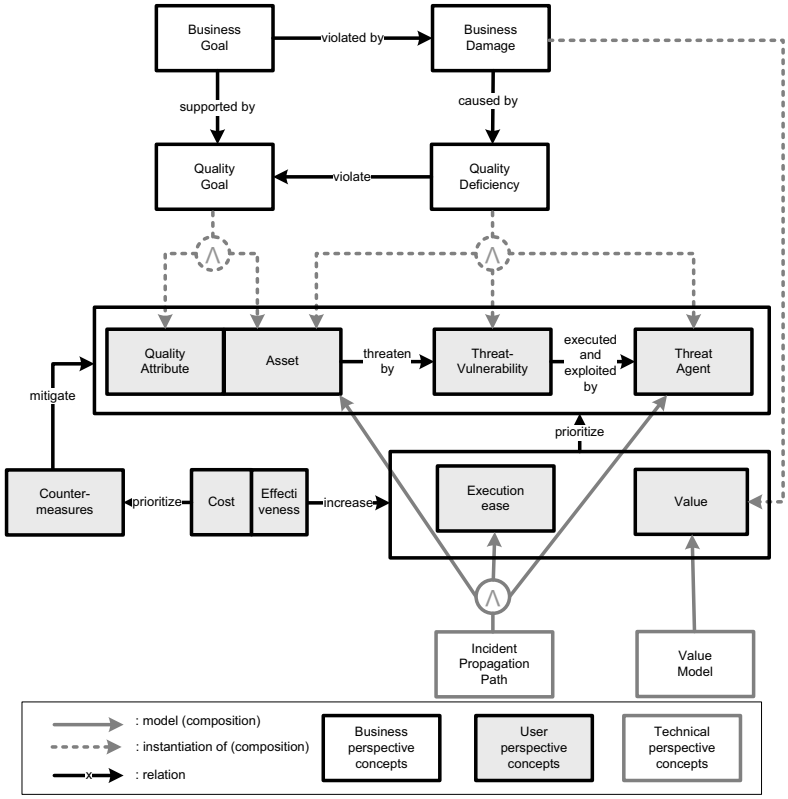

Figure 6.1: Meta-model showing the concepts and their interrelations

intentionally or unintentionally executes a threat. A threat agent can be characterized in terms of his motivation, goal and attributes, e.g. a disgruntled employee.

Threats are actions which cause a quality deficiency that causes the violation of a quality goal, e.g. data theft threatens the confidentiality of data.

Vulnerabilities are a property of the assets, the IT system or its environment that can be exploited by threat agents. A vulnerability can be misused, and this could yield the violation of a quality goal. Identifying the vulnerabilities and determining the assets that are threatened by them help analysts determine the effectiveness of countermeasures that mitigate them. Vulnerabilities can be "lack of technical change management" or also wanted properties of the system such as "Single-Sign On", if they can enhance the execution ease of a threat.

Misuse Cases (MCs) describe as scenarios how a threat agent may cause a quality deficiency. The MC takes the perspective of the user and describes what happens at the

interface between user and system. The MCs are prioritized based on their *execution ease* and the *impact* they cause to the asset(s).

Incident Propagation Paths (IPPs) are descriptions of MCs from the technical perspective. In some cases an IPP consists of several interconnected steps, i.e. a threat agent causing a quality deficiency on an asset by executing one or more threats, which exploit vulnerabilities of several assets. Such IPP scenarios are important for humans to imagine the flow of events including the causes and consequences of incidents. Like the MCs, the IPPs are prioritized based on their execution ease and the impact they have. There may be several IPPs realizing the same MC.

The *execution ease* of an attack is an estimation of the effort required to carry out an attack. It it is determined by the hardest vulnerability that needs to be exploited to carry out the attack. In our approach, the execution ease is considered to be correlated to the likelihood that a threat is actually executed by the "strongest" threat agent.

Countermeasures are mitigation, detection or prevention mechanisms. They partly or completely counteract a threat/vulnerability pair or a threat agent, and transform the asset they are applied to into a more secure asset. Countermeasures are expressed as quality requirements on the IT system.

The *cost* of a countermeasure includes the implementation cost and the cost of ownership. Depending on the depth of the assessment we either use partially ordered scale or the real costs. In case the real costs are used then the risk expert may calculate the implementation cost based on required man hours and average labor cost per hour.

The *effectiveness* of a countermeasure is given by the expected risk reduction it achieves. Most countermeasures either influence the impact or the execution ease of an IPP.

6.5 The RiskREP method

In this section, we describe the steps of the RiskREP method and the activities associated to them. Due to the dynamic nature of IT systems, security requirements elicitation and prioritization is an iterative process. This process consists of four steps:

Step 1: finding quality goals;

Step 2: analyzing security risks;

Step 3: defining countermeasures; and

Step 4: prioritizing countermeasures.

The information that the RiskREP method uses is elicited from three stakeholder categories, i.e. business owner, IT manager and security officer. Two additional stakeholders are the requirements engineering expert and the risk expert, who elicit the necessary information from semi-structured interviews with the other stakeholders and execute the RiskREP method.

6.5.1 Step 1: Finding quality goals

RiskREP begins by identifying the business goals (BG). For this, the RE expert asks the business owners to define their goals. After identifying the BGs, the RE expert makes an estimate of the business damages (BD) by considering what may violate the achievement of these business goals. Then, she identifies quality deficiencies (QD) that may cause a BD. For this, she analyzes quality-related deficiencies of the IT system for the context in which the system will be used. Once the QDs are identified, she derives quality attributes (QA) that need to be protected from the QDs. Finally, she derives quality goals (QG) from QAs by relating affected assets.

6.5.2 Step 2: Analyzing security risks

The aim of this step is to analyze the security risks related to each QG. For this, the risk expert first identifies possible misuse cases (MC) that may threaten the QGs and then makes an estimate of their execution ease (meaning: how easy or difficult it is for the threat agent to carry them out) and their impact.

To estimate the execution ease of each MC, the risk assessment expert first forms a list of possible threat agents, a list of threats, a list of vulnerabilities, a list of information assets and a list of IT assets. For making these lists she refers to her previous risk assessment experiences, available threat and vulnerability databases, and the documents delivered by the IT manager (e.g. IT-architectural drawings and system specifications). Second, she and the security officer (based on these lists) specify the vulnerabilities, the threats and the threat agents that may execute them for the target of assessment, and estimate how incidents might propagate. In RiskREP, we model how an incident may propagate through the IT architecture using *Incident Propagation Paths* (IPPs). The risk expert draws a number of IPPs based on a structured thinking process: she first draws the assets representing the entry points of the system. Then she gradually connects further assets by considering (a) physical and logical connections among the assets and (b) vulnerability-threat pairs associated with the destination component. We consider an IPP to be complete when the asset that the MC refers to is reached. Finally, the risk expert (based on the IPPs) makes an estimation of how easy it is for each threat agent to accomplish the IPPs. We call this the *execution ease* of an IPP. If there are several IPPs realizing the same MC, then we select the easiest path as the path that the attacker follows to execute the MC.

Once the execution ease of the MCs are estimated, the risk expert assesses the *value* of each QG by following *value models*, like the TD model [76] for availability and the DCRA model [4] for confidentiality. These values depend on the related BGs and the degree in which each QG contributes to the satisfaction of a BG. These values are the basis for estimating the *impact* or damage caused by the MC respectively IPP to these QGs.

6.5.3 Step 3: Defining countermeasures

In this step we define a list of countermeasures for each MC. The stakeholders involved here are the security officer, who knows the effects of countermeasures on threat and vulnerability pairs, and the RE expert.

We first compose a set of *countermeasures* by taking them from existing checklists. These checklists are part of RiskREP and contain general countermeasures for 167 threat/vulnerability pairs. In this step of RiskREP, we bring these general measures to a concrete, realizable level by specifying which component each of them applies to and how. We determine which countermeasure can mitigate, prevent, or detect which MCs (and to what level) by referring to the threats and vulnerabilities of each MCs. There are n-m-relationships among MCs and countermeasures, which are best presented in a table. Finally we quantify the cost of each countermeasure.

6.5.4 Step 4: Prioritizing countermeasures

In this step, we prioritize the MCs and the countermeasures. We prioritize the MCs based on their *risk*, whereas we prioritize countermeasures based on their *added value*, i.e. effectiveness and cost. Since a countermeasure's added value is created by reducing MC risk, we approximate the value it adds based on the execution ease reduction and the impact reduction it achieves and comparing these to the additional costs it causes. The risk reduction (effectiveness) is estimated by imagining the system with and without the countermeasure applied and without.

Ease and impact, as well as effectiveness and cost are incomparable entities. Thus, we do not add, multiply or subtract them from each other (as other authors do). Instead, we say that the risk of an MC mc_i is *superior* to the risk of another MC mc_j if both execution ease and impact of mc_i are superior to the execution ease and impact of mc_j. We also say that the added value of a countermeasure c_i is superior to the added value of another countermeasure c_j if the risk reduction accomplished by c_i is higher than the risk reduction accomplished by c_j and/or the cost of c_i is lower than the cost of c_j. When the execution ease of mc_i and the impact of mc_j are both superior (or vice versa), then we consult the stakeholders to determine the superior MC. A similar reasoning applies to the selection of countermeasures.

By applying countermeasures to MCs, we reduce the risk. However, applying countermeasures usually means increased cost. Therefore, RiskREP aims at finding the ideal set of countermeasures to be applied in addition to the countermeasures implemented in the current system. The best set of countermeasures is the set of not yet implemented countermeasures with minimum total cost and maximum risk reduction. These values can be optimized by considering several sets of countermeasures. In practice, the security budget of the system is often the main delimiter for the ideal set of countermeasures.

Countermeasures interact with each other. For instance, some may be overlapping, or diminish each other's effectiveness. Therefore, RiskREP also needs to take into

consideration the results of these interactions to identify the ideal set of countermeasures to be implemented. We call the effectiveness of a set of countermeasures when applied together the *combined effect* of that set of countermeasures. To determine the combined effect of two countermeasures, we interview the security officer. We furthermore address the combined effects of more than two countermeasures by flattening them into pairs of countermeasures. That is, assuming that we have three countermeasures c_1, c_2 and c_3, we argue that the combined effect of applying c_1, c_2 and c_3 together equals to adding the combined effect of c_1 and c_2 with the combined effects of c_2 and c_3, and of c_1 and c_3.

6.6 Case study description

This section, describes the case study we use to validate the feasibility of RiskREP. The target system is the student administration portal developed at the University Braunschweig (TU). The project team's motivation for participating in our security analysis case study was to learn more about the risk level of the system and get ideas for potential improvements.

The TU offers to its students and teaching personnel various online services, such as e-learning platforms, registration for exams, downloading of documents, and an email account. The portal *TUgether* integrates all these services, and allows the students to sign on via one individually configurable interface. The portal itself does not offer any content but is just an entry point to data. On the other hand, the content remains accessible also without the portal. The aim of this project is to offer as much added value to the users as possible within the project budget. Thus, development effort and cost had to be optimized. One major objective is that all students should eventually use the portal.

The first phase of the TUgether project was meant to select the portal framework product which satisfies requirements best. To choose this framework, more than 80 (functional and non-functional) requirements were specified and about 70 products were taken into consideration. Out of the 80 requirements, 9 were security-related, including "privacy", but also technical means such as "backup possibility". At the time of this case study, the portal was in pilot operation.

We want to stress that the scope of this case study is limited to security (confidentiality, integrity and availability) requirements of student information that is managed by or accessed via the portal. The case study example is a real software project, but not too complex. This allows us to apply the RiskREP method as a proof-of-concept in a real project.

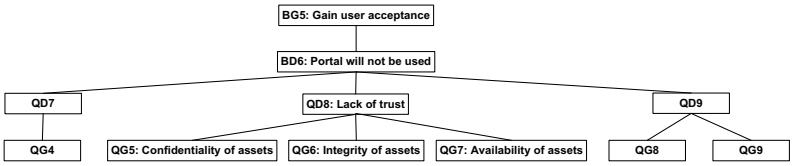

Figure 6.2: Top-down way of eliciting business perspective concepts of RiskREP

6.7 The case: Applying RiskREP

Here we illustrate how we executed the steps of RiskREP in the case described in Section 6.6

6.7.1 Step 1: Finding quality goals

The business goals had been defined by the management level, even before we started our case study. In particular, we could deduce them from a project report. Figure 6.2 plots the connections between the security-related business perspective concepts of the TUgether portal.

In the project report there is only one security-related business goal *BG5: Gain user acceptance*. Starting with this business goal we constructed the goal and damage graph by connecting it to the business damages that threaten it. The BG5 is related to one business damage, i.e. *BD6: Portal will not be used*. Then, the RE expert identified three quality deficiencies that may cause BD6, i.e. *QD7: User unfriendliness*, *QD8: Lack of trust*, and *QD9: Lack of added value*. Because of the scope of our case study we analyzed only QD8 further. Lack of quality attributes confidentiality, integrity and availability may lead to QD8. Accordingly, the expert derived three high-level quality goals, i.e. *QG5: Confidentiality of assets*, *QG6: Integrity of assets*, and *QG7: Availability of assets*.

6.7.2 Step 2: Analyzing security risks

One of the MCs that threatens QG6 is *MC5: Manipulation of account data*. For this case, the risk expert agreed with the security officer to use five threat agents, i.e. *user, hacker, portal admin, portal developer* and *service developer*. Then, the risk expert and the IT manager drew the IT architecture of the system (see Figure 6.3) and listed critical information assets. The IT assets of the TUgether portal related to MC5 are *TUgether portal server, LDAP server* and *Development server*. Finally, the IT manager estimated (1) the impact of each MC based on the information assets that might be disclosed by it, and (2) the execution ease of each MC based on the most likely threat/vulnerability pair that applies to this case. Here we use a scale from 1 (low) to 3 (high) to quantify execution ease and impact. For instance, the execution ease of MC5 is *1.5* and its impact is *1*. We have discussed IPPs when specifying the MCs, but since in this (not too

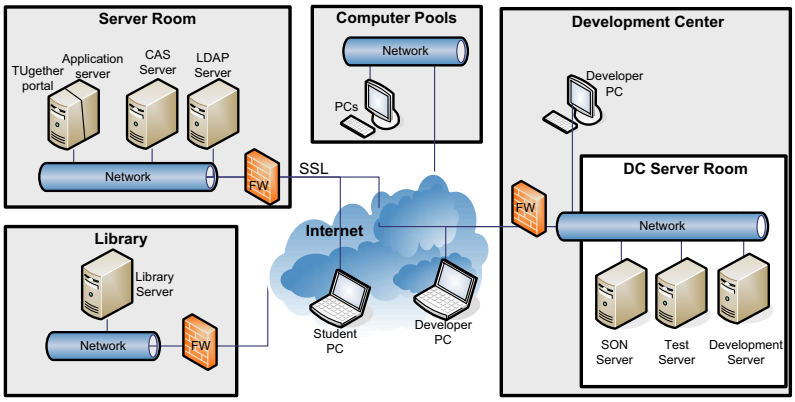

Figure 6.3: IT architecture of the student portal (FW: Firewall, DC: Data Center, CAS: Central Authentication Service, and SON: Personal Development Server)

Table 6.3: Some MCs and their attributes

MC ID	**risk (ease,impact)**	**Threat agent**	**Threat**	**Vulnerability**
MC5: manipulation of account data	(1.5,1)	hacker	data get lost or are manipulated during transfer	Portal does not manage data and therefore data synchronization between portal and services is necessary
MC9: no logout in computer pool	(1,3)	user	does not log out after having used the portal on a computer in the public computer pool	no access control to computer pools

complex) case IPPs are self-evident, we did not specify them explicitly. We estimated the execution ease of each MC intuitively on a scale from 1 (low) to 3 (high). This estimation demanded knowledge about technical architecture and context of use. In total we identified ten MCs realted to QG6, two of which are presented in Table 6.3. The table shows for each MC the threat agent, vulnerability and threat combination, as well as its risk.

Table 6.4: Some countermeasures for mitigating MCs

	Cost	**Misuse Cases**				
		MC 1	MC 2	...	MC 9	MC 10
C1: standardized interfaces (LDAP, CMS,...)	2	mitigates				
C2: timeout and login of user	1				partially mitigates	
C10: security measures taken by the included services	0				partially mitigates	

6.7.3 Step 3: Defining countermeasures

Table 6.4 reports the results of this step for our case. Here, we quantified the cost of each countermeasure on a scale of 0 to 3 points, where 0 stands for no cost, 1 for the cost of changing the settings of applications, 2 for the cost of installing and maintaining freely available countermeasures and 3 the cost of for purchasing, installing and maintaining countermeasures.

6.7.4 Step 4: Prioritizing countermeasures

In the case study we used the simplest scales for cost, execution ease and impact, i.e. -1, 0, or +1. This way it is easy to estimate and less prone to mistakes. If necessary, RiskREP allows using more sophisticated scales. We furthermore defined a countermeasure's effectiveness as follows: if a countermeasure neither affects the impact nor the execution ease of an MC, then its effectiveness is 0; if it decreases either impact or execution ease, then it's effectiveness is 1; if it decreases both, it is 2. We present the interactions among the countermeasures (combined effects) that the security officer estimated for TUgether with a two-dimensional matrix (see Table 6.5). The matrix is sparse and not symmetric, because it is possible that countermeasure c_1 influences c_2, but not vice versa. In this case study it contains 10 interactions, whereas among the 10 countermeasures 90 different interactions would be theoretically possible. To determine which countermeasures should be implemented first, i.e. to prioritize them, we applied a heuristic approach using categories of MC risks and countermeasures added values.

When prioritizing the MCs according to their risk, i.e. execution ease and impact, we want to distinguish between those which have low execution ease while causing high damage and vice versa. Therefore, we used the following categories:

- ignore: execution ease and impact are low;

Table 6.5: Combined effects of countermeasures

Countermeasure	C1	C2	...	C10
C1			...	
C2			...	
...	...	...	...	...
C10	- - 0		...	

- rare, but detrimental: execution ease is low, but impact is high;

- frequent, but harmless: execution ease is high, but impact is low;

- catastrophic: both are high, or one is average and the other high; and

- average: both are average, or one is average and the other low.

We classified countermeasures according to their cost and effectiveness as follows (see Table 6.6):

- no effect: both execution ease and impact are not modified

- contra-effective: both execution ease and impact are increased, or one is increased and the other one is not modified;

- counter-effective: execution ease is increased as impact is reduced or vice versa;

- low-hanging fruit: cost is 0 and either only execution ease or impact is reduced, or both are reduced;

- cost-efficient: cost is 1 and either only execution ease or impact is reduced, or both are reduced;

- cost-effective: cost is 2 and both execution ease and impact are reduced;

- expensive: cost is 2 or more and either only execution ease or impact is reduced;

- expensive effectiveness: cost is 3 and both execution ease and impact are reduced;

To choose the optimal set of countermeasures, we did not use a formula that optimizes the systems added value automatically, but rather decided for a countermeasure selection strategy together with the stakeholders. In this case the strategy is to improve countermeasure effectiveness and cost. Accordingly we suggested the stakeholder to implement all "low hanging fruit" countermeasures. Furthermore, since defining the categories also influences the strategy, we asked for stakeholders' approval after defining them. This way of choosing the countermeasures to be implemented is a heuristical one which allows to make decisions transparently and based on objective criteria, but is still simple and easy to execute.

Table 6.6: Categories of countermeasure effects

ease	impact	cost	category
+1	+1	0	contra-effective
+1	+1	1	contra-effective
+1	+1	2	contra-effective
+1	+1	3	contra-effective
+1	0	0	contra-effective
+1	0	1	contra-effective
+1	0	2	contra-effective
+1	0	3	contra-effective
+1	-1	0	counter-effective
+1	-1	1	counter-effective
+1	-1	2	counter-effective
+1	-1	3	counter-effective
0	+1	0	contra-effective
0	+1	1	contra-effective
0	+1	2	contra-effective
0	+1	3	contra-effective
0	0	0	no-effect
0	0	1	no-effect
0	0	2	no-effect
0	0	3	no-effect
0	-1	0	low hanging fruit
0	-1	1	cost-efficient
0	-1	2	expensive
0	-1	3	expensive
-1	+1	0	counter-effective
-1	+1	1	counter-effective
-1	+1	2	counter-effective
-1	+1	3	counter-effective
-1	0	0	low hanging fruit
-1	0	1	cost-efficient
-1	0	2	expensive
-1	0	3	expensive
-1	-1	0	low hanging fruit
-1	-1	1	cost-efficient
-1	-1	2	cost-efficient
-1	-1	3	expensive effectiveness

6.8 Validation

The case study presented in the previous section shows that RiskREP provides a systematic method for analyzing the security of an information system and for determining which countermeasures should be put in place.

To apply RiskREP, one needs to have knowledge of the IT architecture of the system under consideration, and of the functionalities it supports, e.g. permitted actions of users, administrators and developers and how data are exchanged between the system components. RiskREP uses this knowledge to infer where data can be lost, manipulated or disclosed to unauthorized persons.

It took us four hours for jointly analyzing and prioritizing the QG6-related requirements of the TUgether platform (integrity of assets). Considering the large amount of information gathered during this time, we consider RiskREP to be an efficient and effective method.

In this case study, we could not observe whether RiskREP indeed helps to separate communication with the business owner, the IT manager and the security officer, because our contact person actually covers all these roles. We could find most of the information needed for step 1 in the project report. This report had been written (from a management perspective) by the project management team. This confirms that step 1 in fact models the information which is relevant for management.

We found that RiskREP helps to structure the discussion. The templates and checklists helped not to forget anything important. Our contact person said that the scenarios were very helpful for the analysis, and that the analysis gave them new ideas, whereas all the results of their former discussions were also found by the RiskREP analysis.

Concluding, we should mention that the case study was supported by simple tools: (1) drawing tools for the tree graphic produced in step 1 and for presenting the system architecture, (2) several spreadsheet tables for the qualitative and quantitative analysis of MCs and countermeasures. These tables also support the testing of different sets of countermeasures.

6.9 Concluding remarks

This chapter presents RiskREP, a new method for the systematic elicitation and prioritization of security (quality) requirements. It has been constructed by integrating the methods MOQARE and CRAC++. We have applied RiskREP to a web portal in order to assess the portals' security and to identify potential improvement measures.

RiskREP contains all of the following features:

- systematic processing;
- differentiation between business and quality goals;
- it considers both intentional use and misuse;
- it considers different stakeholder views;
- systematic estimation of asset value and incident likelihood;
- requirements prioritization based on costs; and
- it considers requirements' effectiveness as well as the effects of combining requirements.

These features showed to have positive effects on the analysis. We believe that the strengths of RiskREP include: step-by-step guidance of the analysis; checklists of threats; time-efficient analysis; and transparent prioritization of security requirements.

Security requirements can be used to derive test cases for security analysis and compliance monitoring. RiskREPs countermeasures describe what the system will do

and therefore can be used as test criteria. The MCs and IPPs describe misuse scenarios from the user perspective or the technical perspective respectively. These scenarios end in a system misuse and some sort of damage when a threat is executed. When the countermeasures are effective, they prevent this damage or reduce its execution ease or the damage caused. Consequently, the MCs can be used as test cases for security-related black box tests and the IPPs as test cases for white box tests. Measuring execution ease and impact is also important in order to verify whether the implemented countermeasure has the expected effect. The test cases priorities are related to the risks of the corresponding MC or IPP: the higher its risk, the more important it is to test a scenario. In future work, we want to derive security test cases and monitoring criteria from MCs and IPPs, in order to see how easy and straightforward this can be done and whether these test cases make sense for security testing and monitoring.

Chapter 7

Conclusion and Future Work

We now summarize the contributions of this dissertation in relation to the research aims and questions discussed in Chapter 1. We furthermore highlight open issues and future research directions in the area of model-based risk assessment.

7.1 Reviewing the research question

We claim that in many organizations confidentiality risks are not assessed and evaluated accurately. This is due to three reasons. First, accurate risk assessment techniques rely on event histories, which are (especially for confidentiality) most of the time not available. Second, assessing confidentiality risks with currently available model-based methods is too resource-demanding to be adopted by many organizations. As an alternative to assessing risks with a model-based method a checklist-based method is used. Although for uncomplicated and well-known IT systems checklist-based methods assess risks relatively more cost-efficiently than model-based methods, for complex systems the checklist-based methods do not deliver accurate enough results in a cost-efficient way. Moreover, to form a checklist one has to conduct a number of model-based assessments on similar systems and extract security patterns. This is a very costly process. Last, checklist-based methods are error-prone, because they assess risks mainly according to subjective opinions of the risk assessors.

Based on these observations, in Chapter 1 we formulated the following research aim:

> *To develop an IT confidentiality risk assessment and evaluation method that is:*
>
> - *accurate enough for business applications;*
> - *cost-efficient for business-criticality of the IT system; and*
> - *more inter-subjective than present methods.*

Our contributions are the partial solutions we provide to this research aim. We answer the research goals (presented in Chapter 1) as follows:

G1. How can we assess confidentiality risks of complex IT systems accurately? We address this goal first by extending eTVRA, a model-based risk assessment method, in Chapter 2. Extended eTVRA contributes to cost-efficiently eliciting and structuring IT security-relevant information by applying safety analysis methods, i.e. it uses SWOT analysis to set the scope, HAZOP and functional architecture walkthrough for structuring the information elicitation sessions and TVA for structuring the findings of these semi-structured interviews. Then, we compare extended eTVRA with a checklist-based method with respect to accuracy and cost-efficiency. We came to the following knowledge contribution: with respect to assessing security risks of complex IT systems model-based methods produce more accurate results, are more generally applicable and more cost-efficient than checklist-based methods.

Second, we introduce DCRA in Chapter 3, a model-based confidentiality RA method. DCRA determines how robust an IT system is with respect to confidentiality. It models the IT system based on the IT architecture, which allows a risk assessor to determine how confidentiality incidents may propagate through the system components and what the potential global impact of each incident is. DCRA contributes to the accurate analysis and assessment of confidentiality risks based on the IT architecture.

G2. How can we assess confidentiality risks of IT systems cost-efficiently? To address this goal we first introduce CRAC in Chapter 4. Like the DCRA method, CRAC is based on the IT architecture, but in addition it also elicits necessary input information, and allows to assess and compare risks of distributed IT systems. CRAC operates with ordinal scale values and models (1) information flow (IFP) for eliciting the business value of IT components and (2) attack propagation (APP) for determining the difficulty of different attackers accessing information assets available on these IT components. IFPs and APPs contribute to the goal of cost-efficiently eliciting IT security-relevant information on complex and incompletely documented IT systems.

Second, we introduce RiskREP in Chapter 6, which describes stepwise how to identify quality goals (confidentiality requirements) from business goals and how to link them to the IT security risks. It furthermore delivers the best set of requirements (security controls) based on the risks each requirement encounters, the costs it introduces and the business goals it contributes to. It contributes to the cost-efficiency of IT security risk evaluation by allowing an optimal distribution of the security budget over alternative countermeasures that can be applied to cope with the identified risks.

G3. How can we assess and control confidentiality risks of outsourced IT systems inter-subjectively? To increase the inter-subjectivity of assessment results we present CRAC in Chapter 4. CRAC extracts the necessary information from the available IT-architectural and functional documents, and formalizes the risk assessment activities. Formalization of the risk assessment activities contributes to the inter-subjectivity of the results. It furthermore models the system based on the IT architecture it relies on. Since the alternative IT systems have different IT architectures, the results that CRAC delivers are comparable based on the components that compose each IT architecture.

Furthermore, to control confidentiality risks of outsourced IT systems we present CRAC++ in Chapter 5. CRAC++ extends CRAC for specifying the confidentiality requirements that need to be included in an outsourcing contract. CRAC++ assesses the confidentiality risks of an outsourced IT system based on confidentiality requirements of the outsourcer and the outsourcee, and compares the assessment results based on the IT architecture. Finally it specifies those requirements that cause the confidentiality risk to exceed the outsourcer's risk threshold. CRAC++ contributes to the confidentiality risk assessment by providing a mechanism for controlling the confidentiality risks that arise due to the outsourcing of IT systems.

Based on the findings of our work we state that we achieve the research aim successfully. Now we present and discuss the main contributions of the dissertation that lead us to this statement as follows.

C1: in the absence of explicit risk knowledge tacit knowledge can be practically extracted with techniques from the safety domain as well as by functional architecture walkthroughs.

C2: the IT architecture of a system can be used as a basis for cost-efficient yet accurate confidentiality risk assessment.

C3: alternative IT systems can be compared by predefining and agreeing on ordinal risk scales and assessing risks based on the IT architecture.

C4: organizations can control confidentiality risks of outsourced IT systems by assessing the risks that may arise due to differences between security requirements of business partners and extending service contracts with these requirements.

C5: organizations can achieve a business-optimal distribution of their security budget by combining knowledge from requirements engineering and risk management.

These contributions and the research method used to deliver them have been progressively made public in several publications as presented in Table 7.1.

7.2 Limitations and future work

Our contributions open several possible future research directions.

Validating risk indicators In practice it is common to validate risk scales by asking stakeholders whether they make sense. This way of validation is subjective, thus error prone. If the metrics are wrong, then they may lead system owners to distribute security investments inefficiently. Therefore, when validating risk assessment methods it is important to validate whether the risk indicators reflect the correct value of risk. Measurement theory (as used in different domains such as physics and software development) can be used as a tool for objective validation of risk indicators.

Table 7.1: Contributions, validation and publications

Contribution	Description	Validation	Publication
C1	information elicitation	action research and lab demo	[2, 3, 7]
C2	IT architecture-based risk assessment	action research and lab demo	[1, 2, 4, 7]
C3	alternative comparison	action research	[2, 7]
C4	outsourcing SLAs	action research	[1, 2, 6, 7]
C5	linking business goals	action research	[5]

Collaborative attack analysis According to the Verizon Data Breach Investigation Report 2010 [79] 27% of all confidentiality breaches are executed by multiple attackers. However, currently available RA methods model attacks from a single attacker's perspective. It is necessary to extend the attack models so that they can also model collaborative attacks.

Log analysis Incident and event logs may provide quantitative incident data for less subjective risk assessment. Although most of the time these logs are available, they are rarely analyzed. This is due to the difficulty of discovering breaches in the logs, especially in the case of stolen credentials. However, if attacks can be translated into the changes they would cause on logs, then logs can be analyzed by looking for those changes. For this it is necessary to analyze these changes and provide computational tool support for conducting the analysis.

Appendix A

Application of the CRAC Method to the Invoicing Service.

In this appendix we indicate how CRAC can be applied to a larger, real case. The target of this assessment is the invoicing service described in Chapter 3.

Table A.1 reports the information assets we identified, together with their confidentiality levels, which are based on the estimated (monetary) loss the business would suffer from if they were disclosed to unauthorized people. The most important information assets are customer call records, raw call records, phone contract information and phone line information which have to be kept confidential because of laws and liability issues. The disclosure of the employee mail has a lower but still significant impact, whereas the disclosure of the other information assets do not to lead to an immediate loss.

Table A.1: Information assets in the scope

asset	confidentiality level	homogeneity property
raw call records	high	homogenious
user call records	high	homogenious
phone contract info	high	homogenious
phone line info	high	homogenious
test data sets	low	nonhomogenious
application design specifications	low	homogenious
software test documentation	low	nonhomogenious
encryption keys	low	homogenios
employee mail	medium	homogenious

Figure A.1 shows the architecture graph of the invoicing service. Different types of dashed edges indicate the information flow paths from an information source to a physically or logically connected component. (To visualize the volume of information flow, we annotate these edges with the maximum percentage of instances that can be retrieved.) For instance, the operational traffic Oracle is the information source for the user call records. 5% of the user call records can flow to the post processing application and 20% to the invoicing application. Furthermore, all instances of the user call records can be retrieved from the traffic viewer application. The solid edges indicate the attack paths. For instance, we estimate that it is *very likely* that an attacker who has access to the traffic viewer server will also be able to access the traffic viewer application server. However it is *unlikely* for him to be able to disclose the user call records stored on the operational traffic Oracle. The telecom provider did not want to differentiate between possible attackers. Therefore for this case we illustrate all possible attack paths over the architecture graph.

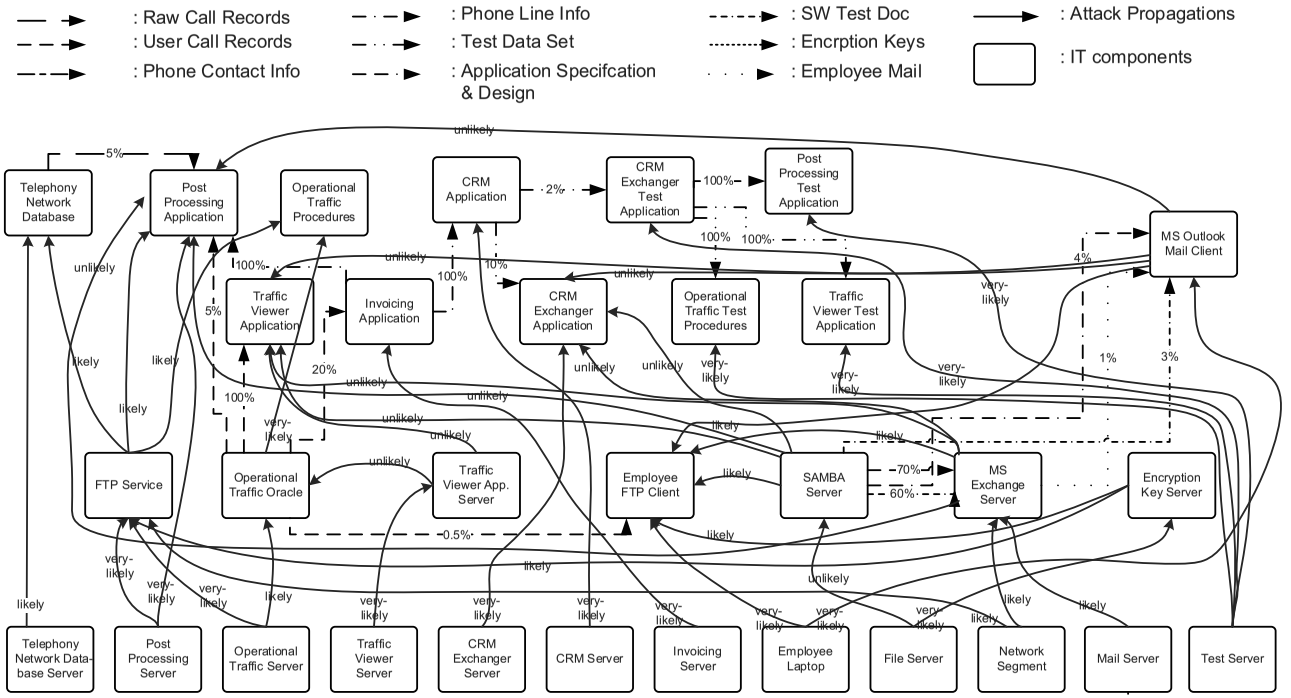

Figure A.1: Architecture graph of the invoicing process

Table A.2 reports the total impact and reachability level of those components containing various information assets. For instance, the Post Processing Application contains 5% of the Raw Call Records, 5% of the User Call Records, and 100% of the Phone Line Info. By "aggregating" the impacts of these information assets we estimate the total impact of the Post Processing Application as *high*. Furthermore, we assessed five APPs that lead to the Post Processing Application.

Table A.2: Total impact and reachability levels

component	total impact	reachability
post processing application	high	very-likely
CRM application	high	very-likely
CRM exchanger application	high	very-likely
invoicing application	high	very-likely
telephony network DB	high	likely
operational traffic Oracle	high	likely
traffic viewer application	high	likely
CRM Exchanger test application	medium	very-likely
MS Exchange server	medium	likely
employee FTP client	low	very-likely
post processing test application	low	very-likely
MS outlook mail client	low	very-likely
traffic viewer test application	low	very-likely
encryption key server	low	very-likely
operational traffic test procedures	low	likely
SAMBA server	low	unlikely

Appendix B

CRAC and CRAC++ Evaluation Metrics

In the following we describe how we calculated the metrics used for evaluating the CRAC (Chapter 4) and CRAC++ (Chapter 5) methods.

We analyze the concepts of the CRAC method. Table B.1 shows which concepts of the CRAC method are optional (M2), adjustable (M3) and inter-subjective (M4). Table B.2 and Table B.3 do the same for respectively the CRAMM and checklist-based methods.

Table B.1: Concepts of the CRAC method with respect to M2, M3 and M4

Concepts	**M2**	**M3**	**M4**
Information asset		X	X
Confidentiality level		X	X
Homogeneity	X		X
IT component		X	X
Vulnerability		X	
Threat agent		X	X
Number of instances that can be retrieved	X	X	X
Impact			X
Total impact			X
Attack propagation graph			
Attack path			X
Attack propagation likelihood (1)			X
Attack propagation likelihood (2)			X
Attack propagation likelihood (3)			X
Competencies and conditions	X	X	
Risk			X
Mitigation level		X	X

Table B.2: Concepts of the CRAMM method with respect to M2, M3 and M4

Concepts	**M2**	**M3**	**M4**
Assets			X
Asset value		X	X
Threats		X	X
Extent of vulnerabilities		X	
Risk		X	X
Level of threats			
Countermeasures			X
Applications			X
Nr. of persons using the application			X
Locations			X
Multi functional assets			X
Quantity of physical assets			X
Class of physical assets			X
Class of software assets			X
Links between assets			X
Asset model			X
Potential impact scenario		X	
Threat source			X
Financial value			X
Scale value		X	
Valuation scenario		X	
Likelihood			X
Asset value		X	X
Threat and vulnerability questions	X	X	
Measures			

Table B.3: Concepts of the checklist-based method with respect to M2, M3 and M4

Concepts	**M2**	**M3**	**M4**
Threat type		X	X
Business impact	X	X	
Vulnerabilities		X	
Sensitivity period	X	X	X
Final business impact level			X
Impact	X		
Data type	X		X
Percentage of data	X		X
User type	X		X
Percentage of users	X		X
Interfaces			X
Threats			
Final residual risk			X
Severity			
Measures			X
Mitigation level			

Finally, Table B.4 reports the concepts of CRAC++ and the corresponding concepts of the checklist-based method used by the Company in Section 5.4. Bold letters in the table indicate that the concept is either objectively determined or based on an information set that is accepted by all stakeholders and the risk assessors for all related assessments. An empty cell indicates that the method does not have a corresponding concept.

Table B.4: The concepts of CRAC++ and the checklist-based method used by the Company

Concepts of CRAC++	**Concepts of Checklist**
Confidentiality requirements	**Measures**
Confidentiality threshold	
Information assets	**Data**
Confidentiality value	**Data type**
Homogeneity	
IT component	**Interfaces**
Information flow graph	
Number of instances	**Percentage of data**
Impact	Impact
Total impact	Business impact
Critical components	
Vulnerabilities	Vulnerabilities & Threats
Threat agents	**User type**
Competencies	
Attack propagation graph	
Ease of exploiting vulnerabilities	
Effectiveness	Mitigation level
Ease of accessing a component	
Bottleneck	
Protection level	Severity
	Sensitivity period
	Final Business impact level
	Percentage of users
	Threat type

Author References

Refereed Conferences

[1] A. Morali and R.J. Wieringa. Risk-based confidentiality requirements specification for outsourced it systems. In *Proceedings of the 18th IEEE International Requirements Engineering Conference (RE'10)*, pages 199–208. IEEE Computer Society, 2010.

[2] A. Morali, E. Zambon, S. Etalle, and R.J. Wieringa. CRAC: Confidentiality Risk Analysis and IT-Architecture Comparison. In *Proceedings of the 6th IEEE/IFIP International Conference on Network and Service Management (CNSM'10)*, pages 322–325. IEEE Computer Society Press, 2010.

[3] A. Morali, E. Zambon, S.H. Houmb, K. Sallhammar, and S. Etalle. Extended eTVRA vs. security checklist: Experiences in a value-web. In *31st International Conference on Software Engineering (ICSE'09) Companion Volume*, pages 130–140. IEEE, 2009.

International Workshops

[4] A. Morali, E. Zambon, S. Etalle, and P. Overbeek. It confidentiality risk assessment for an architecture-based approach. In *Third IEEE International Workshop on Business-Driven IT Management (BDIM 2008)*, pages 31–40. IEEE Computer Society Press, 2008.

CTIT Technical Report

[5] A. Morali, A. Herrmann, and S. Etalle. RiskREP: Risk-Based Security Requirements Elicitation and Prioritization. Technical Report (submitted to RE'10) TR-CTIT-10-28, Centre for Telematics and Information Technology, University of Twente, 2010.

[6] A. Morali and R.J. Wieringa. Risk-based confidentiality requirements specification for outsourced it systems (extended version). Technical Report TR-CTIT-10-09, Centre for Telematics and Information Technology, University of Twente, 2010.

[7] A. Morali, E. Zambon, S. Etalle, and R.J. Wieringa. CRAC: Confidentiality Risk Analysis and IT-Architecture Comparison (extended version). Technical Report TR-CTIT-09-30, Centre for Telematics and Information Technology, University of Twente, 2009.

General References

[8] FIPS PUB 199. Standards for Security Categorization of Federal Information and Information Systems. Technical report, NIST - National Institute of Standards and Technology, 2004.

[9] R. Anderson and M. Kuhn. Low cost attacks on tamper resistant devices. *Security Protocols*, 1361/1998:125–136, 1998.

[10] Y. Asnar and P. Giorgini. Modelling risk and identifying countermeasures in organizations. In *CRITIS'06: Proc. of 1st Int. Workshop on Critical Information Infrastructures Security*, pages 55–66. Springer, 2006.

[11] R.L. Baskerville. Distinguishing action research from participative case studies. *Journal of Systems and Information Technology*, 1(1):25–45, March 1997.

[12] E. Becher, Z. Benenson, and M. Dornseif. Tampering with motes: Real-world physical attacks on wireless sensor networks. In *Proceeding of the 3rd International Conference on Security in Pervasive Computing*, pages 104–118, 2006.

[13] T. Berners-Lee, R. Cailliau, A. Luotenen, H.F. Nielsen, and A. Secret. The world-wide web. *Communications of the ACM*, 37(8):76–82, 1994.

[14] F. Braber, I. Hogganvik, M.S. Lund, K. Stølen, and F. Vraalsen. Model-based security analysis in seven steps — a guided tour to the coras method. *BT Technology Journal*, 25:101–117, January 2007.

[15] R. Breu, F. Innerhofer-Oberperfler, and A. Yautsiukhin. Quantitative assessment of enterprise security system. In *Third International Conference on Availability, Reliability and Security (ARES 2008)*, pages 921–928. IEEE Computer Society, 2008.

[16] A. Dorofee C. Alberts. *Managing Information Security Risks: The OCTAVE (SM) Approach*. CMU, 2003.

[17] Chemical Industry Safety and Health Council. *A Guide to Hazard and Operability Studies*. Chemical Industries Association (1977), 1992.

[18] H. Chivers, J.A. Clark, and P.-C. Cheng. Risk profiles and distributed risk assessment. *Computers & Security*, 28(7):521 – 535, 2009.

[19] M.F. Chudleigh and J.R. Catmur. Safety Assessment of Computer Systems Using HAZOP and Audit Techniques. In *Proceedings of Safety of Computer Control Systems (SAFECOMP'92)*, pages 285–292. Pergamon Press, 1992.

[20] Z. Ciechanowicz. Risk analysis: requirements, conflicts and problems. *Computers & Security*, 16(3):223–232, 1997.

[21] R. Cocchiara. Beyond disaster recovery: becoming a resilient business: An object-oriented framework and methodology. Technical Report G510-6482-01, IBM, 2007. http://ibm.com/services/its/resilience.

[22] A. Dardenne, A. van Lamsweerde, and S. Fickas. Goal-directed requirements acquisition. *Science, Computing and Programming*, 20(1-2):3–50, 1993.

[23] E. Dubois, P. Heymans, N. Mayer, and R. Matulevicius. A systematic approach to define the domain of information system security risk management. In S. Nurcan et al., editor, *Intentional Perspectives on Information Systems Engineering*, pages 289 – 306. Springer-Verlag, 2010.

[24] K. Eagles, K. Markantonakis, and K. Mayes. A Comparative Analysis of Common Threats, Vulnerabilities, Attacks and Countermeasures Within Smart Card and Wireless Sensor Network Node Technologies. *Information Security Theory and Practices. Smart Cards, Mobile and Ubiquitous Computing Systems*, pages 161–174, 2007.

[25] G. Elahi and E. Yu. Modeling and analysis of security trade-offs - A goal oriented approach. *Data Knowledge Engineering*, 68:579–598, 2009.

[26] G. Elahi, E. Yu, and N. Zannone. A vulnerability-centric requirements engineering framework: analyzing security attacks, countermeasures, and requirements based on vulnerabilities. *Requirements Engineering*, 15(1):41–62, 2010.

[27] B. Fabian, S. Gürses, M. Heisel, T. Santen, and H. Schmidt. A comparison of security requirements engineering methods. *Requirements Engineering*, 15:7–40, 2010.

[28] L. Favre, editor. *UML and the Unified Process*. IGI Publishing, Hershey, PA, USA, 2003.

[29] C. Forde, J. Varnus, L. Fehskens, A. Josey, G. Doherty, and C. Fox. *TOGAF Version 9: Enterprise Edition*. The Open Group, 2009.

[30] S. Gritzalis, A. Yannacopoulos, C. Lambrinoudakis, P. Hatzopoulos, and S.K. Katsikas. A probabilistic model for optimal insurance contracts against security risks and privacy violation in it outsourcing environments. *International Journal of Information Security*, 6(4):197–211, 2007.

[31] L. Grunske and D. Joyce. Quantitative risk-based security prediction for component-based systems with explicitly modeled attack profiles. *Journal of Systems and Software*, 81(8):1327–1345, 2008.

[32] B. Haley, C. Laney, D. Moffett, and B. Nuseibeh. Using trust assumptions with security requirements. *Requirements Engineering*, 11(2):138–151, 2006.

[33] C. Haley, R. Laney, J. Moffett, and B. Nuseibeh. Security requirements engineering: A framework for representation and analysis. *IEEE Trans. Softw. Eng.*, 34(1):133–153, 2008.

[34] A. Herrmann and B. Paech. MOQARE: misuse-oriented quality requirements engineering. *Requirements Engineering*, 13(1):73–86, 2008.

[35] C.D. Huang, R.S. Behara, and Q. Hu. Managing Risk Propagation in Extended Enterprise Networks. *IT Professional*, 10(4):14–19, 2008.

[36] C.D. Huang and J. Goo. Rescuing IT Outsourcing: Strategic Use of Service-Level Agreements. *IT Professional*, 11(1):50–58, 2009.

[37] T.R. Ingoldsby. Understanding risk through attack tree analysis. *Computer Security Journal*, 20(2):33–59, 2004.

[38] F. Innerhofer-Oberperfler and R. Breu. Using an enterprise architecture for it risk management. In *Proceedings of the 5th Information Security South Africa Conference (ISSA'06)*, page 12, 2006.

[39] S. Islam and S.H. Houmb. Integrating risk management activities into requirements engineering. In *Proceedings of the 4th International Conference on Research Challenges in Information Science (RCIS'10)*, pages 299–310. IEEE Computer Society, 2010.

[40] J. Jürjens. *Secure Systems Development with UML*. Springer Academic Publishers, 2004.

[41] Y. Karabulut, F. Kerschbaum, F. Massacci, P. Robinson, and A. Yautsiukhin. Security and trust in it business outsourcing: a manifesto. *Electronic Notes in Theoretical Computer Science*, 179:47 – 58, 2007. Proc. of the 2nd Int. Workshop on Security and Trust Management (STM 2006).

[42] R. Kazman, M. Klein, P. Clements, and N.L. Compton. Atam: Method for architecture evaluation. Technical Report CMU/SEI-2000-TR-004, CMU Software Engineering Institute, 2000.

[43] M. Krause and H.F. Tipton. *Handbook of Information Security Management*. CRC Press LLC, Auerbach Publishers Inc., 1998.

[44] A. Lenstra and T. Voss. Information Security Risk Assessment, Aggregation, and Mitigation. In *Proceedings of the Information Security and Privacy: Australasian Conference (ACISP'04)*, pages 391–401, 2004.

[45] F. Majorczyk, E. Totel, L. Mé, and A. Saïdane. Anomaly Detection with Diagnosis in Diversified Systems using Information Flow Graphs. In *Proceedings of the 22nd IFIP International Information Security Conference (SEC'08)*, pages 301–315, 2008.

[46] K.J. Mayer and N.S. Argyres. Learning to Contract: Evidence from the Personal Computer Industry. *Organization Science*, 15(4):394–410, 2004.

[47] N. Mayer, E. Dubois, and A. Rifaut. Requirements engineering for improving business/it alignment in security risk management methods. In *Proceedings of the 3rd International Conference Interoperability for Enterprise Software and Applications (I-ESA'07)*, page 12. I-ESA, 2007.

[48] D. Mellado, E. Fernández-Medina, and M. Piattini. A common criteria based security requirements engineering process for the development of secure information systems. *Computer Standards & Interfaces*, 29(2):244–253, 2007.

[49] R.A. Miura-Ko, B. Yolken, J. Mitchell, and N. Bambos. Security Decision-Making among Interdependent Organizations. In *Computer Security Foundations Symposium, IEEE*, pages 66–80. IEEE Computer Society, 2008.

[50] A.P. Moore, R.J. Ellison, and R.C. Linger. Attack Modeling for Information Security and Survivability. Technical report, Carnegie Mellon University, 2001.

[51] J. Mylopoulos, L. Chung, S. Liao, H. Wang, and E. Yu. Exploring alternatives during requirements analysis. *IEEE Software*, 18:92–96, 2001.

[52] S.L. Osborn. Information flow analysis of an RBAC system. In *Proceedings of the seventh ACM symposium on Access control models and technologies*, pages 163–168, 2002.

[53] M. C. Paulk, C. V. Weber, B. Curtis, and M. B. Chrissis. *The capability maturity model: guidelines for improving the software process*. Addison-Wesley Longman Publishing Co., Inc., 1995.

[54] C. Phillips and L.P. Swiler. A graph-based system for network-vulnerability analysis. In *Proceedings of the 1998 workshop on New security paradigms (NSPW'98)*, pages 71–79. ACM, 1998.

[55] L. Poppo and T.R. Zenger. Do formal contracts and relational governance function as substitutes or complements? *Strategic Management Journal*, 23:707–725, 2002.

[56] E.M. Power and R.L. Trope. Averting security missteps in outsourcing. *IEEE Security and Privacy*, 3(2):70–73, 2005.

[57] N.H. Roberts, W.E. Vesely, D.F. Haasl, and F.F. Goldberg. *Fault Tree Handbook*. System and Reliability Reseach Office of U.S. Nuclear Regulation Commition, 1981.

[58] J.E. Rossebø, S. Cadzow, and P. Sijben. eTVRA, a Threat, Vulnerability and Risk Assessment Method and Tool for eEurope. In *Proceedings of the The Second*

International Conference on Availability, Reliability and Security (ARES'07), pages 925–933. IEEE Computer Society, 2007.

[59] RTS/TISPAN-07006-TECH. Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Methods and Protocols; Part 1: Method and Proforma for Threat, Risk, Vulnerability Analysis. Technical Report TS 102 165-1 V4.2.1, European Telecommunications Standards Institute (ETSI), 2006.

[60] R. Sabherwal. The role of trust in outsourced is development projects. *ACM Communications*, 42(2):80–86, 1999.

[61] B. Schneier. Attack Trees. *Dr. Dobb's Journal*, 12(24):21–29, 1999.

[62] M. Schotten and E. Scherer. Design of co-ordination schemes in the networked enterprise. In *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics (SMC'10)*, volume 1, pages 313–318. IEEE Computer Society, 1998.

[63] G. Selimis, N. Sklavos, and O. Koufopavlou. Crypto processor for contactless smart cards. In *Proceedings of the 12th IEEE Mediterranean Electronical Conference* , volume 2, pages 803–806. IEEE Computer Society, 2004.

[64] G. Sindre and A.L. Opdahl. Eliciting security requirements with misuse cases. *Requirements Engineering*, 10(1):34–44, 2005.

[65] D.I.K. Sjøberg, B. Andal, E. Arisholml, T. Dybå, M. Jørgensenl, A. Karahasanovic, and M. Vokáccaron. Challenges and recommendations when increasing the realism of controlled software engineering experiments. In R. Conradi and A.I. Wang, editors, *Empirical Methods and Studies in Software Engineering*, pages 24–38. Springer, 2003. LNCS 2765.

[66] T. Srivatanakul, J.A. Clark, and F. Polack. Effective Security Requirements Analysis: HAZOP and Use Cases. In K. Zhang and Y. Zheng, editors, *Information Security*, volume 3225 of *Lecture Notes in Computer Science*, pages 416–427. Springer, 2004.

[67] D.H. Stamatis. *Failure mode and effect analysis FMEA from theory to execution*. American Society for Quality Press, 2003.

[68] H.F. Tipton and M. Krause. *Information Security Management Handbook*. Auerbach Publications, Boca Raton, New York, 2007.

[69] A. Vallecillo. *DINTEL Edition on Software Engineering. No. 3.*, chapter RM-ODP: The ISO Reference Model for Open Distributed Processing, pages 69–99. DINDEL, 2001.

[70] P.A.T. van Eck, H.M. Blanken, and R.J. Wieringa. Project GRAAL: Towards operational architecture alignment. *International Journal of Cooperative Information Systems*, 13(3):235–255, 2004.

[71] A. van Lamsweerde, S. Brohez, R. De Landtsheer, and D. Janssens. From system goals to intruder anti-goals: Attack generation and resolution for security requirements engineering. In *Proc. of RHAS Workshop*, pages 49–56. Essener Informatik Beitraege, Bd.6, 2003.

[72] R. Wieringa. 16th IEEE International Requirements Engineering Conference, Tutorial on Requirements Engineering Research Methodology: Principles and practice. `http://wwwhome.cs.utwente.nl/~roelw/DesignScienceMethodology-handout.pdf`, 2008.

[73] R. Wieringa, N. Maiden, N. Mead, and C. Rolland. Requirements engineering paper classification and evaluation criteria: A proposal and a discussion. *Journal of Requirements Engineering*, 11(1):102–107, 2006.

[74] R.J. Wieringa. Design science as nested problem solving. In *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology (DESRIST'09)*, pages 1–12. ACM, 2009.

[75] R. Winther, O.A. Johnsen, and B.A. Gran. Security Assessments of Safety Critical Systems Using HAZOPs. In *Proceedings of the 20th International Conference on Computer Safety, Reliability and Security (SAFECOMP'01)*, pages 14–24. Springer-Verlag, 2001.

[76] E. Zambon. *Towards Optimal IT Availability Planning: Methods and Tools*. PhD thesis, University of Twente, Enschede, The Netherlands, 2011.

Web References (Last Accessed: December 2010)

[77] BS IEC 61882:2001. Hazard and operability studies (HAZOP studies). Application guide. `http://products.ihs.com/cis/Doc.aspx?AuthCode=&DocNum=254734`, 2001.

[78] American Institude of Certified Public Accountants Auditing Standards Board. Statement on auditing standards; 70 Auditing Standards (SAS70). `http://umiss.lib.olemiss.edu:82/record=b1038093`, 2008.

[79] W.H. Baker, A. Hutton, C.D. Hylender, C. Novak, C. Porter, B. Sartin, P. Tippett, and J.A. Valentine. 2010 data breach investigations report: A study conducted by the verizon business risk team in cooperation with the united states secret services. `http://www.verizonbusiness.com/resources/reports/rp_2010-data-breach-report_en_xg.pdf`, 2010.

[80] Basel II: Revised international capital framework. http://www.bis.org/publ/bcbsca.htm, 2005.

[81] Bundesdatenschutzgesetz: § 42a Informationspflicht bei unrechtmiger Kenntniserlangung von Daten. http://bundesrecht.juris.de/bdsg_1990/__42a.html.

[82] ISO Technical Management Board. ISO 31000:2009 Risk management - Principles and guidelines. http://infostore.saiglobal.com/store/Details.aspx?productID=1378614, 2009.

[83] P. Bowen, J. Hash, and M. Wilson. Information Security Handbook: A Guide for Managers. http://csrc.nist.gov/publications/nistpubs/800-100/SP800-100-Mar07-2007.pdf, 2006.

[84] British Government's Central Computer and Telecommunications Agency. CRAMM: Risk Analysis and Management methodology. http://www.cramm.com/, 2008.

[85] French Certification Body Certificat. Common Criteria For Information Technology Security Evaluation: Smart Card Integrated Circuit Protection Profile (PP/9806). http://www.ssi.gouv.fr/site_documents/pp/pp9806.pdf, 1998.

[86] French Certification Body Certificat. Common Criteria For Information Technology Security Evaluation: Protection Profile Smart Card Integrated Circuit With Embedded Software (PP/9911). www.ssi.gouv.fr/site_documents/pp/pp9911.pdf, 1999.

[87] Control Objectives for Information and related Technology. http://www.isaca.org/, 2007.

[88] International Electrotechnical Commission. International Standard ISO/IEC 9126, Information technology - Software product evaluation - Quality characteristics and guidelines for their use. http://www.iso.org, 1991.

[89] International Electrotechnical Commission. ISO/IEC 15408-3:2008 Information technology – Security techniques – Evaluation criteria for IT security – Part 3: Security assurance components. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=46413, 2007.

[90] International Electrotechnical Commission. ISO/IEC 15408-2:2008 Information technology – Security techniques – Evaluation criteria for IT security – Part 2: Security functional components. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=46414, 2008.

[91] International Electrotechnical Commission. ISO/IEC 15408-1:2009 Information technology – Security techniques – Evaluation criteria for IT security – Part 1: Introduction and general model. http://www.iso.org/iso/catalogue_detail.htm?csnumber=40612, 2009.

[92] Health Insurance Portability and Accountability Act (HIPAA) of 1996 (P.L.104-191). http://aspe.hhs.gov/admnsimp/pl104191.htm, 1996.

[93] ISO/IEC FDIS 27001:2005(E) Information technology – Security techniques – Information security management systems – Requirements. http://www.iso.org, 2005.

[94] Information techniques - security techniques - code of practice for information security management. http://www.iso.org/iso/catalogue_detail?csnumber=50297, February 2005. (Previously known as ISO/IEC 17799:2005).

[95] Information technology – Security techniques – Information security risk management. http://www.iso.org/iso/catalogue_detail?csnumber=42107, 2008.

[96] NIST: National Vulnerability Database. http://nvd.nist.gov/, 2008.

[97] Joint Technical Committee OB-007. Risk Management: AS/NZS 4360. http://infostore.saiglobal.com/store/details.aspx?ProductID=381579, 2004. (Superseeded by ISO 31000:2009.).

[98] Sarbanes-Oxley Act of 2002. http://www.sarbanes-oxley.com/, 2002.

[99] G. Stoneburner, A. Goguen, and A. Feringa. Risk Management Guide for Information Technology Systems. http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf, 2002.

[100] The Zachman Institute for Framework Advancement. Zachman Framework. http://www.zifa.com/, 2007.

[101] U.S. Department of Transportation - Federal Aviation Administration. Appendix G: FAA Order 8040.4. http://www.faa.gov/library/manuals/aviation/risk_management/ss_handbook/media/app_g_1200.PDF, 1998.

[102] S. Viveros and S. Yeardsley. McAfee, Inc. Research Shows Global Recession Increasing Risks to Intellectual Property. http://www.mcafee.com/us/about/press/corporate/2009/20090129_063500_j.html, 2009.

Titles in the IPA Dissertation Series since 2005

E. Ábrahám. *An Assertional Proof System for Multithreaded Java -Theory and Tool Support-* . Faculty of Mathematics and Natural Sciences, UL. 2005-01

R. Ruimerman. *Modeling and Remodeling in Bone Tissue*. Faculty of Biomedical Engineering, TU/e. 2005-02

C.N. Chong. *Experiments in Rights Control - Expression and Enforcement*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2005-03

H. Gao. *Design and Verification of Lock-free Parallel Algorithms*. Faculty of Mathematics and Computing Sciences, RUG. 2005-04

H.M.A. van Beek. *Specification and Analysis of Internet Applications*. Faculty of Mathematics and Computer Science, TU/e. 2005-05

M.T. Ionita. *Scenario-Based System Architecting - A Systematic Approach to Developing Future-Proof System Architectures*. Faculty of Mathematics and Computing Sciences, TU/e. 2005-06

G. Lenzini. *Integration of Analysis Techniques in Security and Fault-Tolerance*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2005-07

I. Kurtev. *Adaptability of Model Transformations*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2005-08

T. Wolle. *Computational Aspects of Treewidth - Lower Bounds and Network Reliability*. Faculty of Science, UU. 2005-09

O. Tveretina. *Decision Procedures for Equality Logic with Uninterpreted Functions*. Faculty of Mathematics and Computer Science, TU/e. 2005-10

A.M.L. Liekens. *Evolution of Finite Populations in Dynamic Environments*. Faculty of Biomedical Engineering, TU/e. 2005-11

J. Eggermont. *Data Mining using Genetic Programming: Classification and Symbolic Regression*. Faculty of Mathematics and Natural Sciences, UL. 2005-12

B.J. Heeren. *Top Quality Type Error Messages*. Faculty of Science, UU. 2005-13

G.F. Frehse. *Compositional Verification of Hybrid Systems using Simulation Relations*. Faculty of Science, Mathematics and Computer Science, RU. 2005-14

M.R. Mousavi. *Structuring Structural Operational Semantics*. Faculty of Mathematics and Computer Science, TU/e. 2005-15

A. Sokolova. *Coalgebraic Analysis of Probabilistic Systems*. Faculty of Mathematics and Computer Science, TU/e. 2005-16

T. Gelsema. *Effective Models for the Structure of pi-Calculus Processes with Replication*. Faculty of Mathematics and Natural Sciences, UL. 2005-17

P. Zoeteweij. *Composing Constraint Solvers*. Faculty of Natural Sciences, Mathematics, and Computer Science, UvA. 2005-18

J.J. Vinju. *Analysis and Transformation of Source Code by Parsing and*

Rewriting. Faculty of Natural Sciences, Mathematics, and Computer Science, UvA. 2005-19

M.Valero Espada. *Modal Abstraction and Replication of Processes with Data*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2005-20

A. Dijkstra. *Stepping through Haskell*. Faculty of Science, UU. 2005-21

Y.W. Law. *Key management and link-layer security of wireless sensor networks: energy-efficient attack and defense*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2005-22

E. Dolstra. *The Purely Functional Software Deployment Model*. Faculty of Science, UU. 2006-01

R.J. Corin. *Analysis Models for Security Protocols*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2006-02

P.R.A. Verbaan. *The Computational Complexity of Evolving Systems*. Faculty of Science, UU. 2006-03

K.L. Man and R.R.H. Schiffelers. *Formal Specification and Analysis of Hybrid Systems*. Faculty of Mathematics and Computer Science and Faculty of Mechanical Engineering, TU/e. 2006-04

M. Kyas. *Verifying OCL Specifications of UML Models: Tool Support and Compositionality*. Faculty of Mathematics and Natural Sciences, UL. 2006-05

M. Hendriks. *Model Checking Timed Automata - Techniques and Applications*. Faculty of Science, Mathematics and Computer Science, RU. 2006-06

J. Ketema. *Böhm-Like Trees for Rewriting*. Faculty of Sciences, VUA. 2006-07

C.-B. Breunesse. *On JML: topics in tool-assisted verification of JML programs*. Faculty of Science, Mathematics and Computer Science, RU. 2006-08

B. Markvoort. *Towards Hybrid Molecular Simulations*. Faculty of Biomedical Engineering, TU/e. 2006-09

S.G.R. Nijssen. *Mining Structured Data*. Faculty of Mathematics and Natural Sciences, UL. 2006-10

G. Russello. *Separation and Adaptation of Concerns in a Shared Data Space*. Faculty of Mathematics and Computer Science, TU/e. 2006-11

L. Cheung. *Reconciling Nondeterministic and Probabilistic Choices*. Faculty of Science, Mathematics and Computer Science, RU. 2006-12

B. Badban. *Verification techniques for Extensions of Equality Logic*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2006-13

A.J. Mooij. *Constructive formal methods and protocol standardization*. Faculty of Mathematics and Computer Science, TU/e. 2006-14

T. Krilavicius. *Hybrid Techniques for Hybrid Systems*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2006-15

M.E. Warnier. *Language Based Security for Java and JML*. Faculty of Science, Mathematics and Computer Science, RU. 2006-16

V. Sundramoorthy. *At Home In Service Discovery*. Faculty of Electrical

Engineering, Mathematics & Computer Science, UT. 2006-17

B. Gebremichael. *Expressivity of Timed Automata Models*. Faculty of Science, Mathematics and Computer Science, RU. 2006-18

L.C.M. van Gool. *Formalising Interface Specifications*. Faculty of Mathematics and Computer Science, TU/e. 2006-19

C.J.F. Cremers. *Scyther - Semantics and Verification of Security Protocols*. Faculty of Mathematics and Computer Science, TU/e. 2006-20

J.V. Guillen Scholten. *Mobile Channels for Exogenous Coordination of Distributed Systems: Semantics, Implementation and Composition*. Faculty of Mathematics and Natural Sciences, UL. 2006-21

H.A. de Jong. *Flexible Heterogeneous Software Systems*. Faculty of Natural Sciences, Mathematics, and Computer Science, UvA. 2007-01

N.K. Kavaldjiev. *A run-time reconfigurable Network-on-Chip for streaming DSP applications*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2007-02

M. van Veelen. *Considerations on Modeling for Early Detection of Abnormalities in Locally Autonomous Distributed Systems*. Faculty of Mathematics and Computing Sciences, RUG. 2007-03

T.D. Vu. *Semantics and Applications of Process and Program Algebra*. Faculty of Natural Sciences, Mathematics, and Computer Science, UvA. 2007-04

L. Brandán Briones. *Theories for Model-based Testing: Real-time and Coverage*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2007-05

I. Loeb. *Natural Deduction: Sharing by Presentation*. Faculty of Science, Mathematics and Computer Science, RU. 2007-06

M.W.A. Streppel. *Multifunctional Geometric Data Structures*. Faculty of Mathematics and Computer Science, TU/e. 2007-07

N. Trčka. *Silent Steps in Transition Systems and Markov Chains*. Faculty of Mathematics and Computer Science, TU/e. 2007-08

R. Brinkman. *Searching in encrypted data*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2007-09

A. van Weelden. *Putting types to good use*. Faculty of Science, Mathematics and Computer Science, RU. 2007-10

J.A.R. Noppen. *Imperfect Information in Software Development Processes*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2007-11

R. Boumen. *Integration and Test plans for Complex Manufacturing Systems*. Faculty of Mechanical Engineering, TU/e. 2007-12

A.J. Wijs. *What to do Next?: Analysing and Optimising System Behaviour in Time*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2007-13

C.F.J. Lange. *Assessing and Improving the Quality of Modeling: A Series of Empirical Studies about the UML.*

Faculty of Mathematics and Computer Science, TU/e. 2007-14

T. van der Storm. *Component-based Configuration, Integration and Delivery*. Faculty of Natural Sciences, Mathematics, and Computer Science,UvA. 2007-15

B.S. Graaf. *Model-Driven Evolution of Software Architectures*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2007-16

A.H.J. Mathijssen. *Logical Calculi for Reasoning with Binding*. Faculty of Mathematics and Computer Science, TU/e. 2007-17

D. Jarnikov. *QoS framework for Video Streaming in Home Networks*. Faculty of Mathematics and Computer Science, TU/e. 2007-18

M. A. Abam. *New Data Structures and Algorithms for Mobile Data*. Faculty of Mathematics and Computer Science, TU/e. 2007-19

W. Pieters. *La Volonté Machinale: Understanding the Electronic Voting Controversy*. Faculty of Science, Mathematics and Computer Science, RU. 2008-01

A.L. de Groot. *Practical Automaton Proofs in PVS*. Faculty of Science, Mathematics and Computer Science, RU. 2008-02

M. Bruntink. *Renovation of Idiomatic Crosscutting Concerns in Embedded Systems*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2008-03

A.M. Marin. *An Integrated System to Manage Crosscutting Concerns in Source Code*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2008-04

N.C.W.M. Braspenning. *Model-based Integration and Testing of High-tech Multi-disciplinary Systems*. Faculty of Mechanical Engineering, TU/e. 2008-05

M. Bravenboer. *Exercises in Free Syntax: Syntax Definition, Parsing, and Assimilation of Language Conglomerates*. Faculty of Science, UU. 2008-06

M. Torabi Dashti. *Keeping Fairness Alive: Design and Formal Verification of Optimistic Fair Exchange Protocols*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2008-07

I.S.M. de Jong. *Integration and Test Strategies for Complex Manufacturing Machines*. Faculty of Mechanical Engineering, TU/e. 2008-08

I. Hasuo. *Tracing Anonymity with Coalgebras*. Faculty of Science, Mathematics and Computer Science, RU. 2008-09

L.G.W.A. Cleophas. *Tree Algorithms: Two Taxonomies and a Toolkit*. Faculty of Mathematics and Computer Science, TU/e. 2008-10

I.S. Zapreev. *Model Checking Markov Chains: Techniques and Tools*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2008-11

M. Farshi. *A Theoretical and Experimental Study of Geometric Networks*. Faculty of Mathematics and Computer Science, TU/e. 2008-12

G. Gulesir. *Evolvable Behavior Specifications Using Context-Sensitive Wildcards*. Faculty of Electrical Engineering,

Mathematics & Computer Science, UT. 2008-13

F.D. Garcia. *Formal and Computational Cryptography: Protocols, Hashes and Commitments*. Faculty of Science, Mathematics and Computer Science, RU. 2008-14

P. E. A. Dürr. *Resource-based Verification for Robust Composition of Aspects*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2008-15

E.M. Bortnik. *Formal Methods in Support of SMC Design*. Faculty of Mechanical Engineering, TU/e. 2008-16

R.H. Mak. *Design and Performance Analysis of Data-Independent Stream Processing Systems*. Faculty of Mathematics and Computer Science, TU/e. 2008-17

M. van der Horst. *Scalable Block Processing Algorithms*. Faculty of Mathematics and Computer Science, TU/e. 2008-18

C.M. Gray. *Algorithms for Fat Objects: Decompositions and Applications*. Faculty of Mathematics and Computer Science, TU/e. 2008-19

J.R. Calamé. *Testing Reactive Systems with Data - Enumerative Methods and Constraint Solving*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2008-20

E. Mumford. *Drawing Graphs for Cartographic Applications*. Faculty of Mathematics and Computer Science, TU/e. 2008-21

E.H. de Graaf. *Mining Semi-structured Data, Theoretical and Experimental Aspects of Pattern Evaluation*. Faculty of Mathematics and Natural Sciences, UL. 2008-22

R. Brijder. *Models of Natural Computation: Gene Assembly and Membrane Systems*. Faculty of Mathematics and Natural Sciences, UL. 2008-23

A. Koprowski. *Termination of Rewriting and Its Certification*. Faculty of Mathematics and Computer Science, TU/e. 2008-24

U. Khadim. *Process Algebras for Hybrid Systems: Comparison and Development*. Faculty of Mathematics and Computer Science, TU/e. 2008-25

J. Markovski. *Real and Stochastic Time in Process Algebras for Performance Evaluation*. Faculty of Mathematics and Computer Science, TU/e. 2008-26

H. Kastenberg. *Graph-Based Software Specification and Verification*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2008-27

I.R. Buhan. *Cryptographic Keys from Noisy Data Theory and Applications*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2008-28

R.S. Marin-Perianu. *Wireless Sensor Networks in Motion: Clustering Algorithms for Service Discovery and Provisioning*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2008-29

M.H.G. Verhoef. *Modeling and Validating Distributed Embedded Real-Time Control Systems*. Faculty of Science, Mathematics and Computer Science, RU. 2009-01

M. de Mol. *Reasoning about Functional Programs: Sparkle, a proof assistant for*

Clean. Faculty of Science, Mathematics and Computer Science, RU. 2009-02

M. Lormans. *Managing Requirements Evolution*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2009-03

M.P.W.J. van Osch. *Automated Model-based Testing of Hybrid Systems*. Faculty of Mathematics and Computer Science, TU/e. 2009-04

H. Sozer. *Architecting Fault-Tolerant Software Systems*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2009-05

M.J. van Weerdenburg. *Efficient Rewriting Techniques*. Faculty of Mathematics and Computer Science, TU/e. 2009-06

H.H. Hansen. *Coalgebraic Modelling: Applications in Automata Theory and Modal Logic*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2009-07

A. Mesbah. *Analysis and Testing of Ajax-based Single-page Web Applications*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2009-08

A.L. Rodriguez Yakushev. *Towards Getting Generic Programming Ready for Prime Time*. Faculty of Science, UU. 2009-9

K.R. Olmos Joffré. *Strategies for Context Sensitive Program Transformation*. Faculty of Science, UU. 2009-10

J.A.G.M. van den Berg. *Reasoning about Java programs in PVS using JML*. Faculty of Science, Mathematics and Computer Science, RU. 2009-11

M.G. Khatib. *MEMS-Based Storage Devices. Integration in Energy-Constrained Mobile Systems*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2009-12

S.G.M. Cornelissen. *Evaluating Dynamic Analysis Techniques for Program Comprehension*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2009-13

D. Bolzoni. *Revisiting Anomaly-based Network Intrusion Detection Systems*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2009-14

H.L. Jonker. *Security Matters: Privacy in Voting and Fairness in Digital Exchange*. Faculty of Mathematics and Computer Science, TU/e. 2009-15

M.R. Czenko. *TuLiP - Reshaping Trust Management*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2009-16

T. Chen. *Clocks, Dice and Processes*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2009-17

C. Kaliszyk. *Correctness and Availability: Building Computer Algebra on top of Proof Assistants and making Proof Assistants available over the Web*. Faculty of Science, Mathematics and Computer Science, RU. 2009-18

R.S.S. O'Connor. *Incompleteness & Completeness: Formalizing Logic and Analysis in Type Theory*. Faculty of Science, Mathematics and Computer Science, RU. 2009-19

B. Ploeger. *Improved Verification Methods for Concurrent Systems*. Faculty

of Mathematics and Computer Science, TU/e. 2009-20

T. Han. *Diagnosis, Synthesis and Analysis of Probabilistic Models*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2009-21

R. Li. *Mixed-Integer Evolution Strategies for Parameter Optimization and Their Applications to Medical Image Analysis*. Faculty of Mathematics and Natural Sciences, UL. 2009-22

J.H.P. Kwisthout. *The Computational Complexity of Probabilistic Networks*. Faculty of Science, UU. 2009-23

T.K. Cocx. *Algorithmic Tools for Data-Oriented Law Enforcement*. Faculty of Mathematics and Natural Sciences, UL. 2009-24

A.I. Baars. *Embedded Compilers*. Faculty of Science, UU. 2009-25

M.A.C. Dekker. *Flexible Access Control for Dynamic Collaborative Environments*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2009-26

J.F.J. Laros. *Metrics and Visualisation for Crime Analysis and Genomics*. Faculty of Mathematics and Natural Sciences, UL. 2009-27

C.J. Boogerd. *Focusing Automatic Code Inspections*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2010-01

M.R. Neuhäußer. *Model Checking Nondeterministic and Randomly Timed Systems*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2010-02

J. Endrullis. *Termination and Productivity*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2010-03

T. Staijen. *Graph-Based Specification and Verification for Aspect-Oriented Languages*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2010-04

Y. Wang. *Epistemic Modelling and Protocol Dynamics*. Faculty of Science, UvA. 2010-05

J.K. Berendsen. *Abstraction, Prices and Probability in Model Checking Timed Automata*. Faculty of Science, Mathematics and Computer Science, RU. 2010-06

A. Nugroho. *The Effects of UML Modeling on the Quality of Software*. Faculty of Mathematics and Natural Sciences, UL. 2010-07

A. Silva. *Kleene Coalgebra*. Faculty of Science, Mathematics and Computer Science, RU. 2010-08

J.S. de Bruin. *Service-Oriented Discovery of Knowledge - Foundations, Implementations and Applications*. Faculty of Mathematics and Natural Sciences, UL. 2010-09

D. Costa. *Formal Models for Component Connectors*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2010-10

M.M. Jaghoori. *Time at Your Service: Schedulability Analysis of Real-Time and Distributed Services*. Faculty of Mathematics and Natural Sciences, UL. 2010-11

R. Bakhshi. *Gossiping Models: Formal Analysis of Epidemic Protocols*. Faculty

of Sciences, Department of Computer Science, VUA. 2011-01

B.J. Arnoldus. *An Illumination of the Template Enigma: Software Code Generation with Templates*. Faculty of Mathematics and Computer Science, TU/e. 2011-02

E. Zambon. *Towards Optimal IT Availability Planning: Methods and Tools*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2011-03

L. Astefanoaei. *An Executable Theory of Multi-Agent Systems Refinement*. Faculty of Mathematics and Natural Sciences, UL. 2011-04

J. Proença. *Synchronous coordination of distributed components*. Faculty of Mathematics and Natural Sciences, UL. 2011-05

A. Morali. *IT Architecture-Based Confidentiality Risk Assessment in Networks of Organizations*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2011-06