Helsinki University of Technology Networking Laboratory

Teknillinen korkeakoulu Tietoverkkolaboratorio

Espoo 2007 Report 5/2007

TRUST MANAGEMENT FOR MOBILE COMPUTING PLATFORMS

Zheng Yan

Dissertation for the degree of Doctor of Science in Technology to be presented with due permission of the Department of Electrical and Communications Engineering for public examination and debate in Auditorium S5 at Helsinki University of Technology (Espoo, Finland) on the 14th of December, 2007, at 12 o'clock noon.

Helsinki University of Technology
Department of Electrical and Communications Engineering
Networking Laboratory

Teknillinen korkeakoulu
Sähkö- ja tietoliikennetekniikan osasto
Tietoverkkolaboratorio

Distributor:
Helsinki University of Technology
Networking Laboratory
P.O. Box 3000
FI-02015 TKK

Tel. +358-9-451 2461
Fax +358-9-451 2474

ISBN 978-951-22-9119-9
ISSN 1458-0322

Multiprint Oy
Espoo 2007

HELSINKI UNIVERSITY OF TECHNOLOGY
P. O. BOX 1000, FI-02015 TKK
http://www.tkk.fi

ABSTRACT OF DOCTORAL DISSERTATION

Author Zheng Yan

Name of the dissertation

Trust Management for Mobile Computing Platforms

Date of manuscript October 23, 2007 | Date of the dissertation December 14, 2007

☐ Monograph | ☒ Article dissertation (summary + original articles)

Department Department of Electrical and Communications Engineering
Laboratory Networking Laboratory
Field of research Trust Management and Trusted Computing
Opponent(s) Associate Professor Christian Damsgaard Jensen (Technical University of Denmark, Denmark)
Supervisor Professor Raimo Kantola (Helsinki University of Technology)

Abstract

Providing a trustworthy mobile computing platform is crucial for mobile communications, services and applications. In this dissertation, we study methodologies and mechanisms that can be used to provide a trustworthy mobile computing platform. We also present an autonomic trust management solution for a component software middleware platform targeting at an embedded device, such as a mobile phone.

In the first part of the dissertation, we firstly overview the literature background of trust modeling and trust management. We propose research methodologies on the basis of a conceptual architecture of a trusted mobile environment. Further, we present a methodology to bridge disjoint trusted domains in mobile computing and communications into a trustworthy system.

The second part of the dissertation contains a mechanism to sustain trust among computing platforms. The mechanism builds up a trust relationship based on the Root Trust (RT) module at a trustee platform and ensures trust sustainability according to pre-defined conditions. These conditions are approved at the time of trust establishment and enforced through the use of the pre-attested RT module until the intended purpose is fulfilled. Through applying this mechanism, we introduce a Trusted Collaboration Infrastructure (TCI) for peer-to-peer devices in order to establish trust collaboration among distributed peers. In addition, this mechanism contributes to a mobile Virtual Private Network (VPN) for trusted mobile enterprise networking.

The third part of the dissertation presents an autonomic trust management solution that can manage trust adaptively in a middleware component software platform. We develop a formal trust model to specify, evaluate, set up and ensure trust relationships that exist among system entities. We further present a trust management architecture that supports the implementation of the above model and adopts a number of algorithms for autonomic trust management at system runtime. In particular, special control modes can be applied into the platform to ensure trustworthiness. We develop a methodology for trust control mode prediction and selection on the basis of an adaptive trust control model in order to support autonomic trust management.

Keywords Trust management, trusted computing, trust model, security, component software

ISBN (printed)	978-951-22-9119-9	ISSN (printed)	1458-0322
ISBN (pdf)	978-951-22-9120-5	ISSN (pdf)	1458-0322
ISBN (others)		Number of pages	102 p. + app. 111 p.

Publisher Networking Laboratory, Helsinki University of Technology

Print distribution

☒ The dissertation can be read at http://lib.tkk.fi/Diss/2007/isbn9789512291205

PREFACE

I would like to express my thanks to my supervisor Prof. Raimo Kantola who offered me this opportunity to conduct my doctoral study at the Helsinki University of Technology. I would like to express my deepest appreciation and sincere gratitude to him and both reviewers for their invaluable comments on this dissertation as a whole.

I am grateful to the Nokia Research Center for supporting my research on trust modeling and management. The work presented in this dissertation is fully sponsored by and conducted at the Nokia Research Center from 2003 to 2006 in several projects. I would like to extend my gratitude to my ex-superiors Mr. Petteri Saarinen, and later on Mr. Olli Immonen who trusted and supported me to complete this work, as well as their valuable comments on my publications. Particularly, I appreciate Dr. Valtteri Niemi's comments and crucial instruction towards the finalization of the dissertation and Prof. N.Asokan's valuable comments and suggestions, especially on the contents related to trusted computing and communications. During the period of the dissertation revision, the discussions with Dr. Christian Prehofer greatly helped me elaborate the contents with regard to autonomic trust management for component software platform.

I would like to thank my co-authors, Dr. Piotr Cofta for our rememberable cooperation on two paper publications, Ronan MacLaverty for the cooperation in European Union ITEA Trust4All project, and Dr. Silke Holtmanns for the incentive and cooperation of a book chapter about trust modeling and management. In particular, I benefited a lot through the discussions with many project members of the Trust4All.

I feel deeply indebted to my husband, Dr. Peng Zhang, without whom I would not have considered to enroll in doctoral research. Without his professional instruction, encouragement and assistance, it would have been impossible for me to overcome many difficulties and frustrations, especially at the beginning when I had no idea how to write a scientific paper. He also contributed and co-authored a number of my paper publications related to digital content protection, peer-to-peer systems, enterprise networking, and ad hoc networks. His patience and attitude towards life and research help me comprehend more the significance of scientific work: an individual exercise of spirit and wisdom. His endless support on both of my life and work has been a great source of inspiration in the completion of this dissertation.

I should thank my son, Kuan, who is the key motivation of my career and life. He brings me a lot of happiness and hopes. Life is becoming more significant and colorful because of him. Meanwhile, I am so moved by my parents, sister and brother, as well as my sister-in-law, for their constant caring, encouragement and support throughout my life. I would like to express my special thanks to Miss Ren Yanhong for her help in taking care of Kuan for fourteen months, during which most of the results presented in this dissertation were achieved.

Finally, I would like to thank my friends: Liu Yaohui and Han Shengnan for doctoral experience exchange and encouragement with each other; Li Changhong,

Ma Xiao, Zhang Xiachang and Yang Mu for enjoying the fun of playing badminton; Feng Juan for practicing yoga together; Wang Yue's family, Gu Jun's family, Yu Lixin's family, Fu Yan's family, Wang Yi's family, Ma Yaoyu's family and Wang Yafeng for sharing wonderful time with my family and their helps in our life. I have to say all above aided my doctoral research in special ways. I would like to take this chance to express my thanks to them.

A lot of people played important roles for helping me finish this work within six years, including a one-year maternity leave. For many of those not mentioned above, I would like to give my warmest thanks.

Helsinki, November 2006

Zheng Yan

CONTENTS

Preface **5**

Contents **7**

List of Publications **11**

Author's Contribution **13**

List of Abbreviations **15**

1 Introduction **17**
1.1 Trust and its derivatives..…..……...…….. 17
1.2 Computing platforms…….. 17
1.2.1 Mobile computing platform……... 17
1.2.2 Trusted computing platform…….. 17
1.2.3 Component software platform……... 18
1.3 Motivations of trust management for mobile computing platforms... 18
1.4 Research issues……... 19
1.4.1 Methodology development…… 19
1.4.2 Personal trusted device…….. 19
1.4.3 Trusted mobile communications…… 20
1.5 Research objectives……. 21
1.6 Contribution to knowledge…….. 21
1.7 Outline of the thesis……. 23

2 Background **25**
2.1 Factors of trust…… 25
2.2 Trust modeling…… 26
2.2.1 Characteristics of trust……... 26
2.2.1 Trust models…….. 27
2.3 Trust management…….. 28
2.4 Trust evaluation mechanism……... 29
2.5 Emerging trends…….. 31
2.5.1 An integrated solution…… 31
2.5.2 Autonomic trust management……... 31
2.5.3 Cross-domain benefits...……. 32
2.6 Summary……. 32

3 Research methodologies **33**
3.1 A conceptual architecture based research method…….. 33
3.1.1 Definitions – concept circle……... 33
3.1.2 Theory and modeling methodology – theory circle……... 34
3.1.3 Trust models and standards – practice circle……. 34

3.1.4 Mobile applications and systems – application circle…….. 35
3.2 A methodology to bridge different domains of trust…….. 35
3.2.1 Introduction…… 35
3.2.2 Methodology…….. 36
3.3 An Illustration……. 37
3.4 Summary……. 38

4 Trusted Computing Platform Based Solutions for Mobile Computing 41
4.1 Motivation…….. 41
4.2 Related work…….. 41
4.3 Problem statement…….. 42
4.4 A mechanism for trust sustainability among platforms……. 43
4.4.1 Trust form……. 44
4.4.2 Root trust module……. 44
4.4.3 Mechanism for trust sustainability……... 45
4.5 Trust collaboration in P2P systems…… 47
4.5.1 Trusted collaboration infrastructure for P2P……. 47
4.5.2 Trust collaboration…… 49
4.5.3 Deployment…… 52
4.5.4 Remarks…… 52
4.6 Trust management in mobile enterprise networking…….. 52
4.6.1 Problem statement……. 53
4.6.2 Trust management in mobile VPN……..... 54
4.6.3 Remarks……. 58
4.7 Summary……. 58

5 Autonomic Trust Management for a Component Software Platform 59
5.1 Introduction……. 59
5.2 Related work……... 60
5.3 Trust issues in component software…… 62
5.4 Requirements and approaches to autonomic trust management……. 62
5.5 A Formal Trust model…….…… 65
5.6 Trust management architecture……... 69
5.6.1 Platform structure……... 69
5.6.2 Trust management framework…… 70
5.7 Trust assessment at runtime…… 71
5.7.1 Notations and definitions……... 71
5.7.2 Trust assessment algorithms…….. 73
5.7.3 General criteria support……. 74
5.8 Adaptive trust control modeling and control mode selection……. 75
5.8.1 Fuzzy cognitive map…….. 75
5.8.2 Trust control modeling……..…… 76
5.8.3 Trust control mode prediction and selection…..……. 79
5.8.4 Adaptive trust control model adjustment…….... 80
5.8.5 Examples and simulation results……..…….. 81
5.8.6 Further discussion…… 84
5.9 Remarks…….. 87
5.10 Summary……. 87

6 Conclusions and Future Work **89**

References **93**

Errata **101**

Appendices **103**
Individual articles

LIST OF PUBLICATIONS

[1] Zheng Yan and Silke Holtmanns, "Trust Modeling and Management: from Social Trust to Digital Trust", book chapter of *Computer Security, Privacy and Politics: Current Issues, Challenges and Solutions*, IGI Global, 2007.

[2] Zheng Yan, "A Conceptual Architecture of a Trusted Mobile Environment", In *Proceedings of IEEE 2nd International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing (SecPerU06)*, held in conjunction with IEEE International Conference on Pervasive Services 2006, pp. 75-81, France, June 2006.

[3] Zheng Yan and Piotr Cofta, "Methodology to Bridge Different Domains of Trust in Mobile Communications", In *Proceedings of the 1st International Conference on Trust Management (iTrust2003)*, LNCS Vol. 2692/2003, pp. 211-224, Greece, May 2003.

[4] Zheng Yan and Piotr Cofta, "A Mechanism for Trust Sustainability among Trusted Computing Platforms", In *Proceedings of the 1st International Conference on Trust and Privacy in Digital Business (TrustBus2004)*, LNCS Vol. 3184/2004, pp. 11-19, Spain, September 2004.

[5] Zheng Yan and Peng Zhang, "Trust Collaboration in P2P Systems Based on Trusted Computing Platforms", *WSEAS Transactions on Information Science and Applications*, Issue 2, Vol. 3, pp. 275-282, February 2006.

[6] Zheng Yan and Peng Zhang, "A Trust Management System in Mobile Enterprise Networking", *WSEAS Transactions on Communications*, Issue 5, Vol. 5, pp. 854-861, May 2006.

[7] Zheng Yan and Ronan MacLaverty, "Autonomic Trust Management in a Component Based Software System", In *Proceedings of the 3rd International Conference on Autonomic and Trusted Computing (ATC2006)*, LNCS Vol. 4158/2006, pp. 279-292, China, September 2006.

[8] Zheng Yan, "A Methodology to Predict and Select Control Modes for a Trustworthy Platform", *WSEAS Transactions on Computer*, Issue 3, Vol. 6, pp. 471-477, March 2007.

AUTHOR'S CONTRIBUTION

Publication 1: This book chapter was written by the present author. The idea and main contents are the work of the present author. Dr. Silke Holtmanns input the contents about her publications and provided valuable comments and revision on the book chapter as a whole.

Publication 2: This paper is independent work of the author.

Publication 3: This paper is joint work with Dr. Piotr Cofta. The idea to bridge different domains of trust is by the present author. Dr. Piotr Cofta contributed to the three approaches to identify bridging solutions, the application about mobile trust interaction in a mobile terminal and the presentation of the whole paper.

Publication 4: This paper was written by the present author. The idea and result to sustain trust among trusted computing platforms are by the present author. Dr. Piotr Cofta conducted final revision.

Publication 5: This paper was written by the present author. It is almost independent work of the present author. The idea about device management is from Dr. Peng Zhang.

Publication 6: This paper was written by the present author. The idea and result to apply the mechanism of trust sustainability into mobile enterprise networking are by the present author. Dr. Peng Zhang contributed to the problem statements and the trust management solution for a mobile virtual private network.

Publication 7: This paper was written by the present author. The idea and results about trust modeling, trust management architecture for component software, and Subjective Logic based trust evaluation algorithms are by the present author. Ronan MacLaverty contributed to the contents about related work, system structure and the statement of trust issues and provided final revision.

Publication 8: This paper is independent work of the author.

LIST OF ABBREVIATIONS

BIOS	Basic Input/Output System
CA	Certificate Authority
CBS	Component Based Software
CS	Component Software
DoS	Denial of Service
DRM	Digital Rights Management
EU	European Union
FCM	Fuzzy Cognitive Map
FL	Fuzzy Logic
GRID	Global Resource Information Database
GSM	Global System for Mobile Communications
HMI	Human-Machine Interaction
HW	Hardware
ITEA	Information Technology for European Advancement
JVM	Java Virtual Machine
JXTA	Juxtapose
MANET	Mobile Ad Hoc Network
MID	Mobile Information Device
MIDP	Mobile Information Device Profile
NN	Neural Network
OMA	Open Mobile Alliance
OS	Operating System
P2P	Peer-to-Peer
PDA	Personal Digital Assistant
PKI	Public Key Infrastructure
PTD	Personal Trusted Device
QA	Quality Attribute
QoS	Quality of Service
RE	Runtime Environment
RT	Root Trust
SE	Secure Element
SIM	Subscriber Identity Module
SKIP	Simple Key-Management for Internet Protocols

SL	Subjective Logic
SP	Service Provider
SW	Software
TA	Trust Authority
TC	Trusted Computing
TCB	Trusted Computing Base
TCG	Trusted Computing Group
TCI	Trusted Collaboration Infrastructure
TCPA	Trusted Computing Platform Alliance
TMC	Trusted Mobile Communications
TMCP	Trusted Mobile Computing Platform
TPM	Trusted Platform Module
TSS	TPM Software Stack
TTP	Trusted Third Party
UML	Unified Modeling Language
VPN	Virtual Private Network
WIM	WAP Identity Module
WLAN	Wireless Local-Area Network
XML	eXtensible Markup Language

1 INTRODUCTION

This dissertation studies trust management for mobile computing platforms. Concretely, we study methodologies and mechanisms to provide a trustworthy computing platform for mobile devices. Further, we seek solutions to support trusted communications and collaboration among those platforms in a distributed and dynamic system.

1.1 Trust and Its Derivatives

The concept of trust has been studied in disciplines ranging from economics to psychology, from sociology to medicine, and to information science. It is hard to say what trust exactly is because it is a multidimensional, multidiscipline and multifaceted concept. We can find various definitions of trust in the literature. Common to these definitions are the notions of confidence, belief, faith, hope, expectation, dependence, and reliance on the goodness, strength, reliability, integrity, ability, or character of a person or thing.

Generally, a trust relationship involves two parties: a trustor and a trustee. The trustor is the person or entity who holds confidence, belief, faith, hope, expectation, dependence, and reliance on the goodness, strength, reliability, integrity, ability, or character of another person or thing, which is the object of trust - the trustee.

1.2 Computing Platforms

1.2.1 Mobile Computing Platform

A *computing platform* is a framework, either in hardware or software, which allows software to run. A typical mobile computing platform includes a mobile device's architecture, operating system, or programming languages and their runtime libraries. Generally, a mobile computing platform contains three layers: an application layer that provides features to a user; a middleware layer that provides functionality to applications; and, a foundational platform layer that includes the OS and provides access to lower-level hardware.

1.2.2 Trusted Computing Platform

A *trusted computing platform* is a computing platform that behaves in a way as it is expected to behave for an intended purpose. For example, the most important work about the trusted computing (TC) platform is conducted in the Trusted Computing Group (TCG) [Tcg03]. It defines and promotes open standards for hardware-enabled trusted computing and security technologies, including hardware building blocks and software interfaces, across multiple platforms, peripherals, and devices.

TCG specified technology enables more secure computing environments without compromising functional integrity, privacy, or individual rights.

1.2.3 Component Software Platform

A *component software platform* is a type of computing platform that supports the execution of software components. The concept of software component builds on prior theories of software objects, software architectures, software frameworks and software design patterns, and the extensive theory of object-oriented programming and object-oriented design of all these. It is expected that a software component, like the idea of a hardware component, can be ultimately made interchangeable and reliable. The component software platform can play as a concrete middleware layer inside a mobile computing platform.

1.3 Motivations of Trust Management for Mobile Computing Platforms

Trust management is becoming an important issue for the mobile computing platforms. Firstly, mobile commerce and mobile services hold the yet unfulfilled promise to revolutionize the way we conduct our personal, organizational and public business. Some attribute the problem to the lack of a mobile computing platform that all the players may trust enough. Nowadays, it is very hard to build up a long-term trust relationship among manufactures, service/application providers and mobile users. This could be the main reason that retards the further development of mobile applications and services.

On the other hand, new mobile networking is raising with the fast development of mobile ad hoc networks (MANET) and local wireless communication technology. It is more convenient for mobile users to communicate in their proximity to exchange digital information in various circumstances. However, the special characteristics of the new mobile networking paradigms introduce additional challenges on security [ZhH99, YZV03]. This introduces special requirements for the mobile computing platform to embed trust management mechanisms for supporting trustworthy mobile communications.

More interesting and strange phenomena is that current mobile systems are designed based on the assumptions that a) the user trusts his/her device totally; or b) the user has to trust a service provider; or c) the user has no choice except using some manufacture's device in order to deploy some mobile applications or mobile services. Generally, the systems are not designed considering the users' trust preferences or standards, thus the systems produced are hard to be finally accepted by the end users.

In addition, the growing importance of the third party software in the domain of component software platforms introduces special requirements on trust. Particularly, the system's trustworthiness is varied due to component joining and leaving. How to manage trust in such a platform is crucial for embedded devices, such as mobile phones.

All of the above problems influence further development of mobile applications and services targeting at different areas, such as mobile enterprise, mobile

networking and mobile computing. The key reason is that we lack a trust management solution for mobile computing platforms.

1.4 Research Issues

In this section, we identify the research issues that are worthy of special efforts.

1.4.1 Methodology Development

First of all, we need methodologies to model trust for an intended purpose, and thus to build up a trusted system by applying the model. There are various methodologies that can be applied for solving different issues. Some trust models are based on sound technologies, e.g. PKI [Per99]. A big number of trust models are built up targeting at some trust properties, such as reputations, recommendations and risk [XiL03, LiS05]. Many trust models have been constructed for various computing paradigms such as GRID computing, ad hoc networks, peer-to-peer networks, and multi-agent systems, etc. [ZWW05, ThB06, LVW04]. In those models, some are computational, others are linguistic or graphic. For example, in [Jos99], Subjective Logic is used to assess trust values based on the triplet representation of trust. In [Man98], linguistic trust metrics are used for reasoning out trust with provided rules. In the context of the "Web of Trust," many trust models are built upon a graph where the resources/entities are nodes and trust relationships are edges, such as in [ReS98].

Although a variety of trust models are available, it is still not well understood what fundamental criteria the trust models must follow. Without a good answer to this question, the design of trust models is still at an empirical stage [SYH06]. Current work focuses on concrete solutions in special systems. Particularly, there are no feasible trust modeling methodologies available that can be applied into the mobile computing platform in a common way. Thus, we lack general instructions when we are designing, analyzing and developing a trusted mobile system.

1.4.2 Personal Trusted Device

The second issue is how to provide a personal trusted device (PTD) for the mobile users. This is because establishing a trusted mobile environment requires a trusted mobile computing platform as a corner stone.

Three problems need to be solved herein. Firstly, we need a trusted mobile computing platform capable of trustworthy code interaction in an efficient way. The mobile device platform layer should provide essential security services, such as authenticated booting, encryption service, secure storage, privacy support and digital rights management. Trusted Computing Group (TCG) set its goal towards maintaining the privacy of the platform owner while providing a ubiquitous interoperable mechanism to validate the identity and integrity of a computing platform [Tcg03]. However, due to the small size of mobile devices and a different computing platform structure from personal computers, this technology needs to be adapted for hand-held products. This work is still on-going at the TCG.

Secondly, the platform's middleware runtime layer should provide a mechanism to support trustworthy cooperation of multiple software components, thus ensuring that device applications can be executed as required and expected regarding system dependability, security and adaptability. The device platform should be capable of monitoring system performance and adaptively arranging limited device resources (such as power, memory, network capability and CPU) in order to fulfill trust requirements of different applications and services even in a dynamically changed context. The platform could further overcome system threats with a preventable and tolerant measure. It should be adaptive to the changes in the software running environment due to new components execution and old one's deletion, as well as the changes raised outside the device. Based on our literature study, little work is conducted in this area, especially for the mobile devices [ZMZ05].

Thirdly, we need the PTD to ensure the trust relationship established for an intended purpose and sustained until the purpose is fulfilled. This is crucial for trusted mobile commerce and services. In one word, the PTD will become the mobile user's trust authority in mobile commerce and communications. The TC platform was proposed to improve the trust between users and their devices. The TCG's TC technology ensures this through a set of hardware and software mechanisms. However, current work on TC platform lacks solutions for trust sustaining among computing platforms, so that trust loyalty might be broken after a period of time. This problem may influence the completion of a trustworthy transaction or service conducted between two platforms.

1.4.3 Trusted Mobile Communications

The third issue is how to provide trusted mobile communications in both a dynamically changed public domain and an organization's enterprise domain.

In the public domain, future mobile networking is most possibly in an ad hoc style randomly organized by mobile devices. Operation in an ad hoc network introduces new security problems. The ad hoc networks are generally more prone to physical security threats. The possibility of eavesdropping, spoofing, denial-of-service, and impersonation attacks increases. But security approaches used for the fixed networks are not feasible due to the salient characteristics of the ad hoc networks. New threats, such as attacks raised from malicious nodes, are hard to defend against. New security mechanisms are needed to adapt to the special characteristics of the ad hoc networks. A trust evaluation based security solution we developed could be an effective approach for data protection, secure routing and other network activities [YZV03]. It can also cooperate with a TC platform based solution to provide improved trust in a MANET. Such combined solutions are seldom studied in the literature and practiced in industry.

Mobile peer-to-peer computing has emerged as a significant paradigm for providing distributed services, in particular collaboration for content sharing and distributed computing. Generally, a mobile P2P system consists of a decentralized and self-organizing network of autonomous devices that interact as peers. Each peer acts as both client and server to share its resources with other peers. However, this computing paradigm suffers from several drawbacks that obstruct its wide adoption. Lack of trust between peers is one of the most serious issues, which causes security challenges in the P2P systems. Building up trust collaboration among the system peers is a key issue to overcome.

In the enterprise domain, trust plays a key role in the context of VPN. However, providing advanced trust into mobile VPN networks has proven to be problematic. Generally, mobile enterprise networking is composed of various devices provided by different vendors with different security support. Trust management of confidential digital contents at different enterprise devices in different domains (e.g. a public networking domain and a virtual private network domain that are either trusted or distrusted) is a new challenge worth special efforts.

The new paradigm of mobile communications introduces additional requirements of trust management for the mobile computing platforms.

1.5 Research Objectives

The objective of this research is to study trust management for mobile computing platforms. This dissertation concentrates especially on the following aspects:

- Study the state-of-the-art of trust management and specify its emerging trends in order to propose trust management solutions for the mobile computing platforms;
- Develop suitable methodologies for trust management that can instruct the designing, analyzing and developing of a trusted mobile system;
- Embed dynamic support of trust into the mobile computing platform in order to ensure trusted mobile communications and collaboration; integrate trust evaluation and management into traditional security technologies;
- Propose and develop a solution of autonomic trust management for the component software platform targeting at runtime trust support.

1.6 Contribution to Knowledge

Trust management is an important issue for the mobile computing platforms. This dissertation studies methodologies and mechanisms of providing a trustworthy computing platform for mobile devices. What is more, we seek solutions to support trusted communications and collaboration among those platforms in a distributed and dynamic system. The main contributions of each publication are summarized below.

- Publication 1 provides a comprehensive review of trust perspective, trust modeling, trust evaluation and trust management. Based on the study on the state-of-the art, it identifies emerging and future trends.
- Publication 2 presents a conceptual architecture towards establishing a trusted mobile environment. The contributions of this paper are a) specifying the architecture of a trusted mobile environment; b) developing the conceptual architecture and explaining key motivations behind the location of

every element in the architecture; and c) evaluating the architecture by applying it into a mobile peer-to-peer system.

- Publication 3 introduces a methodology to bridge the domains of trust in mobile computing and communications. This methodology benefits the system analysis and design for finding trust issues and identifying security problems. The concrete approaches for bridging the trust gaps among domains instruct how to seek a concrete solution regarding trust management for mobile computing platforms.

- Publication 4 presents a mechanism to sustain the trust among computing platforms on the basis of the root trust (RT). This mechanism extends the trust model from static to dynamic. Thus, it develops the notion of using trust management not only for trust assessment but also for trust sustainability. The proposed mechanism could be applied in many real applications to ensure trusted services and communications. It could work as an extension of future TC platform to support various applications with better flexibility.

- Publication 5 introduces a perspective of building up trust collaboration in a P2P system based on the TC platform. It is a concrete application of the trust sustainability mechanism. Through a uniform TC platform compatible P2P device architecture − Trusted Collaboration Infrastructure (TCI), many security challenges can be overcome. It is a concrete example of integrating both a trust evaluation solution and a pure security solution. In addition, the proposed TCI based P2P system can also support automatic network resource management as well as privacy. It provides a series of platform mechanisms for people to select in order to realize personal protection. Therefore, it broadly supports trust collaboration in P2P networks.

- Publication 6 presents another concrete application of the trust sustainability mechanism. It illustrates how to apply this mechanism into the context of a mobile VPN in order to provide a trusted mobile enterprise solution. This solution realizes trust management on mobile enterprise devices at both VPN connection and disconnection.

- Publication 7 develops a trust management solution for the dynamically changing component software middleware platform based on the trust expression using Subjective Logic. It defines a formal trust model to specify, evaluate, set up and ensure trust relationships amongst platform entities. Based on this trust model, an architecture of autonomic trust management can be designed to adopt a number of algorithms to enable the trust assessment at runtime and autonomic trust management on the basis of auto-selection of trust control mechanisms. The trust at the system runtime is better addressed with the above emerging properties.

- Publication 8 proposes a methodology for the trust control mode prediction and selection aiming at autonomic trust management for a component software platform. The simulation results show that this method is effective

for predicting and selecting the suitable trust control modes. It also helps improving the control mode configurations, especially when there is no solution from the prediction. In addition, this method is flexible for supporting any system entity's autonomic trust management.

1.7 Outline of the Thesis

The rest of the thesis consists of several independent, but closely related chapters.

Chapter 2 presents the work published in Publication 1. We introduce the state-of-the-art in conceptualizing trust, trust modeling, trust evaluation and trust management and identify emerging trends in this area.

Chapter 3 summarizes the work published in Publication 2 and Publication 3. We develop a conceptual architecture based research method and a methodology to bridge different domains of trust in order to solve trust related issues in mobile computing and communications. This work contributes to the research issue described in Section 1.3.1.

Chapter 4 reports the results published in Publication 4, Publication 5 and Publication 6. We proposed a TC platform based mechanism for trust sustainability among platforms. This mechanism can be further applied into the P2P systems to achieve trust collaboration among peer computing platforms. It can also be used to realize trust management in mobile enterprise networking. The work described in this Chapter contributes to the research issues described in Section 1.3.2 and Section 1.3.3, respectively.

Chapter 5 reports the results about the autonomic trust management for a component software platform. We develop a formal trust model to specify, evaluate, set up and ensure trust relationships that exist among platform entities. We further present an autonomic trust management architecture that adopts a number of algorithms for trust assessment and maintenance during component execution. In addition, we propose a methodology for trust control mode prediction and selection based on an adaptive trust control model in order to support autonomic trust management. The results reported in this Chapter contribute to the research issue described in Section 1.3.2. The publications associated with this Chapter are Publications 7, Publication 8, [YaP07], [Yan07], [Yan06] and [YaP07a].

Chapter 6 concludes the thesis and proposes future work.

2 BACKGROUND

We introduce the state-of-the-art in conceptualizing trust, trust modeling, trust evaluation and trust management and identify emerging trends in this area.

2.1 Factors of Trust

We can find various definitions of trust in the literature [BoH91, Gam90, MDS95, GrS02, McC03, McC00, CKW03, Mui03, ALR04, LJT04, Den93, FaC05]. In summary, it is widely understood that trust itself is a comprehensive concept, which is hard to narrow down. Trust is subjective because the level of trust considered sufficient is different for each entity. It is the subjective expectation of the trustor on the trustee related to the trustee's behaviors that could influence the trustor's belief. Trust is also dynamic as it is affected by many factors that are hard to monitor. It can further develop and evolve due to good experience about the trustee. It is sensitive to be decayed caused by bad experience. More interestingly, from the digital system point of view, trust is a kind of assessment on the trustee based on a number of trust referents, e.g. competence, security, and reliability, etc.

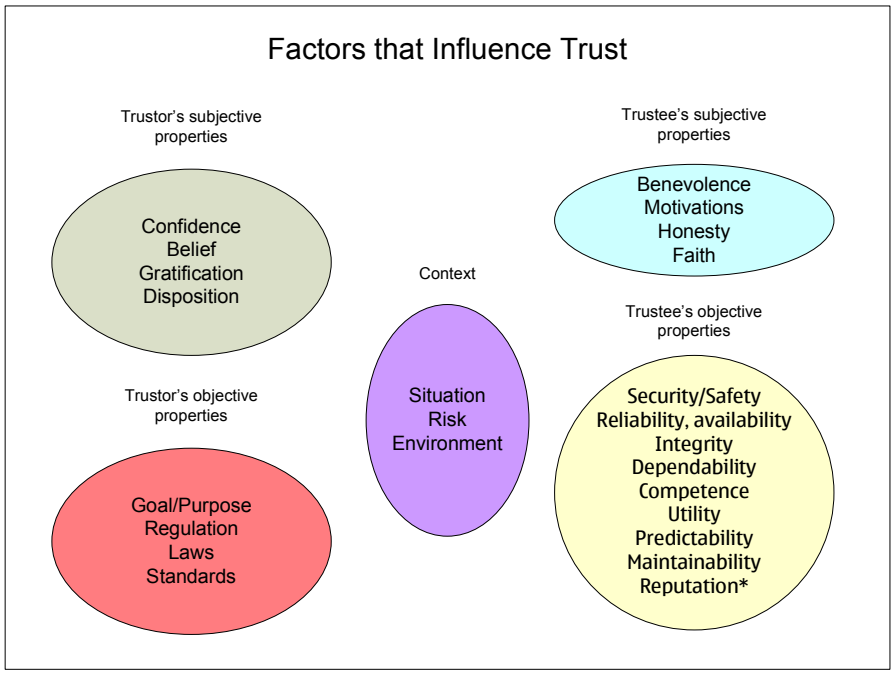

Figure 2.1: Factors that influence trust

Based on our study, we hold the opinion that trust is influenced by a number of factors. Those factors can be classified into five viewpoints, as shown in Figure 2.1:

- Trustee's objective properties, such as trustee's security and dependability. In particular, reputation is a public assessment of the trustee considering its earlier behavior.
- Trustee's subjective properties, such as trustee's honesty.
- Trustor's subjective properties, such as trustor's disposition to trust.
- Trustor's objective properties, such as the standards or policies specified by the trustor for a trust decision.
- Context that the trust relationship resides in, such as specified situation, risk, etc. The context contains any information that can be used to characterize the situation of involved entities [Dey01].

From the digital system point of view, we pay more attention to the objective properties of both the trustor and the trustee. For social human interaction, we consider more the trustee's subjective and objective properties and the trustor's subjective properties. For economic transactions, we study more about the context for risk management. The context of trust is a very important factor that influences trust. It also specifies the background or situation where trust exists.

2.2 Trust Modeling

2.2.1 Characteristics of Trust

Despite the diversity among the existing definitions of trust, and despite that a precise definition is missing in the literature, there is a large confluence on what properties the concept of trust satisfies. We report here the most significant characteristics of trust, which play as the important guidelines for trust modeling.

a) *Trust is directed:* trust is an oriented relationship between the trustor and the trustee.
b) *Trust is subjective:* Trust is inherently a personal opinion. According to [GrS00], trust is considered a personal and subjective phenomenon that is based on various factors or evidence, and that some of those may carry more weight than others.
c) *Trust is context-dependent:* In general, trust is a subjective belief about an entity in a particular context.
d) *Trust is measurable:* Trust values can be used to represent the different degrees of trust an entity may have in another. "Trust is measurable" also provides the foundation for trust modeling and computational evaluation.
e) *Trust depends on history:* This property implies that past experience may influence the present level of trust.
f) *Trust is dynamic* [GrS00]*:* Trust is usually non-monotonically changed with time. It may be refreshed periodically, may be revoked, and must be able to adapt to the changing conditions of the environment in which the trust decision was made. Trust is sensitive to many factors, events, or changes of context. In order to handle this dynamic property of trust, solutions should take into account the notion of learning and reasoning. The dynamic adaptation of the trust relationship between two entities requires a sophisticated trust management approach.

g) *Trust is conditionally transferable:* Information about trust can be transmitted/received along a chain (or network) of recommendations.
h) *Trust can be a composite property:* "trust is really a composition of many different attributes: reliability, dependability, honesty, truthfulness, security, competence, and timeliness, which may have to be considered depending on the environment in which trust is being specified" [GrS00]. Compositionality is an important feature for making trust calculations.

2.2.2 Trust Models

The method to specify, evaluate, set up and ensure trust relationships among entities for calculating trust is referred to as a trust model. Trust modeling is the technical approach used to represent trust for the purpose of digital processing.

A trust model aims to process and/or control trust digitally. Most of the modeling work is based on the understanding of trust characteristics and considers some factors influencing trust. Current work covers a wide area including ubiquitous computing, distributed systems (e.g. P2P systems, ad hoc networks, GRID virtual organization), multi-agent systems, web services, e-commerce (e.g. Internet services), and component software. For example, trust models can be classified into various categories according to different criteria, as shown in Table 2.1.

Table 2.1: Taxonomy of trust models

Classification criteria	**Categories**		**Examples**
Based on modeling method	Models with linguistic description		[BFL96] and [TaT98]
	Models with graphic description		[ReS98]
	Models with mathematic description		[XiL04] and [SYH06]
Based on modeled contents	Single-property modeling		[JHK05], [XiL04], and [SYH06]
	Multi-property modeling		[ZMZ05], [WaV05], and [Publication 7]
Based on the expression of trust	Models with binary rating		
	Models with numeral rating	Continuous rating	[Mau96] and [XiL04]
		Discrete rating	[LJT04]
Based on the dimension of trust expression	Models with single dimension		[Mau96] and [XiL04]
	Models with multiple dimensions		[ThB06] and [Jos99]

Although a variety of trust models are available, it is still not well understood what fundamental criteria trust models must follow. Without a good answer to this question, the design of trust models is still at an empirical stage [SYH06]. Current work focuses on concrete solutions in special systems. We would like to advocate

that a trust model should reflect the characteristics of trust, consider the factors that influence trust, and thus support trust management in a feasible way.

It is widely accepted that trust is influenced by reputations (i.e. the public evidence on the trustee), recommendations (i.e. a group of entities' evidence on the trustee), the trustor's past experience and context (e.g. situation, risk, time, etc.). Most of the work has focused on trust valuation or level calculation without any consideration of ensuring or sustaining trust for the fulfillment of an intended purpose. We still lack comprehensive discussions with regard to how to automatically take an essential action based on the trust value calculated. Except the context, all the above items are assessed based on the quality attributes of the trustee, the trust standards of the trustor and the context for making a trust or distrust conclusion. A number of trust models have considered and supported the dynamic nature of trust. So far, some elements of context are considered, such as time, context similarity, etc. The time element has been considered in many pieces of work, such as [WaV05b, XiL04]. However, no existing work gives a common consideration on all factors that influence trust, as shown in Figure 2.1.

2.3 Trust Management

As defined in [GrS00], trust management is concerned with: collecting the information required to make a trust relationship decision; evaluating the criteria related to the trust relationship as well as monitoring and re-evaluating existing trust relationships; and automating the process. We think that extension of this definition is needed in order to manage trust in a computing platform. We will discuss more about this in Chapter 5.

Various trust management systems have been described in the literature. One important category is reputation based trust management systems. Trust and reputation mechanisms have been proposed in various fields such as distributed computing, agent technology, GRID computing, component software, economics and evolutionary biology. Examples are FuzzyTrust system [SHZ05], the eBay user feedback system (www. ebay.com) [ReZ02], Trustme - a secure and anonymous protocol for trust [SiL03], IBM propagation system of distrust [GuK04], PeerTrust model developed by Li Xiong and Ling Liu [XiL04], Eigen-Trust algorithm [KSG03], TrustWare - a trusted middleware for P2P applications [LiS05], a scheme for trust inference in P2P networks [LSB03], a special reputation system to reduce the expense of evaluating software components [Her01] and Credence developed at Cornell - a robust and decentralized system for evaluating the reputation of files in a peer-to-peer file-sharing system [WaS05].

Reputation-based trust research stands at the crossroads of several distinct research communities, most notably computer science, economics, and sociology. As defined by Aberer and Despotovic [AbD01], reputation is a measure that is derived from direct or indirect knowledge on earlier interactions of entities and is used to assess the level of trust an entity puts into another entity. Thus, reputation based trust management (or simply reputation system) is a specific approach to trust management. Using a reputation system, Alice establishes trust in Bob based on the experience that Alice and others have had with Bob.

Reputation schemes can be classified in two different categories depending on what sort of reputation they utilize. Global reputation is the aggregation of all

available assessments by other entities that have had interactions with a particular entity, and thus it has an n-to-1 relationship. On the other hand, local reputation of an entity is each entity's own assessment based on past history of interaction with a particular entity, thus it is a 1-to-1 relationship.

2.4 Trust Evaluation Mechanisms

Trust evaluation is a technical approach of representing trustworthiness for digital processing, in which the factors influencing trust will be evaluated by a continuous or discrete real number, referred to as a trust value. Embedding a trust evaluation mechanism into trust management is necessary for providing trust intelligence in future computing platforms.

Trust evaluation is the main aspect in the research for the purpose of digitalizing trust. A number of theories about trust evaluation can be found in the literature. For example, Subjective Logic was introduced by Jøsang [Jos01]. It can be used for trust representation, evaluation and update. It has a sound mathematical foundation in dealing with evidential beliefs rooted in Shafer's theory and the inherent ability to express uncertainty explicitly. Trust valuation can be calculated as an instance of Opinion in Subjective Logic. An entity can collect the opinions about other entities both explicitly via a recommendation protocol and implicitly via limited internal trust analysis using its own trust base. It is natural that the entity can perform an operation in which these individual opinions can be combined into a single opinion to allow a relatively objective judgment about other entity's trustworthiness. It is desirable that such a combination operation shall be robust enough to tolerate situations where some of the recommenders may be wrong or dishonest. Another situation with respect to trust valuation includes combining the opinions of different entities on the same entity together using a Bayesian Consensus operation; aggregation of an entity's opinions on two distinct entities with logical AND support or with logical OR support. A real description and demo can be found in [SL].

In particular, Subjective Logic is a theory about opinion that can represent trust. Its operators mainly support the operations between two opinions. It doesn't consider context support, such as time based decay, interaction times or frequency; trust standard support like importance weights of different trust factors. Concretely, how to generate opinions on recommendations based on credibility and/or similarity and how to overcome attacks on trust evaluation are beyond the theory of SL. These need to be further developed in real practice.

Fuzzy Cognitive Maps (FCM) could be regarded as a combination of Fuzzy Logic and Neural Networks [Kos86]. In a graphical illustration, FCM seems to be a signed directed graph with feedback, consisting of nodes and weighted arcs. Nodes of the graph stand for the concepts that are used to describe the behavior of the system and they are connected by signed and weighted arcs representing the causal relationships that exist between the concepts. We will introduce the FCM in details in Chapter 5 (Section 5.8) before applying it to propose an adaptive trust control model.

A FCM can be used for evaluating trust. In this case, the concept nodes are trustworthiness and the factors that influence trust. The weighted arcs represent influencing relationships among those factors and the trustworthiness. The FCM is convenient and practical for implementing and integrating trustworthiness and its

influencing factors [CFP03]. In addition, some work makes use of the fuzzy logic approach to develop an effective and efficient reputation system [SHZ05].

Semiring is introduced in [ThB06]. The authors view the trust inference problem as a generalized shortest path problem on a weighted directed graph *G(V, E)* (*trust graph*). The vertices of the graph are the users/entities in the network. A weighted edge from vertex *i* to vertex *j* corresponds to the *opinion* that the trustor has about the trustee. The weight function is $l(i, j): V \times V \rightarrow S$, where *S* is the opinion space. Each opinion consists of two numbers: the *trust* value, and the *confidence* value. The former corresponds to the trustor's estimate of the trustee's trustworthiness. On the other hand, the confidence value corresponds to the accuracy of the trust value assignment. Since opinions with a high confidence value are more useful in making trust decisions, the confidence value is also referred to as the *quality* of the opinion. The space of opinions can be visualized as a rectangle *(ZERO_TRUST, MAX_TRUST)*×*(ZERO_CONF, MAX_CONF)* in the Cartesian plane (*S = [0, 1]* × *[0, 1]*). Using the theory of semirings, two nodes in an ad hoc network can establish an indirect trust relation without previous direct interaction. The semiring framework is also flexible to express other trust models.

Generally, two versions of the trust inference problem can be formalized in an ad hoc network scenario. The first is finding the trust-confidence value that a source node *A* should assign to a destination node *B*, based on the intermediate nodes' trust-confidence values. Viewed as a generalized shortest path problem, it amounts to finding the generalized distance between nodes *A* and *B*. The second version is finding the most trusted path between nodes *A* and *B*. That is, find a sequence of nodes that has the highest aggregate trust value among all trust paths starting at *A* and ending at *B*. In the trust case, multiple trust paths are usually utilized to compute the trust distance from the source to the destination, since that will increase the evidence on which the source bases its final estimate. The first problem is addressed with a "distance semiring", and the second with a "path semiring". They use two operators to combine opinions: One operator (denoted $\otimes$) combines opinions along a path, i.e., *A*'s opinion for *B* is combined with *B*'s opinion for *C* into one indirect opinion that A should have for *C*, based on *B*'s recommendation. The other operator (denoted $\oplus$) combines opinions across paths, i.e., *A*'s indirect opinion for *X* through path *p1* is combined with *A*'s indirect opinion for *X* through path *p2* into one aggregate opinion. Then, these operators can be used in a general framework for solving path problems in graphs, provided they satisfy certain mathematical properties, i.e., form an algebraic structure called a semiring.

[SYH06] presents an information theoretic framework to quantitatively measure trust and model trust propagation in ad hoc networks. In the proposed framework, trust is a measure of uncertainty with its value represented by entropy. The authors develop four axioms that address the basic understanding of trust and the rules for trust propagation. Based on these axioms two trust models are introduced: entropy-based model and probability-based model, which satisfy all the axioms.

[XiL04] presents five trust parameters used in PeerTrust, namely, feedback a peer receives from other peers, the total number of transactions a peer performs, the credibility of the feedback sources, a transaction context factor, and a community context factor. By formalizing these parameters, a general trust metric is presented. It combines these parameters in a coherent scheme. This model can be applied into

a decentralized P2P environment. It is effective against dynamic personality of peers and malicious behaviors of peers.

2.5 Emerging Trends

Herein, we provide insights about emerging trends in trust management.

2.5.1 An Integrated Solution

Theoretically, there are two basic approaches for building up a trust relationship. We name them as a 'soft trust' solution and a 'hard trust' solution [Publication 2]. The 'soft trust' solution provides trust based on trust evaluation according to subjective trust standards, facts from previous experiences and history. The 'hard trust' solution builds up trust through structural and objective regulations, standards, as well as widely accepted rules, mechanisms and sound technologies (e.g. PKI and TC platform). Possibly, both approaches are applied in a real system. They can cooperate and support with each other to provide a trustworthy system. 'Hard trust' provides a guarantee for the 'soft trust' solution to ensure the integrity of its functionality. 'Soft trust' can provide a guideline to determine which 'hard trust' mechanisms should be applied and at which moment. It provides intelligence for selecting a suitable 'hard trust' solution.

An integrated solution is expected to provide a trust management framework that applies both the 'hard trust' solution and the 'soft trust' solution. This framework should support data collection and management for trust evaluation, trust standards extraction from the trustor (e.g. a system user), and experience or evidence dissemination inside and outside the system, as well as a decision engine to provide guidelines for applying different 'hard trust' mechanisms for trust management purposes. How to design a light-weight and effective trust management framework is a practical challenge, especially for the device platforms with limited resources.

In addition, how to store, propagate and collect information for trust evaluation and management is seldom considered in the existing work, thus making it an issue in real implementation.

Apart from the above, the question of human-machine interaction with regard to trust is an interesting topic that requires special attention. Human-machine interaction is crucial to transmit user's trust standards to the machine and the machine needs to provide its assessment of trust to its user and explain it in a friendly way.

2.5.2 Autonomic Trust Management

There is a trend that all the processing for trust management is becoming autonomic. This trend benefits from the digitalization of trust. Since trust relationships are dynamically changed, this requires trust management to be context-aware and intelligent to handle the context changes. In addition, the trust model itself should be adaptively adjusted in order to match and reflect the real

system situation. Context-aware trust management is a developing research topic and adaptive trust model optimization could be an emerging research opportunity.

2.5.3 Cross-Domain Benefits

We can estimate that trust management will not only benefit security, but also other properties of the system, such as privacy, usability, dependability and Quality of Service. Combining trust management with other management technologies (e.g. resource management, power management, identity management, risk management and fault management, etc.) or applying it into other areas could produce cross-domain benefits. The outcome system would be more intelligent and provide better performance.

2.6 Summary

This Chapter introduced former work on trust presented in the literature. We summarized the factors influencing trust and the characteristics of trust. Trust modeling is actually based on these. Quite a number of research projects on trust modeling, evaluation and management have been conducted in the area of distributed systems and e-commerce. A more extended survey of the literature is provided in Publication 1. A number of recent achievements model trust using a mathematical approach. Thus, it is possible to conduct digital trust management for the emerging technologies, including mobile computing platforms.

However, current work mostly focuses on theoretic study. It lacks experience on how the proposed approaches work in practice. Most of existing solutions are special system driven. They have not considered how to provide a generic solution, and thus make trust management benefit not only one specific system, but also other digital systems. In addition, current work lacks effort to study human-machine or human-platform interaction for the purpose of trust management, which is one of the most important issues that require special considerations in practice, especially for devices with limitations.

Regarding the emerging trends, we believe an integrated solution is promising and could combine a traditional security solution with newly developed trust evaluation based management together. This integrated solution should handle trust management in an automatic way and cooperate with other technologies to offer better system performance.

The understanding gained from the literature study instructs our work towards solving special issues of trust regarding the mobile computing platform.

3 RESEARCH METHODOLOGIES

We develop a conceptual architecture based research method and a methodology to bridge different domains of trust in order to solve trust related issues in mobile computing and communications.

3.1 A Conceptual Architecture Based Research Method

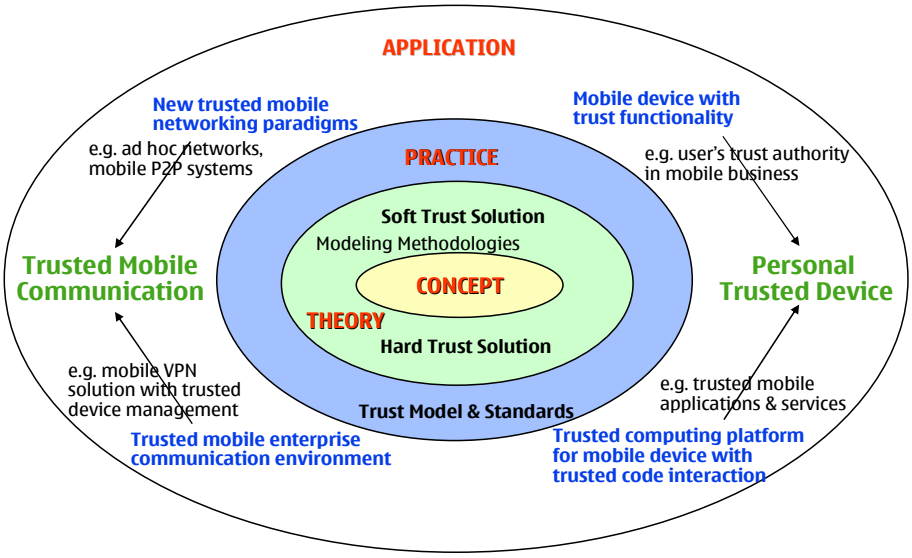

Figure 3.1: A conceptual architecture of trusted mobile computing

Trust is a very complicated phenomenon attached to multiple disciplines and influenced by many subjective and objective factors. Therefore, it is essential to define a conceptual architecture to clarify target scenarios, thus narrow down our study and make it easy to focus on the concrete issues in different aspects of trusted mobile computing.

We propose an onion structure, as shown in Figure 3.1. It is composed of four circles. A concept circle is at the core of the onion. This circle defines a series of concepts about trust, its derivatives and its related terms. Based on the working concepts, theories and modeling methodologies can be built upon, forming a theory circle. Outside the theory circle, there is a practice circle. The practice circle applies theories and methodologies into various trust models and standards for supporting trust in real applications and systems that form its outer circle: an application circle.

3.1.1 Definitions – Concept Circle

Due to multiplicity of meanings associated with the word 'trust' and its derivatives, it is essential to establish a certain set of definitions that can be used throughout one

intended purpose. A series of working definitions of trust and its derivatives build up the core part of the proposed architecture.

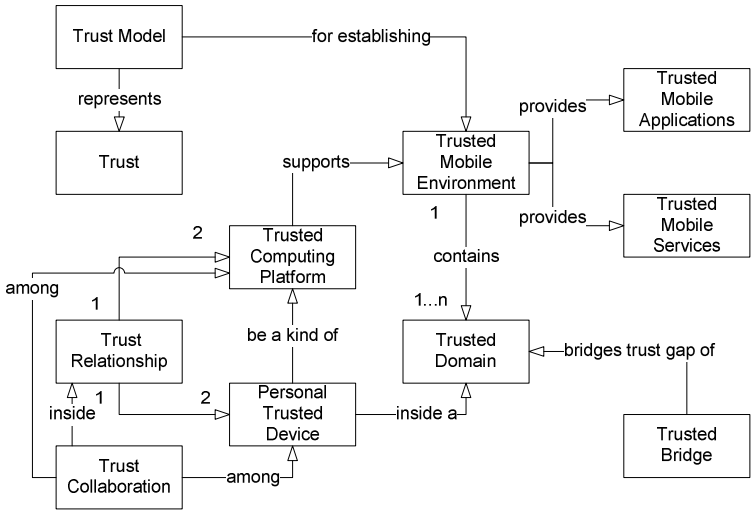

Figure 3.2: Relationships of concepts

Towards our purpose, we can define a number of concepts, for examples, *trust, trust modeling, trust model, trusted mobile environment, trusted domain, trusted bridge, Trusted Computing platform, Personal Trusted Device,* and *trust relationship*. Their relationships are depicted in Figure 3.2.

3.1.2 Theory and Modeling Methodology – Theory Circle

Theoretically, there are two basic approaches for building up a trust relationship: a 'soft trust' solution and a 'hard trust' solution, as shown in Figure 3.1.

There are various ways of trust modeling targeting at different scenarios. Trust modeling is crucial for applying trust management approaches into mobile systems. Regarding the trust analysis and modeling, we need special methodologies. In order to apply a 'hard trust' solution, it is essential to analyze default trust relationships among system entities and study potential changes of the trust relationships after the system initiation. Thereby, trust solutions to overcome trust gaps in the underlying system could be designed based on the existing regulations, standards, and widely accepted rules and technologies. For the 'soft trust' approach, it is important to clarify the border of entities or domains among which the trust evaluation is needed. Based on timely trust evaluation, decisions could be made to apply appropriate mechanisms for ensuring the trust relationship.

3.1.3 Trust Models and Standards – Practice Circle

Based on the theory and methodology established, we can design the trust models for mobile applications and systems. Thus corresponding standards can be made in industry to support real applications.

For example, the trust model of the Trusted Computing (TC) platform as defined in [Tcg03] is that the basic trust of every entity is rooted from sound hardware security – a 'hard trust' solution. Based on this root trust, trust can be further built on local OS and application software through authenticated booting. Trust on a remote platform can be built based on the attestation of expected platform configurations.

3.1.4 Mobile Applications and Systems – Application Circle

This circle considers mobile applications and systems. We divide the application circle into four directions, as shown in Figure 3.1. Each direction implies motivations for potential business.

- A trusted mobile computing platform with trusted code interaction: This direction aims at providing a TC platform for mobile devices in order to support mobile applications and services in a secure and trustworthy way. It also ensures trustworthy device internal operation in a dynamically changed context (e.g. the middleware component software platform).
- A mobile device with trust functionalities: This direction tries to provide the trust functionalities into mobile devices (e.g. a DRM solution). With the new trust features, the devices will become more intelligent in interacting with the users and behave as their trust advisors to help them make trust related decisions in mobile communications and personal business. With the TC platform support and the embedded trust functionalities, the future mobile device could become a Personal Trusted Device (PTD) that could be the user's trust authority for various usages.
- New trusted mobile networking paradigms: This direction aims to support new mobile networking paradigms, such as MANETs and mobile P2P systems. These new networking paradigms hold special characteristics that introduce new challenges to security and trust.
- A trusted mobile enterprise communication environment: This direction is towards building up a trusted environment for mobile working and enterprise management, e.g. a mobile Virtual Private Network (VPN) solution with trusted device management. With the trusted mobile networking in both public domain and enterprise domain, a Trusted Mobile Communication (TMC) environment could be supported.

3.2 A Methodology to Bridge Different Domains of Trust

3.2.1 Introduction

Trust is such a subjective and dynamic concept that different entities can hold different opinions on it even while facing the same situation [Gam90]. Based on different trust perception, different trusted domains can be formed in the area of mobile computing and communications. For example, a trusted domain that contains a security element (such as a smart card) and its issuer is formed if the issuer trusts the security element due to its tamper resistance.

Taking a mobile computing or communication system as an example, we can find many cases in which a system is actually formed by a number of trusted domains. Communication is actually conducted among and across those domains. Inside each trusted domain, the trust relationships exist among the domain entities. But among the domains, a significant problem may arise from the fact that the domains deficient of trust must cooperate in order to provide a complete service even though they may not share the same concept of trust. Specifically, frequent security problems among those domains may be caused by the deficiency of trust among domains. This deficiency is probably one of the major barriers that prevent the proliferation of mobile computing, communications and services.

Based on the above analysis, a mobile system can be modeled as consisting of a number of trusted domains. Inside a trusted domain, the trust relationship exists. While among the domains, trust is lacking, and needs special technologies to build up. This trust domain based modeling methodology falls into the theory circle. It is also related to the practice circle because a new trust model for a mobile system is generated based on the methodology.

There are several methods to bridge the trust gap among trusted domains, e.g. legal, contractual, and risk management based solutions. We believe that technology is one of the most important methods. In the following, a technical method to bridge the trust gap is provided. We propose a methodology to analyze and bridge the trusted domains.

3.2.2 Methodology

In any mobile system, we can always specify the system as a number of trusted domains. The communications, transactions or collaboration are actually conducted among those domains. Inside each trusted domain, the domain entities trust the domain components according to their trust statements, for whatever reasons they find appropriate. Among the trusted domains, it is expected that trust must be usually created and constructed logically and rationally. We propose a methodology to analyze the trust domains and to create the trusted bridge, effectively enabling the domains to form a complete solution. A *trusted bridge* is a component or a set of components that is/are trusted by more than one domain. Therefore such component(s) can work as a bridge to establish trust or bridge trust gaps among those domains. The proposed methodology is summarized as follows.

1. Model the mobile system by separating it into a number of trusted domains formed by different entities.
2. Analyze each domain in order to extract the trust statements and list existing domain components.
3. For each pair of disjoint domains that must trust each other for the purpose of a given intention, seek a bridging solution that can satisfy both domains.
4. Form the trusted bridge by finding or creating a suitable component (or components), or by establishing bridging domains, depending on needs.

There are several approaches to identify the bridging solution and to introduce the trusted bridge, depending on the trust statements within the trusted domains as well as on non-technical limitations. A more extended discussion of the approaches is provided in Publication 3. Following is a short list of those.

a. Use an existing component

The system analysis itself may lead to the discovery that there is already an existing component that may be trusted by more than one domain and thus can behave as the trusted bridge.

b. Create a new component

If the bridging component does not exist, it is possible to create it. Some components may conform to only one trust statement so that they require the statement to be identical in both domains. Some components may conform to more than one statement so that they can be used to bridge the domains with different statements.

c. Create a separate domain

If there is no potential component that may satisfy the domains (e.g. the trust statements are significantly different), the solution may be to create a separate domain such that its domain components fulfill statements from both disjoint domains. Such a domain may share existing or new components with all the domains it is bridging. We call the created domain a bridging domain.

3.3 An Illustration

In this section, we evaluate the conceptual architecture's expressiveness and advantages by applying it into a mobile peer-to-peer system. We also apply the methodology described in Section 3.2 to solve the trust deficiency among different trusted domains.

Mobile peer-to-peer computing has emerged as a significant paradigm for providing distributed services, in particular collaboration for content sharing and distributed computing. Generally, a mobile P2P system consists of a decentralized and self-organizing network of autonomous devices that interact as peers. Each peer acts as both client and server to share its resources with other peers. However, this computing paradigm suffers from several drawbacks that obstruct its wide adoption. Lack of trust between peers is one of the most serious issues, which causes security challenges in the P2P systems. Building up trust collaboration among the system peers is a key issue to overcome.

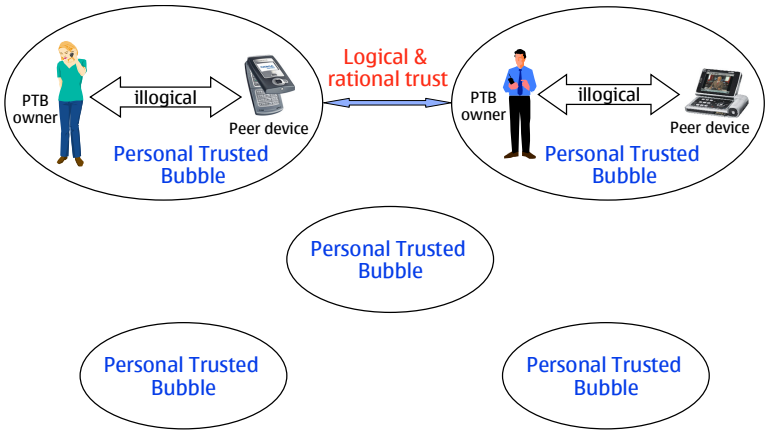

Figure 3.3: Trust model of a mobile peer-to-peer system

We presented a Trusted Collaboration Infrastructure (TCI) for a mobile P2P system in [Publication 5], which will be described in Chapter 4. This infrastructure combines both a 'soft trust' solution and a 'hard trust' solution in order to support the trust collaboration among the mobile peers. We applied the concepts defined in the concept circle. By using the modeling methodology introduced in Section 3.2, the system can be modeled as a number of trusted domains – trust bubbles. As shown in Figure 3.3, each peer device is independently located inside a personal trusted bubble: the basic unit that represents a peer. Inside the bubble, the owner of the peer device trusts the device based on the TC platform technology. The device is responsible for communication with other peers. Among bubbles, logical and rational trust relationships should be attested. In order to build up the trust collaboration among the bubbles, we applied both a 'hard trust' solution and a 'soft trust' solution.

The 'hard trust' solution uses an improved TC platform technology that can ensure the trust sustainability. A trust relationship can be established between a trustor device and a trustee device based on the device platform attestation and the registration of trust conditions at the trustee device's TC platform components. With the TC platform components inside the peer device, a trustee device can ensure trust sustainability according to pre-defined conditions. The conditions are approved by both the trustor device and the trustee device at the time of trust establishment. They can be further enforced through the use of the pre-attested TC platform components at the trustee device until the intended collaboration is fulfilled. The TC platform components are built upon a secure hardware chip, which is very hard to be broken, even by the trustee itself. This solution falls into approach (a) – use existing component: the TC platform component trusted by both peer devices, especially after the remote attestation.

Regarding the 'soft trust' solution, the trust evaluation mechanisms embedded in each peer device can anticipate potential risks and make the best decision on any security related issues in the P2P communications and collaboration. The trust evaluation results can help generating feasible conditions for sustaining the trust relationship. This mechanism is very helpful in fighting against attacks raised by malicious peers that hold a correct platform certificate and valid data for trusted platform attestation. This solution falls into approach (b) – create new component: a trust evaluation module to support partial trust.

Through defining the basic concepts and using the modeling methodology, we model the system trust and clarify where problems exist. By making use of the technologies specified in the theory circle, we can establish trusted mobile communications in a mobile P2P system. The presented architecture facilitates our work in understanding, analyzing and solving the trust issues. The methodology to bridge different domains of trust further helps us to seek the technical solutions for trust collaboration among peer devices.

3.4 Summary

Trust is playing and will continuously play an important role in the mobile domain. In order to support further success of mobile communications, applications and services, it is significant to study trust issues for providing a trusted mobile computing platform. This platform aims to offer trusted interaction among the

mobile computing platforms and their internal components, to support trust collaboration among platforms and to provide trust-intelligence to the users of mobile devices.

We presented a conceptual architecture to clarify the structure of trust issues in different aspects. Based on this architecture, we can specify a number of key motivations. A more extended discussion of these motivations is provided in Publication 2. We also introduced a methodology to bridge the domains of trust in mobile computing and communications. We proposed that any system analysis and design could include modeling the system as composed of different trusted domains that may reflect various reasons for trust. Inside the domain, the trust relationships have been established, while among domains the trust is deficient. Thus in this way, it is easier to identify the trust and security problems hidden inside the system. In order to bridge the trust gaps, we proposed three approaches that can be used to develop a trust bridging solution. Furthermore, we apply our methodologies into a mobile P2P system to demonstrate their applicability, expressiveness and advantages.

4 TRUSTED COMPUTING PLATFORM BASED SOLUTIONS FOR MOBILE COMPUTING

We proposed a TC platform based mechanism for trust sustainability among platforms. In this Chapter, this mechanism is further applied into P2P systems to achieve trust collaboration among peer computing platforms. We also show how to use the mechanism to realize trust management in mobile enterprise networking.

4.1 Motivation

With the rapid growth of internetworking and electronic commerce, trust plays a crucial role in cyberspace in order to provide various digital services [ChW03]. However, establishing a trust relationship in cyberspace is more complicated than in the social world. This is because communication in the cyberspace relies not only on human beings but also on digital components. Moreover, it is more difficult to accumulate accurate information for trust purpose in digital communications than in a social context. Generally, it is reasonably easy to initiate trust based on many existing technologies and structural regulations, but hard to sustain the trust during the fulfilment of a whole service or an intended purpose.

Trust in digital information society, called digital trust, introduces two major challenges. The first one is establishing trust between users and their devices (e.g., PC and mobile phone) that is necessary to start the communication. With the increasing complexity of device computing platforms and various kinds of software running on them, it is very difficult for the users to verify that their devices work properly. Trusted computing (TC) platform has been proposed to solve this problem [Tcg03, ELM03].

Another particular challenge is that trust has to be sustained over time. For example, trustor A's trust on trustee B at one moment does not mean A can or will trust B at the next moment. The trust relationship built at the beginning of the communication should be maintained at least until the service is completed. It is essential to monitor and control the trust relationship in order to sustain trust for the final success of the service. This Chapter will mainly focus on solving this particular challenge that has not been yet properly explored.

4.2 Related Work

As introduced in Chapter 2, there is a large range of existing work on trust in information technology. The concept of trust is defined in various ways in the literature. It is widely understood that the trust itself is a comprehensive concept, which is hard to narrow down. It is subjective because the level of trust considered sufficient is different for each entity. The trust is also dynamic as it is affected by many factors and easily influenced by a bad experience.

Many people believe that some metrics should be defined to state various degrees of trust [XiL04]. A number of computational trust models were presented by different authors in [Man00, LMA02, JoK98, Jos99, Jos01]. These models evaluate trust based on the trustor's direct or indirect experience. However, these models only pay attention to the influence of previous knowledge on trust, but ignore future changes that may destroy the established trust. Thereby, they lack support for cases that demand trust for a longer period of time.

Also a lot of work has been done on trust management [JoT03, BIK03, GrS03]. Trust management systems provide trust assessments based on some trust root, e.g. policy assertion and trust specifications. However, they focus on how to evaluate trust and have not considered a mechanism to sustain the trust relationship in order to support the fulfilment of an intended purpose.

Another important topic in the literature is Digital Rights Management (DRM) [Pas02]. It deals with client-side control of the usage of digital information. The trust model of a traditional DRM solution can be described as a reference monitor (generally a software application) existing at a user's system for controlling usage of disseminated digital information in lieu of an information issuer. Not only does DRM pose significant technical and operational challenges but none of the existing DRM solutions considers how to sustain the trust relationship.

The results presented in this Chapter are highly related to the work on trusted computing platforms [Dav02, Vau03, ELM03, BaS03, Fel03]. All work on TC platforms is based on hardware security and cryptography for providing a root trust (RT) module at a digital computing platform. However, as described in the next section, current work on the TC platform still lacks support on trust sustaining over the network. This is the key problem that we try to solve. We believe that trust management in cyberspace should assure not only trust assessment, but also trust sustainability.

4.3 Problem Statement

The intention of this section is to clarify one of the problems of the current TC platform [Tcg03]. In the TC platform trust is built upon a root trust, which is enforced by sound technologies, and realized through secure hardware [Dav02, Vau03]. Every time a computer is reset, the root trust module steps in, checks itself, and then verifies the OS loader (e.g. BIOS) before letting the boot-up continue. Through checking the integrity metrics of different components, the OS loader is assumed to verify the operating system, then, the operating system is assumed to verify every piece of software, and so on. A remote computing platform can be trusted by challenging its integrity metrics, verifying and comparing them with expected values that represent components that are trusted enough to perform the intended purpose. If compared values match the expected values, trusted interaction with the remote computing platform can be commenced. Anomalous metrics indicate that the platform is not operating as expected and further communication with the platform should be reconsidered.

However, trust in the remote computing platform neither necessarily remains intact for an extended period of time, nor does it remain intact after hardware or software configuration changes. Actually, as the trusted computing platform is built up during system boot, the root trust module can only verify OS within the

previously identified configurations, thus failing to verify trust for any newly added hardware or software components. This also means that trust on the remote platform cannot be sustained even though the platform could have been trusted at some moment. Therefore, one disadvantage of the current TC platform paradigm is that it does not provide a dynamic solution and is thus unable to sustain its protection in a changeable environment.

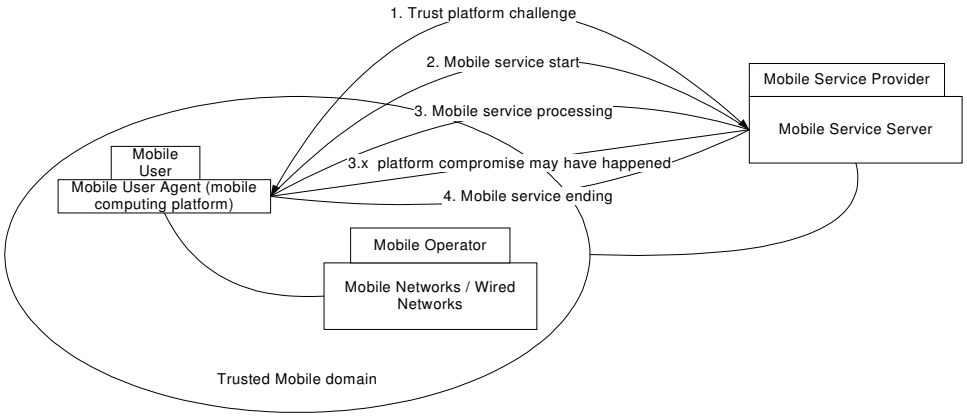

Figure 4.1: An example of trust in mobile services

In order to illustrate the problem, we take a mobile service as an example. The term 'mobile service' can be vaguely defined as a service that is provided to mobile users via mobile computing platforms [Publication 3]. Specifically, the mobile computing platforms such as mobile phones are considered to be the user agents of mobile services. As shown in Figure 4.1, a mobile phone already has the trust relationship with its operator through the existence of SIM (Subscriber Identity Module) and relevant authentication methods. A mobile Service Provider (SP) stays out of the usual trust relationship. Based on the TC platform technology, it is possible for both the mobile SP server and the mobile computing platform to verify each other as trusted computing platforms at the beginning of the service. However, as time passes, the SP server cannot guarantee that trust is sustained since hardware or malicious software can be installed in the mobile computing platform.

One simple solution is to periodically re-challenge the remote platform. This however requires frequent communications between the remote platform and the server, which are neither feasible nor economical in the mobile environment. Further, the remote device bears the burden of frequent and unnecessary computationally-intensive operations. Still, this method may be subject to some forms of the man-in--middle attacks [HaD05].

4.4 A Mechanism for Trust Sustainability among Platforms

In order to overcome the above problem, we introduce a mechanism for sustaining trust among computing platforms. We first present a trust formula used in the mechanism, and then the RT module on which the mechanism is based.

4.4.1 Trust Form

The proposed mechanism uses the following trust formula: "Trustor A trusts trustee B for purpose P under condition C based on root trust R". The difference between this formula and others is in the element C - conditions to trust. The element C is defined by A to identify the rules for sustaining the trust for purpose P, the conditions and methods to get signal of distrust behaviours, as well as the mechanism to restrict any changes at B that may influence the trust relationship. The root trust R is the foundation of A's trust on B and its sustainability. Since A trusts B based on R, it is rational for A to sustain its trust on B based on R controlled by the conditions decided by A. The R is an existing component trusted by entities located at different domains of trust. Thus, it can be used for bridging trust deficiency for building up a long term trust relationship among the computing platforms. This formula makes it possible to extend one-moment trust over a longer period of time.

4.4.2 Root Trust Module

The proposed mechanism is based on a root trust (RT) module that is also the basis of the TC platform. The RT module could be an independent module embedded in the computing platform. It could also be a build-in feature in the current TC platform's Trusted Platform Module (TPM) and related software [Vau03].

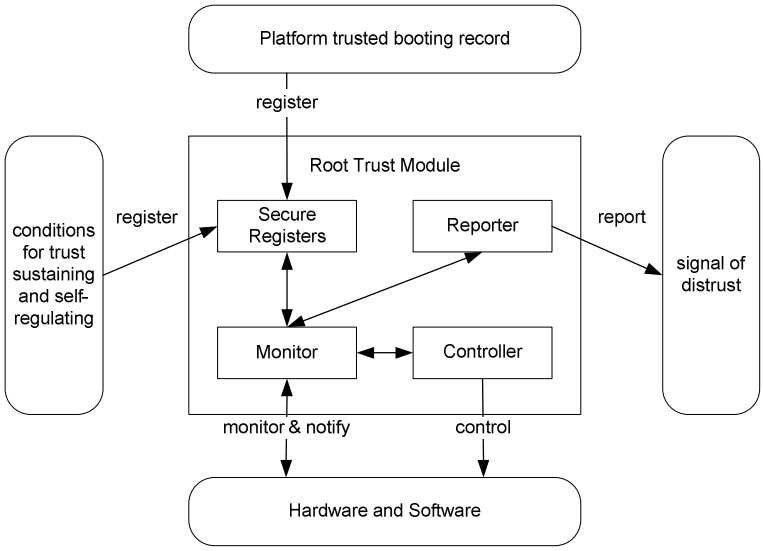

Figure 4.2: Root trust module

The RT module at the trustee is most possibly a hardware-based security module. It has capability to register, protect and manage the conditions for trust sustaining and self-regulating. It can also monitor any computing platform's change including any alteration or operation on hardware, software and their configurations. The RT module is responsible for checking changes and restricting

them based on the trust conditions, as well as notifying the trustor accordingly. Figure 4.2 illustrates the basic structure of this module.

There are two ways to know the platform changes. One is an active method, that is, the platform hardware and software notify the RT module about any changes for confirmation. The other way is a passive method, that is, the RT module monitors the changes at the hardware and the software. At the booting time, the RT module registers the hash codes of each part of platform hardware and software. It also periodically calculates their run-time values and checks if they are the same as those registered. If there is any change, the RT module will check with the registered trust conditions and decide which measure should be taken.

The RT module can be designed and implemented inside a secure main chip of the mobile computing platform. The secure main chip provides a secure environment to offer security services for the operating system (OS) and software applications and some security enforcement mechanisms (e.g. system integrity booting and device identity) [PaL]. It also provides cryptographic functions and a secure storage. The RT module functionalities are implemented by a number of protected applications. The protected applications are small applications dedicated to performing security critical operations inside the secure environment. They have strict size limitations and more resemble function libraries. The protected applications can access any resources in the secure environment. They can also communicate with normal applications in order to offer security services. New protected applications can be added to the system at any time. The secure environment software controls loading and execution of the protected applications. Only the signed protected applications are allowed to run.

4.4.3 Mechanism for Trust Sustainability

As postulated, the trust relationship is controlled through the conditions defined by the trustor, which are executed by the RT module at the trustee on which the trustor is willing to depend. The reasons for the trustor to depend on the RT module at the trustee can be various. Herein, we assume that the RT module at the trustee can be verified by the trustor as its expectation for some intended purpose and cannot be compromised by the trustee or other malicious entities later on. This assumption is based on the work done in industry and in academy [Dav02, Vau03, ELM03, BaS03].

As shown in Figure 4.3, the proposed mechanism comprises the following procedures.

a) Root trust challenge and attestation to ensure the trustor's basic trust dependence at the trustee in steps 1- 2;
b) Trust establishment by specifying the trust conditions and registering them at the trustee's RT module for trust sustainability in steps 3-6;
c) Sustaining the trust relationship through the monitor and control of the RT module in steps 7-8;
d) Re-challenge the trust relationship if necessary when any changes against trust conditions are reported.

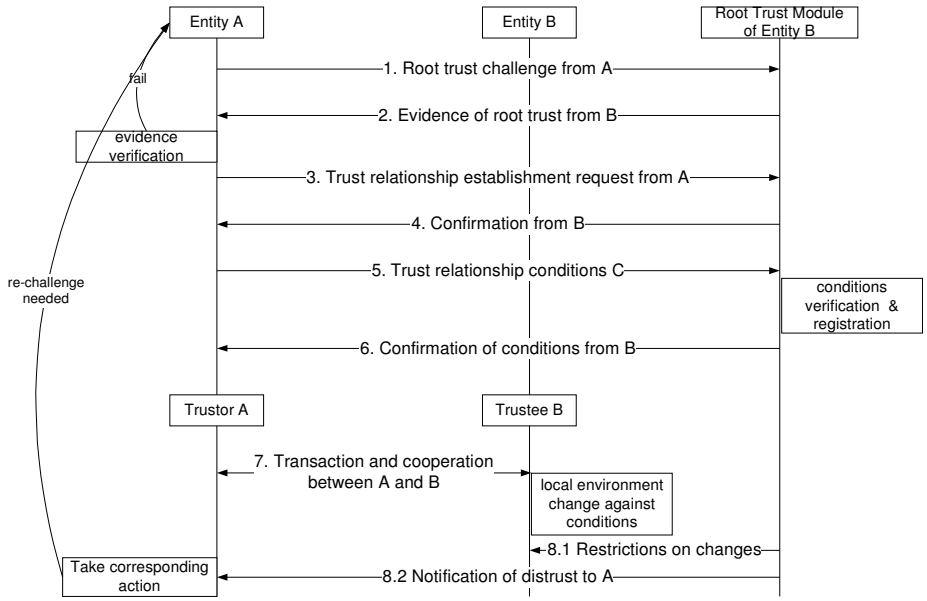

Figure 4.3: Protocol for trust sustainability

As we can see in the above protocol, trust is based on the trustor's dependence on the RT module. Although the RT module is located at the trustee, its execution for trust maintenance and sustainability is based on the agreed conditions and rules approved by both the trustee and trustor at the time trust is established.

Notably, step 8.2 is an option, which is applied based on the negotiation of the trust relationship establishment. If the requirement of distrust notification is not presented in the trust relationship conditions, the step 8.2 will not be applied. If there is such a requirement, corresponding technologies or mechanisms for information protection should be further clarified by the trustor and be agreed by both the trustor and the trustee in the step 5 and 6. We can make use of public key encryption or secret key encryption to protect the notification. We can also use some existing protocol (e.g. SKIP) to implement the step 8.2.

In order to defend against the attacks raised by information capturing and destroying, the trustee can wait for the trustor's response after it sends the notification. If there is no response within an expected period, the trustee can take corresponding measures, which are specified in the trust relationship conditions and approved by both the trustor and the trustee at the trust relationship establishment. Trust is a subjective concept. Based on the conditions, the trustor has reasons to sustain its trust on the trustee until the fulfillment of the intended purpose. The corresponding measures specified in the trust relationship conditions for any distrust situation and any abnormal situation are thought as trusted by the trustor.

Optionally, the trust conditions could be certified through a Trusted Third Party (TTP). The trustor can send its private policies to the TTP. The TTP combines the private policies and general policies together in order to generate the conditions tailored for the trustor. It then issues the certified conditions to the trustor. Typical example conditions are "the integrity of the platform is not changed" and "additional software can be installed only if it is certified by a specified trusted

authority". The trust conditions can be described using XML (eXtensible Markup Language). A detailed design is provided in [YaC03].

Taking a mobile ad hoc network (MANET) as an example, it is possible to ensure the trustworthy communications among a number of nodes for an intended purpose (e.g. routing from a source node to a destination) by imposing identical trust conditions (e.g. the integrity of the platform is not changed and extra software applications are restricted to install) in the node computing platforms. At the beginning, the initial trust relationships are established based on the Root Trust module challenge and attestation between each communication node pairs. If the trust attestation fails, the trust relationship can not be built up. After the initial trust relationships have been established, the RT module can ensure the trust relationships based on the requirements specified in the trust conditions. Particularly, if the RT module detects any malicious behavior or software at the trustee device, it will reject or block it. If the RT module finds that the node platform is attacked, the trustor node platform could be notified. In addition, a trust evaluation mechanism can be embedded into the RT module or its protected components in the node computing platform in order to evaluate other nodes' trustworthiness based on experience statistics, the reputation of the evaluated node, node policies, an intruded node list and transformed data value. Any decision related to security (e.g. a secure route selection) should be based on trust analysis and evaluation among network nodes. Detailed discussion about this 'soft trust' solution is provided in [YZV03]. In particular, the trust evaluation results can greatly help in designing suitable trust conditions for trust sustainability during node communications. It could also help in selecting the most trustworthy node in the ad hoc networking.

In the following two sections, we will present two use cases and illustrate how this mechanism benefits solving trust issues in P2P systems and mobile enterprise networking, respectively.

4.5 Trust Collaboration in P2P Systems

Peer-to-peer computing has emerged as a significant paradigm for providing distributed services, in particular collaboration for content sharing and distributed computing. However, this computing paradigm suffers from several drawbacks that obstruct its wide adoption. Lack of trust between peers is one of the most serious issues, which causes a number of security challenges in P2P systems. Publication 5 studies the feasibility of building up trust collaboration in the P2P systems based on the mechanism introduced in Section 4.4. We introduce a Trusted Collaboration Infrastructure (TCI) for peer-to-peer computing devices. Through applying the TCI, trust collaboration can be established among distributed peers through the control of the TC platform components. Based on analysis, we conclude that the TC platform technology is a promising solution that can overcome many P2P security challenges and thus realize trust collaboration among P2P peers.

4.5.1 Trusted Collaboration Infrastructure (TCI) for P2P

Based on the trust model presented in Section 3.3, we further propose a trusted collaboration infrastructure (TCI) for the P2P system. In this infrastructure, each peer device is TC platform compatible and has an internal architecture as shown in Figure 4.4.

There are three layers in this architecture. A platform layer contains TC platform components specified in [Tcg03] (e.g. TPM) and an operating system that is booted and executed in a trusted status, which is attested and ensured by the TC platform components.

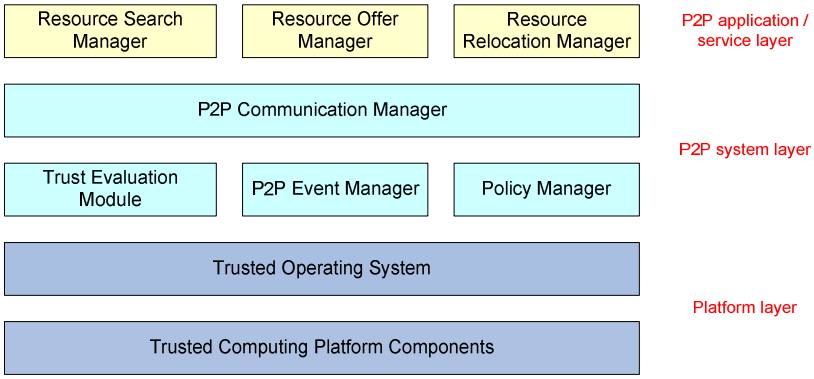

Figure 4.4: Architecture of P2P peer device in TCI

A P2P system layer contains common components required for trusted P2P communications. Those components are installed over the platform layer and ensured running in a trusted status. This is realized through trusted component installation and alteration-detection mechanism supported by the platform layer. A communication manager is responsible for various P2P communications (e.g., the communications needed for the P2P system joining and leaving). A trust evaluation module is applied to evaluate the trust relationship with any other peer before any security related decision is made. The trust evaluation module cooperates with a policy manager and an event manager in order to work out a proper trust evaluation result. The policy manager registers various local device policies regarding P2P applications and services. It also maintains subjective policies for trust evaluation. The event manager handles different P2P events and cooperates with the trust evaluation module in order to conduct proper processing.

A P2P application/service layer contains components for P2P services. Taking resource sharing as an example, this layer should contain components such as a resource-search manager, a resource-offer manager and a resource-relocation manager. The resource-search manager is responsible for searching demanded resources in the P2P system. The resource-offer manager provides shared resources according to their copyright and usage rights. The offered resources could be encapsulated through the encryption service of the TC platform. The encryption offered by the encryption service is attached to some special configurations as mandatory requirements for decryption. The resource-relocation manager handles remote resource accessing and downloading. The downloaded resources are firstly checked with no potential risk, and then stored at the local device.

Like the system layer, all the components in this layer are attested by the platform layer (e.g. trusted OS) as trusted for execution. Any malicious change could be detected and rejected by the platform layer.

4.5.2 Trust Collaboration

Trust collaboration is defined as interaction; communication and cooperation are conducted according to the expectation of involved entities. For example, the shared contents in the P2P systems should be consumed and used following the content originator's or right-holder's expectation without violating any copyrights. In peer-to-peer systems, the trust collaboration requires autonomous control on resources at any peer. The trust collaboration in the proposed P2P system infrastructure fulfills the following trust properties.

- Each peer device can verify that another peer device is working in its expected status.

Building up on the TC platform technology, each peer device with the underlying architecture can ensure that every component on the device is working in a trusted status. It can also challenge any other device and attest that it is working in its expected status, as shown in Figure 4.5 (step 1 and 2). This is done through digitally certifying the device configurations.

Two levels of certifying are provided. One is certifying the OS configuration. On this level, the system uses a private key only known by the RT module to sign a certificate that contains the configuration information, together with a random challenge value provided by a challenger peer device. The challenger provided that it generates the random challenge value can verify that the certificate is valid and up-to-date, so it can know what the device's OS configuration is.

In many cases, there is a strong desire to certify the presence and configuration of application programs. Application configurations are certified through a two-layer process. The RT module certifies that a known OS version is running and then the OS can certify the applications' precise configuration.

- Trust relationship established at the beginning of the collaboration between peers can be sustained until the collaboration is fulfilled for some intended purpose based on trust conditions.

As shown in Figure 4.5, the trust relationship can be established between a trustor device and a trustee device based on the trust platform attestation (step 1-2) and the registration of trust conditions at the trustee device's TC platform components, e.g. the RT module (step 3-4). Through applying the mechanism described in Section 4.4, a trustee device can ensure the trust sustainability according to pre-defined conditions (step 5-6). The conditions are approved by both the trustor device and the trustee device at the time of trust establishment. They can be further enforced through the use of the pre-attested TC platform components at the trustee device until the intended collaboration is fulfilled.

One example of the trust conditions is shown in Figure 4.6. The example trust conditions specify that a) upgrading of P2P applications is only allowed for the 'TrustIssuer' certified applications; b) the changes for any hardware components in the computing platform is disallowed; and c) any changes for the rest of software in the computing platform are disallowed. All of above conditions can be ensured

through integrity check by the Root Trust module based trusted computing components and secure software installation mechanism that can verify the certificate of a software application before the installation.

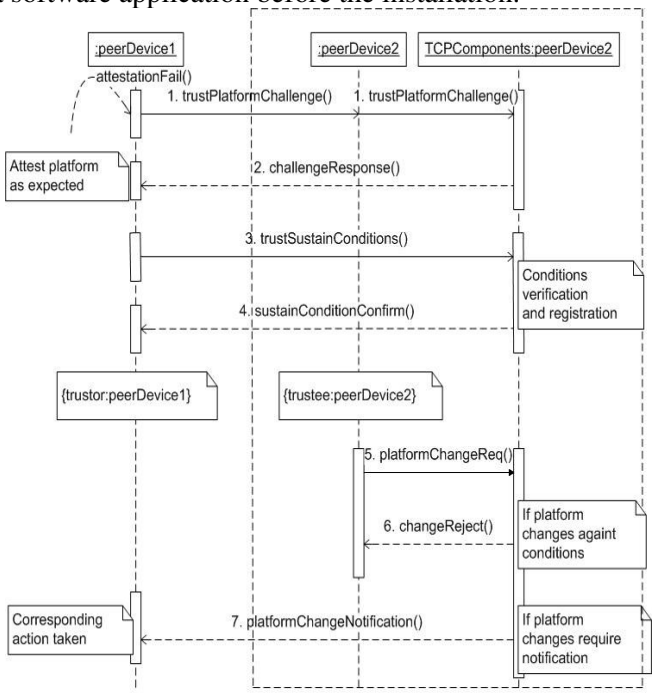

Figure 4.5: Trust collaboration in P2P system

```
<?xml version="1.0" encoding="UTF-8"?>
<trustConditions identity='P2PApp'>
 <Restrict>
  <operation>
   <name> upgrade </name>
   <runner> owner </runner>
   <exception>
        <condition> certified </condition>
        <issuer> 'TrustIssuer' </issuer>
   </exception>
  </operation>
 </Restrict>

 <PlatformRunEnvironment>
  <Hardware name="all">
   <Restrict>
    <operation>
     <name> change </name>
    </operation>
   </Restrict>
  </Hardware>

  <Software name="rest">
   <Restrict>
    <operation>
     <name> change </name>
    </operation>
   </Restrict>
  </Software>
 </PlatformRunEnvironment>
</trustConditions>
```

Figure 4.6: An example of trust conditions for trust collaboration in P2P system

Through applying this mechanism, there are ways to automatically control the remote environment as trusted. Optionally, it is also possible to inform the trustor peer about any distrust behavior of the trustee according to pre-defined conditions (step 7). Therefore, it is feasible for the trustor peer to take corresponding measures to confront any changes that may affect the continuation of trust for the purpose of a successful P2P service.

- Each peer can manage the trust relationship with other peers and therefore it can make the best decision on security issues in order to reduce potential risks.

Based on the trust evaluation mechanisms [YZV03, FDK02, KSP01, Jos01, Kos86, JIB05, LVW04] embedded in the trust evaluation module, each peer can anticipate potential risks and make the best decision on any security related issues in the P2P communications and collaboration. The trust evaluation results can help generating feasible conditions for sustaining the trust relationship. In particular, the trust evaluation is conducted in the expected trust environment, thus the evaluation results are generated through protected processing. This mechanism is very helpful in fighting against attacks raised by malicious peers that hold a correct platform certificate and valid data for trusted platform attestation.

- Resources are offered under expected policies.

This includes two aspects. One is that the resources are provided based on copyright restrictions. Those contents that cannot be shared should not be disclosed to other peers. The other is that the resources are provided with some limitations defined by the provider. The encryption services offered by the TC platform can cooperate with the resource-offer manager to provide protected resources and ensure copyrights and usage rights. Regarding the encryption services, refer to [Publication 5].

- Resources are relocated safely and consumed as the provider expects.

The trust attestation mechanism offered by the TC platform can support the resource-relocation manager to attest that the downloaded contents are not malicious code. In addition, the resources are used in an expected way, which is specified according to either copyrights or pre-defined usage restrictions. This can be ensured by the TC platform encryption mechanism before and during content consuming.

- Personal information of each peer is accessed under expected control.

The resource-offer manager in the proposed architecture can cooperate with the TC platform components to encapsulate the personal information based on the policies managed by the policy manager. Only trusted resource-search manager can access it. The trusted resource-search manager is an expected P2P application component that can process the encapsulated personal information according to the pre-defined requirements specified by the personal information owner.

With the TC platform components in the TCI, any P2P device component can only execute as expected and process resources in the expected ways. Furthermore, with the support of trust evaluation and trust sustainability, the peers could collaborate in the most trustworthy way.

4.5.3 Deployment

The TCI is a device based infrastructure. One essential requirement for deploying the TCI is that the device is TC platform compatible. Once the TCI is deployed in every P2P system device or part of the system devices, it can automatically support the system trust collaboration through the components and mechanisms provided by the TCI.

Due to the above reason, the TCI can be flexibly integrated into any distributed system, such as a peer-to-peer system or a Grid computing system. It can support peer based or node based trust collaboration in a dynamically changed decentralized system. For different purposes, different components can be downloaded and installed at the application / service layer. The preferred software middleware platform for the TCI could be component-based software architecture that interfaces with the TC platform functionalities and provides necessary mechanisms to support components' execution in a trustworthy way. We will further discuss how to ensure a trustworthy component software platform in Chapter 5.

More promisingly, the P2P paradigm will be a valuable extension of current enterprise networking, especially for a mobile enterprise networking. The TCI can also be applied for enterprise device management in a P2P scenario. It can protect confidential resources from being accessed by unauthorized peers. In addition, it ensures the enterprise peer device to behave as expected in a P2P networking even though it is disconnected with the enterprise network. This is the result of [Publication 6] and will be presented in Section 4.6.

4.5.4 Remarks

TC platform technologies are under development in the industry and academy in order to provide more secure and better trust support for future digital devices. The TC platform tries to solve existing security problems by hardware trust. Although it is still in its infancy and may be vulnerable to some hardware attacks [Hua02], it has advantages over many software-based solutions.

We introduced a perspective of building up trust collaboration in a P2P system based on the TC Platform, which is discussed in a more extended way in Publication 5. Through a uniform TC platform compatible P2P device architecture – TCI, many security challenges can be overcome. In addition, the proposed TCI based P2P system can also support automatic network resource management as well as privacy. It provides a series of platform mechanisms for people to select for the purpose of personal protection. Therefore, it can support trust collaboration in the P2P systems that lack trust. It has potential advantages over other solutions; especially when the TCG standard is deployed and many industry digital device vendors (e.g. Microsoft, IBM, HP, Intel, etc.) will offer compatible hardware and software in the future.

4.6 Trust Management in Mobile Enterprise Networking

How to manage trust in mobile enterprise networking among various mobile devices is problematic for companies using mobile enterprise solutions. This

section presents a trust management system in an enterprise virtual private network (VPN) also based on the mechanism for trust sustainability.

4.6.1 Problem Statement

Trust plays a key role in the context of virtual private networking (VPN). However, providing advanced trust into VPN networks has proven to be problematic in mobile domains. This is mainly caused by two reasons.

First, current VPN networks lack the means to enable trust among mobile computing platforms from different manufactures. For example, an application can be trusted by Manufacture A's devices but may not be recognized by Manufacture B's devices. Moreover, from a VPN management point of view, it is difficult to manage the security of a large number of computing platforms. This problem is more serious in mobile security markets. Since different mobile device vendors provide different security solutions, it is difficult or impossible for mobile enterprise operators to manage the security of diverse devices in order to successfully run security-related services.

Second, no existing VPN system ensures that the data or components on a remote user device can only be controlled according to the enterprise VPN operator's security requirements, especially during VPN connection and disconnection. The VPN server is unaware as to whether the user device platform can be trusted or not although user verification is successful. Especially, after the connection is established, the device could be compromised, which could open a door for attacks. Particularly, data accessed and downloaded from the VPN can be further copied and forwarded to other devices after the VPN connection has been terminated. The VPN client user could conduct illegal operations using various ways, e.g. disk copy of confidential files and sending emails to other people. Nowadays, the VPN operators depend on the loyalty of the VPN client users to address this potential security problem. In addition, a malicious application or a thief that stole the device could also try to compromise the integrity of the device.

Regarding the problems described above, no good solutions could be found in the literature. Related work did not consider the solutions of the problems described above [Her99, WSC88, Reg03, ChM02]. For example, a trust management solution based on KeyNote for IPSec in [BIK02] could ensure trust during VPN connection in the network-layer. A security policy transmission model was presented to solve security policy conflicts for large-scale VPN in [SLW03]. But the proposal could not help in solving the trust sustainability after the VPN connection and disconnection. Past work focused on securing network connection, not paying much attention to the necessity to control VPN terminal devices [HAM05]. In addition, security or trust policy of the VPN operator should be different regarding different VPN client devices, which raises additional requirements for trust management in enterprise networking.

The following sections present a trust management system based on a virtual private network in order to enhance trust in mobile enterprise networking. Our focus will be on how to support confidential content management and how to overcome the diversity support of security in different devices manufactured by different vendors. The discussion is based on the mechanism for trust sustainability among computing platforms. We illustrate how to apply this mechanism into mobile virtual private networks.

4.6.2 Trust Management in Mobile VPN

We provide a solution for enhancing trust in a mobile VPN system. In this case, a VPN trust management server is the trustor, while a VPN client device is the trustee. A trust relationship could be established between them. The VPN trust management server identifies the client device and specifies the trust conditions for that type of device at the VPN connection. Thereby, the VPN client device could behave as the VPN operator expects. Additional trust conditions could be also embedded into the client device in order to control VPN-originated resources (e.g. software components or digital information originated from the VPN). Therefore, those resources could be managed later on as the VPN operator expects even if the device's connection with the VPN is terminated. Even though the VPN client device is not RT module based, the trust management server can identify it and apply corresponding trust policies in order to restrict its access to confidential information and operations.

```
<?xml version="1.0" encoding="UTF-8"?>
<trustConditions deviceStatus='VPNDisconnect'>
 <Restrict>
  <operation>
   <name> print </name>
   <runner> owner </runner>
   <targets>
        <fileResource> EnterpriseIntranet </fileResource>
   </targets>
  </operation>

  <operation>
   <name> forward </name>
   <runner> owner </runner>
   <targets>
        <fileResource> EnterpriseIntranet </fileResource>
   </targets>
   <method> any </method>
  </operation>
 </Restrict>

 <PlatformRunEnvironment>
  <Hardware type="all">
   <Restrict>
    <operation>
     <name> change </name>
    </operation>
   </Restrict>
  </Hardware>

  <Software type="all">
   <Restrict>
    <operation>
     <name> change </name>
     <runner> owner </runner>
    </operation>
   </Restrict>
  </software>
 </PlatformRunEnvironment>
</trustConditions>
```

Figure 4.7: An example of trust conditions for trust management in a mobile enterprise networking

A simple example of trust conditions for trust management in a mobile enterprise networking is illustrated in Figure 4.7. The example trust conditions specify that a) printing and forwarding files achieved from the enterprise Intranet are disallowed when the device disconnects the Intranet; b) the changes for any hardware components in the computing platform are disallowed; and c) the changes by the device owner for any software in the computing platform are disallowed, too. All of above conditions can be ensured through the Root Trust module based trusted computing technology.

System structure

The proposed mobile VPN system comprises a plurality of client devices, gateways and servers, part or all of which are RT module based platforms. The system provides the management of RT based platforms in the network, and enables verification among the platforms.

Figure 4.8 illustrates the proposed mobile VPN system used in mobile networks (e.g., GSM networks). In the figure, the mobile VPN users use their mobile devices to connect to their enterprise VPN and access VPN services (e.g., emails, file sharing, etc.). The mobile devices connect to the Internet through some wireless access technology (e.g., WLAN). The VPN trust management server manages the trust policies for the mobile devices. Notably, the server may reside inside the VPN or in the Internet (protected by a firewall). It instructs how the mobile devices can use their RT module and for what operations. Meanwhile, the server is able to push/pull the trust conditions to the mobile devices in a secure, fast and convenient way (e.g., through SSL). With the help of the server, the mobile devices can more securely and easily set up trust relationships with other trusted entities including other client devices and VPN network devices. Therefore, they are able to easily set up and maintain the trust relationships during VPN operations and even beforehand and afterwards.

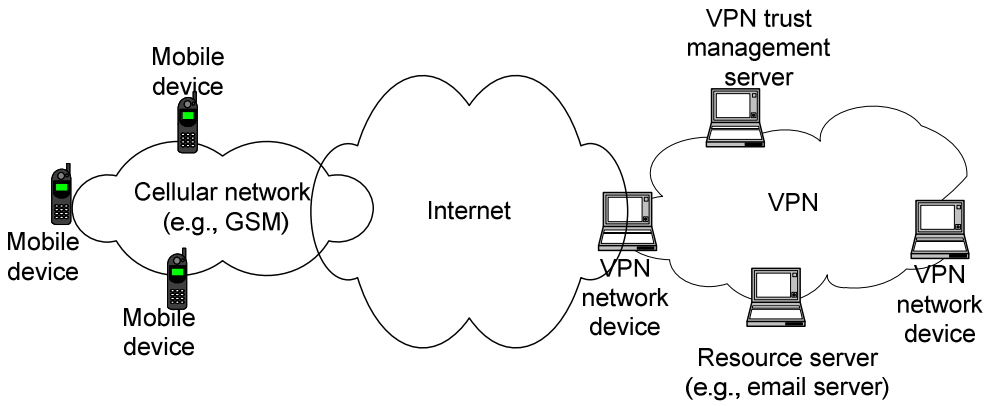

Figure 4.8: Mobile VPN system structure

In particular, with the trust conditions received from the trust management server, the RT module with other necessary modules (e.g., secure storage) in the mobile device is able to keep and maintain the trust relationship, e.g., allow or refuse to install a software, etc.

Although we only mention one trust management server, the server itself may consist of a number of servers that make the system working in practice. For example, a PKI server that generates certificates for the mobile client devices can be included into the system if needed.

System functions

The proposed system provides four major functions. Firstly, the system provides for a trust management server that manages the RT information (e.g., certificates) of various computing platforms in the network. The trust management server stores the RT information of the platforms in a local storage and is able to provide the RT information of any platform to other platforms upon requests. It also maintains the trust conditions on different platforms according to the security policies applied by the VPN operator. Those trust conditions are attached to the RT information of different devices and indicate the expected conditions that the device platform has to fulfill for trust establishment and management. The trust conditions can be configured at the trust management server in order to ensure and maintain the trust relationships with different vendor devices. In addition, the trust management server collects distrust notifications/warnings from the client devices and decides whether to terminate the VPN connection of the client device.

Secondly, the RT module based platform of the system is able to request the RT information and the trust conditions of local platforms or remote platforms from the trust management server. In requesting the RT information, the platform is also able to challenge and verify the remote platforms. By applying the trust conditions into the RT module, the challenging platform can ensure that the remote platform will work as expected according to the VPN operator's specifications.

Thirdly, the RT module based platform is able to manage the trustworthiness of the platform all the time, e.g., verifying codes when the codes are installed and loaded, and verifying the RT module of remote platforms before/during communication. The platforms in the system also ensure that the VPN client device platform is the VPN operator trusted platform for the duration of the VPN connection. It restricts the distrusted changes of the device hardware and software according to the VPN's connection requirements (i.e. trust conditions); therefore, a trusted VPN connection is ensured throughout the entirety of the connection.

Fourthly, with the RT module, more security related services can be provided. For example, in order to prevent crucial data (e.g. confidential files saved locally from the VPN) from being accessed in the VPN disconnection status, the usage of the data can be controlled by the RT module. This aspect is especially significant in that the employees of a company can safely use their company devices, in which company confidential data is stored, in an extranet environment (e.g., the Internet) without the potential for disclosing the crucial data to network hackers. Without this level of protection, the company devices are vulnerable to hackers via the Internet. They are also vulnerable to malicious applications and employees without loyalty.

In general, the system proposes a trust management solution in a mobile enterprise VPN context. The system aims to manage trust-related operations among devices in the enterprise network so that building up trust across devices and between different components of a device (e.g., between applications and OS) is possible. In particular, the system ensures the execution of local platforms and remote device platforms as VPN operator's expectation by applying the trust

conditions into those platforms and maintaining the trust relationship through the RT module control. Therefore, the system solves the existing problems in a mobile enterprise VPN context. In addition, the system offers an advanced control on confidential data on the basis of the RT module after the VPN connection is terminated. Therefore, it offers enhanced trust with better security for an enterprise VPN and increases the user's confidence in VPN services.

Implementation

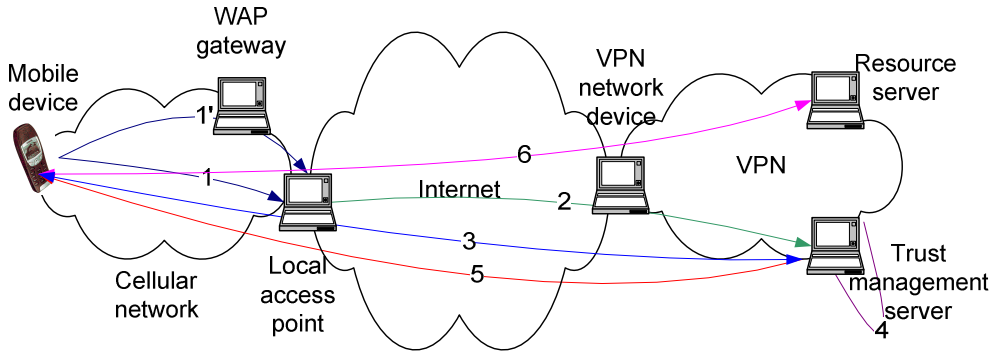

Figure 4.9: An example implementation for getting trust conditions

Trust management of the proposed VPN system is driven by trust conditions issued by the trust management server and sent to the VPN client devices. Figure 4.9 illustrates an example implementation through which a mobile device with the RT module can get the trust conditions from the trust management server. The conditions are embedded into the device for trust management purpose. The implementation consists of the following steps.

1. A mobile device connects (or accesses via WAP) to a local access point.
2. The local access point forwards the connection request to the VPN trust management server. The device may also be able to connect to the trust management server directly without passing through the access point.
3. The trust management server challenges the device over a secure channel (e.g., SSL) for authentication. Also, the device may require information from the trust management server for server authentication. Once the authentication succeeds, the device sends its information to the server upon request. The device information may include a platform configuration certificate, and the mobile device unique platform ID.
4. The trust management server verifies that the above documents can be trusted.
5. Then, the trust management server issues all kinds of files to the device. The files may include, for example, connection configurations, trust conditions for the underlying VPN connection and disconnection, trust conditions on the device for local networking (e.g. P2P enterprise networking), and trust conditions for the contents originated from the enterprise resources.
6. The device can use these files to connect to the intranet services. It also registers the conditions into its RT module based platform for trust management in the context of mobile VPN.

4.6.3 Remarks

By deploying the mechanism for trust sustainability among computing platforms, a VPN system can be managed according to the enterprise operator's expectation. With the proposed system, problems that retard the deployment of mobile enterprise networking can be solved. No matter connected or disconnected, the mobile devices behave as trusted due to the RT module control. In addition, various devices with different security solutions could work together under unified management of the trust management server. A more extended discussion of the issues is provided in Publication 6.

4.7 Summary

This Chapter presented a mechanism for sustaining trust among computing platforms on the basis of Root Trust. The formula of trust used takes the form "A trusts B for P under C based on R". The formula creates trust based on the attestation of the RT module at the trustee and controls its sustainability according to the pre-defined conditions C. Those conditions are approved by both the trustor and the trustee at the time of trust establishment and enforced through the use of the pre-attested RT module until the intended purpose is fulfilled.

This work extends the trust model from static to dynamic. Thus, it develops the notion of using trust management not only for trust assessment but also for trust sustainability. The proposed mechanism could be applied in many real applications for trusted services and communications, for example, trust collaboration in P2P systems and trust management in mobile enterprise networking. It could work as an extension of the trusted computing platform to support various applications with enhanced flexibility.

5 AUTONOMIC TRUST MANAGEMENT FOR A COMPONENT SOFTWARE PLATFORM

In this Chapter, we discuss autonomic trust management for a component software platform. We develop a formal trust model to specify, evaluate, set up and ensure trust relationships that exist among platform entities. We further present an autonomic trust management architecture that adopts a number of algorithms for trust assessment and maintenance during component execution. In addition, we propose a mechanism for trust control mode prediction and selection on the basis of an adaptive trust control model in order to support autonomic trust management.

5.1 Introduction

The growing importance of software in the domain of mobile systems introduces special requirements on trust due to the nature of applications they provide. In particular, this applies when the software is component based and varies due to components joining and leaving the system. However, the lack of a trustworthy software platform could be the main reason that retards the further development of mobile applications and services.

From a system point of view, trust is the assessment of a trustor on how well the observed behavior (quality attributes) of a trustee meets the trustor's own standards for an intended purpose [Den93]. From this, the critical characteristics of trust can be summarized, it is: subjective, different for each individual in a certain situation; and dynamic, sensitive to change due to the influence of many factors. Obviously, it does not suffice to require the trustor (e.g. most possibly a digital system user) to make a lot of trust related decisions because that would destroy any attempt at user friendliness. For example, the user may not be informed enough to make sound decisions. Thus, establishing trust is quite a complex task with many optional actions to take. Rather trust should be managed automatically following a high level policy established by the trustor, for example a software component or the user of a component software platform. We call such trust management autonomic. Autonomic trust management automatically processes evidence collection, trust evaluation, and trust (re-)establishment and control. We need a proper mechanism to support autonomic trust management not only on trust establishment, but also on trust sustaining. This is important for a component software platform that should support trustworthy downloading and executing of the software components.

A number of studies on trusted computing and management have been conducted in the industry and reported in the literature. For example, TCG (Trusted Computing Group) aims to build up a trusted computing device on the basis of a secure hardware chip [Tcg03, Vau03, Dav02, ELM03, BaS03]. Some trust management systems focus on protocols for establishing trust in a particular context, generally related to security requirements. Others make use of a trust policy language to allow the trustor to specify the criteria for a trustee to be considered trustworthy [GrS00]. However, the focus on the security aspect of trust

tends to assume that the other non-functional requirements [BMM05], such as availability and reliability, have already been addressed.

Recently, many mechanisms and methodologies are developed for supporting trusted communications and collaborations among computing nodes in a distributed system (e.g. an Ad Hoc Network, a P2P system and a GRID computing system) [ZWW05, ThB06, LVW04, SYH06]. These methodologies are based on digital modeling of trust for trust evaluation and management. We found that these methods are not very feasible for supporting trust on a device software platform.

At present, there is no common framework to enable trust management in a commercial component software system (CSS), even though there is a pressing need to support a range of new applications. This framework must support autonomic trust management through trust assessment and maintenance over the dynamic component software system, consisting of different functionalities provided by various disparate companies. We need technologies for the development and validation of trusted systems based on the integration of multi-party software while at the same time reducing the cost and integration time. Meanwhile, we argue that trust can be controlled according to its evaluation result. Special control modes can be applied into the software platform in order to ensure a trustworthy system. A trust control mode contains a number of control mechanisms or operations, e.g. encryption, authentication, hash code based integrity check, access control mechanisms, duplication of process, and a man-in-middle solution for improving availability, etc. It can be treated as a special configuration of trust management that can be provided by the system.

The work presented in this Chapter is conducted in EU ITEA Trust4All project [RST]. This project aims to build up a trustworthy middleware architecture in order to support easy and late integration of software from multiple suppliers and still have dependable and secure operation of the resulting system.

5.2 Related Work

Trust has been recognized as an important factor for component software. A number of interesting solutions have been proposed to ensure its trustworthiness [Her01, Her03, ZMZ05, ZJM05, HaR06].

Herrmann developed a special reputation system based on a component user's experience, other users' experiences and the third trusted party's certificate in order to reduce the expense of evaluating components [Her01, Her03]. His work is one of the first to study trust management for component software. He is the first, as what we know, to apply runtime observation based method to collect valuable information for trust evaluation on a software component. In [Her01], he applied an approach which takes the experience of other users with a component and employed the concept of trust management to calculate trust values from good and bad evaluations with it. Particularly, a trust information service was introduced to collect expertise and make it available to component users and certification authorities. The expertise is gained from certification of a component as well as monitoring it during deployment. From these evaluations a trust value is generated and offered to parties interested to purchase the component. The runtime monitoring was implemented by a secure wrapper. It is a piece of code extending a component, while the wrapper does not change the behavior of the component. It

monitors the component interface for security flaws. In addition, the intensity of the runtime observations about a component can be adjusted due to the current trust value of the component. In some urgent distrust situation, the security wrapper can aid to seal the component. In [Her03], Herrmann further extended his work to prevent that a component user sends wrong reports resulting in a bad trust value of the component by discounting a recommendation with the trust value in the recommender. The total trust value of a particular component is calculated by application of Jøsang's Subjective Logic [JoK98, Jos01].

Our work reported in this Chapter focuses on component execution time trust management. We aim to conduct runtime holistic trust management in a component software platform based on the system's competence in an autonomic way. We apply a centralized trust management framework to conduct runtime observation based autonomic trust management in order to release the development burden that is needed per each component and support interoperability. The trust assessment is based on observing a number of quality attributes of the trustee entity for the purpose of adaptively recognizing the real system's situation to conduct autonomic trust management. This observation is conducted by the trust management framework embedded in the component software runtime layer. It is a system centralized observation solution, not a secure wrapper based distributed solution. The total trust value is calculated by aggregating the values of different quality attributes together according to the preference set by the trustor entity. The trust value expression and generation is also based on the Subjective Logic [JoK98, Jos01], while the total trust value aggregation is implemented by applying a new operator. The autonomic trust management is implemented through control mode prediction and selection mechanisms on the basis of an adaptive trust control model with the consideration of trust control mechanisms' influence on trust.

A framework for dynamic re-configuration of different qualities from the view of trust was constructed in [ZMZ05, ZJM05] providing common mechanisms in middleware to ease the burden for trust component developers. Comparing with previous works, it focused on a trust perspective to satisfy various QoS demands of different users, and built a five-layer trust management framework, which not only provides common trust management facilities for trust components, but also supplies components for dynamical (re-)configuration of multi-properties. Based on the framework, the authors presented an algorithm to adjust dynamically all the involved trust properties according to predefined policies when the environment changes. The solution proposed in [ZMZ05, ZJM05] supports multiple properties of trust. The centralized trust management in middleware is similar to our solution, but with different design since our design supports auto-selection of trust control mechanisms. Also, the trust evaluation function in [ZMZ05, ZJM05] relies on users to customize. It is usually time-consuming and prone to errors. Some automation functions are needed in the trust management framework to reduce more the burdens of developers. Regarding the dynamic reconfiguration of component trust properties, it lacks necessary support to evaluate if trust can be managed based on the system's competence. The adjustment based on predefined policies lacks flexibility and can not predict cross-influence of various trust mechanisms on different trust properties. In this Chapter, we propose a trust assessment based autonomic trust management solution in order to overcome the above problems and further release the burden of component software developers.

The on-going TrustSoft project aims to study a holistic approach to software trustworthiness through certifying multiple quality attributes of the software [HaR06]. The methodology of trust management applied in this project is different from the solution presented in this Chapter.

5.3 Trust Issues in Component Software

Due to dynamic changes in the component software system and due to multiple vendors that may offer software components with similar functionalities, we need to develop trust management mechanisms for the component software system. For the component-centered aspect we must consider trust at several decision points: at download time and during execution. At a component download time, we need to consider whether a software provider can be trusted to offer a component. Furthermore, we need to predict whether the component is trustworthy for installation. More necessarily, when the component is executed, we have to ensure that it can cooperate well with other components and the system provides expected performance and quality. The trust relationship changes during the above procedure.

When discussing a component software platform, the execution of components in relation to other entities of the system needs to be taken into account. Even though the component is trustworthy in isolation, the new joined component could cause problems because it will share system resources with others. This influence will impact the trustworthiness of the system. Consequently, the system needs mechanisms to control its performance, and to ensure its trustworthiness in an autonomic way even if the internal and external environments change. Additionally, some applications (e.g. a health care service) need special support for trust because they have high priority requirements, whereas game playing applications, while exhibiting similar functionality (e.g. a network connection) will not have the same priority. Therefore, system-level trustworthiness is dependent on the application domain, so the system needs a trust management framework that supports different trust requirements from the same or different components. This Chapter presents autonomic trust management for a component software platform mainly focusing on system runtime and embedded intelligence to predict and select control modes for supporting autonomic trust management.

5.4 Requirements and Approaches to Autonomic Trust Management

From the component software platform point of view, autonomic trust management includes the following four aspects.

- Trust establishment: the process for predicting trustworthiness and establishing a trust relationship between a trustor and a trustee. Trust establishment is required when a component or a bundle of components is downloaded and installed at the system. It is also required when a new component starts to run.
- Trust monitoring: the trustor or its delegate monitors the performance of the trustee. The monitoring process aims to collect useful evidence for trust assessment.

- Trust assessment: the process for evaluating the trustworthiness of the trustee by the trustor or its delegate. The trustor assesses the current trust relationship and decides if this relationship should be changed or not. If it is changed, the trustor will make a decision which measure should be taken.
- Trust control and re-establishment: if the trust relationship will be broken or is broken, the trustor will find reasons and take corresponding measures to control or re-establish the trust relationship.

A number of requirements can be summarized in order to support autonomic trust management for a component software platform. Firstly, the platform should handle the requests with different trust priority adaptively. This can be solved by a system architecture design supporting collaboration between the trust management framework and the resource management framework through component trust modeling. Secondly, for trust crash, the device should react adaptively as expected within some limited time. Trust assessment based evaluation on selected control modes can be applied to solve this issue. Finally, the platform should be intelligent for trust management. "Which trust control mechanism is good for improving which quality attributes in what kind of context" should be well addressed. The effectiveness of trust control modes should be predicted for the selection and deployment of the best modes.

Particularly, a component software platform is composed of a number of entities, e.g. a component (composition of components), an application, a sub-system and the whole platform system. The trustworthiness of an entity depends on a number of Quality Attributes (QAs) of the entity. The quality attributes can be the entity's trust properties (e.g. security, availability and reliability) and/or recommendations or reputations with regard to them. The taxonomy of Quality Attributes is shown in Figure 5.1. We mainly aim to support security and dependability ensured trustworthiness.

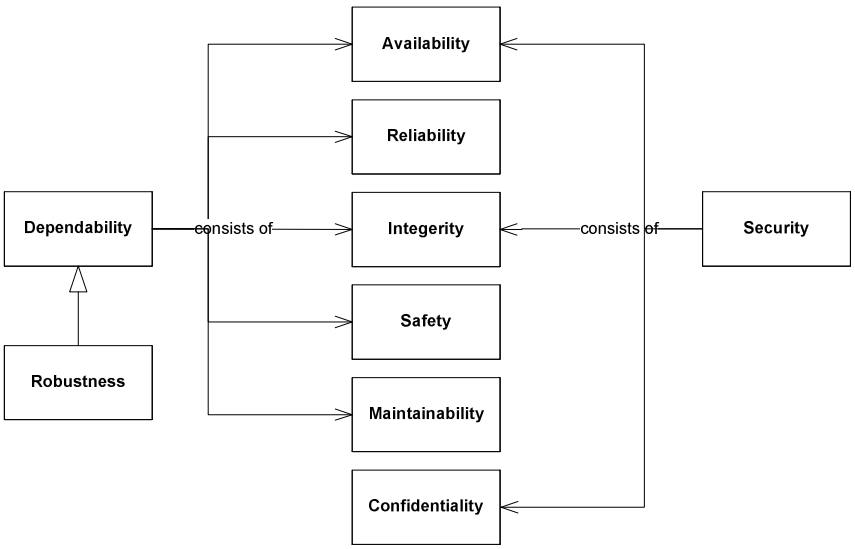

Figure 5.1: Taxonomy of quality attributes

The QA is reflected by some matrices, the parameters of the matrices can be monitored. Thus the QA can be evaluated at system runtime. For example, the

availability of a system entity can be reflected by this entity's response time and uptime percentage. The decision or assessment of trust is conducted based on the trustor's (e.g. a component consumer) subjective criteria and the trustee entity's quality attributes, as well as influenced by context information. The context includes any information that can be used to characterize the situation of the involved entities. The quality attributes of the entity can be controlled or improved via applying a number of trust control modes. Thus, special control modes can ensure the trustworthiness of the system entity, especially at component download time and runtime. The relationships of those factors related to platform trust are illustrated in Figure 5.2.

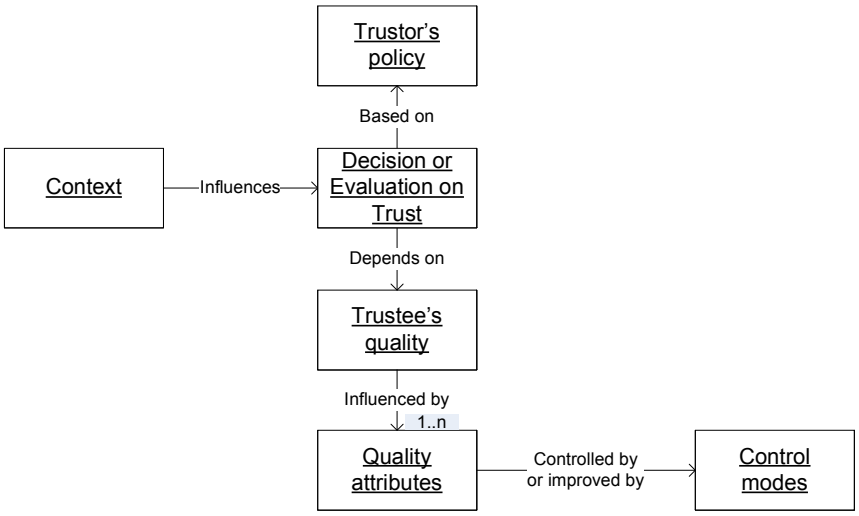

Figure 5.2: Factors related to platform trust

Based on the above understanding, we propose a procedure to conduct autonomic trust management in the component software platform targeting at a trustee entity specified by a trustor entity, as shown in Figure 5.3.

Trust control mode prediction is a mechanism to anticipate the performance or feasibility of applying some control modes before taking a concrete action. It predicts the trust value supposed that some control modes are applied before the decision to initiate those modes is made. Trust control mode selection is a mechanism to select the most suitable trust control modes based on the prediction results.

For a trustor, the trustworthiness of its specified trustee can be predicted regarding various control modes supported by the system. The control mode can be treated as a special configuration of trust management that can be provided by the system. Based on the prediction results, a suitable set of control modes could be selected to establish the trust relationship between the trustor and the trustee. Further, a runtime trust assessment mechanism is triggered to evaluate the trustworthiness of the trustee through monitoring its behavior based on the instruction of the trustor's criteria. According to the runtime trust assessment results in the underlying context, the system conducts trust control model adjustment in order to reflect the real system situation if the assessed trustworthiness value is below an expected threshold (refer to Section 5.8). This threshold is generally set

by the trustor to express its real expectation on the assessment. Then, the system repeats the procedure. The context-aware or situation-aware adaptability of the trust control model is crucial to re-select suitable control modes in order to fulfill autonomic trust management.

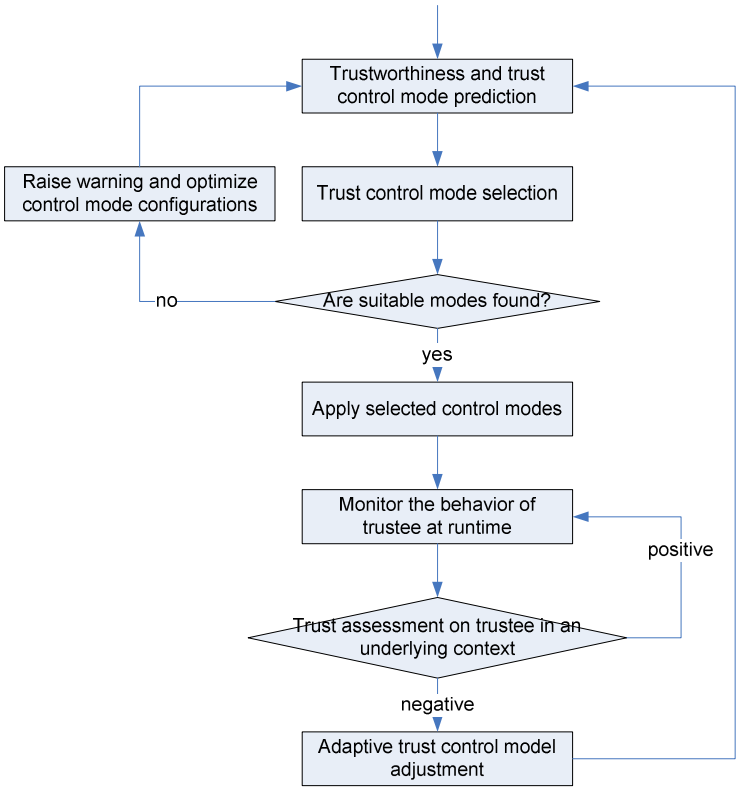

Figure 5.3: Autonomic trust management procedure at runtime

In order to implement the above approach, we need firstly to develop a trust model that can specify, evaluate, set up and ensure the trust relationships among system entities. In addition, a trust management architecture is required to adopt a number of algorithms that can realize the approach. These algorithms should be developed based on the trust model and include trust prediction for component software execution, trust assessment at runtime, control mode prediction and selection and adaptive trust control model adjustment. In the following sections, we will discuss the above issues in details.

5.5 A Formal Trust Model

In this section, we present a formal trust model that can support autonomic trust management. It contains a sub-model to present trust relationships among system entities, a sub-model to specify the information related to trust management for a software component, a sub-model for trust assessment and a sub-model for trust management.

A component software system can be represented as a structure (E, R, M), where E represents the set of the system entities, R the set of trust relationships between the entities, M the set of trust management mechanisms for the management of such trust relationships.

The system entities can be any parties that are involved in or related to the component software system or platform. These entities include a platform/system user, a component consumer, a component provider, a service, a component (composition of components), an application, a sub-system and a system, as well as an operation or a mechanism provided by the system.

An application is a software entity that provides a set of functions to a user. A component is a unit of trading that may contain multiple services. A service is a unit of software instantiation that is contained in a component and conforms to a component model. A system is a combination of a platform, a set of components, a runtime environment (RE) and a set of applications that can provide a user with a set of functions. The platform provides access to the underlying hardware. The relationships among the above entities are described in Figure 5.4.

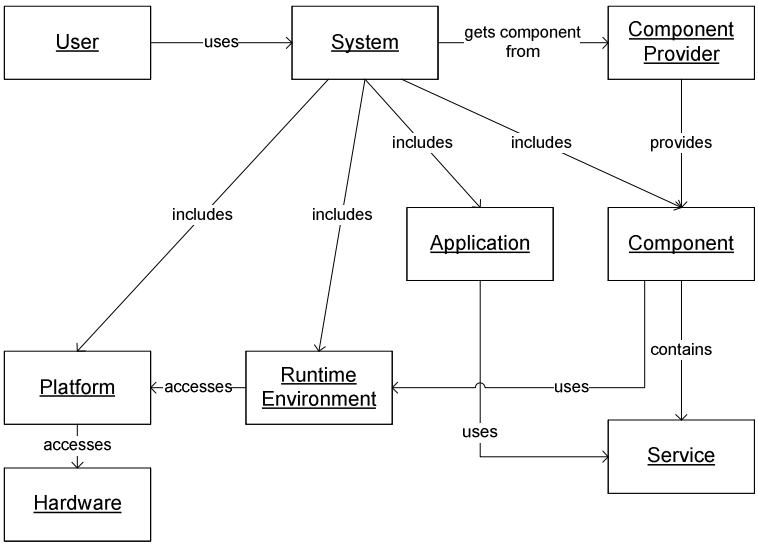

Figure 5.4: Relationships of component software system entities

A model of trust relationship (TR)

A trust relationship between two entities in the component software system can be specified as a 6-tuple $TR = \{tr, te, py, ct, ev, op(b, d, u)\}$ which asserts that entity *tr* trusts entity *te* with regard to *tr*'s trust policy *py* in the context *ct*, based on the evidences about *te*: *ev*, which is collected from the experience of the trustor (e.g. trustor's experiences about a number of quality attributes) and/or the experiences collected from other entities (e.g. recommenders) [Her03, XiL04, JHK05], and *op(b,d,u)* indicates the trust valuation. It is the probabilities of opinion on *te* regarding belief (*b*), disbelief (*d*) and uncertainty (*u*) [JoK98, Jos01]. Particularly, *b*, *d*, and *u* satisfy: $b + d + u = 1$, and $0 \le b, d, u \le 1$. Herein, belief means the probability that the entity *te* can be trusted by the entity *tr*; disbelief means the probability that

te can not be trusted by *tr*; and uncertainty fills the void in the absence of both belief and disbelief. It is important to note that the trust value can be expressed in other forms, e.g. a single-dimension value, depending on the method used for the trust value calculation.

Variable *py* represents a subset of the set (PY) of all policies regarding trust management. $PY = \{to, ir, ep, cr\}$, where *to* is a threshold opinion for trust, $ir = \{ir_{qa\_1}, ir_{qa\_2}, ir_{qa\_3}, ..., ir_{qa\_n}\}$ is the importance rates of different quality attributes *qa* of the *te*, and *ep* is the policy about the evidence used for the trust assessment. Variable *cr* ($cr = \{tv_{qa\_1}, tv_{qa\_2}, tv_{qa\_3}, ..., tv_{qa\_n}\}$) is the criteria for setting positive points p_{qa_i} or negative points n_{qa_i} on different quality attributes (refer to Section 5.7 for details). It specifies the trusted values or value scopes of different factors reflecting the quality attributes of the trustee.

A model of software component (CTM)

A component trust model can be described as $CTM = \{opr, (res, con), tl, per, req, rul\}$, where $opr \in OPR$, the set of all operations provided by the component services. This model describes the trust specifications of all the operations implemented by the services in the component. The trust request level (*tl*) indicates the importance of the operation (*opr*). *res* and *con* specify the resource (*res*) consumption requirements (*con*) that the operation required in order to provide the performance described by *per*. Particularly, *res* ∈ {*memory*, *cpu*, *bus*, *net*,…}. Variable *per* represents a subset of the set ($PER = \{(qa_i, tv_{qa\_i})\}$) of all possible quality attributes regarding the underlying operation and the promised trusted value scopes. Variable *req* specifies the requirements for cooperating with other services, e.g. the trust policy to call a service. In addition, composition rules (*rul*) are criteria for composing this model with other component trust models.

The component trust models are attached to the software component. They can be composed based on the composition rules. They have several usages. At download time, they can be used to help the system to predict whether a component may have some trust influence on the system, e.g. whether the required resources may exceed the system's competence. At execution time, they are used by the system to predict trustworthiness of a number of cooperated components and arrange resources for the services, especially when the resources are limited. A more detailed discussion of the issues is provided in [Yan07]. In addition, the component trust models could help the system trust management framework to predict trustworthiness and monitor the performance of the services (e.g. the composed performance *per* could play as the default trust criteria for trust assessment at runtime), thus evaluate if the component's services and the subsystem containing the component are trusted or not. The assessment result plays as a trigger for autonomic trust management, especially trust control and enforcement. In addition, the component trust model could also be used to reason about problems of some services in a component.

A model of trust assessment (TA) and a model of trust management (ATM)

The trust assessment can be expressed as a 6-tuple $TA = (tr, te, ct, ev, py, CTM)$, which asserts that *tr* evaluates *te*'s trust in the context *ct*, based on the *tr*'s policy *py*, and according to evidence *ev* and the component trust model *CTM*. The output of trust assessment is a set of opinions on trustworthiness and a number of quality attributes. Trust management can be expressed as a 3-tuple $ATM = (TR, TA, M)$, which asserts that mechanisms *M* are applied for the trust relationship *TR* according to the trust assessment *TA*. In Section 5.8, we propose an adaptive trust control model that is a concrete implementation of *ATM* with a number of algorithms' support.

The trust management mechanisms compose a set $M = \{T_e, T_m, T_a, T_c\}$, where T_e is the set of mechanisms used for trust establishment and re-establishment, e.g. a component container to isolate a distrusted component from its environment; T_m is the set of mechanisms applied for monitoring and collecting the evidences or the factors regarding the quality attributes of the trustee, e.g. a dynamically initiated monitor instantiation at the trust management framework to report a component service's runtime performance; T_a is the set of mechanisms for trust evaluation, e.g. a runtime trust assessment mechanism and a trust prediction mechanism for component software download and execution regarding system competence and component cooperation [Yan07]; and T_c is the set of mechanisms for controlling trust in order to sustain the trust relationship, e.g. encryption, authentication, hash code based integrity check, access control mechanisms, duplication of process, reconfiguration of component linkage, man-in-middle solution for improving availability, etc. The mechanisms in T_e and T_c are classified in terms of different quality attributes. Thus the system knows which mechanism should be considered in order to ensure or support a quality attribute.

Relationships of models

The above introduced sub-models can cooperate with each other to support solving the trust issues related to component software.

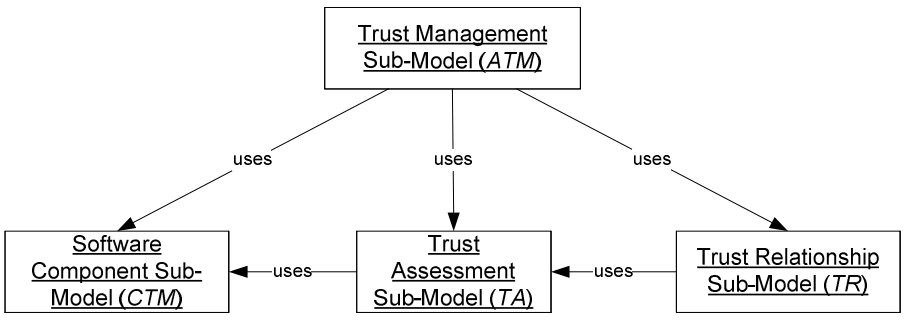

Figure 5.5: Relationships among four sub-models

At the component download time, the sub-model of software component (*CTM*) is used to predict if the component is trustworthy for downloading based on system resource and reliability analysis [Yan07]. *TA* can also be applied to study the reputation of a software component, (e.g. [Her01, Her03]) for component

procurement. We think the above two levels of evaluations should be supported by the system and conducted based on the trustor's preference. During the component execution, *CTM* can be applied to check system resource and reliability before dynamically linking a number of components together [Yan07]. Additionally, we make use of *TA* to conduct trust assessment based on runtime observation on the behaviour of the trustee entity (e.g. a software component) with regard to the *CTM* of the component and the trustor's policy. Different from the prior arts, the assessment results further trigger autonomic trust management based on the sub-model of trust management (*ATM*) in order to ensure and sustain the trust relationship. In particular, *TR* and *CTM* can be used to compare trust relationships for auto-configuring a set of components that could cooperate with each other in order to provide expected services. The relationships of these four sub-models are depicted in Figure 5.5.

5.6 Trust Management Architecture

5.6.1 Platform Structure

The mobile computing platform generally consists of a layered architecture with three layers: an application layer that provides features to the user; a component-based middleware layer that provides functionality to applications; and, the fundamental platform layer that provides access to lower-level hardware. Using components to construct the middleware layer divides this layer into two sub-layers: a component sub-layer that contains a number of executable components and a runtime environment (RE) sub-layer that supports component development.

Placing trust management inside this architecture means linking the trust management framework with other frameworks responsible for component management (including download), security management, system management and resource management. Figure 5.6 describes interactions among different functional-blocks inside the RE sub-layer. The trust management framework is responsible for the assessment of trust relationships and trust management operations, system monitoring and autonomic trust managing. The download framework requests the trust framework for trust assessment of a component to decide whether to download the component and which kind of mechanisms should be applied to this component. When a component service needs cooperation with other components' services, the execution framework will be involved, but the execution framework will firstly request the trust management framework for decision. The system framework takes care of system configurations related to the components. The trust management framework is located at the core of the runtime environment sub-layer. It monitors the system performance and instructs the resource framework to assign suitable resources to different processes. This allows the trust management framework to shutdown any misbehaving component, and to gather evidence on the trustworthiness of a system entity. Similarly, the trust management framework controls the security framework, to ensure that it applies the necessary security mechanisms to maintain a trusted system. So briefly, the trust management framework acts like a critical system manager, ensuring that the system conforms to

its trust policies. This architecture ensures the implementation of both the 'hard trust' solution and the 'soft trust' solution.

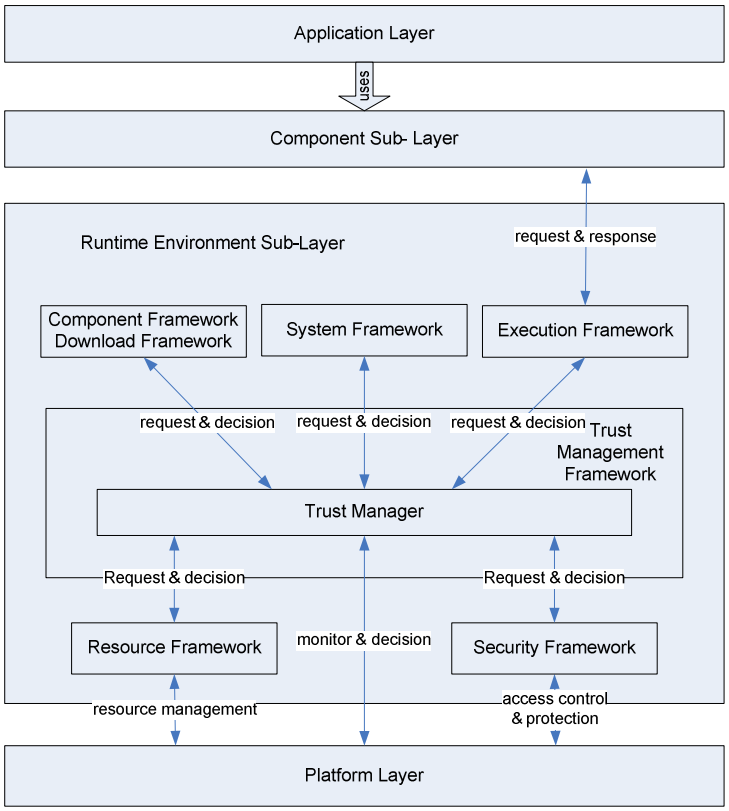

Figure 5.6: Relationships among trust management framework and other frameworks

5.6.2 Trust Management Framework

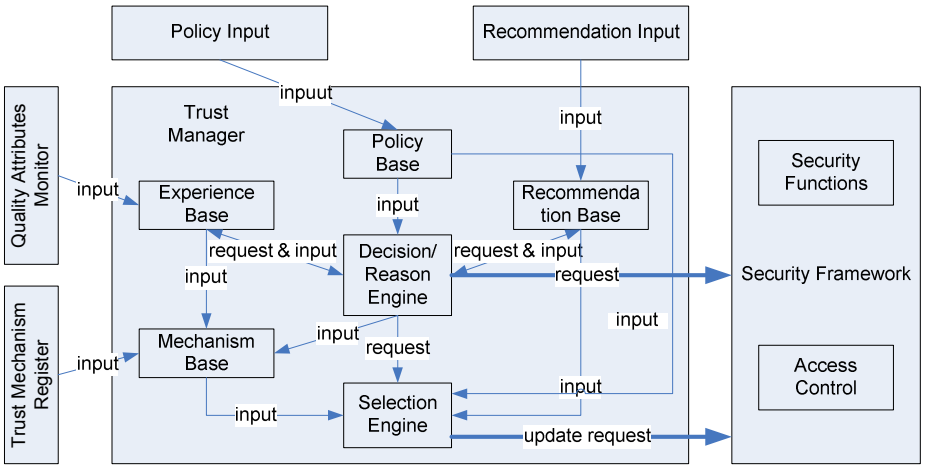

Figure 5.7: The structure of trust management framework

Figure 5.7 illustrates the structure of the trust management framework. In Figure 5.7, the trust manager is responsible for trust assessment and trust related decision-making, it closely collaborates with the security framework to offer security related management. The trust manager is composed of a number of functional blocks. The trust policy base saves the trust policy (*py*) regarding making trust assessments and decisions. The recommendation base saves various recommendations. The experience base saves the evidence *ev* collected from the component software platform itself in various contexts. The decision/reason engine is used to make trust decision by requests from other frameworks (e.g. the download framework and the execution framework). It combines information from the experience base, the recommendation base and the policy base to conduct the trust assessment. It is also used to identify the reasons of trust problems. The mechanism base registers a number of mechanisms in T_e and T_c for trust control and establishment that are supported by the platform. It is also used to store the trust control models as described in Section 5.8. The selection engine is used to select suitable mechanisms to ensure the platform's trustworthiness in a special context. It also conducts adaptive adjustments on the trust control model.

In addition, the recommendation input is the interface for collecting recommendations. The policy input is the interface for the system entities to input their policies. The trust mechanism register is the interface to register trust mechanisms that can be applied in the system. The quality attributes monitor is the functional block used to monitor the system entities' performance regarding those attributes that may influence trust. The trust manager cooperates with other frameworks to manage the trustworthiness in the whole system.

5.7 Trust Assessment at Runtime

The main functionality provided by the decision/reason engine is the trust assessment. There are several existing mechanisms that can be applied for assessing trust through evidence. Here subjective logic (SL) [Jos01] has been chosen as the formal base for trust assessment because of its sound mathematical foundation in dealing with evidential beliefs; and the inherent ability to express uncertainty explicitly. The SL consists of a belief model called opinion and a set of operations for aggregating opinions. Herein, we apply a simplified scheme of the SL as in [JoK98, LVW04, Twi03].

5.7.1 Notations and Definitions

We develop and use several new SL operators to illustrate how to assess trust based on the formal trust model in the trust management framework.

Notation 1:

1. An opinion $\omega = (b, d, u) \in \Omega$, where $b + d + u = 1$, and $b, d, u \in [0,1]$. b is the belief, d is the disbelief, u is the uncertainty of the opinion and Ω is the set of all opinions. In particular, ω_i is the opinion on the quality attribute qa_i, and ω_A is an opinion about a proposition A;

2. QA is the set of quality attributes *qa* that may influence an entity's opinion on a proposition;
3. W is the set of weights w of different opinions, such that w_i is the importance rate of ω_i on the quality attribute qa_i, or w_i is the weight of opinion o_i on an entity, where $o \in \mathrm{O}$ – the set of opinions on an entity. In particular, $w_i \in [0,1]$ and $\sum w_i = 1$;
4. $op(x)$ is an opinion on *x*, $op(x.y)$ is an opinion on the quality attribute *y* of *x*;
5. *x.y.z* is the value of parameter *z* of *y* in *x*; e.g. $py.cr.tv_{qa\_i}$: parameter tv_{qa_i} of cr in py.

Definition 1.
Let Ω_a *be the set of opinions,* W *their weights w and* $\Omega_a \subset \Omega$*, such that* $\omega = (b,d,u) \in \Omega_a$.
Then $\Sigma : (W \times \Omega_a) \times (W \times \Omega_a) \to \Omega$ *is the weighted summation operator on opinions and for a finite set of n opinions,* $\omega_\Sigma = (b_\Sigma, d_\Sigma, u_\Sigma)$ *is the summary opinion such that* $b_\Sigma = \sum_{i=1}^{n} w_i b_i$; $d_\Sigma = \sum_{i=1}^{n} w_i d_i$; $u_\Sigma = \sum_{i=1}^{n} w_i u_i$.

The weighted summation operator can be used to combine the opinions on a number of the trustee's quality attributes. The combination is based on the importance rates of the attributes. It also makes it easy to consider other influencing factors of trust through weighting, such as time element, similarity of recommendations to trustor's own experience and similarity of different contexts, etc. It is also flexible to support applying other forms of trust value expression.

The weighted summation operator can be applied to a finite set of *m* opinions $o \in \mathrm{O}$ on an entity with weights $w \in W$. Then $o_\Sigma = (b_\Sigma, d_\Sigma, u_\Sigma)$, where $b_\Sigma = \sum_{i=1}^{m} w_i b_i$; $d_\Sigma = \sum_{i=1}^{m} w_i d_i$; $u_\Sigma = \sum_{i=1}^{m} w_i u_i$.

The weighted summation operator can also be used to aggregate the opinions about a trustee generated by different entities or by the same entity at different times. The combination is also based on weights.

Notably, the operator $\sum$ sits beyond the original theory of the SL. It supports a special case that all ω_i hold the same *base rate a* (the default value is 0.5) [Jos01]. In addition, based on experimental study, it can provide similar results to the results through applying the SL original operators $\bigoplus_{i=1}^{m} \{\omega_i \otimes (w_i, 1-w_i, 0)\}$. That is using the Discounting operator $\otimes$ to weight ω_i with discounting opinion $(w_i, 1-w_i, 0)$, and then applying Bayesian Consensus operator $\oplus$ to aggregate all discounted ω_i together [TLU06]. Thereby, the operator $\sum$ provides a shortcut for the aggregation of trust values expressed as SL opinions.

Definition 2.
Let $\omega = (b,d,u)$.

Then $\theta:(p,n,r)\rightarrow\omega$ *is the opinion generator, such that* $b+d+u=1$, $b=p/(p+n+r)$, $d=n/(p+n+r)$ *, and* $u=r/(p+n+r)$.
Particularly, p is the positive points of evidence on a proposition, n is the negative points of evidence on the proposition, $r\geq 1$ *is a parameter controlling the rate of loss of uncertainty, which can be used to tune the use of uncertainty in the model for the requirements of different scenarios (we often take* $r=2$ *).*

The operator θ is used to generate an opinion based on positive and negative evidence [JoK98]. Note that other definitions on calculating *b, d* and *u* can also be applied.

Definition 3.
Given two opinions $\omega_A=(b_A,d_A,u_A)$ *and* $\omega_B=(b_B,d_B,u_B)$ *, we define the comparison operator* $\geq_{op}$ *as an opinion comparison operator, whereby* $\omega_A\geq_{op}\omega_B$ *holds, if* $b_A>b_B;d_A<d_B;u_A<u_B$. *And we say that opinion* $\omega_A=(b_A,d_A,u_A)$ *is over a threshold presented by* $\omega_B=(b_B,d_B,u_B)$.

The operator $\geq_{op}$ is used to compare two opinions, especially to decide if an opinion is over a threshold presented by another opinion and order a number of opinions [LVW04]. Note that $\geq_{op}$ is a partial order operator.

5.7.2 Trust Assessment Algorithm

At runtime, the quality attribute monitor monitors the trustee's performance with respect to its quality attributes. The monitoring is based on or driven by the criteria or policies set by the trustor entity who registers itself firstly at the trust management framework in order to manage trust of its specified trustee. Monitoring based method for trust (re-)evaluation was defined in the concept of trust management by Grandison and Sloman [GrS00]. A runtime observation based method through applying a security wrapper to simulate a component contract situation for collecting useful information for trust evaluation on the software component was proposed in [Her01]. In the experience base, for each quality attribute, if the monitored performance is better than the criteria (saved in the policy base), the positive points of that attribute are increased by 1. If the monitored result is worse than the criteria, the negative points of that attribute are increased by 1. The opinion of each quality attribute can be generated based on the opinion generator θ [JoK98]. In addition, based on the importance rates of different quality attributes, a combined opinion on the trustee can be calculated by applying the weighted summation operator. By comparing to the trust threshold opinion (*to*), the decision engine can decide if the trustee is still trusted or not. The algorithm for trust assessment at runtime is described below.

Initialization
    te: the assessed target (a system or subsystem or a service)
    py(to, ir, ep, cr): the policy on *te*:
    $n_{qa\_i}=p_{qa\_i}=0$; $r_{qa\_i}=2$; $(i=1,...,n)$
    op(te.qa_i) = (0,0,1); op(te) = (0,0,1)

1. Monitor *te*'s performance regarding *te*'s quality attributes in specified period *t*.
2. For $\forall qa\_i(i=1,...,n)$,

   If the monitored result is better than $py.cr.tv_{qa\_i}$, p_{qa_i}++;

   Else, n_{qa_i}++
3. For $\forall qa\_i(i=1,...,n)$, calculate the opinion:

   $op(te.qa\_i)=\theta(p_{qa\_i},n_{qa\_i},r_{qa\_i})$.
4. Based on the importance rates on different attributes, calculate the combined opinion: $op(te)=\sum_{i=1}^{n}\{ir_{qa\_i},op(te.qa\_i)\}$.
5. If $op(te)\geq_{op} py.to$, make trust decision; else, make distrust decision.

The assessment is conducted based on a time window or monitoring times, which can be specified by the trustor. This algorithm has been tested, implemented and integrated into the EU ITEA Trust4All platform as one of the mechanisms for trust assessment on the platform entities. The Trust4All platform is a Linux based middleware platform for component software.

There are a number of operators provided by the SL to conduct operations on the SL opinions. Herein, we use the newly defined weighted summation operator. There are several reasons. Firstly, we can not find any existing SL operator that can directly support aggregating a number of opinions on one entity's quality attributes together. Secondly, as specified above, the weighted summation operator is a shortcut operation of applying the SL original operators. That is using the Discounting operator $\otimes$ to weight ω_i with discounting opinion $(w_i,1-w_i,0)$, and then applying Bayesian Consensus operator $\oplus$ to aggregate all discounted ω_i together [TLU06]. Thirdly, using the weighted summation operator could make the trust assessment autonomic. The weights or important rates can be automatically generated based on the high level policies of the trustor, for example, time distance or the deviation of context similarity and opinion similarity. Finally, using the weighted summation operator is compatible with the trust control model as will be introduced in Section 5.8. Thereby, the trust assessment at runtime introduced herein can cooperate with the trust control mode prediction and selection as well as adaptive trust control model adjustment in order to provide an autonomic trust management solution.

5.7.3 General Criteria Support

In the proposed formal trust model, we support the criteria for setting positive points p_{qa_i} or negative points n_{qa_i} on different quality attributes. Particularly, it is also possible to support general criteria that specify the conditions of trust related to several quality attributes. The trust opinions calculated based on the general criteria can also be aggregated based on the importance rates.

In addition, a tracing factor (*tf*) can be introduced to support a special case that when the opinion on a trust influencing factor (e.g. a quality attribute) is below some threshold, trust will be totally lost. We set $tf=0$ if it happens. The final trust

value will be tailored as $\omega_f = \omega_f(b_f, d_f, u_f)$, where $b_f = b_\Sigma * tf = 0$, $d_f = b_\Sigma + d_\Sigma$, $u_f = u_\Sigma$. If trust is partially lost in this case, $0 < tf < 1$, and $b_f = b_\Sigma * tf$, $d_f = b_\Sigma(1-tf) + d_\Sigma$, $u_f = u_\Sigma$.

5.8 Adaptive Trust Control Modeling and Control Mode Selection

For autonomic trust management of a component software platform, the trust control mode prediction and selection are important functionalities with regard to the automatic processing of trust. In this section, we firstly introduce the Fuzzy Cognitive Map that plays as the foundation of the adaptive trust control model proposed in Section 5.8.2. Then we present the algorithms used for control mode prediction and selection, and context-aware adaptive model adjustment. We also report our simulation results published in [YaP07] followed by further discussion on a number of issues related to the flexibility and effectiveness of the model.

5.8.1 Fuzzy Cognitive Map

A Fuzzy Cognitive Map (FCM) could be regarded as a combination of Fuzzy Logic and Neural Networks. In a graphical illustration, FCM is a signed directed graph with feedback, consisting of nodes and weighted arcs. Nodes of the graph stand for the concepts that are used to describe the behavior of the system and they are connected by signed and weighted arcs representing the causal relationships that exist between the concepts, as shown in Figure 5.8. It must be mentioned that all the values in the graph are fuzzy, so concepts take values in the range between [0, 1] and the weights of the arcs are in the interval [-1, 1]. The FCM makes clear the interconnections and influences between concepts, It also permits updating in the construction of the graph, such as the adding or deleting of an interconnection or a concept [SGG].

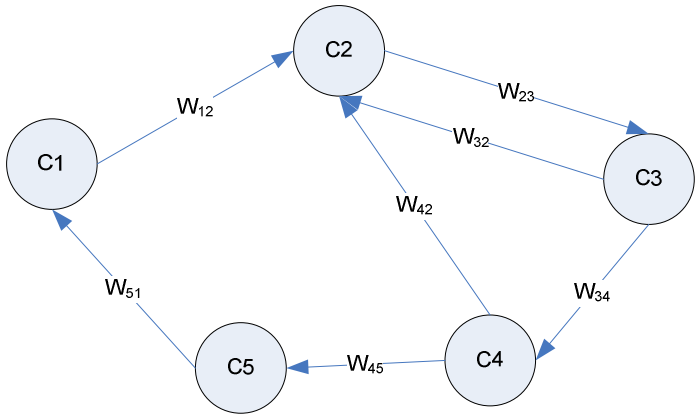

Figure 5.8: A simple Fuzzy Cognitive Map

A Fuzzy Cognitive Map consists of nodes-concepts and arcs between concepts. Each concept represents a characteristic of the system; in general it stands for

events, actions, goals, values, trends of the system that is modeled as an FCM. Each concept is characterized by a number A_i that represents its value and it results from the transformation of the real value of the system's variable, for which this concept stands.

Between concepts, there are three possible types of causal relationships that express the type of influence from one concept to the others. The weights of the arcs between concept C_i and concept C_j could be positive ($W_{ij} > 0$) which means that an increase in the value of concept C_i leads to the increase of the value of concept C_j , and a decrease in the value of concept C_i leads to the decrease of the value of concept C_j. Or there is negative causality ($W_{ij} < 0$) which means that an increase in the value of concept C_i leads to the decrease of the value of concept C_j and vice versa. Or there is no causality ($W_{ij} = 0$) which means that an increase or decrease in the value of concept C_i has no any influence on the value of concept C_j

In addition to the graphical form of the FCM there is its algebraic representation. It consists of a $1 \times n$ state vector **A** which includes the values of the n concepts and an $n \times n$ weight matrix **W** which gathers the weights W_{ij} of the interconnections between the n concepts of the FCM. The matrix **W** has n rows and n columns where n equals the total number of distinct concepts of the FCM and the matrix diagonal is zero since it is assumed that no concept causes itself.

The value of each one concept is influenced by the values of the connected concepts with the appropriate weights and by its previous value. So the value A_i for each concept C_i is calculated by the following rule:

$$A_i = f\left(\sum_{\substack{j=1 \\ j \neq i}}^{n} A_j W_{ji}\right) + A_i^{old}, \quad (1)$$

where A_i is the activation level of concept C_i at time t+1, A_j is the activation level of concept C_j at time t, A_i^{old} is the activation level of concept C_i at time t, and W_{ji} is the weight of the interconnection between C_j and C_i, and f is a threshold function.

$$\mathbf{A}_{new} = f(\mathbf{A}_{old} \circ \mathbf{W}) + \mathbf{A}_{old} \quad (2)$$

So the new state vector $\mathbf{A}_{new}$ is computed by multiplying the previous state vector $\mathbf{A}_{old}$ by the weight matrix **W**. The new vector shows the effect of the change in the value of one concept in the whole Fuzzy Cognitive Map. But, equation (2) includes also, the old value of each concept, and so the FCM possesses memory capabilities and there is a smooth change after each new cycling of the FCM.

The Fuzzy Cognitive Map is a useful method in modeling and control of complex systems which will help the designer of a system in decision analysis and strategic planning. It appears to be an appealing tool in the description of the supervisor of complex control systems, which can be complemented with other techniques and will lead to more sophisticated control systems [SGG].

5.8.2 Trust Control Modeling

The Fuzzy Cognitive Map is a good method to analyze systems that are otherwise difficult to comprehend due to the complex relationships between their components. In this section, we introduce an adaptive trust control model via applying the theory of the FCM in order to illustrate the relationships among trust, its influencing

factors and the control modes used for managing it. Let us first introduce some notations.

Notation 2:

QA_i	the *i*th quality attribute;
C_j	the *j*th control mode;
w_i	the importance rate of QA_i;
V_{QA_i}	the value of QA_i;
V_{C_j}	the value of C_j;
T	the value of trustworthiness;
f	the sigmoid threshold function;
cw_{ji}	the influence factor of control mode C_j on QA_i;
B_{Cj}	the selection factor of the control mode C_j;
T^{old}	the old value of trustworthiness;
$V_{QA_i}^{old}$	the old value of QA_i;
$V_{C_j}^{old}$	the old value of C_j;
ΔT	the change of trustworthiness value;
S_k	the *k*th composition of control modes;
tr	the selection threshold;
SF_{S_k}	the selection factor of S_k;
T_k	the trustworthiness value regarding S_k;
$V_{QA_i,k}$	the value of QA_i regarding S_k;
δ	the accepted change of trustworthiness value;
ω	is a unit deduction factor;
$V_{QA_i}\_monitor$	the value of QA_i generated through real system observation;
$V_{QA_i}\_predict$	the value of QA_i generated by prediction;
n	the total number of quality attributes;
m	the total number of control modes;
K	the total number of the composition of control modes;
σ	the accepted error between $V_{QA_i}\_monitor$ and $V_{QA_i}\_predict$;
d_k	the distance of $V_{QA_i,k}$ and T to tr;

A platform entity's trustworthiness is influenced by a number of quality attributes $QA_i (i = 1,...,n)$. These quality attributes are ensured or controlled through a number of control modes supported by the platform system $C_j (j = 1,...,m)$. A control mode contains a number of control mechanisms or operations that can be provided by the system. We assume that the control modes are not exclusive and that combinations of different modes are used.

The model can be described with a graphical illustration using a Fuzzy Cognitive Map, as shown in Figure 5.9. There are three layers of nodes in the graph. The node in the top layer is the trustworthiness of the platform entity. The nodes located in the middle layer are the quality attributes of the entity, which have direct influence on the entity's trustworthiness. The nodes at the bottom layer are

control modes that could be supported and applied inside the system. These control modes can control and thus improve the quality attributes. Therefore, they have indirect influence on the trustworthiness of the entity.

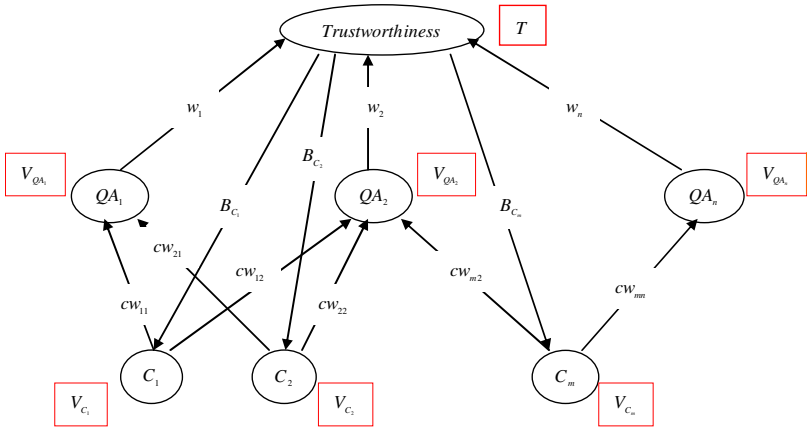

Figure 5.9: Graphical modeling of trust control

Note that $V_{QA_i}, V_{C_j}, T \in [0,1]$, $w_i \in [0,1]$, and $cw_{ji} \in [-1,1]$. T^{old}, $V_{QA_i}^{old}$ and $V_{C_j}^{old}$ are old value of T, V_{QA_i}, and V_{C_j}, respectively. $\Delta T = T - T^{old}$ stands for the change of trustworthiness value. B_{C_j} reflects the current system configuration on which control modes are applied. The trustworthiness value T can be described as:

$$T = f\left(\sum_{i=1}^{n} w_i V_{QA_i} + T^{old}\right), \quad (3)$$

such that $\sum_{i=1}^{n} w_i = 1$. Where w_i is a weight that indicates the importance rate of the quality attribute QA_i regarding how much this quality attribute is considered at the trust decision or assessment. The weight w_i can be decided based on the trustor's criteria. We apply the Sigmoid function as the threshold function *f*: $f(x) = \frac{1}{1+e^{-\alpha x}}$ (e.g. $\alpha = 2$), to map node values V_{QA_i}, V_{C_j}, T into [0, 1]. The value of the quality attribute is denoted by V_{QA_i}. It can be calculated according to the following formula:

$$V_{QA_i} = f\left(\sum_{j=1}^{m} cw_{ji} V_{C_j} B_{C_j} + V_{QA_j}^{old}\right), \quad (4)$$

where cw_{ji} is the influence factor of control mode C_j on QA_i, cw_{ji} is set based on the impact of C_j on QA_i. Positive cw_{ji} means a positive influence of C_j on QA_i. Negative cw_{ji} implies a negative influence of C_j on QA_i. B_{Cj} is the selection factor of the control mode C_j, which can be either 1 if C_j is applied or 0 if C_j is not applied.

The value of the control mode can be calculated using

$$V_{C_j} = f\left(T \cdot B_{C_j} + V_{C_j}^{old}\right), \quad (5)$$

where T is the value of trustworthiness and B_{Cj} is the selection factor of the control mode C_j.

5.8.3 Trust Control Mode Prediction and Selection

The control modes are predicted through evaluating all possible modes and their compositions based on the proposed model using the prediction algorithm described below. As a standard for predicting new modes, we introduce a constant δ, which is the accepted ΔT that controls the iteration of the prediction.

- For every composition of control modes, i.e. $\forall S_k (k = 1,..., K)$, while $\Delta T_k = T_k - T_k^{old} \geq \delta$, do

  $V_{C_j,k} = f\left(T_k \cdot B_{C_j,k} + V_{C_j,k}^{old}\right)$

  $V_{QA_i,k} = f\left(\sum_{j=1}^{m} cw_{ji} V_{C_j,k} B_{C_j,k} + V_{QA_j,k}^{old}\right)$

  $T_k = f\left(\sum_{i=1}^{n} w_i V_{QA_i,k} + T_k^{old}\right)$

  The control modes are selected based on the control mode prediction results:

  - Calculate selection threshold $tr = \frac{1}{K}\sum_{k=1}^{K} T_k$;
  - Compare $V_{QA_i,k}$ and T_k of S_k to tr, set selection factor $SF_{S_k} = 1$ if $\forall V_{QA_i,k} \geq tr \wedge T_k \geq tr$; set $SF_{S_k} = -1$ if $\exists V_{QA_i,k} < tr \vee \exists T_k < tr$;
  - For $\forall SF_{S_k} = 1$, calculate the distance of $V_{QA_i,k}$ and T_k to tr as $d_k = \min\{|V_{QA_i,k} - tr|, |T_k - tr|\}$; For $\forall SF_{S_k} = -1$, calculate the distance of $V_{QA_i,k}$ and T_k to tr as $d_k = \max\{|V_{QA_i,k} - tr|, |T_k - tr|\}$ only when $V_{QA_i,k} < tr$ and $T_k < tr$;
  - If $\exists SF_{S_k} = 1$, select the best winner with the biggest d_k; else $\exists SF_{S_k} = -1$, select the best loser with the smallest d_k.

Herein, the selection threshold (tr) is the average of trust value T_k of all $S_k (k = 1,..., K)$, i.e. $tr = \frac{1}{K}\sum_{k=1}^{K} T_k$. $S_k (k = 1,..., K)$ can be expressed by the control mode selection factors B_{C_j}, which represent which control mode is selected and applied in the system. The selection factor $SF_{S_k} = 1$ means that all the predicted $V_{QA_i,k}$ and T_k are above the threshold tr. While $SF_{S_k} = -1$ means there is some predicted $V_{QA_i,k}$ and T_k below the threshold tr. The selection algorithm selects the best control modes based on the absolute difference between $V_{QA_i,k}$, T_k and tr. For $\forall SF_{S_k} = 1$, it records the absolute difference between $V_{QA_i,k}$, T_k and tr as the minimum $d_k = \min\{|V_{QA_i,k} - tr|, |T_k - tr|\}$. For $\forall SF_{S_k} = -1$, it records the absolute difference between $V_{QA_i,k}$, T_k and tr as the maximum $d_k = \max\{|V_{QA_i,k} - tr|, |T_k - tr|\}$, only when $V_{QA_i,k} < tr$ and $T_k < tr$. Thus, the algorithm can select the best winner if $\exists SF_{S_k} = 1$. Even though, there is no choice available, it is also possible for the algorithm to

select the best loser with the biggest $V_{QA_i,k}$ and T_k below tr. Selecting the best loser is significant for the system to optimize the configurations of the control modes in order to re-predict and re-select a proper set of control modes.

5.8.4 Adaptive Trust Control Model Adjustment

It is important for the trust control model to reflect the real system situation and context precisely. The influencing factors of each control mode should be context-aware. The trust control model should be dynamically maintained and optimized in order to reflect the real system situation. Thereby, it is sensitive to indicate the influence of each control mode on different quality attributes in a dynamically changed context. For example, when some malicious behaviors or attacks happen, the currently applied control modes can be found not feasible based on trust assessment. In this case, the influencing factors of the applied control modes should be adjusted in order to reflect the real system situation. Then, the system can automatically re-predict and re-select a set of new control modes in order to ensure the trustworthiness. In this way, the system can avoid using attacked or useless trust control modes in a special context. As can be seen from the above analysis, an adaptive trust control model is vital for supporting autonomic trust management in a component software platform.

We apply observation based trust assessment as described in Section 5.7, which can play as the feedback for adaptive model adjustment. Herein, we use $V_{QA_i}\_monitor$ and $V_{QA_i}\_predict$ to stand for V_{QA_i} generated based on real system observation (i.e. the trust assessment result) and by prediction, respectively. Concretely, the influencing factor cw_{ji} can be further adjusted based on two schemes in order to make it match real system situation. One of the schemes is an equal adjustment scheme. It holds a strategy that each control mode has the same impact on the deviation between $V_{QA_i}\_monitor$ and $V_{QA_i}\_predict$. In this scheme, all related cw_{ji} will be adjusted equally. The other is an unequal adjustment scheme. It holds a strategy that the control mode with the biggest absolute influencing factor always impacts more on the deviation between $V_{QA_i}\_monitor$ and $V_{QA_i}\_predict$. In this scheme, we always select the biggest absolute influencing factor to adjust. Which one should be applied depends on experimental experience on the control mode's influence on the quality attributes. In the schemes, ω is a unit deduction factor and σ is the accepted deviation between $V_{QA_i}\_monitor$ and $V_{QA_i}\_predict$. We suppose C_j with cw_{ji} is currently applied in the system. The equal adjustment scheme is:

- While $\left|V_{QA_i}\_monitor - V_{QA_i}\_predict\right| > \sigma$, do

a) If $V_{QA_i}\_monitor < V_{QA_i}\_predict$, for $\forall cw_{ji}$,

$cw_{ji} = cw_{ji} - \omega$, if $cw_{ji} < -1, cw_{ji} = -1$;

Else, for $\forall cw_{ji}$,

$cw_{ji} = cw_{ji} + \omega$, if $cw_{ji} > 1, cw_{ji} = 1$;

b) Run the control mode prediction function.

The unequal adjustment scheme is the following:

- While $\left|V_{QA_i}\_monitor - V_{QA_i}\_predict\right| > \sigma$, do

a) If $V_{QA_i}\_monitor < V_{QA_i}\_predict$, for $\max(\left|cw_{ji}\right|)$,

$cw_{ji} = cw_{ji} - \omega$, if $cw_{ji} < -1, cw_{ji} = -1$ (warning);

Else, $cw_{ji} = cw_{ji} + \omega$, if $cw_{ji} > 1, cw_{ji} = 1$ (warning);

b) Run the control mode prediction function.

5.8.5 Examples and Simulation Results

We conducted simulations to prove the above modeling and algorithms. The simulations are based on a practical example, as shown in Figure 5.10. The trustworthiness of the trustee is influenced by three quality attributes: QA_1 - Security; QA_2 - Availability; QA_3 - Reliability, with important rates $w_1 = 0.6$, $w_2 = 0.2$, and $w_3 = 0.2$, respectively. There are three control modes that could be provided by the system:

- C_1: security mode 1 with light encryption and light negative influence on availability.
- C_2: security mode 2 with strong encryption, but medium negative influence on availability.
- C_3: fault management mode with positive improvement on availability and reliability.

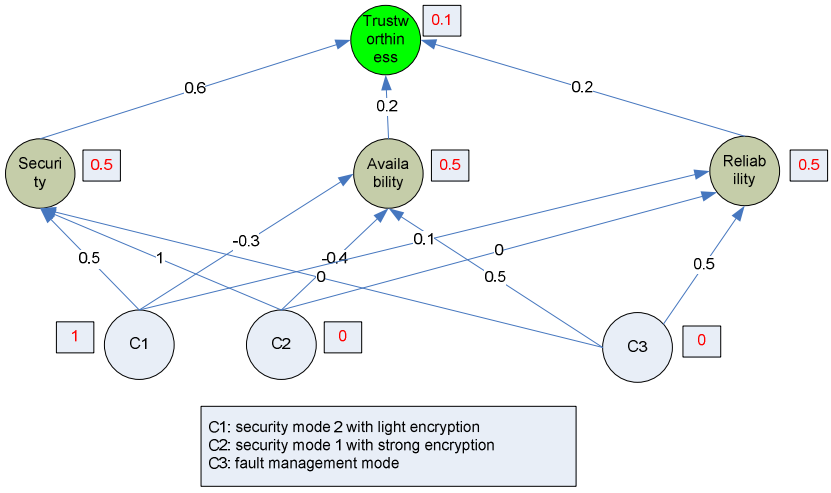

Figure 5.10: Simulation configurations

The influence of each control mode on the quality attributes is specified by the arc weights. Their initial values can be set based on the experimental results tested at the control mode development. The values in the square boxes are initial values of the nodes. In practice, the initial values can be set as asserted ones or expected ones, which are specified in the trustor's criteria profile. Actually, the initial values have no influence on the final results of the prediction and selection based on the simulation experience.

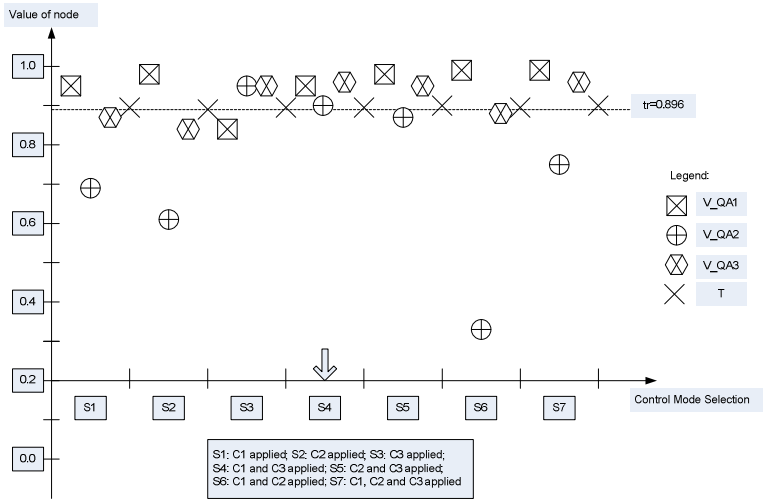

Figure 5.11: Control mode prediction and selection result ($\alpha = 2$ and $\delta = 0.0001$)

The simulation results are shown in Figure 5.11. In this case, there are seven control mode compositions: S_1 ($B_{C_1} = 1; B_{C_2} = 0; B_{C_3} = 0$); S_2 ($B_{C_1} = 0; B_{C_2} = 1; B_{C_3} = 0$); S_3 ($B_{C_1} = 0; B_{C_2} = 0; B_{C_3} = 1$); S_4 ($B_{C_1} = 1; B_{C_2} = 0; B_{C_3} = 1$); S_5 ($B_{C_1} = 0; B_{C_2} = 1; B_{C_3} = 1$); S_6 ($B_{C_1} = 1; B_{C_2} = 1; B_{C_3} = 0$); S_7 ($B_{C_1} = 1; B_{C_2} = 1; B_{C_3} = 1$). We can see that S_4 (the composition of C_1 and C_3) is the best choice since both the quality attribute values and the trustworthiness value are above the threshold.

If S_4 is applied but the assessed values of quality attributes based on runtime observation are not the same as the predicted ones (e.g. $V_{QA_1}\_predict = 0.946$, $V_{QA_2}\_predict = 0.899$; $V_{QA_3}\_predict = 0.956$), the trust control model should be adjusted in order to reflect real system context. Supposed that the assessed $V_{QA_i}\_monitor$ are: $V_{QA_1}\_monitor = 0.92$, $V_{QA_2}\_monitor = 0.70$, and $V_{QA_3}\_monitor = 0.956$. In this case, the security attribute is a bit worse than prediction and the availability attribute is definitely predicted incorrectly. The mismatch indicates that the underlying model parameters do not reflect real system situation precisely. This could be caused by some attacks happening at the control mechanisms in S_4 with regard to ensuring the availability, or raised by limited resources shared by many system entities, or due to weaker influence of S_4 on the availability in practice than prediction. We conducted model adjustment based on the equal and unequal schemes, respectively. The adjustment simulation results are shown in Table 5.1. Both schemes can adjust the model with similar predicted V_{QA_i} to the assessment results, as shown in Table 5.2. The deviation between $V_{QA_1}\_predict$ and $V_{QA_i}\_monitor$ can be controlled through parameter σ. As can be seen from the simulation results, both schemes can adjust the influencing factors to make the prediction values of QA $V_{QA_1}\_predict$ match the assessment results $V_{QA_i}\_monitor$ generated through observation.

Table 5.1: Trust control model adjustment results ($\sigma = 0.002, \omega = \sigma / 20$)

Influencing factors cw_{ji}	Original values of cw_{ji}	Adjusted values of cw_{ji} based on equal adjustment scheme	Adjusted values of cw_{ji} based on unequal adjustment scheme
cw_{11}	0.5	0.41	0.32
cw_{12}	-0.3	-0.54	-0.58
cw_{13}	0.1	0.1	0.1
cw_{21}	1.0	1.0	1.0
cw_{22}	-0.4	-0.4	-0.4
cw_{23}	0.0	0.0	0.0
cw_{31}	0.0	-0.089	0.0
cw_{32}	0.5	0.26	0.30
cw_{33}	0.5	0.5	0.5

Table 5.2: Prediction results after model adjustment

QA names	Old prediction values	Predicted values after applying equal adjustment scheme	Predicted values after applying unequal adjustment scheme
QA_1	0.9463	0.9220	0.9220
QA_2	0.8993	0.7015	0.7015
QA_3	0.9563	0.9563	0.9563

We further run the control mode prediction and selection functions with two sets of adjusted cw_{ji} listed in Table 5.1, respectively. The results show the system can not offer a good selection. This means that the system needs to re-configure its control modes in order to improve its trustworthiness. In both cases, the selection function indicates the best loser is S_3. We can further optimize the control model S_3 for providing satisfied trust management configuration. The prediction and selection results after the model adjustment are shown in Figure 5.12.

Furthermore, the simulation results show that the initial values of nodes have no influence on the simulation results. The importance rates have an impact on the final values of trustworthiness, and thus influence the control mode prediction and selection since the threshold for selection is the average of the trustworthiness values. The performance of the control model prediction is influenced by the number of control modes (i.e. *K*) and α. Generally, we expect $K \leq 4$, otherwise the performance of the prediction could be low. That is also an important reason we apply the trust control mode, not the concrete trust control mechanisms into the model. In addition, the prediction results could also help optimize the configurations of the control modes. This is because the prediction results indicate the values of quality attributes. If some value of a quality attribute is below the threshold, the platform needs to re-configure the control modes in order to have more positive influence on this quality attribute. This is very useful if there is no solution from the control mode selection. In particular, this trust control model can also be applied to compare the feasibility or quality of different control mode configurations based on the prediction. With regard to the adaptive model

adjustment, both schemes can adjust the influencing factors to make the prediction values of QA match the assessment results generated through observation.

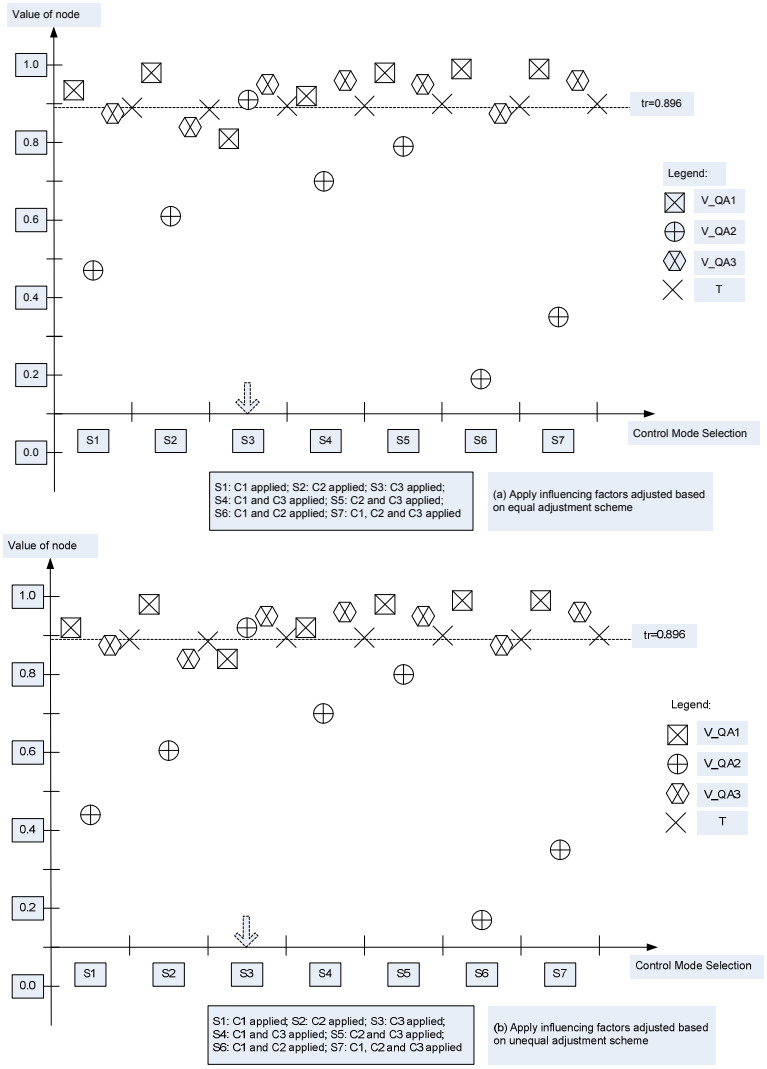

Figure 5.12: Control mode prediction and selection results after model adjustment ($\alpha = 2$ and $\delta = 0.0001$) (a) model adjusted based on equal adjustment scheme; (b) model adjusted based on unequal adjustment scheme

5.8.6 Further Discussion

Cooperation with trust assessment

The adaptive trust control model will cooperate with the runtime trust assessment mechanism. In Section 5.7, we proposed a Subjective Logic based trust assessment mechanism. The output of trust assessment is a set of values which are expressed as opinions ($V_{QA_i} = (b_i, d_i, u_i)$ and $T = (b, d, u)$).

In order to conduct trust control model adjustment, we need to map assessed values based on observation from three dimensions to a single dimension. We suggest to use formula $V_{QA_i} = \alpha b_i + u_i$ and $T = \alpha b + u$, where $\alpha = 0.5$, as defined in [Jos01]. Apart from this, the mapped evaluation values need further transfer to the model values through using sigmoid function $f(x) = \frac{1}{1+e^{-\alpha x}}$ (e.g. $\alpha = 2$) before running the model adjustment function.

In addition, the proposed adaptive trust control model can cooperate with various trust evaluation mechanisms. What we need to do is to map the evaluation values to the value scope of the proposed model.

Two adjustment schemes

We proposed two adjustment schemes. Both can adjust the model to reflect a real system situation. The equal adjustment scheme holds a strategy that each control mode has the same impact on the deviation between $V_{QA_i}\_monitor$ and $V_{QA_i}\_predict$. While the unequal adjustment scheme holds a strategy that the control mode with the biggest absolute influencing factor always impacts more on the deviation between $V_{QA_i}\_monitor$ and $V_{QA_i}\_predict$. Which one should be applied depends on experimental experiences on the control mode's influence on the different quality attributes. We can set an indicator for each control mode to specify the preference. In particular, if $cw_{ji} = 0$ means no influence on QA_i, the unequal adjustment scheme is preferred.

Resource consideration

For some devices with limited resources, we should add additional checking steps in the implementation regarding resource management. Two checks are needed. One is conducted before running the prediction functions in order to find all possibly supported control modes. The other check is needed after the selection in order to ensure the resources required by the selected control modes can be satisfied by the system. If not, we could select the second best solution. Otherwise, the system will raise a warning.

Effectiveness of trust control model

The adaptive trust control model is proposed in order to support autonomic trust management for the component software platform. It is crucial that this model is effective. We analyze the effectiveness of the proposed model through four aspects as below.

Correctness: showing how correct or comprehensive the trust model presents the trustworthiness. The proposed model considers not only direct influencing factors of trust, such as multiple quality attributes of the trustee, but also the impact of trust control mechanisms on trustworthiness. It reflects objective factors (e.g. the quality attributes), subjective factors (e.g. the important rates) and context that influence the trust decision. The context is reflected based on the model adjustment according

to the runtime observation results, thus overcome the difficulty of modeling context completely and precisely. This model is an extension of many existing trust models [ZMZ05, ThB06, SYH06, LVW04, ZWW05], thus it is more comprehensive.

Preciseness: showing how close the trust model reflects a real situation. The proposed model can be adjusted based on the runtime trust assessment results. The preciseness of the adjustment can be controlled by the parameter σ, the accepted deviation between $V_{QA_i}\_monitor$ and $V_{QA_i}\_predict$. The smaller σ is, the more precise the model. Herein, we assume that $V_{QA_i}\_monitor$ is a reliable value since it is generated based on runtime system monitoring and the trust assessment mechanism is protected by trusted computing technology.

Robustness: how robust the trust model regarding malicious behaviors, mistakes and various attacks. The gist of our model is that it is adaptively adjusted based on the real system situation or context. Once there are some problems happening that may influence the trustworthiness, this model can be adaptively adjusted to reflect them. It will be possible to learn the performance or effectiveness of the applied control modes through this model. Thereby, it is possible to re-select other control modes to re-establish or ensure the trustworthiness. Even though there is no solution for trust management, the platform user will be informed. We note that the robustness is also related to the control mode configurations. This model could additionally help the platform administrator optimizing the configurations of the control modes.

Adaptability (Timeliness): how fast the model can reflect the real situation and act accordingly. The proposed model can be dynamically maintained according to the real system context. For example, new control modes can be added and ineffective ones can be removed. The parameters of the model (e.g. cw_{ji}) can be adjusted based on the model adjustment result. The adaptability is controlled by the parameters α, K, δ, σ, and $\omega = \sigma / 20$. The parameters α, K and δ influence the speed of prediction. The smaller the parameter α, K and/or the bigger the parameter δ are, the faster the speed of prediction. But generally, δ can not be set very big since it will influence correctness. The parameter K can not be set very big since it will impact the prediction performance. Our suggested value is $K \leq 4$. The parameters σ and ω are applied to control the speed of the model adjustment. The bigger the parameter σ is, the faster the adjustment, but the worse the preciseness of the adjustment. With regard to the parameter ω, the bigger it is, the faster the adjustment. But ω can not be set too big since this may lead to missing a solution (i.e. the algorithm cannot provide an adjustment). Based on our simulation, we suggest setting $\omega = \sigma / 20$. We should select σ properly in order to keep preciseness and meanwhile ensure adaptability. In summary, adaptability is the most important factor that influences the effectiveness of the trust control model.

Usability: how usable is the model with regard to human-computer interaction in practice. The proposed model applies a number of subjective policies of the trustor (e.g. the importance rates and the trust threshold). Generally, the trustor can be a system user, the system or a component. If the trustor is a user, he/she can set the

policy through a user interface based on a default or suggested setting. If the trustor is the system, it can get a certified policy from a Third Trusted Party (TTP). If the trustor is a component, it can set its policy based on the *CTM* of the trustor component and the trustee component. There are many issues about usability, which could be an interesting topic worth further study.

5.9 Remarks

The methodologies described in Chapter 3 also instruct working out an autonomic trust management solution for the component software platform.

By using the modeling methodology introduced in Section 3.2, the system can be modeled as a number of trusted domains – system entities. In order to build up the trust relationship among these entities, we applied both a 'hard trust' solution and a 'soft trust' solution.

The 'hard trust' solution uses an embedded trust management framework that plays as the trustor entity's delegate to manage the trustworthiness of the trustee entity. This trust management framework also supports applying a number of trust control mechanisms that can be used to ensure or sustain the trust relationships among the system entities. One important category of the trust control mechanisms is security related mechanisms, which include such mechanisms as encryption, decryption, access control mechanisms, authentication, hash code based integrity check, etc.

Regarding the 'soft trust' solution, the trust assessment mechanism embedded in the decision/reason engine can assess the trustworthiness of a specified trustee entity based on runtime observation. In addition, the control mode prediction and selection mechanisms and the mechanisms for adaptive trust control model adjustment that are embedded in the selection engine can further support and enhance the autonomic management for the platform trustworthiness. This solution falls into approach (b) – create new component: a trust management framework to support autonomic trust management. In practice, this framework cooperates with other frameworks (e.g. resource management framework and security framework) to realize the whole system's trust management. Therefore, this work is a significant attempt regarding the three emerging trends of trust modeling and management.

The work described in Chapter 4 aims to improve the existing trusted computing technology at the platform layer of the mobile computing platform. It can provide a secure and integrated environment for the component software platform that is installed and run at the platform middleware layer. The mechanism for trust sustainability could be used to restrict any malicious changes that may influence the security of the component software platform.

5.10 Summary

This Chapter presented an autonomic trust management solution for a component software platform. It is also a significant attempt with regard to the emerging trends described in Chapter 2. We have identified the trust issues in the component

software system and briefly introduced our approach. We then developed a formal trust model to specify, evaluate, set up and ensure trust relationships amongst system entities. Based on this trust model, we further designed an autonomic trust management architecture that can adopt a number of algorithms to implement autonomic trust management. This design is compatible with the component software system architecture. Thus it can be easily deployed in practice in order to enhance the trustworthiness of the component software. In addition, the proposed trust management architecture will enable trust management for both system users and system internal entities since it supports managing the trust relationship between any two entities in the component software system. Therefore, it supports trust management for 'All'. A more extended discussion is provided in Publication 7 and [YPN07].

Furthermore, we proposed an adaptive trust control model for the trust control mode prediction and selection aiming at autonomic trust management for the component software platform. We made use of a Fuzzy Cognitive Map to model the relationships among the trust control modes, the quality attributes of the platform entity and its trustworthiness. In this model, the importance factors are set based on the trustor's preference. The influencing factors of the control modes can be adaptively adjusted according to the trust assessment in order to reflect real system context and situation. Based on this model, we proposed the algorithms to conduct the control mode predication and selection. The simulation results show that this model is effective for predicting and selecting the suitable trust control modes for the system. It also helps improving the control mode configurations, especially when there is no solution from the prediction. In addition, this model is flexible to support any system entity's autonomic trust management. The system entity can be a system component, a sub-system or the whole system. Thereby, this model can also support auto-selection of the trust control modes for all system entities. A more extended discussion of the issues is provided in Publication 8 and [YaP07].

Desirable emerging properties can be obtained by applying the proposed trust management solution into a mobile computing platform with component software support. These include enabling the trust assessment at runtime based on system monitoring of a number of quality attributes of the assessed entity; autonomic trust management on the basis of the performance prediction of trust control modes, the auto-selection of trust control modes and the adaptive adjustment of the trust control model through cooperation with the trust assessment. These emerging properties allow addressing trust at the system runtime better.

6 CONCLUSIONS

Trust is playing and will continuously play an important role in mobile computing and communications. The rapid growth of new mobile networking paradigms and mobile Internet services is introducing new requirements and challenges to mobile computing platforms. Trust management is becoming an important issue for the mobile computing platforms.

This dissertation studies methodologies and mechanisms of providing a trustworthy computing platform for mobile devices. In addition, we seek solutions to support trusted communications and collaboration among those platforms in a distributed and dynamic system. This dissertation contributes in four aspects.

Firstly, it provides a comprehensive review of trust, trust modeling, trust evaluation and trust management. Based on the study of the state-of-the art, it identifies the emerging trends. The understanding generated from the literature study instructs our work towards solving special issues of trust regarding the mobile computing platforms.

Secondly, this dissertation studies research methodologies for the purpose of our research objectives. We presented a conceptual architecture to clarify the structure of trust issues in the mobile domain and specify a number of key motivations. We also introduced a methodology to bridge the domains of trust in mobile computing and communications. This methodology benefits system analysis and design for finding trust issues and identifying security problems. The concrete approaches for bridging the trust gaps among domains instruct how to seek a concrete solution regarding trust management for mobile computing platforms. Furthermore, we applied our methodologies into practice to demonstrate their applicability, expressiveness and advantages.

Thirdly, the dissertation presents a mechanism to sustain trust among computing platforms on the basis of Root Trust (RT). It creates trust based on the attestation of the RT module at the trustee and controls its sustainability according to the pre-defined conditions. Those conditions are approved by both the trustor and the trustee at the time of trust establishment and enforced through the use of the pre-attested RT module until the intended purpose is fulfilled. This work extends the trust model from static to dynamic. Thus, it develops the notion of using trust management not only for trust assessment but also for trust sustainability. The proposed mechanism could be applied in many real applications for trusted services and communications, for example, trust collaboration in P2P systems and trust management in mobile enterprise networking. It could work as an extension of future TC platform to support various applications with enhanced flexibility.

Finally, the dissertation develops an autonomic trust management solution for a component software system. It is also a significant attempt with regard to the three emerging trends. We developed a formal trust model to specify, evaluate, set up and ensure trust relationships amongst system entities. Based on this trust model, we designed an autonomic trust management architecture that can adopt a number of algorithms for autonomic trust management. These algorithms are designed based on an adaptive trust control model. Desirable emerging properties can be obtained by applying the proposed trust management solution into a mobile computing

platform with component software support. These include enabling the trust assessment at runtime based on system monitoring of a number of quality attributes of the assessed entity; autonomic trust management on the basis of the performance prediction of trust control modes, the auto-selection of trust control modes and the adaptive adjustment of the trust control model through cooperation with the trust assessment. These emerging properties allow addressing trust at the component software system runtime better.

Particularly, the proposed methodologies and mechanisms in this dissertation can cooperate altogether to provide a meaningful approach to manage trust of mobile computing platforms. First of all, the conceptual architecture based research method provides us a clear guideline of research steps and helps us analyze research motivations in the area of mobile computing. At the design stage of a trustworthy mobile system or platform, the methodology for bridging different domains of trust can help system designers working out a trustworthy system architecture that can overcome potential trust deficiencies among system components [YaZ07]. In order to support trustworthy collaborations and communications among mobile computing platforms in a mobile system, the mechanism to sustain trust among computing platforms on the basis of Root Trust can be applied to satisfy the trust requirements and conditions of the trustor platform on the trustee platform. Furthermore, the approach proposed for autonomic trust management can be deployed for building up a trustworthy platform through autonomic platform management on any trust relationship among platform entities based on the platform's competence according to high-level trust policies. From trust problem discovery, the analysis and design of a trustworthy mobile system to trustworthy mobile platform or system execution and collaboration, this dissertation provides a series of methodologies or mechanisms for managing trust for mobile computing platforms in a dynamically changed environment.

More importantly, the mechanisms proposed in Chapter 4 and Chapter 5 can cooperate together to realize autonomic trust management across mobile computing platforms [YaP07a]. In this case, the trust conditions are set based on the trust policies or criteria specified by the trustor entity located in one mobile computing platform, while the trustee entity is located in another remote platform. The trust policies or criteria are registered at the trustee platform's trust management framework and ensured through the trust sustaining mechanism based on the Root Trust module. The autonomic trust management on the trustee entity is conducted based on the approach proposed in Chapter 5 according to the trustee platform's competence. If trust can not be managed, the trustor entity could be notified. This integrated approach is significant to automatically manage holistic trust reflected by multiple quality attributes for mobile Internet services and applications. In addition, the proposed methodologies and mechanisms in this dissertation can be further extended to apply into any digital computing platform or system.

Notably, this dissertation proposed two methods for trust sustainability. One is Root Trust module based 'hard trust' solution. This method is more suitable for supporting trustworthy communications or collaborations among a number of mobile computing platforms in a distributed system. Another method is trust evaluation based autonomic trust management, which is a 'soft trust' solution. This method provides intelligence to ensure an open platform's trustworthiness according to the platform's competence. These two methods can be applied

independently or cooperate together to make the mobile computing platforms more trustworthy.

Trust management for mobile computing platforms is a wide research area. This dissertation mainly focuses on discussing several important topics related to it. There are still many interesting issues worth further study and deep research. In what follows, we propose several pieces of work we would like to conduct in the near future.

Regarding the TC platform based solution for trustworthy communications, immediate future work includes performance study of the proposed mechanism and implementation of the trust sustainability mechanism on a mobile computing platform in order to support mobile P2P collaboration and trusted mobile enterprise networking.

Regarding autonomic trust management for the component software platform, we will further optimize the algorithms and study the performance of model adjustment schemes and attempt to implement the adaptive trust control model and related algorithms in the Trust4All platform. What is more, we will further extend the runtime trust management to the download trust management, and thus achieve a comprehensive trust management solution for component software. On the other hand, how to support multiple trustors' trust management requests is an interesting and crucial issue worthy of special study.

Apart from those mentioned above, we found that little work has been conducted regarding human-device interaction in order to support trust management, especially in the mobile domain. Embedding personal criteria of trust regarding different events into the device requires interaction between the end user and his/her device. This would require a friendly user interface for the device to collect useful information for trust evaluation and present the evaluation results in a comprehensive manner to the user. Human-device interaction for trust management could be an interesting research topic worth our efforts.

REFERENCES

[AbD01] K. Aberer, and Z. Despotovic, "Managing trust in a peer-to-peer information system," in *Proceedings of ACM Conf. Information and Knowledge Management (CIKM)*, 2001.

[ALR04] A.Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Transactions on Dependable and Secure Computing*, Vol. 1, Issue 1, pp.11-33, Jan. 2004.

[BaS03] A. Baldwin, and S. Shiu, "Hardware security appliances for trust," in *Proceedings of the First International Conference of Trust Management* (iTrust 2003), LNCS 2692, pp. 46-58, Crete, Greece, May 2003.

[BFL96] M. Blaze, J. Feigenbaum, and J. Lacy, "Decentralized trust management," in *Proceedings of IEEE Symposium on Security and Privacy*, pp.164-173, May 1996.

[BIK02] M. Blaze, J. Ioannidis, and A. D. Keromytis, "Trust management for Ipsec," *ACM Transactions on Information and System Security*, Vol. 5, No. 2, pp.95-118, May 2002.

[BIK03] M. Blaze, J. Ioannidis, and A. D. Keromytis, "Experience with the KeyNote trust management system: applications and future directions," in *Proceedings of the First International Conference of Trust Management* (iTrust 2003), LNCS 2692, pp. 284-300, Crete, Greece, May 2003.

[BMM05] S. Banerjee, C. A. Mattmann, N. Medvidovic, and L. Golubchik, "Leveraging architectural models to inject trust into software systems", ACM SIGSOFT Software Engineering Notes , in *Proceedings of the 2005 workshop on software engineering for secure systems—building trustworthy applications SESS '05*, Vol. 30, Issue 4, 2005.

[BoH91] S. Boon, and J. Holmes, "The dynamics of interpersonal trust: Resolving uncertainty in the face of risk," in R. Hinde, 1991.

[CFP03] C. Castelfranchi, R. Falcone, and G. Pezzulo, "Integrating trustfulness and decision using fuzzy cognitive maps," in *Proceedings of the First International Conference of Trust Management* (iTrust 2003), LNCS 2692, pp. 195-210, Crete, Greece, May 2003.

[ChM02] K. H. Cheung, and J. Misic, "On virtual private network security design issues," *Computer Networks*, Vol. 38, No. 2, pp. 165-179, Feb. 2002.

[ChW03] K. Chopra, and W. A. Wallace, "Trust in electronic environments," in *Proceedings of the 36th Hawaii International Conference on System Sciences* (HICSS'03), 2003.

[CKW03] C. L. Corritore, B. Kracher and S. Wiedenbeck, "On-line trust: concepts, evolving themes, a model," *International Journal of Human-Computer Studies*, Trust and Technology, Vol. 58, Issue 6, pp. 737-758, June 2003.

[Dav02] P. T. Davis, "TCPA: who can you trust," *EDPACS: the EDP Audit, Control and Security Newsletter*, Dec. 2002.

[Den93] DE Denning, "A new paradigm for trusted systems", in *Proceedings of the IEEE New Paradigms Workshop*, 1993.

[Dey01] A. K. Dey, "Understanding and using context", *Personal and Ubiquitous Computing Journal*, Vol. 5, pp. 4-7, 2001.

[ELM03] P. England, B. Lampson, J. Manferdelli, M. Peinado, and B. Willman, "A trusted open platform," *IEEE Computer Society*, pp. 55-62, July 2003.

[FaC05] R. Falcone, and C. Castelfranchi, "Socio-cognitive model of trust," *Encyclopedia of Information Science and Technology*, pp. 2534-2538, 2005.

[FDK02] P. Fenkam, S. Dustdar, E. Kirda, G. Reif, and H. Gall, "Towards an access control system for mobile peer-to-peer collaborative environments", in *Proceedings of Eleventh IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pp. 95-100, June 2002.

[Fel03] E. W. Felten, "Understanding trusted computing: will its benefits outweigh its drawbacks?" *IEEE Security & Privacy*, Vol.1, No.3, pp. 60-62, May-June 2003.

[Gam90] D. Gambetta, "Can We Trust Trust?" In, Trust: Making and breaking Cooperative Relations, Basil Blackwell, Oxford, 1990.

[GrS00] T. Grandison, and M. Sloman, "A survey of trust in internet applications," *IEEE Communications and Survey*, Fourth Quarter, 3(4), pp. 2-16, 2000.

[GrS03] T. Grandison, and M. Sloman, "Trust management tools for internet applications," in *Proceedings of the First International Conference of Trust Management* (iTrust 2003), LNCS 2692, pp. 91-107, Crete, Greece, May 2003.

[GuK04] R. Guha, and R. Kumar, "Propagation of trust and distrust," in *Proceedings of the 13th international conference on World Wide Web*, pp. 403-412, ACM Press, 2004.

[HaD05] T. Hardjono, and L. R. Dondeti, "Security in Wireless LANS and MANS (Artech House Computer Security)", Artech House Inc., 2005.

[HAM05] H. Hamed, E. Al-Shaer, and W. Marrero, "Modeling and verification of IPSec and VPN security policies", *13th IEEE International Conference on Network Protocols*, pp. 259 – 278, Nov. 2005.

[HaR06] W. Hasselbring, and R. Reussner, "Toward trustworthy software systems," *IEEE Computer*, Vol. 39, Issue 4, pp. 91-92, April 2006.

[Her99] E. Herscovitz, "Secure virtual private networks: the future of data communications," *International Journal of Network Management*, Vol. 9, Issue 4, Aug. 1999.

[Her01] P. Herrmann, "Trust-Based procurement support for software components," in *Proceedings of the 4th International Conference of Electronic Commerce Research* (ICECR04), pp. 505-514, Dallas, 2001.

[Her03] P. Herrmann, "Trust-based protection of software component users and designers," in *Proceedings of the First International Conference of Trust Management* (iTrust 2003), LNCS 2692, pp. 75-90, Crete, Greece, May 2003.

[JHK05] H. Jameel, L. X. Hung, U. Kalim, A. Sajjad, S. Lee; and Y. Lee, "A trust model for ubiquitous systems based on vectors of trust values," in *Proceedings of the seventh IEEE International Symposium on Multimedia*, pp. 674-679, Dec. 2005.

[Hua02] A. B. Huang, "The trusted OC: skin-deep security", *Computer*, Vol. 35, No.10, pp. 103-105, Oct. 2002.

[JIB05] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision Support Systems*, 2005.

[JoK98] A. Jøsang and S. J. Knapskog, "A metric for trusted systems," in *Proceedings of the 21st National Security Conference*, NSA 1998.

[Jos99] A. Jøsang, "An algebra for assessing trust in certification chains," in J.Kochmar, editor, *Proceedings of the Network and Distributed Systems Security Symposium* (NDSS'99), The Internet Society, 1999.

[Jos01] A. Josang, "A logic for uncertain probabilities," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, Vol. 9, No. 3, pp. 279-311, 2001.

[JoT03] A. Jøsang, and N. Tran, "Trust management for e-commerce," in *Proceedings Virtual Banking 2000*, DSTC University report, 2003.

[Kos86] B. Kosko, "Fuzzy Cognitive Maps," *International Journal Man-Machine Studies*, Vol.24, pp. 65-75, 1986.

[KSG03] S. Kamvar, M. Scholsser, and H. Garcia-Molina, "The EigenTrust algorithm for reputation management in P2P networks," in *Proceedings of 12th International Conference of World Wide Web*, May 2003.

[KSP01] G. Kortuem, J. Schneider, D. Preuitt, T. G. C. Thompson, S. Fickas, and Z. Segall, "When peer-to-peer comes face-to-face: collaborative peer-to-peer computing in mobile ad hoc networks," in *Proceedings of the First International Conference on Peer-to-Peer Computing* (P2P01), Aug. 2001.

[LiS05] Z. Liang, and W. Shi, "PET: A PErsonalized trust model with reputation and risk evaluation for P2P resource sharing", in *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, pp. 201b-201b, Jan. 2005.

[LJT04] Z. Liu, A. W. Joy, and R. A. Thompson, "A dynamic trust model for mobile ad hoc networks," in *Proceedings of 10th IEEE International Workshop on Future Trends of Distributed Computing Systems* (FTDCS 2004), pp. 80-85, May 2004.

[LMA02] M. Lik, M. Mojdeh, and H. Ari, "A computational model of trust and reputation," in *Proceedings of the 35th Annual Hawaii International Conference on System sciences*, Jan. 2002.

[LSB03] S. Lee, R. Sherwood, and B. Bhattacharjee, "Cooperative peer groups in NICE," in *Proceedings of IEEE Conference on Computer Communications* (INFOCOM 03), IEEE CS Press, pp. 1272-1282, 2003.

[LVW04] C. Lin, V. Varadharajan, Y. Wang, and V. Pruthi, "Enhancing Grid security with trust management", in *Proceedings of IEEE International Conference on Services Computing* (SCC 2004), pp. 303-310, Sept. 2004.

[Man98] D. W. Manchala, "Trust metrics, models and protocols for electronic commerce transactions,"in *Proceedings of 18th IEEE International Conference on Distributed Computing System*, pp. 312-321, May 1998.

[Man00] D. W. Manchala, "E-commerce trust metrics and models," *IEEE Internet Computing*, Vol.4, No.2, pp. 36-44, 2000.

[Mau96] U. Maurer, "Modeling a public-key infrastructure," in *Proceedings of European Symposium on Research in Computer Security*, LNCS 1146, pp. 325–350, 1996.

[McC00] D. H. McKnight, and N. L. Chervany, "What is trust? a conceptual analysis and an interdisciplinary model," in *Proceedings of the 2000 Americas Conference on Information Systems* (AMCI2000), Aug. 2000.

[McC03] D. H. McKnight, and N. L. Chervany, "The meanings of trust," UMN university report, 2003. http://www.misrc.umn.edu/wpaper/wp96-04.htm

[MDS95] R. C. Mayer, J. H. Davis and F. D. Schoorman, "An integrative model of organizational trust," *Academy of Management Review*, Vol. 20, No 3, pp. 709-734, 1995.

[Mui03] L. Mui. *Computational Models of Trust and Reputation: Agents, Evolutionary Games, and Social Networks*. PhD thesis, Massachusetts Institute of Technology, 2003.

[PaL] Personal contact with Lauri Paatero with regard to the concept of secure environment, 2003.

[PaS02] J. Park, and R. Sandhu, "Towards usage control models: beyond traditional access control," in *Proceedings of the seventh ACM Symposium on Access Control Models and Technologies*, California, USA, 2002.

[Per99] R. Perlman, "An overview of PKI trust models", *IEEE Network*, Vol.13, No.6, pp. 38-43, Nov.-Dec. 1999.

[Reg03] K, Regan, "Secure VPN design considerations," *Network Security*, pp. 5-10, May 2003.

[ReS98] M. K. Reiter, and S. G. Stubblebine, "Resilient authentication using Path independence," *IEEE Transaction on Computer*, Vol. 47, No. 12, pp. 1351-1362, Dec. 1998.

[ReZ02] P. Resnick, and R. Zeckhauser, "Trust among strangers in Internet transactions: Empirical analysis of eBay's reputation system," in *Advances in Applied Microeconomics: The Economics of the Internet and E-Commerce,* Vol. 11, In M. Baye, Ed., Elsevier, pp. 127-157, Nov. 2002.

[RST] Robocop, Space4U and Trust4All website: https://nlsvr2.ehv.campus.philips.com/

[SGG] C. D. Stylios, V. C. Georgopoulos, and P. P. Groumpos, "The use of fuzzy cognitive maps in modeling systems", http://med.ee.nd.edu/MED5/PAPERS/067/067.PDF

[SHZ05] S. Song, K. Hwang, R. Zhou, and Y.-K. Kwok, "Trusted P2P transactions with fuzzy reputation aggregation," *IEEE Internet Computing*, Vol. 9, Issue 6, pp. 24-34, Nov.-Dec. 2005.

[SiL03] A. Singh, and L. Liu, "TrustMe: anonymous management of trust relationships in decentralized P2P systems," in *IEEE International Conference on Peer-to-Peer Computing*, pp. 142-149, 2003.

[SL] http://sky.fit.qut.edu.au/~josang/sl/demo/Op.html.

[SLW03] R. Shan, S. Li, M. Wang, and J. Li, "Network security policy for large-scale VPN," *International Conference on Communication Technology Proceedings*, *ICCT* 2003, Vol. 1, pp. 217-220, April 2003.

[SYH06] Y. Sun, W. Yu, Z. Han, and K.J.R. Liu, "Information theoretic framework of trust modeling and evaluation for ad hoc networks," *IEEE Journal on Selected Area in Communications*, Vol. 24, Issue 2, pp. 305-317, Feb. 2006.

[TaT98] Y. Tan, and W. Thoen, "Toward a generic model of trust for electronic commerce," *International Journal of Electronic Commerce*, Vol.5, No.2, pp. 61-74, 1998.

[Tcg03] TCG TPM Specification v1.2, 2003. https://www.trustedcomputinggroup.org/specs/TPM/

[ThB06] G. Theodorakopoulos, and J.S. Baras, "On trust models and trust evaluation metrics for ad hoc networks," *IEEE Journal on Selected Areas in Communications*, Vol. 24, Issue 2, pp. 318-328, Feb. 2006.

[TLU06] S. Toivonen, G. Lenzini, and I. Uusitalo, "Context-aware trustworthiness evaluation with indirect knowledge," *Models of Trust for the Web* (MTW'06), Edinburgh, Scotland, May 2006.

[Twi03] A. Twigg, "A subjective approach to routing in P2P and ad hoc networks," in *Proceedings of the First International Conference of Trust Management* (iTrust 2003), LNCS 2692, pp. 225-238, Crete, Greece, May 2003.

[Vau03] S.J. Vaughan-Nichols, "How trustworthy is trusted computing?" *Computer*, Vol. 36, Issue 3, Mar. 2003.

[WaS05] K. Walsh, and E. G. Sirer, "Fighting peer-to-peer SPAM and decoys with object reputation," in *Proceedings of the Third Workshop on the Economics of Peer-to-Peer Systems* (P2PECON), pp. 138-143, Philadelphia, PA, Aug. 2005.

[WaV05] Y. Wang, and V. Varadharajan, "Trust^2: developing trust in peer-to-peer environments**,"** *IEEE International Conference on Services Computing*, Vol. 1, pp. 24-31, July 2005.

[WSC88] D. Wood, V. Stoss, L. Chan-Lizardo, G. S. Papacostas, and M. E. Stinson, "Virtual private networks," *International Conference on Private Switching Systems and Networks*, pp. 132-136, June 1988.

[XiL03] L. Xiong, and L. Liu, "A reputation-based trust model for peer-to-peer e-commerce communities," *IEEE International Conference on E-Commerce*, CEC 2003, pp. 275-284, 2003.

[XiL04] L. Xiong, and L. Liu, "PeerTrust: supporting reputation-based trust for peer-to-peer electronic communities**,"** *IEEE Transactions on Knowledge and Data Engineering*, Vol. 16, Issue 7, pp.843-857, July 2004.

[YaC03] Z. Yan, and P. Cofta**, "**A method or system to establish and maintain conditional trust by stating signal of distrust," *Patent Application*, 10/637,813 (US), 2003.

[Yan06] Z. Yan, "Predicting trustworthiness for component software," *Patent Application*, PCT/IB2007/053466, 2006.

[Yan07] Z. Yan, "Predicting trustworthiness for component software", In *Proceedings of IEEE 3rd International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing (SecPerU07)*, held in conjunction with IEEE International Conference on Pervasive Services 2007, pp. 1-6, Turkey, July 2007.

[YaP07] Z. Yan, and C. Prehofer, "An adaptive trust control model for a trustworthy software component platform", In *Proceedings of the 4th International Conference on Autonomic and Trusted Computing (ATC2007)*, LNCS 4610, pp. 226-238, China, July 2007.

[YaP07a] Z. Yan, and C. Prehofer, "Autonomic trust management for a trustworthy system," *Patent Application*, 059864.01347 (US), 2007.

[YaZ07] Z. Yan, and P. Zhang, "Bridging disjoint trusted domains into a trustworthy system", In *Proceedings of the 7th WSEAS International Conference on Applied Informatics and Communications (AIC'07)*, Greece, Aug. 2007.

[YPN07] Z. Yan, C. Prehofer, and V. Niemi, "Trust4All: a trustworthy middleware platform for component software", In *Proceedings of the 7th WSEAS International Conference on Applied Informatics and Communications (AIC'07)*, Greece, Aug. 2007.

[YZV03] Z. Yan, P. Zhang, and T. Virtanen, "Trust evaluation based security solution in ad hoc networks," In *Proceedings of the 7th Nordic Workshop on Secure IT Systems (NordSec03)*, Norway, Oct. 2003.

[ZhH99] L. Zhou, and Z.J. Haas, "Securing ad hoc networks", *IEEE Network*, 13(6), pp. 24-30, Nov./Dec. 1999.

[ZhY03] P. Zhang, and Z. Yan, "Virtual private network based on root-trust module computing platforms, *Patent Application*, 60/519,343 (US), 2003.

[ZJM05] M. Zhou, W. Jiao, and H. Mei, "Customizable framework for managing trusted components deployed on middleware," in *Proceedings of 10th IEEE International Conference Engineering of Complex Computer Systems ICECCS 2005*, pp. 283-291, June 2005.

[ZMZ05] M. Zhou, H. Mei, and L. Zhang, **"**A multi-property trust model for reconfiguring component software," *Fifth International Conference on Quality Software QAIC2005*, pp. 142-149, Sept. 2005.

[ZWW05] Z. Zhang, X. Wang, and Y. Wang, "A P2P global trust model based on recommendation," in *Proceedings of 2005 International Conference on Machine Learning and Cybernetics*, Vol. 7, pp. 3975-3980, Aug. 2005.