Chapter 8
Autonomic Trust Management in Mobile Environments

ABSTRACT

Autonomic trust management is the technology to automatically evaluate, establish, maintain, reevaluate, reestablish, and sustain dynamically changed trust relationships to adapt various contexts or situations. This chapter introduces an autonomic trust management solution in mobile environments by applying both trusted computing and trust evaluation technologies. The authors apply this solution to a number of mobile application scenarios in order to illustrate its applicability.

1. INTRODUCTION

1.1. Autonomic Trust Management

Autonomic Trust management is the technology to automatically evaluate, establish, maintain, and reevaluate, reestablish and sustain dynamically changed trust relationships to adapt various contexts or situations in order to continuously provide system trustworthiness (Yan & Prehofer, 2011). Yan and MacLaverty (2006) proposed that autonomic trust management includes the following four aspects and these four aspects are processed in an automatic way:

- **Trust establishment**: The process for establishing a trust relationship between a trustor and a trustee;
- **Trust monitoring**: The trustor or its delegate monitors the performance or behaviour of the trustee. The monitoring process aims to collect useful evidence for trust assessment of the trustee;
- **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 is changed;

DOI: 10.4018/978-1-4666-4765-7.ch008
Trust control and re-establishment:
If the trust relationship will be broken/changed or is broken/changed, the trustor will take corresponding measures to control or re-establish the trust relationship.

As we can see from the above, autonomic trust management can be achieved through trust modeling and evaluation.

Various trust modeling and management mechanisms are described in the literature for different systems, such as P2P systems (Kamvar, Scholsser & Garcia-Molina, 2003; Lee, Sherwood & Bhattacharjee, 2003; Liang & Shi, 2005; Singh & Liu, 2003; Song, Hwang, Zhou & Kwok, 2005; Walsh & Sirer, 2005; Xiong & Liu, 2004), e-commerce (Guha, Kumar, Raghavan & Tomkins, 2004), and web services (Resnick & Zeckhauser, 2002). It is widely accepted that trust is influenced by reputation (i.e., public evidence on the trustee), recommendations (i.e., a group of entities’ evidence on the trustee), the trustor’s past experience, and context. Most work focuses on trust evaluation (i.e., trust assessment), but does not consider how to ensure or sustain trust for the fulfillment of an intended purpose. It still lacks comprehensive discussions with regard to how to automatically take an essential action according to a trust value. Although a number of trust models consider the dynamic nature of trust and context’s influence, current literature does not adequately address context-aware adaptation of trust in some domains, such as services, component software, mobile applications, cloud computing and so on (Hall, Heimbigner, Van Der Hoek & Wolf, 1997; Herrmann, 2001; Herrmann, 2003; Malek, Esfahani, Menasce, Sousa & Gomaa, 2009; Mikic-Rakic, Malek & Medvidovic, 2008; Suryanarayana, Diallo, Erenkrantz & Taylor, 2006; Theodorakopoulos & Baras, 2006; Xiong & Liu, 2004; Zhou, Jiao & Mei, 2005; Zhou, Mei & Zhang, 2005). Most of existing solutions focus on a specific system that could be very different with each other. Particularly, due to the complexity and difference of context in different systems, a trust management solution, especially an autonomic trust management solution for one system could become inappropriate for another system. Recently, many solutions were developed for supporting trusted communications and collaborations among computing nodes in a distributed system, e.g., a P2P system (Zhang, Wang & Wang, 2005), an ad hoc network (Sun, Yu, Han & Liu, 2006; Theodorakopoulos & Baras, 2006) and a GRID computing system (Lin, Varadharajan, Wang & Pruthi, 2004). We found, however, that these methods are not feasible for supporting autonomic trust management, since most of them only concern trust assessment and evaluation. Current solutions generally ignore trust control and re-establishment by adapting dynamically changed environments and trust relationships, thus they cannot support or provide autonomic trust management.

The dynamic characteristic of trust is pushing trust management to become autonomic. This requires that trust management should handle the context’s influence adaptively and intelligently. In addition, the trust model itself should be adaptively adjusted to reflect the real situation of a system. Context-aware trust management is a developing research topic and adaptive trust model optimization is an emerging research opportunity.

1.2. Research Issues

Nowadays, trust management for mobile computing platforms is becoming an important issue in mobile computing environments. 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. However, 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 trust, security and privacy. This introduces special requirements for the mobile computing platform to embed trust management mechanisms for supporting trustworthy mobile communications.

However, due to the subjective characteristic of trust, trust management needs to take the trustor’s criteria into consideration. For a mobile system, it is essential for a user’s device to understand the user’s trust criteria in order to behave as her/his agent for trust management. However, most of existing digital systems are not designed to be configured by the users with regard to their trust criteria. Generally, it is not good to require a user to make a lot of trust related decisions because that would destroy usability. Also, 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. Trust should rather be managed automatically following a high level policy established by the trustor or auto-sensed by the device. 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 an embedded device, such as a mobile phone.

Nowadays, cloud computing is seen as the future of mobile (Perez, 2009). Cloud computing virtualizes physical and software resources and provides generic services, e.g., Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), etc. It offers a number of advantages such as scalability, agility and economy efficiency, in comparison of traditional Information Technology (IT) infrastructure (Armbrust, 2009). Meanwhile, contributed by the rapid deployment of broadband wireless networks and fast growth of smart phones, more and more users are using mobile devices to access Internet services. However, cloud computing still faces a number of challenges, one of which is trust, i.e., how a cloud service provider can ensure trust for its services (Dillon, Wu & Chang, 2010). Herein, trust refers to a set of properties including objective ones (e.g., expected transmission rate, delay variance, packet loss, and cost) and subjective ones (user experience, privacy concern and satisfaction degree). There are some existing works on trust assurance for cloud computing, e.g., QoS framework and various QoS mechanisms (Dillon, Wu & Chang, 2010; Lodi, Panzieri, Rossi & Turri, 2007; Stantchev & Schrofer, 2009; Wang, Du, Liu, Xie & Jia, 2010; Ye, Jain, Xia, Joshi, Yen, Bastani, Cureton & Bowler, 2010; Li, Hao, Xiao & Li, 2009; Xiao, Lin, Jiang, Chu & Shen, 2010). However, there still lacks a comprehensive study on trust for mobile cloud services. Notably, mobile cloud services are often affected by many specific factors, e.g., hardware and software limitations of mobile devices, signal strength of mobile networks, mobility of mobile users, etc. Thus, providing trust assurance, which includes QoS assurance for mobile cloud services requires a more advanced infrastructure and more effective mechanisms than traditional cloud services, e.g., based on mobile personal computers and devices.

All of the above problems influence the further development of mobile applications and services targeting at different areas, such as mobile enterprise, mobile networking, mobile terminal software and mobile computing. The key reason is that we lack a trust management solution for mobile computing platforms. This chapter presents
an autonomic trust management solution for the mobile computing platforms, which is based on the trusted computing technology and an adaptive trust control model. This solution supports autonomic trust control on the basis of the trustee device’s specification, which is ensured by a Root Trust module at the trustee device’s computing platform. We also assume several trust control modes, each of which contains a number of control mechanisms or operations, e.g. encryption, authentication, hash code based integrity check, access control mechanisms, etc. A control mode can be treated as a specific configuration of trust management that can be provided by the trustee device. Based on a runtime trust assessment, the rest objective of autonomic trust management is to ensure that a suitable set of control modes are applied in the trustee device in order to provide a trustworthy service. As we have to balance several trust properties in this model, we make use of a Fuzzy Cognitive Map to model the factors related to trust for control mode prediction and selection. Particularly, we use the trust assessment result as a feedback to autonomously adapt weights in the adaptive trust control model in order to find a suitable set of control modes in a specific mobile computing context.

1.3. Chapter Organization

Providing a trustworthy mobile computing platform is crucial for mobile communications, services and applications. This chapter contributes a concrete autonomic trust management solution to the literature regarding the above research issues by making use of both trusted computing and trust evaluation technologies. It studies methodologies and mechanisms of providing a trustworthy computing platform for mobile devices. In addition, solutions to support trusted communications and collaboration among those platforms are proposed in a distributed and dynamic manner. Section 2 of this chapter specifies a mechanism for trust sustainability among the mobile computing platforms based on the trusted computing technology. It plays as the first level of autonomic trust management. Section 3 describes an adaptive trust control model. The trust management mechanism based on this model plays as the second level of the proposed autonomic trust management solution. We demonstrate how the above two mechanisms can cooperate together to provide a comprehensive autonomic trust management solution in Section 4. Section 5 further discusses other related issues, such as standardization and implementation strategies. Finally, conclusions and future work are presented in the last section.

2. AUTONOMIC TRUST MANAGEMENT BASED ON TRUSTED COMPUTING PLATFORM

We propose a mechanism based on a Trusted Computing platform for trust sustainability among computing platforms. This mechanism can be further applied into P2P systems and ad hoc networks to achieve trust collaboration among mobile computing platforms. We also show how to use this mechanism to realize trust management in mobile enterprise networking.

2.1. Trust Form

This mechanism uses the following trust form: “Trustor A trusts trustee B for purpose P under condition C based on root trust R” (Yan & Cofta, 2004). The element C is defined by A to identify the rules or policies for sustaining or autonomic managing trust for purpose P, the conditions and methods to get the signal of distrust behaviors, as well as the mechanism to restrict any changes at B that may influence the trust relationship. It can also contain trust policies used for trust assessment and autonomic trust management at service runtime. The root trust R is the foundation of A’s trust on B and its sustaining. 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 the trustor device. Thus, it can be used to ensure a long term trust relationship among the computing platforms. This form makes it possible to extend one-moment or short-term trust over a longer period of time.

2.2. Root Trust Module

The mechanism is based on a Root Trust (RT) module that is also the basis of the Trusted Computing (TC) platform (TCG, 2003). The RT module could be an independent module embedded in the computing platform. It could be also a build-in feature in the current Trusted Platform Module (TPM) of TC platform and its related software.

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 changes of computing platform 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 1 illustrates the basic structure of this module. It contains a number of secure registers that can register the record of platform trusted booting and the platform configurations. Meanwhile, it can be used to save the conditions for trust sustaining and self-regulating. The conditions are registered into the root trust module at the remote attestation among computing platforms. The RT contains a Monitor that monitors the changes of the platform hardware and software by calculating their hash codes and comparing them with the registered ones. It also receives change requests from platform components (hardware or software). By checking with the conditions for trust sustaining and self-regulating, a Controller in the RT controls the platform changes by allowing or rejecting change requests or reporting the trustor platform a signal of distrust through a Reporter.

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

*Figure 1. Root trust module (Yan & Cofta, 2004)*
Autonomic Trust Management in Mobile Environments

module monitors the changes at the hardware and the software. At the booting time and software installation, the RT module registers the hash codes of each part of platform hardware and software. That is the configurations of the platform are registered at the RT module. It also periodically calculates the run-time values of platform configurations and checks if they are the same as those registered and expected. If there is any change, the RT module will check with the registered trust conditions and decide which measure should be taken.

2.3. Protocol

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 advance of current trusted computing technology and the work done in industry and in academy (TCG, 2003; Vaughan-Nichols, 2003; England, Lampson, Manferdelli, Peinado & Willman, 2003).

As shown in Figure 2, the proposed mechanism comprises the following procedures:

- Root trust challenge and attestation to ensure the trustor’s basic trust dependence at the trustee device in steps 1-2; (Note that if...
the attestation in this step is not successful, the trust relationship between device A and B cannot be established); 

- Trust establishment by specifying the trust conditions and registering them at the trustee’s RT module for trust sustaining in steps 3-6; 
- Sustaining the trust relationship through the monitor and control by the RT module in steps 7-8; 
- Re-challenge the trust relationship if necessary when any changes against trust conditions are reported.

2.4. Example Applications

In this sub-section, we present three use cases to illustrate how this mechanism can be applied to solve trust issues in ad hoc networks, P2P systems and mobile enterprise networking.

2.4.1. Trustworthy Communications in Ad Hoc Networks

The mechanism based on the Trusted Computing platform for trust sustainability among computing platforms can be applied into ad hoc networks to ensure trustworthy communications among a number of nodes for an intended purpose. For example, routing from a source node to a destination can be ensured 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 cannot 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 behaviors or software at the trustee device, it will reject or block it. If the RT module finds that the node platform is attacked, it will notify the trustor node platform. 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 the trustworthiness of other nodes based on statistical experiences, 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 the literature, e.g., a mechanism proposed by Yan, Zhang and Virtanen (2003). 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 Section 3 of this chapter, we further propose a mechanism to automatically ensure the trustworthiness of the trustee device according to runtime trust assessment.

2.4.2. 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.

Based on the mechanism for trust sustainability, we further develop a Trusted Collaboration Infrastructure (TCI) for peer-to-peer computing devices (Yan & Zhang, 2006). In this infrastructure, each peer device is TC platform compatible and has an internal architecture as shown in Figure 3. Through applying the TCI, trust collaboration...
Autonomic Trust Management in Mobile Environments

Figure 3. Architecture of P2P peer device in TCI

![Architecture of P2P peer device in TCI](image)

...can be established among distributed peers through the control of the TC platform components.

There are three layers in the TCI. A platform layer contains TC platform components specified in TCG, 2003 (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.

A P2P system layer contains common components required for trusted P2P communications. Those components are installed over the platform layer and ensured to be run 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 process properly.

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, refer to Chapter 3. The encryption service allows data to be encrypted in such a secure way that it can be decrypted only by a certain machine, and only if that machine is in a certain configuration. 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.,
by a trusted OS) as trusted for execution. Any malicious change could be detected and rejected by the platform layer. For different purposes, different components can be downloaded and installed into the application/service layer. A preferred software middleware platform for the TCI could be a component-based software platform that interfaces with the TC functionalities and provides necessary mechanisms to support trustworthy execution of components.

2.4.2.1. Trust Collaboration

The proposed mechanism supports trust collaboration among P2P peers. 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 expectation of the content originators or right-holders 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 2 (step 1 and 2). This is done through digitally certifying the device configurations.

Two levels of certifying are provided: certifying OS configurations and certifying the applications running on the OS. For certifying the OS, 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 can then verify that the certificate is valid and up-to-date, so it can know the OS configuration of the device.

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 precise configuration of the applications:

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

As shown in Figure 2, 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 TC platform components of the trustee device, e.g., the RT module (step 3-4). Through applying the mechanism described above, 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 could specify that a) upgrading of P2P applications is only allowed for applications certified by a Trusted Third Party; b) the changes of 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 by the Root Trust module
and the secure software installation mechanism that can verify the certificate of a software application before the installation.

The proposed mechanism for autonomic trust management based on trusted computing platform provides a way 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 predefined conditions (refer to 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 (e.g., Yan, Zhang & Virtanen, 2003; Fenkam, et al., 2002; Kortuem, et al., 2001; Jøsang, Ismail & Boyd, 2005; Lin, Varadharajan, Wang & Pruthi, 2004) 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 helps 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 with protection. 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 feature includes two aspects. One is that the resources are provided based on copyright restrictions. The contents that cannot be shared should not be disclosed to other peers. The other is that the resources are provided with 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 (TCG, 2003; Yan & Zhang, 2006a):

- 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 an expected way.
Furthermore, with the support of trust evaluation and trust sustainability, the peers could collaborate in the most trustworthy way.

2.4.3. 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. First, current Virtual Private Networks (VPN) 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 security requirements of the enterprise VPN operator, during both 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 with confidential attachment 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 can be found in the literature. Related work did not consider the solutions of the problems described above (Herscovitz, 1999; Wood, Stoss, Chan-Lizardo, Papacostas & Stinson, 1988; Regan, 2003; Cheung & Misic, 2002). For example, a trust management solution based on KeyNote for IPSec proposed by Blaze, Ioannidis and Keromytis (2002) could ensure trust during VPN connection in the network-layer. A security policy transmission model was presented to solve security policy conflicts in large-scale VPN (Shan, Li, Wang & Li, 2003). But the proposal could not help in solving the trust sustainability after the VPN connection and disconnection. Prior work focused on securing network connection, not paying much attention to the necessity to control VPN terminal devices (Hamed, Al-Shaer & Marrero, 2005). In addition, security or trust policies of the VPN operator should be different regarding different VPN client devices. This fact raises additional challenges for trust management in enterprise networking.

A solution for enhancing trust in a mobile VPN system can be provided based on the mechanism for trust sustainability among computing platforms. We aim to support confidential content management and overcome the diversity support of security in different devices manufactured by different vendors. 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 connection with the VPN is terminated. Even though the VPN client device is not RT module based, the trust management server can attest it and apply corresponding trust policies in order to restrict its access to confidential information and operations (Yan & Zhang, 2006b).

A simple example of trust conditions for trust management in a mobile enterprise networking could 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 on 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.

3. AUTONOMIC TRUST MANAGEMENT BASED ON AN ADAPTIVE TRUST CONTROL MODEL

In this section, we further introduce an adaptive trust control model via applying the theory of Fuzzy Cognitive Map (FCM) in order to conduct autonomic trust management based on trust assessment. This model illustrates the relationships among trust, its influence factors, the control modes used for managing trust, and the trustor’s policies. It supports predicting trustworthiness to select proper trust control modes to ensure trust by adapting to current context. We illustrate how to manage trust adaptively in a middleware component software platform and for mobile cloud services by applying this method.

3.1. Adaptive Trust Control Model

The trustworthiness of a service or a combination of services provided by a device 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 $C_j (j = 1, ..., m)$. A control mode contains a number of control mechanisms or operations that can be provided by the device. We assume that the control modes are exclusive and that combinations of different modes are used.

The adaptive trust control model can be described as a graphical illustration using a FCM, as shown in Figure 4. It is a signed directed graph with feedback, consisting of nodes and weighted arcs. Nodes of the graph are connected by signed and weighted arcs representing the causal relationships that exist between the nodes. There are three layers of nodes in the graph. The node in the top layer is the trustworthiness of a service. The nodes located in the middle layer are its quality attributes, which have direct influence on the service’s trustworthiness. The nodes at the bottom layer are control modes that could be supported and applied by the device. These control modes can control and thus improve the quality attributes. Therefore, they have indirect influence on the trustworthiness of the service. The value of each node is influenced by the values of the connected nodes with the appropriate weights and by its previous value. Thus, we apply an addition operation to take both into account.

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

$$T = f \left( \sum_{i=1}^{n} w_i V_{QA} + T^{old} \right)$$  \hspace{1cm} (1)$$

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 at-
Autonomic Trust Management in Mobile Environments

Figure 4. Graphical modeling of trust control (Yan & Prehofer, 2011)

tribe is considered at the trust decision or assessment. \( w_i \) can be decided based on the trustor’s policies. We apply the Sigmoid function
\[
f(x) = \frac{1}{1 + e^{-\alpha x}} \quad \text{(e.g., } \alpha = 2)\]

to map node values \( V_{QA}, V_{QB}, T \) into [0, 1]. The value of the quality attribute is denoted by \( V_{QA} \). It can be calculated according to the following formula based on Figure 4:
\[
V_{QA} = \sum_{j=1}^{m} c_{w_j} V_{C_j} B_{C_j} + V_{old} \quad \text{(2)}
\]

where \( c_{w_j} \) is the influence factor of control mode \( C_j \) to \( QA \), \( c_{w_j} \) is set based on the impact of \( C_j \) on \( QA \). Positive \( c_{w_j} \) means a positive influence of \( C_j \) on \( QA \). Negative \( c_{w_j} \) implies a negative influence of \( C_j \) on \( QA \). Zero implies no influence of \( C_j \) on \( QA \). Null means no any relationship between \( C_j \) and \( QA \). \( B_{C_j} \) 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. Based on Figure 4, the value of the control mode can be calculated using:
\[
V_{C_j} = f \left( T \cdot B_{C_j} + V_{old} \right) \quad \text{(3)}
\]

### 3.2. Procedure

Based on the above understanding, we propose a procedure to conduct autonomic trust management in a computing platform targeting at a trustee entity specified by a trustor entity, as shown in Figure 5. The entity could be a service or a combination of services or an application. Herein, we apply several trust control modes, each of which contains a number of control mechanisms or operations. The trust control mode can be treated as a special configuration of trust management that can be provided by the platform or system. In this procedure, 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 them is made. Trust control mode selection is a mechanism to select the most suitable trust control modes based on the prediction results. Trust assessment is conducted based on the subjective criteria of the trustor through evaluating the quality attributes of the trustee entity. It is also influenced by the platform context. Particularly, the quality attributes of the entity can be controlled or improved via applying a number of trust control modes, especially at system runtime. In this autonomic trust management solution, we consider such factors that impact trust as the trustee’s objective factors (i.e., quality attributes), the trustor’s objective factors (i.e., the policies of trustor regarding a trust decision, e.g., the importance rates of QA, performance criteria and trust threshold).

For a trustor, the trustworthiness of its specified trustee can be predicted regarding various control modes supported 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 trust value is below an expected threshold. This threshold is generally set by the trustor to express its 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 trust control modes in order to conduct autonomic trust management.

3.3. Algorithms

Based on the adaptive trust control model, a number of algorithms are designed to implement each step of the procedure for autonomic trust management at the service runtime, as shown in Figure 5. These algorithms include trust assessment, trust control mode prediction and selection, and adaptive trust control model adjustment, which were evaluated in (Yan & Prehofer, 2007; Yan, 2008a; Yan & Prehofer, 2011).

3.3.1. Trust Assessment

We conduct trust assessment based on observation. At the trustee service runtime, the performance observer monitors its performance with respect to specified quality attributes. For each quality attribute, if the monitored performance is better than the criteria of the trustor, the positive point \( p \) of that attribute is increased by 1. If the monitored result is worse than the criteria, the negative point \( n \) of that attribute is increased by 1. For evaluating trust at system runtime, we suggest not considering recommendations in the algorithm because the evidence achieved through runtime monitoring is determinate. The trust opinion of each quality attribute can be generated based on an opinion generator, e.g.:

\[
\theta = \frac{p}{p + n + r}, r \geq 1
\]  

In addition, based on the importance rates \( ir \) of different quality attributes, a combined opinion \( \theta_T \) on the trustee can be calculated by applying weighted summation:

\[
\theta_T = \sum ir_i \theta_i
\]

By comparing to a trust threshold opinion \( to \), the system can decide if the trustee is still trusted or not. The runtime trust assessment results play as a feedback to trigger trust control and re-establishment.

3.3.2. Control Mode Prediction and Selection

The control modes are predicted by evaluating all possible modes and their compositions using a prediction algorithm based on Formula 1, 2 and 3. We then select the most suitable control modes based on the above prediction results using a selection algorithm:

**Algorithm 1:** Trustworthiness prediction is used to anticipate the performance or feasibility of all possibly applied trust control modes. Note that a constant \( \delta \) is the accepted \( \Delta T \) that controls the iteration of the prediction;

**Algorithm 1. Trustworthiness prediction**

- For every composition of control modes, i.e., \( \forall S_k (k = 1, ..., K) \), while
  \[
  \Delta T_k = T_k - T_k^{old} \geq \delta, \quad \text{do}
  \]
  \[
  V_{C_{j,k}} = f \left( T_k \cdot B_{C_{j,k}} + V_{C_{j,k}}^{old} \right)
  \]
  \[
  V_{QA_{i,k}} = f \left( \sum_{j=1}^{m} c_{w_i} V_{C_{j,k}} B_{C_{j,k}} + V_{QA_{i,k}}^{old} \right)
  \]
  \[
  T_k = f \left( \sum_{i=1}^{n} w_i V_{QA_{i,k}} + T_k^{old} \right)
  \]
Algorithm 2: Control mode selection is applied to select a set of suitable trust control modes based on the control mode prediction results.

3.3.3. Adaptive Trust Control Model Adjustment

It is important for the trust control model to be dynamically maintained and optimized in order to precisely adapt to the real system situation and context. The influence factors of each control mode should sensitively indicate the influence of each control mode on different quality attributes in a dynamically changed environment. 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 influence factors of the applied control modes should be adjusted in order to reflect this situation. Then, the device can automatically re-predict and re-select a set of new control modes in order to ensure the trustworthiness. In this way, the device can avoid using the attacked or useless trust control modes in the underlying context. Therefore, it is crucial to make the trust control model context-adaptive in order to support autonomic trust management.

We apply two schemes to adjust the influence factors $cw_j$ of the trust control model in order to make it reflect the real system situation. We use $V_{QA} - m$ and $V_{QA} - p$ to stand for $V_{QA}$ generated based on real system observation (i.e., the trust assessment result) and by prediction, respectively. In the schemes, $\omega$ is a unit deduction factor and $\sigma$ is the accepted deviation between $V_{QA} - m$ and $V_{QA} - p$. We suppose $C_j$ with $cw_j$ is currently applied. The first scheme (Algorithm 3) is an equal adjustment scheme, which holds a strategy that each control mode has the same impact on the deviation between $V_{QA} - m$ and $V_{QA} - p$. The second one (Algorithm 4) is an unequal adjustment scheme. It holds a strategy that the control mode with the biggest absolute influence factor always impacts more on the deviation between $V_{QA} - m$ and $V_{QA} - p$.

3.4. Autonomic Trust Management for Component Software Platform

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

Algorithm 2. Control mode selection

- Calculate selection threshold $thr = \frac{\sum_{k=1}^{K} T_k}{K}$;
- Compare $V_{QA,k}$ and $T_k$ of $S_k$ to $thr$, set selection factor $SF_{S_k} = 1$ if $\forall V_{QA,k} \geq thr \land T_k \geq thr$; set $SF_{S_k} = -1$ if $\exists V_{QA,k} < thr \lor \exists T_k < thr$;
- For $\forall SF_{S_k} = 1$, calculate the distance of $V_{QA,k}$ and $T_k$ to $thr$ as $d_k = \min\{|V_{QA,k} - thr|, |T_k - thr|\}$; For $\forall SF_{S_k} = -1$, calculate the distance of $V_{QA,k}$ and $T_k$ to $thr$ as $d_k = \max\{|V_{QA,k} - thr|, |T_k - thr|\}$ only when $V_{QA,k} < thr$ and $T_k < thr$;
- 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 $d_k$.
Autonomic Trust Management in Mobile Environments

Algorithm 3. An equal adjustment scheme

- While \( |V_{QA} - m - V_{QA} - p| > \sigma \), do
  a) If \( V_{QA} - m < V_{QA} - p \), for \( \forall c_{w_{ji}} \),
     \( c_{w_{ji}} = c_{w_{ji}} - \omega \), if \( c_{w_{ji}} < -1, c_{w_{ji}} = -1 \)
     Else, for \( \forall c_{w_{ji}} \),
     \( c_{w_{ji}} = c_{w_{ji}} + \omega \), if \( c_{w_{ji}} > 1, c_{w_{ji}} = 1 \)
  b) Run Algorithms 1 - Trustworthiness Prediction.

Algorithm 4. An unequal adjustment scheme

- While \( |V_{QA} - m - V_{QA} - p| > \sigma \), do
  a) If \( V_{QA} - m < V_{QA} - p \), for
     \( \max(c_{w_{ji}}) \),
     \( c_{w_{ji}} = c_{w_{ji}} - \omega \), if \( c_{w_{ji}} < -1, c_{w_{ji}} = -1 \)
     (warning);
     Else, \( c_{w_{ji}} = c_{w_{ji}} + \omega \), if \( c_{w_{ji}} > 1, c_{w_{ji}} = 1 \) (warning);
  b) Run Algorithms 1 - Trustworthiness Prediction.

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.

We introduce a trust management framework that implements the above described mechanism into the RE sub-layer of the platform middleware. 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 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 management. 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. The trust management framework can shutdown any misbehaving component and gather evidence on the trustworthiness of a system entity. Meanwhile, it also controls the security framework, to ensure that it applies essential security mechanisms to maintain a trusted system. In short, the trust management framework acts like a critical system manager, ensuring that the system conforms to its trust policies. This architecture supports the implementation of both the ‘hard trust’ solution and the ‘soft trust’ solution.

3.4.1. Trust Management Framework

Figure 7 illustrates the structure of the trust management framework. In Figure 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 regarding making trust assessments and decisions. The recommendation
Autonomic Trust Management in Mobile Environments

Base saves various recommendations. The experience base saves the evidence collected from the component software platform itself in various contexts. The decision/reason engine is used to make trust decision when receiving 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 for trust control and establishment that are supported by the platform. It is also used to store the trust control models as described above. The selection engine is used to predict and select suitable mechanisms to ensure the platform trustworthiness in a specific context. It also conducts adaptive adjustments on the trust control model. The selected mechanisms are executed by the RE.

Figure 6. Relationships among trust management framework and other frameworks (Yan & Prehofer, 2011)
In addition, the recommendation input is the interface for collecting recommendations, which are useful to make component installation decision (Yan, 2008a). 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 and supported in the system. The quality attributes monitor is the functional block used to monitor the performance of specified system entity with regard to those attributes that may influence trust. The trust manager cooperates with other frameworks to manage the trustworthiness of the middleware component software platform.

3.5. Autonomic Trust Management for Mobile Cloud Services

Trust management is a crucial issue for mobile cloud services. This section proposes an autonomic trust management system for mobile cloud computing based on the adaptive trust control model. This solution facilitates trust prediction, establishment, assessment and assurance. We introduce the influence of QoS (Quality of Service) properties (i.e., quality attributes) and service modes into the adaptive trust control model, which supports autonomic trust management according to trust assessment based on runtime QoS observation. The effectiveness and benefits of this solution is reported by Zhang and Yan (2011). This method contributes to a practical solution that can react against trust unsatisfactory adaptively at service runtime and handle trust requests with different QoS criteria.

3.5.1. System Architecture

Figure 8 shows trust management architecture based on QoS observation for mobile cloud services. In a mobile device, a QoS agent monitors QoS status at runtime, e.g., percentage of memory and CPU consumption, connection speed, remaining battery percentage and packet loss rate, etc. The QoS status will be reported to a trust management center in cloud side. The trust management center aggregates and analyzes the huge set of QoS data at a data analyzer, and dynamically adjusts resources at a resource controller to meet trust requirements of each mobile cloud service.
Based on the above trust management architecture and the adaptive trust control model, we apply several modes of mobile cloud services. Each mode contains multiple services, mechanisms and resource configuration schemes. A cloud service mode is a specific configuration to guarantee the trust requirements for a cloud service. Notably, the mobile cloud computing platform (offered by a mobile cloud service provider) can provide multiple similarly functioned services that can satisfy the demand of an integrated service. Especially, the trust requirements of a service can be assured by selecting suitable service modes. Herein, the data analyzer is applied to implement trust assessment based on the collected QoS data, while the resource controller is designed to conduct trustworthiness prediction, cloud service selection, control model adjustment and service/resource arrangement. The trust management center is responsible for autonomic trust management of cloud services for mobile devices based on the QoS observation reports from the devices.

In order to ensure the reliability of QoS observation reports from the mobile devices, we can apply the mechanisms introduced in Section 2 of this chapter to ensure the trustworthiness of the QoS agent running in the mobile device. Concretely, the QoS data collector at the cloud service side can remotely attest and ensure the trust of mobile device before and during the execution of cloud services.

3.5.2. Autonomic Trust Management Process

The process of autonomic trust management for mobile cloud services follows the procedure of Figure 5. In this case, trust predication is a mechanism to predict the performance of a set of cloud service modes before selecting a suitable one. Mode selection is a mechanism to select the best service mode based on previous prediction results. Trust assessment is a mechanism to monitor and assess the QoS status according to the trust requirements of a user. For the trust requirements on a service, the trust values can be predicted by assuming a service mode is selected. Based on prediction results, a service mode can be selected and set as the configuration of cloud services. The trust assessment mechanism evaluates the QoS status by collecting the QoS performance data of the cloud service from the mobile device. According to the assessment results, the system...
adjusts the parameters of the adaptive trust control model to reflect real status. The adjustment happens when the evaluation result is below a threshold defined by the user. The process runs over to achieve the autonomic trust management in a dynamic mobile cloud environment. In particular, the trust management supports context awareness by adaptively selecting a proper set of service modes that can always ensure the trust and quality of cloud services.

4. A COMPREHENSIVE SOLUTION FOR AUTONOMIC TRUST MANAGEMENT

An integrated solution can be further proposed by integrating the above two mechanisms together (Yan, 2008b). The trustworthiness of all kinds of mobile systems can be ensured by applying this solution. Taking a mobile pervasive system as an example, we demonstrate how trust can be automatically managed and the effectiveness of our solution.

4.1. A System Model

A mobile system is described in Figure 9. It is composed of a number of mobile computing devices. The devices offer various services. They could collaborate together in order to fulfill an intended purpose requested by a mobile system user. We assumed that the mobile computing device has a Root Trust module as described in Section 2, which supports the mechanism to sustain trust. This module locates at a trusted computing platform with necessary hardware and software support (TCG, 2003). The trusted computing platform protects the Operating System (OS) that runs a number of services (offered by various software components or applications) and a performance observer that monitors the performance of the running services. The service or device could behave as either a trustor or a trustee in the system. Particularly, an autonomic trust management framework (ATMF) is also contained in the trusted computing platform with the support of the RT module. The ATMF is responsible for managing the trustworthiness of the services.

Figure 9. Model of a pervasive system (Yan, 2008b)
4.2. Autonomic Trust Management Framework (ATMF)

The ATMF is applied to manage the trustworthiness of a trustee service by configuring its trust properties or switching on/off the trust control mechanisms, i.e., selecting a suitable set of control modes. Its structure is shown in Figure 10.

The framework contains a number of secure storages, such as an experience base, a policy base and a mechanism base. The experience base is used to store the monitoring results of service performance regarding a number of quality attributes. The experience data could be accumulated locally or contain recommendations of other devices. The policy base registers the trustor’s policies for trust assessment. The mechanism base registers the trust control modes that can be supported by the device in order to ensure the trustworthiness of the services. The ATMF located at the platform layer has secure access to the RT module in order to extract the policies into the policy base for trust assessment if necessary (e.g., if a remote service is the trustor). In addition, an evaluation, decision and selection engine (EDS engine) is applied to conduct trust assessment, make trust decision and select suitable trust control modes.

4.3. A Procedure of Comprehensive Autonomic Trust Management

Based on the above design, we propose a procedure to conduct autonomic trust management targeting at a trustee service specified by a trustor service in the mobile system, as shown in Figure 11.

The device that locates the trustor service firstly checks whether remote service collaboration is required. If so, it applies the mechanism for trust sustaining to ensure that the remote service device will work as its expectation during the service collaboration. The trust conditions about the trustee device can be protected and realized with its RT module. Meanwhile, the trust

Figure 10. Autonomic trust management framework (Yan, 2008b)
policies of the trustor on services will also be embedded into the RT module of the trustee device when the device trust relationship is established. The rest procedure is the same for both remote service collaboration and local service collaboration. After inputting the trust policies into the policy base of the trustee device’s ATMF, autonomic trust management is triggered to ensure trustworthy service collaboration.

The rest procedure is the same as the procedure illustrated in Figure 5. Concretely, we apply several trust control modes, each of which contains a number of control mechanisms or operations. The trust value is predicted supposed that some...
control modes are applied before the decision to initiate those modes is made. The most suitable trust control modes can be selected based on the prediction results. Trust assessment is then conducted based on the trustor’s subjective policies by evaluating the trustee entity’s quality attributes which are influenced by the system context. According to the runtime trust assessment results in the underlying context, the trustee device conducts trust control model adjustment in order to reflect the real system situation if the assessed trustworthiness value is below an expected threshold. The quality attributes of the entity can be controlled or improved via applying a number of trust control modes, especially at the service runtime. The context-aware or situation-aware adaptability of the trust control model is crucial to re-select a suitable set of trust control modes in order to conduct autonomic trust management.

4.4. An Application Example

This section takes a simple example to show how autonomic trust management is realized based on the cooperation of both the trust sustaining mechanism and the adaptive trust control model. The proof of applied algorithms has been reported in (Yan & MacLaverty, 2006; Yan, 2008a; Yan & Prehofer, 2007; Yan & Prehofer, 2011).

We take a mobile pervasive healthcare system as a concrete example. It is composed of a number of services located at different devices. For example, a health sensor locates at a portable mobile device, which can monitor a user’s health status; a healthcare client service in the same device provides multiple ways to transfer health data to other devices and receive health guidelines. A healthcare consultant service locates at a healthcare centre, which provides health guidelines to the user according to the health data reported. It can also inform a hospital service at a hospital server if necessary. The trustworthiness of the healthcare application depends on not only the trustworthiness of each device and service, but also the cooperation of all related devices and services. It is important to ensure that they can cooperate well in order to satisfy trust requirements with each other and their user. For concrete examples, the healthcare client service needs to provide a secure network connection and communication as required by the user. It also needs to respond to the request from the health sensor within expected time and performs reliably without any break in case of an urgent health information transmission. Particularly, if the system deploys additional services that could share resources with the healthcare client service, the mobile healthcare application should be still capable of providing qualified services to its users.

In order to provide a trustworthy healthcare application, the trustworthy collaboration among the mobile device, the healthcare centre and the hospital server is required. In addition, all related services should cooperate together in a trustworthy way. Our example application scenario is the user’s health is monitored by a mobile device that reports his/her health data to the healthcare centre in a secure and efficient way. In this case, the hospital service should be efficiently informed since the user’s health needs to be treated by the hospital immediately. Meanwhile, the consultant service also provides essential health guidelines to the user. Deploying our solution, the autonomic trust management mechanisms used to ensure the trustworthiness of the above scenario are summarized in Table 1 based on a number of example trust conditions and policies. Taking the first example in the Table 1, the trust policies include the requirements on different quality attributes: confidentiality, integrity, availability and reliability in order to ensure the trustworthiness of health data collection in the mobile device.

5. FURTHER DISCUSSIONS

The proposed solution supports autonomic trust management with two levels. The first level imple-
Autonomic Trust Management in Mobile Environments

Table 1. Autonomic trust management for a healthcare application

<table>
<thead>
<tr>
<th>Trustor</th>
<th>Trustee</th>
<th>Example Trust Requirements</th>
<th>Autonomic Trust Management Mechanisms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Health sensor</td>
<td>Healthcare client</td>
<td>Trust policies (data confidentiality: yes; data integrity: yes; service availability – response time: &lt;3sec; service reliability – uptime: &gt;10min)</td>
<td>Control mode prediction and selection, runtime trust assessment, trust control model adjustment and control mode re-selection to ensure the trustworthiness of health data collection;</td>
</tr>
<tr>
<td>Mobile device</td>
<td>Healthcare centre</td>
<td>Trust conditions (device and trust policies integrity: yes)</td>
<td>Trust sustaining mechanism to ensure the integrity of healthcare centre and trust policies for consultant service;</td>
</tr>
<tr>
<td>Healthcare client</td>
<td>Consultant service</td>
<td>Trust policies (authentication: yes; data confidentiality: yes; data integrity: yes; service availability – response time: &lt;30sec; service reliability – uptime: &gt;10hour)</td>
<td>Control mode prediction and selection, runtime trust assessment, trust control model adjustment and control mode re-selection to ensure the trustworthiness of health data reception;</td>
</tr>
<tr>
<td>Healthcare centre</td>
<td>Mobile device</td>
<td>Trust conditions (device and trust policies integrity: yes)</td>
<td>Trust sustaining mechanism to ensure the integrity of mobile device and trust policies for healthcare client service;</td>
</tr>
<tr>
<td>Consultant service</td>
<td>Healthcare client</td>
<td>Trust policies (authentication: yes; data confidentiality: yes; data integrity: yes; service availability – response time: &lt;10sec; service reliability – uptime: &gt;1hour)</td>
<td>Control mode prediction and selection, runtime trust assessment, trust control model adjustment and control mode re-selection to ensure the trustworthiness of guidelines reception;</td>
</tr>
<tr>
<td>Healthcare centre</td>
<td>Hospital server</td>
<td>Trust conditions (device and trust policies integrity: yes)</td>
<td>Trust sustaining mechanism to ensure the integrity of hospital server and trust policies for hospital service;</td>
</tr>
<tr>
<td>Consultant service</td>
<td>Hospital service</td>
<td>Trust policies (authentication: yes; data confidentiality: yes; data integrity: yes; service availability – response time: &lt;10min; service reliability – uptime: &gt;10hour)</td>
<td>Control mode prediction and selection, trust assessment, trust control model adjustment and control mode re-selection to ensure the hospital service’s trustworthiness.</td>
</tr>
</tbody>
</table>

The proposed solution applied the trust sustaining mechanism to stop or restrict any potential risky activities. Thus, it is a more active approach than the existing solutions.

Trusted computing platform technology is developing in both industry and academia in order to provide more secure and better trust support for future digital devices. The technology aims to solve existing security problems by hardware trust. Although it may be vulnerable to some hardware attacks (Huang, 2002), it has advantages over many software-based solutions. It has potential advantages over other solutions as well; especially when the Trusted Computing Group standard (TCG, 2003) is deployed and more and more industry digital device vendors offer TCG-compatible hardware and software in the

ments autonomic trust management among different system devices by applying the mechanism to sustain trust. On the basis of a trusted computing platform, this mechanism can also securely embed the trust policies into a remote trustee device for the purpose of trustworthy service collaboration. This mechanism is mainly implemented at the device platform layer. Regarding the second level, the trustworthiness of the service is automatically managed based on the adaptive trust control model at its runtime. This mechanism can be implemented in either the platform layer or the middleware layer (e.g., a component software middleware layer), depending on the concrete system requirements. Both levels of autonomic trust management can be conducted independently or cooperate together in order to ensure the trustworthiness of the entire mobile system. From this point of view, none of the existing work reviewed in this book provides a similar solution.
future. The proposed solution will have potential advantages when various digital device vendors produce TCG compatible products in the future.

The RT module can be designed and implemented inside a secure main chip in the mobile computing platform. The secure main chip provides a secure environment to offer security functions or services for the operating system (OS) and application software. It also has a number of security enforcement mechanisms (e.g., secure booting, integrity checking, privacy enhancement, and device authentication). Particularly, it provides cryptographic functions and secure storages. The RT module functionalities and the ATMF functionalities can be implemented by a number of protected applications. The protected applications are small applications dedicated to performing security critical operations inside a secure environment. They have strict size limitations and resemble function libraries. The protected applications can access any resource 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 signed protected applications are allowed to run. Notably, the secure registers in the RT module, the policy base, the execution base and the mechanism base could be implemented by a flexible and light secure storage mechanism supported by the trusted computing platform (Asokan & Ekberg, 2008).

6. CONCLUSION

This chapter elaborates the demand of autonomic trust management in mobile environments. We specify a number of research issues regarding mobile computing platforms in mobile environments. In order to solve these issues, we presented a comprehensive and generic autonomic trust management solution based on the trust sustaining mechanism and the adaptive trust control model. The main significance of this solution lies in the fact that it supports two levels of autonomic trust management: between devices as well as between services offered by the devices. This solution can also effectively avoid or reduce risks by stopping or restricting any potential risky activities based on the specification of the trustor. We demonstrated the effectiveness of this solution by applying it into a number of mobile systems, e.g., ad hoc networks, P2P systems, mobile enterprise networking, component software platform, mobile cloud services and mobile pervasive systems. We also discussed the advantages and implementation strategies of this solution.

Regarding future work direction, it is essential to analyze the performance of the proposed solution on the basis of a mobile trusted computing platform. Furthermore, how to automatically extract mobile the trust policies of mobile users based on machine learning through user-device interaction is also an interesting research topic. The usability of autonomic trust management is a crucial issue for the success of practical deployment.

REFERENCES


Autonomic Trust Management in Mobile Environments


APPENDIX

Questions and Exercises

1. What are the advantages and disadvantages of the comprehensive autonomic trust management solution presented in this chapter?
2. What are the characteristics of autonomic trust management?
3. What is the difference of autonomic trust management from normal trust management?