AUTONOMIC TRUST MANAGEMENT FOR A PERVASIVE SYSTEM

Zheng Yan
Nokia Research Center, Itämerenkatu 11-13, Helsinki, Finland

Keywords: Trust management, pervasive computing, autonomic control, trusted computing, and security.

Abstract: A pervasive system allows seamless interactions among various portable and networked processing devices, distributed at all scales throughout everyday routine life. In such an open and dynamic environment, trust becomes a crucial issue to ensure effective collaborations among various devices in order to provide expected services. Many existing trust management solutions for the pervasive systems did not support autonomic control that automatically manages trust requested by a trustor device on a trustee device for the fulfillment of an intended service. This greatly influences the effectiveness of trust management. In this paper, we propose an autonomic trust management solution for the pervasive system on the basis of a trusted computing platform and an adaptive trust control model. We demonstrate how trust can be automatically managed and the effectiveness of our solution by applying it into an example pervasive system. Additional issues such as standardizing pervasive computing devices and implementation strategies are also discussed.

1 INTRODUCTION

A pervasive system allows seamless interactions among various portable and networked processing devices, distributed at all scales throughout everyday routine life. Despite its promising opportunities, the pervasive system is experiencing new technical challenges, as the pervasive computing environment has become vulnerable to new security and privacy threats (Campbell, Al-Muhtadi, Naldurg, Sampemane and Mickunas, 2002). The highly decentralized and distributed nature of pervasive computing environments makes classical, centralized security-managing mechanisms unusable. In such an open and dynamic environment, the communications depend highly on trust among devices (Sun and Denko, 2007). Therefore, trust becomes a crucial issue to ensure effective collaborations among various devices in order to provide expected services.

Quite a number of researches have been conducted in order to manage trust in the pervasive system. Most existing researches are mainly on establishing distinct trust models based on different theories or methods in terms of various scenes and motivations. Generally, these researches apply trust, reputation and/or risk analysis mechanism based on fuzzy logic, probabilistic theory, cloud theory, traditional authentication and cryptography methods and so on to manage trust in such an uncertain environment (Xu, Xin, and Lu, 2007). However, many existing trust management solutions for the pervasive systems did not support autonomic control that automatically manages trust requested by a trustor device on a trustee device for the fulfillment of an intended service. This greatly influences the effectiveness of trust management since trust is both subjective and dynamic.

In this paper, we adopt a holistic notion of trust which includes several properties, such as security, availability and reliability, depending on the requirements of a trustor. Hence trust is defined as the assessment of a trustor on how well the observed behavior that can be measured through a number of quality attributes of a trustee meets the trustor's own standards for an intended purpose (Denning, 1993).

We present an autonomic trust management solution for the pervasive system, which is based on a trusted computing platform and an adaptive trust control model. This solution supports autonomic trust control on the basis of the trustor 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

Yan Z. (2008).
AUTONOMIC TRUST MANAGEMENT FOR A PERVASIVE SYSTEM.
In *Proceedings of the International Conference on Security and Cryptography*, pages 491-500
DOI: 10.5220/0001921104910500

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 special 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 pervasive computing context.

The rest of the paper is organized as follows. Section 2 introduces related work. Section 3 specifies the fundamental technologies which play as the basis of our solution. In Section 4 the autonomic trust management solution for the pervasive system is described. We demonstrate how trust can be automatically managed and the effectiveness of our solution by applying it into an example pervasive system in Section 5. Section 6 further discusses other related issues, such as implementation strategies. Finally, conclusions and future work are presented in Section 7.

2 RELATED WORK

Xu, Xin, and Lu (2007) discussed the essentiality of trust model and management in pervasive computing systems. They presented a hybrid model encompassing a trust model, a security model and a risk model for pervasive computing. Their model is dynamic and lightweight. It is adaptable to the changes of scenarios by choosing different thresholds and factors. This framework supports accepting or rejecting a service request based on the trust and risk values' calculation. Unfortunately, it cannot automatically ensure a trust relationship that is easily changed in a dynamic environment during the fulfillment of the accepted service.

Shand, Dimmock, and Bacon (2004) presented a trust and risk framework to facilitate secure collaboration in ubiquitous and pervasive computer systems. It used a system of trust-evaluated recommendations combined with an explicit risk analysis to control the exchange of personal information between handheld computers. In this work, trust and risk evaluation is used for controlling the access of personal information.

Claycomb and Shin (2006) presented a visual framework for securing impromptu collaboration in a pervasive computing environment. The framework incorporates a method of demonstrative identification of mobile devices, key-based capability list for resource access, and two-dimensional visual barcode technology to support a simple and convenient access control service between mobile devices.

To support the dynamic of trust, Yin, Ray, and Ray (2006) developed a trust model for pervasive computing applications and develop strategies for establishing trust between entities. The model accommodated the notion of different degrees of trust, identified how to determine a trust value, and defined how trust changes over time. When an entity has no information about its counterpart and cannot determine its trust value, a trust negotiation strategy was provided to establish trust. We hold the same motivation as this work towards trust management in a pervasive system, but with a different approach. Our solution doesn't need any negotiation procedure. On the basis of a Root Trust module for trusted computing, the trustor device can specify and ensure its trust conditions and policies for autonomic trust management at the trustee device. Thus, our solution is more efficient without multi-step negotiations involved.

Spanoudakis (2007) outlined a programme of research focusing on the development of a platform for dynamic trust assessment of software services. This platform does not provide any autonomic trust management mechanisms for device collaboration in a pervasive system.

A flexible, manageable, and configurable trust framework for the security of pervasive computing applications was proposed by Wolfe, Ahamed, and Zulkernine (2006). This trust framework minimized the effects of malicious recommendations related to trust from other devices and have the capability to transfer security functionality from devices with limited computing resources to other secure and powerful devices. Within the framework, wireless devices are broken down into different categories based upon available resources and desired security functionalities. A device's categorization determines its security functionalities and interactions with neighboring devices. It realized trust management based on a scheme for categorizing devices,

calculating trust, and facilitating trust-related communications.

As can be seen, none of above work has considered how to support autonomic control and management of trust requested by a trustor device on a trustee device for the fulfillment of an intended service. This greatly influences the effectiveness of trust management. The main problem is that trust could be easily lost due to the dynamic influence of the environment although initial trust can be built up based on the existing solutions.

3 FOUNDAMENTAL TECHNOLOGIES

Our autonomic trust management solution is built upon a mechanism for sustaining trust among computing platforms which is used to satisfy the trustor's trust conditions at the trustee computing device. In order to support autonomic trust management on services, an adaptive trust control model is applied to ensure that the trustee device will perform as the trustor device's expectation during the fulfilment of an intended service. This section briefly introduces these two fundamental technologies.

3.1 A Mechanism to Sustain Trust

3.1.1 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". 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 signal of distrust behaviours, 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 (refer to Section 3.2 and Section 4). 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 trust over a longer period of time.

3.1.2 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 also be a build-in feature in the current TC platform's Trusted Platform Module (TPM) and 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 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 1 illustrates the basic structure of this module.

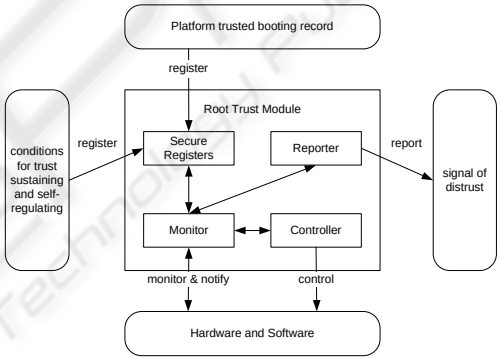

Figure 1: Root trust module.

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.

3.1.3 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 work done in industry and in academy (TCG, 2003, Vaughan-Nichols, 2003, England, et. al., 2003).

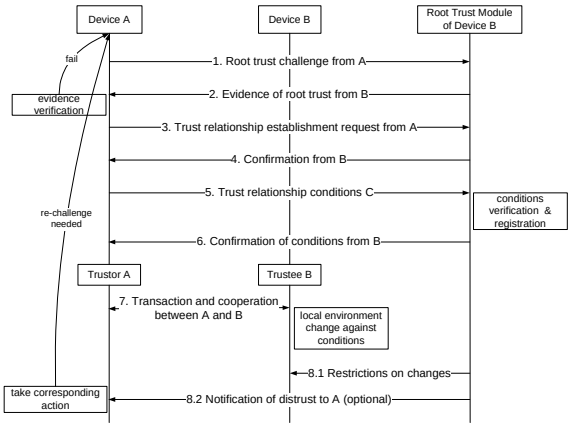

Figure 2: Protocol of trust sustainability.

As shown in Figure 2, 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; (Note that if the attestation in this step is not successful, the trust relationship between device A and B can not be established.)
b) Trust establishment by specifying the trust conditions and registering them at the trustee's RT module for trust sustaining in steps 3-6;
c) Sustaining the trust relationship through the monitor and control by the RT module in steps 7-8;
d) Re-challenge the trust relationship if necessary when any changes against trust conditions are reported.

3.2 An Adaptive Trust Control Model

Herein, we introduce an adaptive trust control model via applying the theory of Fuzzy Cognitive Map (FCM) in order to illustrate the relationships among trust, its influence factors, the control modes used for managing it, and the trustor's policies (Kosko, 1986).

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 model can be described as a graphical illustration using a FCM, as shown in Figure 3. 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 the 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 inside 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.

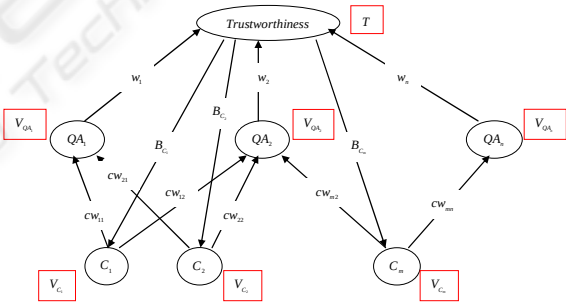

Figure 3: Graphical modeling of trust control.

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 device configurations about which control modes are applied. The trustworthiness value can be described as:

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

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 attribute is considered at the trust decision or assessment. w_i can be decided based on the trustor's policies. We apply the Sigmoid function as a 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:

$$V_{QA_i} = f\left(\sum_{j=1}^{m} cw_{ji} V_{C_j} B_{C_j} + V_{QA_i}^{old}\right) \quad (2)$$

where cw_{ji} is the influence factor of control mode C_j to QA_i, cw_{ji} is set based on the impact of C_j to 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

$$V_{C_j} = f\left(T \cdot B_{C_j} + V_{C_j}^{old}\right) \quad (3)$$

4 AUTONOMIC TRUST MANAGEMENT

In this section, we firstly specify a simple pervasive system model that plays as our working definitions. We then present the design of the autonomic trust management framework, followed by an autonomic trust management procedure with a number of algorithms' support.

4.1 A System Model

A pervasive system is described in Figure 4. It is composed of a number of pervasive computing devices. The devices offer various services. They could collaborate together in order to fulfill an intended purpose requested by a pervasive system user. We assumed that the pervasive computing device has a Root Trust module as described in Section 3.1, 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 the services 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 RT module's support. The ATMF is responsible for managing the trustworthiness of the services.

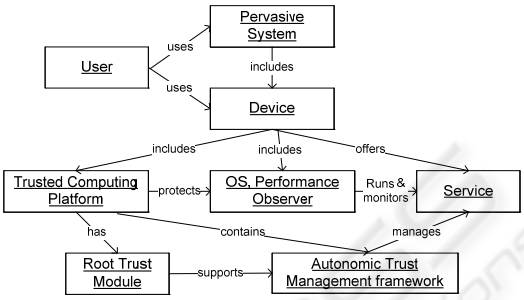

Figure 4: Model of a pervasive system.

4.2 Autonomic Trust Management Framework (ATMF)

As mentioned above, the ATMF is applied to manage the trustworthiness of a trustee service by configuring its trust properties or switch on/off the trust control mechanisms, i.e. selecting a suitable set of control modes. Its structure is shown in Figure 5.

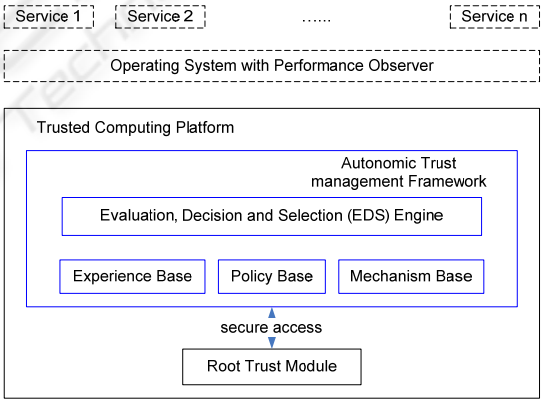

Figure 5: Autonomic trust management framework.

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 service performance monitoring results regarding quality attributes. The experience data could be accumulated locally or 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 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 Autonomic Trust Management Procedure

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 pervasive system, as shown in Figure 6.

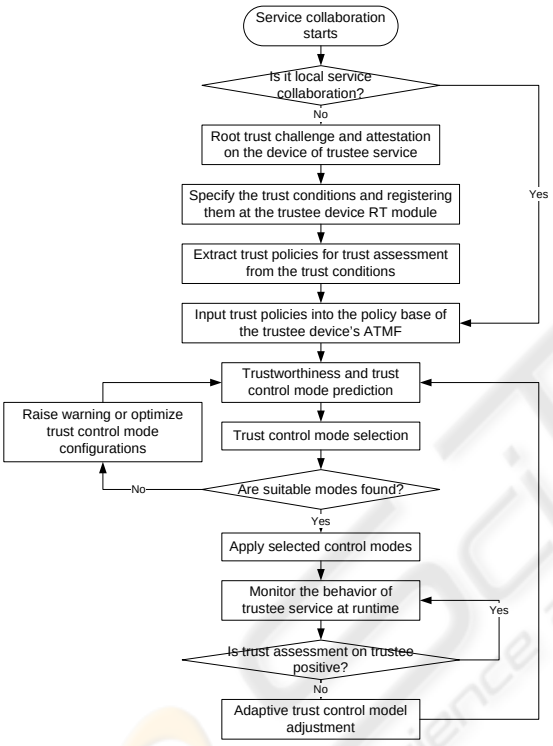

Figure 6: Autonomic trust management procedure.

The device locating 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 through its RT module. Meanwhile, the trustor service's trust policies will also be embedded into the trustee device's RT module 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.

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 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 those modes 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 trustor's subjective policies by evaluating the trustee entity's quality attributes. It is also influenced by the system context. The quality attributes of the entity can be controlled or improved via applying a number of trust control modes, especially at the service runtime.

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 initiate the trust relationship between the trustor and the trustee. Further, a runtime trust assessment mechanism is triggered to evaluate the trustworthiness of the trustee by monitoring its behavior based on the instruction of the trustor's policies. According to the runtime trust assessment results in the underlying context, the trustee's device conducts trust control model adjustment in order to reflect the real system situation if the assessed trustworthiness 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 a suitable set of trust control modes in order to conduct autonomic trust management.

4.4 Algorithms

Based on the adaptive trust control model, we design a number of algorithms to implement each step of the procedure shown in Figure 6 for autonomic trust management at the service runtime. These algorithms include trust assessment, trust control mode prediction and selection, and adaptive trust control model adjustment.

4.4.1 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 trustor's policies, the positive point (*p*) of that attribute is increased by 1. If the monitored result is worse than the policies, the negative point (*n*) of that attribute is increased by 1. The trust opinion of each quality attribute can be generated based on an opinion generator, e.g.

$$\theta = p/(p+n+r), r \geq 1 \quad (4)$$

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

$$\theta_T = \sum ir_i \theta_i \quad (5)$$

By comparing to a trust threshold opinion (*to*), the EDS engine 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.

4.4.2 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) (refer to Appendix A). We then select the most suitable control modes based on the above prediction results with a selection algorithm. Appendix B presents the detailed algorithms based on the adaptive trust control model.

4.4.3 Adaptive Trust Control Model Adjustment

It is important for the trust control model to be dynamically maintained and optimized in order to precisely reflect 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 the real system 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 an underlying context. Therefore, the adaptive trust control model is important for supporting autonomic trust management for the pervasive system. We developed a couple of schemes to adaptively adjust the trust control model in order to achieve the above purpose (refer to Appendix C).

5 AN EXAMPLE APPLICATION

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 our past work (Yan and Prehofer, 2007, Yan and MacLaverty, 2006).

The concrete example is a mobile healthcare application. It is composed of a number of services located at different devices. For example, a health sensor locates at a potable 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 each device and service's trustworthiness, 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 its user's. 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 the mobile device which reports his/her health data to the healthcare centre in a secure and efficient way. In this case, the hospital service should be 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.

Table 1: Autonomic trust management for a healthcare application.

Trustor	Trustee	Example trust requirements	Autonomic trust management mechanisms
Health sensor	Healthcare client	Trust policies (data confidentiality: yes; data integrity: yes; service availability – response time: <3s; service reliability – uptime: >10m)	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
Mobile device	Healthcare centre	Trust conditions (device and trust policies integrity: yes)	Trust sustaining mechanism to ensure the integrity of healthcare centre and trust policies for consultant service
Healthcare client	Consultant service	Trust policies (authentication: yes; data confidentiality: yes; data integrity: yes; service availability – response time: <30s; service reliability – uptime: >10h)	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
Healthcare centre	Mobile device	Trust conditions (device and trust policies integrity: yes)	Trust sustaining mechanism to ensure the integrity of mobile device and trust policies for healthcare client service
Healthcare centre	Hospital server	Trust conditions (device and trust policies integrity: yes)	Trust sustaining mechanism to ensure the integrity of hospital server and trust policies for hospital service
Consultant service	Hospital service	Trust policies (authentication: yes; data confidentiality: yes; data integrity: yes; service availability – response time: <10m; service reliability – uptime: >10h)	Control mode prediction and selection, trust assessment, trust control model adjustment and control mode re-selection to ensure the hospital service's trustworthiness
Consultant service	Hospital service	Trust policies (authentication: yes; data confidentiality: yes; data integrity: yes; service availability – response time: <10m; service reliability – uptime: >10h)	Control mode prediction and selection, trust assessment, trust control model adjustment and control mode re-selection to ensure the hospital service's trustworthiness

6 FURTHER DISCUSSIONS

Our proposed solution supports autonomic trust management with two levels. The first level implements 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. Regarding the second level, the trustworthiness of the service is automatically managed based on the adaptive trust control model at its runtime. Both levels of autonomic trust management can cooperate to ensure the trustworthiness of the entire pervasive system. From this point of view, none of the existing work reviewed provides a similar solution. Our 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 future. Our 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 services for the operating system (OS) and application software. It also has a number of security enforcement mechanisms (e.g. secure booting, integrity checking and device authentication). Particularly, it provides cryptographic functions and secure storage. 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.

In addition, the secure register of 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 and Ekberg, 2008).

7 CONCLUSIONS

In this paper, we presented our arguments for autonomic trust management in the pervasive system. In our brief literature review, we found that related work seldom supported autonomic trust management. We proposed an autonomic trust management solution based on the trust sustaining mechanism and the adaptive trust control model. The main contribution of our 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 risk by stopping or restricting any potential risky activities based on the trustor's specification. We demonstrated the effectiveness of our solution by applying it into an example pervasive system. We also discussed the advantages of and implementation strategies for the solution.

For future work, we will study the performance of our solution through a prototype implementation on the basis of a mobile trusted computing platform.

REFERENCES

Asokan, N., Ekberg, J., 2008. A platform for OnBoard credentials. Financial Cryptography and Data Security 2008.

Campbell, R., Al-Muhtadi, J., Naldurg, P., Sampemane, G., Mickunas, M.D., 2002. Towards security and privacy for pervasive computing. In *proceedings of International Symposium on Software Security*.

Denning, D.E., 1993. A new paradigm for trusted Systems. *Proc. 1992-1993 workshop on New security paradigms*, pp. 36-41.

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

Huang A. B., 2002. The trusted OC: skin-deep security, *Computer*, Vol.35, No.10, pp. 103-5.

Kosko, B., 1986. Fuzzy cognitive maps. *International Journal Man-Machine Studies*, Vol. 24, pp. 65-75.

Shand, B., Dimmock, N., Bacon, J., 2004. Trust for ubiquitous, transparent collaboration. *Wireless Networks*, Vol. 10 Issue 6, pp. 711-721.

Spanoudakis, G., 2007. Dynamic trust assessment of software services. *2nd international workshop on Service oriented software engineering*, pp. 36-40.

Sun, T., Denko, M.K., 2007. A distributed trust management scheme in the pervasive computing environment. *Canadian Conference on Electrical and Computing Engineering*, pp. 1219-1222.

TCG, Trusted Computing Group, 2003. Trusted Platform Module - TPM Specification v1.2. Retrieved May, 2006, from https://www.trustedcomputinggroup.org/specs/TPM/

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

William, C., Shin, D., 2006. A visual Framework for securing impromptu collaboration in pervasive computing. *International Conference on Collaborative Computing: Networking, Applications and Worksharing*.

Wolfe, S.T., Ahamed, S.I., Zulkernine, M.A., 2006. Trust framework for pervasive computing environments. *IEEE International Conference on Computer Systems and Applications*, pp. 312-319.

Xu, W., Xin, Y., Lu, G., 2007. A trust framework for pervasive computing environments. *International*

Conference on Wireless Communications, Networking and Mobile Computing, pp. 2222-2225.

Yan, Z., Cofta, P., 2004. A mechanism for trust sustainment among trusted computing platforms. *The 1st International Conference on Trust and Privacy in Digital Business*, LNCS 3184, pp. 11-19.

Yan, Z., MacLaverty, R., 2006. Autonomic trust management in a component based software system. *ATC'06*, LNCS 4158, pp. 279-292.

Yan, Z., Prehofer, C., 2007. An adaptive trust control model for a trustworthy component software platform. *ATC07*, LNCS 4610, pp. 226-238.

Yin, S., Ray, I., Ray, I., 2006. A trust model for pervasive computing environments. *International Conference on Collaborative Computing: Networking, Applications and Worksharing*.

APPENDIX

A An Algorithm for Control Mode Prediction

This algorithm is used to anticipate the performance or feasibility of all possibly applied trust control modes. Note that a constant δ 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

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

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

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

B An Algorithm for Control Mode Selection

The algorithm below is applied to select a set of suitable trust control modes based on the control mode prediction results.

- Calculate selection threshold $thr = \sum_{k=1}^{K} T_k / K$;
- Compare $V_{QA_i,k}$ and T_k of S_k to thr, set selection factor $SF_{S_k} = 1$ if $\forall V_{QA_i,k} \geq thr \wedge T_k \geq thr$; set $SF_{S_k} = -1$ if $\exists V_{QA_i,k} < thr \vee \exists T_k < thr$;
- For $\forall SF_{S_k} = 1$, calculate the distance of $V_{QA,k}$ and T_k to thr as $d_k = \min\{|V_{QA_i,k} - thr|, |T_k - thr|\}$; For $\forall SF_{S_k} = -1$, calculate the distance of $V_{QA_i,k}$ and T_k to thr as $d_k = \max\{|V_{QA_i,k} - thr|, |T_k - thr|\}$ only when $V_{QA_i,k} < thr$ 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.

C Schemes for Adaptive Trust Control Model Adjustment

The following two schemes are used to adjust the influence factors of the trust control model in order to make it reflect the real system situation. 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. 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. The first scheme is an equal adjustment scheme, which holds a strategy that each control mode has the same impact on the deviation between $V_{QA_i}\_monitor$ and $V_{QA_i}\_predict$. The second one 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 $V_{QA_i}\_monitor$ and $V_{QA_i}\_predict$.

C.1 An Equal Adjustment Scheme

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

C.2 An Unequal Adjustment Scheme

- While $|V_{QA_i}\_monitor - V_{QA_i}\_predict| > \sigma$, do
  a) If $V_{QA_i}\_monitor < V_{QA_i}\_predict$, for $\max(|cw_{ji}|)$, $cw_{ji} = cw_{ji} - \omega$, 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