Trust Model for Mobile Devices in Ubiquitous Environment

Zhefan Jiang and Sangwook Kim

Department of Computer Science, Kyungpook National University,
Sankyuk-dong 1370, Bukgu, Daegu, Korea
{zfjiang,swkim}@cs.knu.ac.kr

Abstract. In ubiquitous computing environment, people carrying their mobile devices (eg., mobile phone, PDA, embedded devices) expect to access locally hosted services or resources anytime, anywhere. These mobile devices have restricted capabilities and security supports. Traditional security management systems used definite access control policies for each role or user in each domain server or agent. But in ubiquitous environment, it is hard to specify authorization policies for mobile users and it is inflexible and unavailable for security management of users or mobile devices. To solve these problems, we need trust-based management mechanism as a reference to security management systems. Trust model contains trust relationship and calculation of trust value. Experiences and recommendations are the factors to calculate trust value. In this paper, we design a trust model to calculate trust value and a trust management architecture which can be running in various domain servers and mobile devices.

1 Introduction

In ubiquitous computing environment, people carrying their mobile devices (eg., mobile phone, PDA, embedded devices) expect to access services or resources anytime, anywhere. But they don't know these services are trustworthy or not. At the same time, service domains don't know how to trust mobile users. Traditional security management systems used definite access control policies for each mobile user or device. But in ubiquitous environment, it is hard to specify authorization policies and it is inflexible and unavailable for security management. Trust-based security management defines a trust model to allow entities to compare the trustworthiness of other entities for security decisions[12]. It captures the dynamic aspects and human intuitions about trust for using in security management. It enhances the existed security management and makes more easier to do collaboration works.

In this paper, we propose a trust model for mobile users. This model is used experience and recommendation as factors to compute trust value. We present new computation method to compute trust value according to transaction history and enhance the recommendation protocol to propagate recommendation requests. We also designed a trust-based management system which can be used in mobile devices or domain servers.

T. Enokido, L. Barolli, and M. Takizawa (Eds.): NBiS 2007, LNCS 4658, pp. 426–434, 2007.

The remainder of this paper is structured as follows. In section 2, we describe some related work. Section 3 shows our trust model which includes the trust relationship and calculation algorithms of trust value. We present our trust management architecture in section 4. Finally, we draw some conclusions and outlines directions for future work.

2 Related Work

In this section, we briefly highlight several existing trust management systems. The basic part in trust management system is the trust model which defines trust relationships and the computation mechanisms for trust value.

The mains trust factors calculating trust value are experiences and recommendations. For instance, in [2,3] mainly used experience between a trust and a trustee. [4,5,7,9] used both experience and recommendation to calculate trust value. VTrust used both experience and recommendation as trust factors, it calculated experience value which given weight for each action[6]. These researches considered that each negative or positive action gave the same effect to evaluate trust value. [8] designed a trust evolution model, they used mathematical and probabilistic model to calculate trust value. In [11], they set more weight to negative actions when they calculated experience value. In [13], the authors distinguished transaction amounts and computes different impact factors when computing trust values. In [13], the authors distinguished transaction amounts and computes different impact factors when computing trust values. These works considered different impact factors to compute trust value with different views and implemented their trust management systems.

In this paper, we propose a dynamic trust model for mobile devices. We consider security level of a target service and give more weight to continuous negative actions to calculate experience value. And we enhanced our recommendation protocol to propagate recommendation re-quests. We also consider security capabilities of mobile devices to reduce risks.

3 Proposed Trust Model

A Trust model defines trust relationship and the computational mechanism for trust value.

3.1 Trust Relationship

Truster(Tr) trusts Trustee(Te) to perform actions to the specific Services(Se) when Contexts(Cs) are satisfied during a TimePeriod(TP). Tr and Te can be users or intelligent devices. As are performed actions to the Tr's resources. TV is a trust value. Trust value is the number in the range [-1, 1]. The value in the positive region is used to express trust and in the negative region is used to express distrust. 1 means complete trust and -1 means complete distrust. 0 indicates trust neutral value.

$$\{ Tr, \quad Te, \quad TV, \quad Se, \quad Cs, \quad TP \}$$

3.2 Experience

Experience is the most important and direct factor to evaluate a trust value. It is calculated by the past interactions between a truster and a trustee. The calculations of the trust value are different from the domain management applications and administrators disposition. This disposition also determines how trust value is updated after interactions[11].

As an interaction result, an action can be a positive action(a+ = 1)or a negative action(a- =1). For calculation of experience value, we consider following several matters. First is the security level of a target service. For instance, one negative action performed on a target service which the security level set to high and one negative action performed on a target service which security level set to low. These two actions have different effects on the trust evaluation. Security level(SL) is a integer number in the range [1, n] according to the service domain or applications. N is the highest security level.

$$Va_j = \max\left\{ \frac{a_j * 2^{cn}}{Total_a} * \frac{SL_j}{SL_n} \quad , \quad -1 \right\} \tag{1}$$

Intuitively, trust is hard to gain, easy to lose[8]. The continuous negative actions give more effect than non-continuous negative actions. And second continuous negative action has given more disappointment than the first negative action to a truster. So we give more weight to continuous negative actions. We use equation(1) to calculate each action value Va_j, which is rewarded or penalized according to past interaction.

a_j is a jth action, it can be positive action or negative action. $Total_a$ is the total action number of 1 period. SL_n is the highest security level in an applying domain and SL_j is a security level of target service which jth action performed. cn is a counter number of continuous negative actions, cn is established by default to 0, cn is increased to 1 when continuous negative actions performed. For instance, if there are two continuous negative actions, cn is increased to 1, if there are three continuous negative actions, cn is increased to 2. After that, a positive action performed, it set to 0 again.

The current trust value according to experience(EV_i) is recalculated according to previous trust value(EV_{i-1}) and the current action value. User can configure weight ß to current action Va, ß is in the range[0,1]. We configure 0.5 as a default weight. The new trust value is calculated according to equation(2):

$$EV_i = EV_{i-1} * (1 - | ß * Va_j |) + ß * Va_j \tag{2}$$

Figure 1 shows the experiments of experience evaluation using 30 actions performed in 1 period. We assume the initial trust value is 0 and the target services are classified into four security levels(unclassified=1, classified=2, secret=3, top secret=4). We can see the trust value was drop faster when the continuous negative actions performed. And trust values are changed according to weight ß.

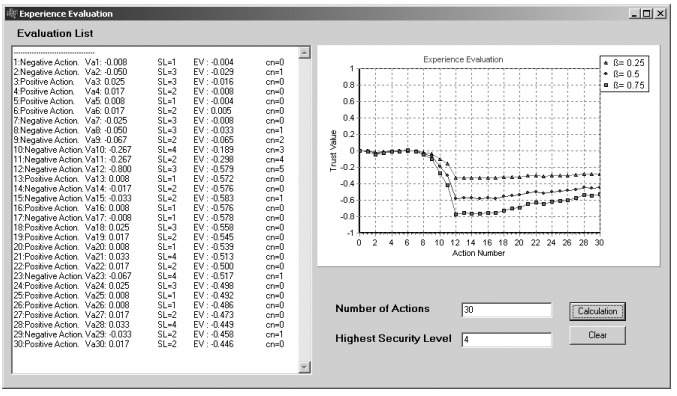

Fig. 1. Experience Evaluation according to Continuous Negative Actions and ß

3.3 Recommendation Protocol

Recommendation is used when there in no experience or insufficient information between a truster(requester) and a target entity(trustee). A truster sends recommendation request messages to recommenders who have a trust value higher than a certain threshold. Recommendation protocol is used to exchange recommendation messages as shown in Figure 2. In our recommendation protocol, a truster can set Hop which the maximum number of cascade propagation of recommendation request. It is only valid when the recommenders do not have trust information about the target entity. For instance, when the requester(Truster) set the Hop as 1, it means that the recommenders can transmit the recommendation requests as a requester to another trust entity and set the Hop as 0 and set the new TP as a TP/2.

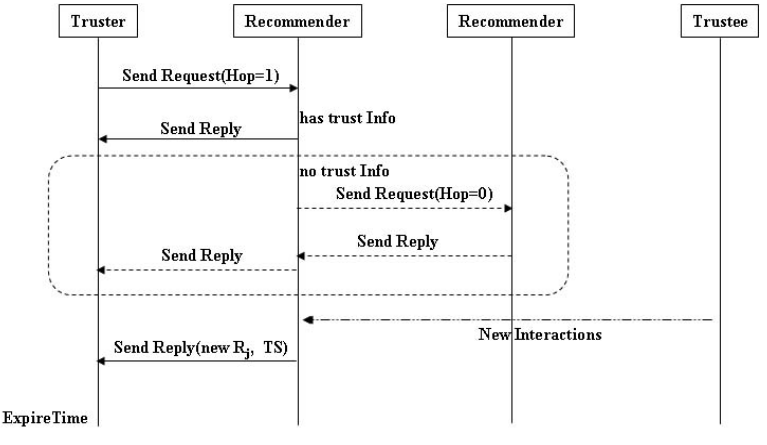

Fig. 2. Recommendation Protocol

Recommendation Request Message: When a truster(requester) receives the interaction request from a trustee(target entity), but the truster doesn't know about trustee or consider the experience information is not enough, then the truster sends recommendation requests to recommenders which trust value(TR_j) is higher than a specific threshold($TV_{threshold}$). Its message format is the following:

RRQ::={Reqster_ID, Req_ID, Rec_ID Te_ID, TP, Hop}

Req_ID is the unique request identifier and Reqster_ID, Rec_ID, Te_ID are identifiers or credentials for a requester, recommender, and trustee entity. TP is the request message's expiration time. Hop is the maximum number of cascade propagation of recommendation request. It is only valid when the recommenders do not have trust information about the target entity. For instance, when the requester set the Hop as 1, it means that the recommenders can transmit the recommendation requests as a requester to another trust entity and set the Hop as 0 and set the new TP as a TP/2. 0 means that the recommenders don't send requests to others any more. RV is a recommendation value for a trustee. R_j is jth recommender's recommendation value. m is the total number of recommenders. Figure 3 shows the algorithm and implementation of a truster and recommender.

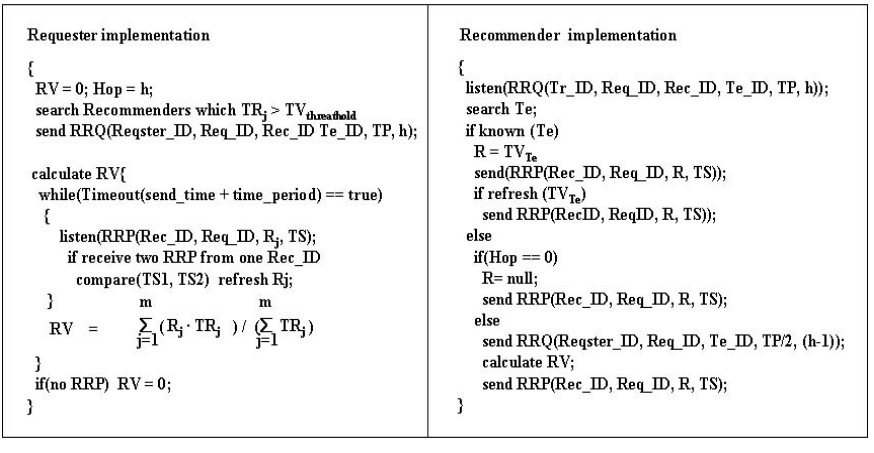

Fig. 3. Implementation of a Requester and a Recommender

Recommendation Reply: It is used to send response back per request. The recommender sends reply message with a trust value of the target entity and TS(Timestamp).

RRP::={Rec_ID, Req_ID, R, TS}

If the recommender get new trust value before the TP expired, it sends a recommendation replay message again which set new R(recommendation value) and TS. The final recommendation value(RV) is used in computing final trust value for the trustee.

3.4 Trust Evaluation

For evaluating the trust value(TV), a truster may assign different weights to the experience factor and recommendation factor according to equation(3). EV is the trust value according to experience with weight = a. RV is the recommendation value according to recommendation. The weights will specify in trust evaluation policy.

$$TV = EV * a + RV * (1-a) \quad (0 <= a <= 1) \tag{3}$$

3.5 Risk Management

In this paper, we propose two schemes to reduce risks. First scheme is comparing the minimum security requirements of target services for interaction and security capabilities of mobile devices.

The security capabilities for mobile devices include secure communication protocol, cryptographic algorithms, authentication schemes. But they are still restricted. Although a truster and trustee has a trust relationship and the trust value is high, if the security capabilities for mobile devices can't be satisfied with the security requirements for the target objects, the access request could be denied. For instance, the domain server specifies that accessing a file transfer service must use SSL protocol, but if the mobile device has not support this secure protocol. So domain server can deny this request.

Second is implicit in trust evaluation to reduce risks. It is a scheme that if a truster has a low trust value for a trustee, this trustee only can access the service which has a low security level. For instance, a truster can configure risk rules as shown in Figure 4.

security level 1	-1	<=	TV	<	-0.5
security level 2	-0.5	<=	TV	<	0
security level 3	0	<=	TV	<	0.5
security level 4	0.5	<=	TV	<=	1

Fig. 4. Risk Rules

If the truster has a trust value which equals -0.6 for a trustee, the trustee only can access the target service which security level is 1.

4 Trust Management System

We design our trust management architecture as shown in Figure 5. It includes a security manager and a trust agent. A trust agent is an assembly of software components for the trust management.

Security manager: It consists of a request analyzer, monitor, access control manager and policy repository. When a trustee sends a service request for interaction, the request analyzer analyzes the request information. The access control manager is responsible for searching the policy repository about the target service's security

level(SL) and security requirements and the minimum trust value. Then it sends these information to the trust agent to evaluate trust value. It makes a decision the request would be granted or denied. The policy repository stores the security policies which specify service name, security level, trust value. The monitor is monitoring the status of domain resources and events.

Trust Agent: It consists of experience, recommendation protocol, trust inference engine, risk manager and a trust policy repository. Experience collects interaction history for each trustee and calculates the experience value. Recommendation protocol sends recommendation requests and receives recommendation reply from recommenders. Trust policy repository stores trust policies and trust values for each trustee.

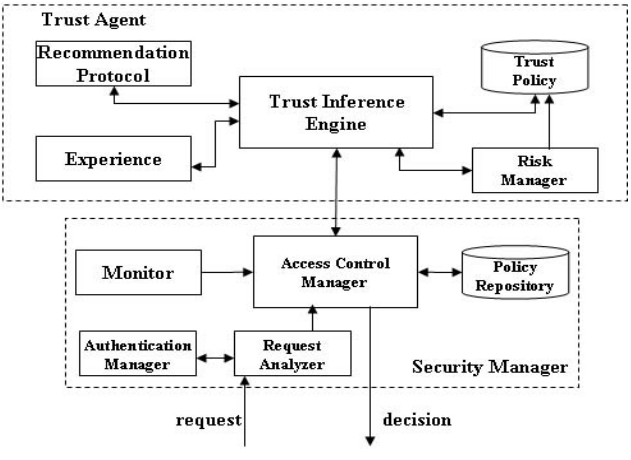

Fig. 5. Trust Management Architecture

Trust policy specifies as following 7 fields like the trust relationship. a is the weight of experience value and ß is the weight of last action value of interaction.

[Trustee, Services, TrustValue, Contexts, ValidTime, a, ß]

The risk manager is responsible for analyzing the risks. The trust inference engine refers to trust policies and computes new trust value. Then it sends the new trust value and risk information to security manager to make a decision for requests.

We have implemented the trust management system. Figure 6 shows the experiment of our trust management system and recommendation process. We used two domain servers which has our trust management system. Domain Server 1 has the trust information for PDA and it calculated the mobile user's trust value according to interaction history. When this mobile user move to another service domain and want to access services, but domain server 2 doesn't have trust information about this mobile user. If domain server 2 trusts domain server 1, domain server 2 sent recommendation request to domains server 1 and made a decision according to recommendation value.

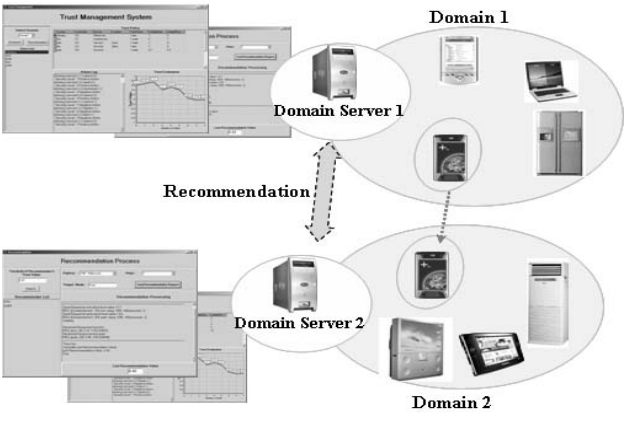

Fig. 6. Trust Management System and Recommendation Process

5 Conclusion and Future Work

We have designed a trust model for mobile users and devices. It enhances the existed security management and makes more easier to do collaboration works. This model is used experience and recommendation as factors to compute trust value. It can express that each action's effects are different according the security level of target service and continuous negative action counters. We also enhance the recommendation protocol to propagate recommendation requests.

We also implemented a trust management system to be used in mobile devices or domain servers. Our model minimizes the risks from the restricted security capabilities of mobile devices. Now we are analyzing the performance of our trust model. Furthermore, we will develop more efficient trust management system for using a various domains.

Acknowledgement

This research was supported by the MIC of Korea, under the ITRC support program supervised by the IITA(IITA-2006-C1090-0603-0026).

References

1. Gambetta, D.: Can we trust trust? In: Gambetta, D. (ed.) Trust, Making and Breaking Cooperative Relations, pp. 213–237. Basil Blackwell, Oxford (1988)
2. Virendra, M., Upadhyaya, S.: Securing Information through Trust Management in Wireless Networks. In: Workshop on Secure Knowledge Management (2004)
3. Shi, J., von Bochmann, G., Adams, C.M.: A Trust Model with Statistical Foundation. In: 18th International Federation for Information Processing (IFIP) World Computer Congress TC1 WG1.7 Workshop on Formal Aspects in Security and Trust (2004)

4. Abdul-Rahman, A., Hailes, S.: A Distributed Trust Model. In: Proceedings of the workshop on New security paradigms (1997)
5. Josang, A., Ismail, R., Boyd, C.: A Survey of Trust and Reputation Systems for Online Service Provision. In: Decision Support Systems (2006)
6. Ray, I., Chakraborty, S.: VTrust: A Trust Management System Based on a Vector Model of Trust. In: International Conference on Information Systems Security (2005)
7. Secure environments for collaboration among ubiquitous roaming entities (SECURE) (2001)
8. Almenarez, F., Marin, A., et al.: Developing a Model for Trust Management in Pervasive Devices. In: Proceedings of the Fourth IEEE Conference on Pervasive Computing and Com-munications Workshop. IEEE Computer Society Press, Los Alamitos (2006)
9. Grandison, T.: Trust Management for Internet Applications. PhD thesis, Imperial College of Science, University of London, Department of Computing (2003)
10. Lin, C., Varadharajan, V., et al.: Security and Trust Management in Mobile Agents: A New Perspective. In: Proceedings of The Second International Conference on Mobile Technology, Application and Systems (November 15-17, 2005)
11. Jameel, H., Hung, L., et al.: A Trust Model for Ubiquitous Systems based on Vectors of Trust Values. In: Proceedings of the Seventh IEEE International Symposium on Multimedia. IEEE Computer Society Press, Los Alamitos (2005)
12. McKnight, D., Chevany, N.: The Meanings of Trust. Working paper, Carlson School of Management, University of Minnesota (1996)
13. Wang, Y., Lin, F.: Trust and Risk Evaluation of Transactions with Different Amounts in Peer-to-Peer E-Commerce Environments. In: The IEEE International Conference on e-Business Engineering. IEEE Computer Society Press, Los Alamitos (2006)