Developing a Model for Trust Management in Pervasive Devices*

Florina Almenárez, Andrés Marín, Daniel Díaz, Juan Sanchéz
Department Telematic Engineering, University Carlos III of Madrid
E-28911 Leganés (Madrid), Spain
{florina, amarin, dds, jjsanchez}@it.uc3m.es

Abstract

Pervasive devices interacting in open and dynamic spaces with each others require a mechanism that allows them acting autonomously in a secure way and protecting their resources. Trust is fundamental to establish communication with other users, because the identity is often uncertain and on one's own does not provide trust information, for instance, could an unknown user be trustworthy? Nowadays, these devices have a so limited security support. So, we propose a simple trust management model to enhance such support, allowing them interact in ad hoc networks and peer-to-peer applications in a secure way. In this paper, our main contribution is a mathematical and a probabilistic model, as well as demonstrating the model feasibility, since it has been assessed through the prototype implementation, which has been tested in a Pocket PC.

1. Introduction

Pervasive Computing involves open and dynamic environments which comprise a multitude of heterogeneous devices with communication, computing and storage capabilities. Such devices interact with each other, forming ad hoc networks, to offer/demand any kind of service. The presence of mobile devices with restricted capabilities, called pervasive devices, is very frequent. For instance, people in the zoo carrying their mobile phone, PDA, digital camera, etc., can use the information services offered by the devices embedded, or even, they can share multimedia information about certain animals among some people interested in it. In this way, they create environments of cooperation and collaboration to benefit from the common knowledge.

In these environments, TRUST arouses a big interest to establish new relationships; just as it happens in the real world. As early as 1994, Marsh defines trust as the basis for the cooperation among autonomous agents [13]. Later, this concept was introduced into public key infrastructures [20], [14], in order to assess the authenticity of public keys. Other approaches are defined in [6, 5, 17] and [8]. However, these models have drawbacks when they are applied to open and dynamic environments, or developed in limited devices.

Pervasive devices require a simple model which provide them certain autonomy to manage their own security, because their security support is scarce and the presence of a central server is not always possible. Thus, the trust management model should allow: a) to establish trust relationships in an autonomous way, reducing the complexity of the ad hoc interactions b) to protect our resources from malicious entities, c) to participate in ad hoc networks and peer-to-peer application in a secure way, d) to minimize the human intervention, e) to make suitable decisions despite the imperfect knowledge (uncertainty), and f) to minimize the resource consumption. In this paper, we propose, accordingly, such distributed trust management model (section 2), which involves the 'TRUSTWORTHINESS' concept, for instance, cryptographic algorithms cannot say whether the owner of a certificate is a 'hacker', but with this model it can. We have developed a prototype, whose implementation is explained in the section 3.

After explaining our contribution, in section 4, we give an overview concerning related work, and security protocols or management protocols of ad hoc networks that can be benefited from our model. Finally, we present conclusions and future work in section 5.

2. Trust Management Model

A trust management model specifies, analyses, establishes, monitors and finishes trust relationships. We have designed a model for trust management in pervasive devices, called **Pervasive Trust Management (PTM)**. Its main perspective is for users offering services, instead of users requesting a service. The latest is suitable for quality of service. It could be integrated within PTM using situational trust [13].

PTM, based on Luhmman's ideas [12], attempts to take

*Thanks to UBISEC (IST-STREP 506926) and EVERYWARE (MCyT N2003-08995-C02-01) projects.

Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06)
0-7695-2520-2/06 $20.00 © 2006 IEEE

the management in the real world to the digital world. In addition, taking the subjective nature of trust into account, the trust management is distributed and ad-hoc. PTM allows pervasive devices to manage their own security, similar to PGP [20]. Each one has its own key pair, a list of trustworthy and untrustworthy users, behavioural information and available certificates. Every user can be her/his own CA and forms a domain with all her/his devices. Trust relationships between CAs are established peer-to-peer, because hierarchical relationships are not specified for open environments. In some cases, we have certificates issued by well-known commercial CAs, which are generally preconfigured as trustworthy in our mobile devices by the manufacturer; therefore, if there are people that hold certificates issued by such CAs, these trust relationships would be used.

We use fuzzy logic to express the trust relationships, that is, a continuous function ranging from 0 to 1, being the extreme cases of complete distrust and complete trust, respectively. In addition, we include intermediate states, for instance, 0.5 would be used as ignorance value. Such relationships fulfil certain properties, such as: reflexive, nonsymmetrical, conditionally transitive, and dynamic [2].

2.1. How PTM works?

Now, let us use Alice and Bob, two unknown users willing to communicate with each other to explain how PTM works. At the beginning, we can establish the trust relationship in a *direct* or *indirect* way:

- In the first case, Alice will trust Bob without intervention of third parties, for that, Alice takes into account some available previous knowledge about Bob, otherwise Alice will use an inference engine to interpret the established rules. Such rules are based on the user's security context.
- The indirect trust relationships are given by recommendations from TTPs. A TTP is a peer who has a trust value higher than a certain threshold. Such recommendations are distributed using a pervasive recommendation protocol (PRP) among close entities or using public key certificates [2]. PRP works in a way similar to an ad-hoc service discovery protocol, whose messages are protected using digital signatures. The degree of trust is obtained by taking an average of all the recommendations weighted by the recommender's trust degree. We use the weighted average because it is very simple, allows distinctions based on the reliability of the source, and obtains results that correspond well with intuitive human judgement [2, 11].

Once the trust relationship is established, Alice calculates the Bob's degree of trust, which would be our *belief* similar to Jsang's model [10]. The belief is expressed as a set of fuzzy propositions, which represent in a qualitative way the ownership degree of a user to the set of trustworthy users. However, this belief might change over time according to the user's behaviour, providing us with a feedback about user's performance during the interaction. Each interaction is considered an *evidence*. The evidence is measured through the actions (positive or negative) performed. Negative actions are measured according to the effect occurred; therefore, we distinguish between wrong actions (bad actions that do not cause any damage or cause mild damages) and malicious actions (attacks).

2.2. Mathematical Trust Evolution Model

The value of trust (T_i) is recalculated in accordance with the previous trust value (T_{i-1}) increased or decreased in accordance with the current behaviour. The difference percentage is given by strictness factor (ω) which is related to the user's disposition regarding the present and the past. A pragmatic user maintains the equilibrium between them, therefore, ω should be a value in the range $[0.25, 0.75]$ instead of takes extreme values. ω value is established by default to 0.5, but it could be configured by the user. The new trust value is calculated according to the following equation:

$$T_i = \begin{cases} T_{i-1} + \omega.V_{a_i}(1 - T_{i-1}) & V_{a_i} > 0 \\ T_{i-1}(1 - \omega + \omega.V_{a_i}) & \text{else} \end{cases} \quad (1)$$

Where:

V_{a_i} is calculated according to the associated weight to each kind of action (W_a), which is rewarded or penalised according to the past behaviour, both positive (a^+) and negative (a^-), as shown equation 2.

$$V_{a_i} = W_{a_i}^{(m)} . \frac{(a^+ - a^-)((a^+ - a^-).\sigma)^{2m}}{(a^+ + a^-)((a^+ - a^-).\sigma)^{2m} + 1} \quad (2)$$

σ is a value in the interval $(0, 0.05]$, which can be calculated from m. This factor determines the increment, which is inversely proportional to m.

The Figure 1 depicts "Trust is hard to acquire and easy to lose". In this figure, we can see the negative behaviour decreases faster than the positive behaviour increases. For instance, with one negative action, trust value is lesser than the value obtained after several positive actions. Besides, we can see that when there are more negative actions, the growth with the positive actions is every smaller time.

2.3. Probabilistic Trust Evolution Model

The calculation of action value includes classic a priori probabilities about the user behaviour. It is well-known that

Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06)
0-7695-2520-2/06 $20.00 © 2006 IEEE

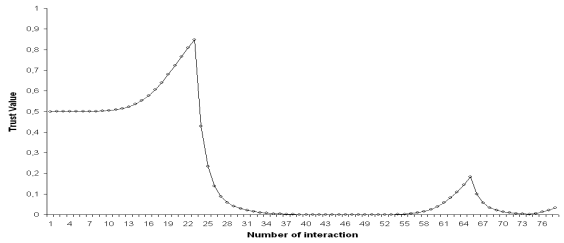

Figure 1. Trust Evolution

from a priori probabilities we can calculate posteriori probabilities, applying the Bayes' theorem (see equation 3) [16].

$$P(a^i|H_{act}) = \frac{P(H_{act}|a^i)P(a^i)}{P(H_{act})} \quad (3)$$

Where H_{act} is the historical behaviour. The density function represents the distribution of these probabilities for binary events in the interval $[0, 1]$, so:

$$P(X|a^+, a^-) = \int_0^1 X^{a^+}(1-X)^{a^-}dx = \frac{a^+!(a^- - a^+)!}{(a^+ + a^- + 1)!} \quad (4)$$

From the equation 4, is deduced:

$$f_x(X|H_{act}) = \frac{(a^+ + a^- + 1)!}{a^+!(a^-)!}X^{a^+}(1-X)^{a^-} \quad (5)$$

In conclusion, we obtain the Beta density function, which is the only function that takes into account binary events to assign belief degrees between 0 and 1. From an aleatory behaviour, we calculate the posteriori probabilities (see Table 1).

The probabilistic model could be of use for evaluating the risk. Risk is implicitly considered within the trust definition. Our model does not include a risk management module, but it could be built from this probabilistic model.

ID	a^+	a^-	$P(a^+\|H_{act})$	$P(a^-\|H_{act})$
a	4	0	1	0
b	4	1	0.80	0.20
c	6	2	0.750	0.250
d	11	3	0.786	0.214
e	11	8	0.579	0.421
f	11	11	0.50	0.50

Table 1. Posteriori Probabilities in according to the user behaviour

Fig. 2 depicts the probability distribution for this behaviour. In this figure, we can see as probability of the positive behaviour tends to 1, but it moves away as negative actions occur. With these values, we can calculate the perceived risk from a user given its historical behaviour.

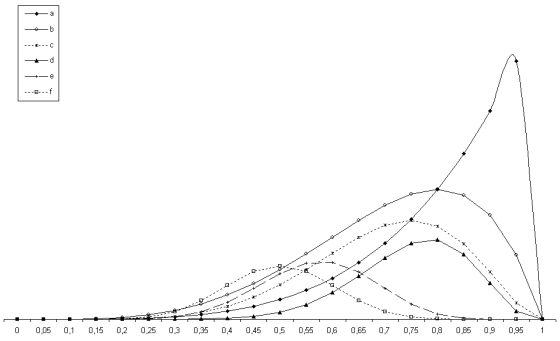

Figure 2. Beta Distribution Function in accordance with the user behaviour in the Table 1

3. Components-based PTM Implementation

PTM meets the security requirements for pervasive devices to act as secure peers, besides being secure clients. PTM is implemented in compliance with a component-based architecture, so that the components can be plugged-in and reconfigured [1] according to the requirements. We take as basis the security architecture supported by the operating systems of pervasive devices, such as Windows CE, Symbian OS, or Palm OS, to implement our prototype.

Our prototype has been using J2ME Personal Profile, OpenSSL [15], and JNI functions. We use this tools in order to guarantee interoperability with all operating systems. OpenSSL provides all cryptographic functions. This prototype has been tested in a Dell Axim X30, HP iPAQ h6340 whose operating system is Windows Mobile 2003. The service using this API was simulated with a graphical interface that generates access requests and returns the response.

In this initial prototype, the trust values are assigned to a public key, because the users are identified by this. Identity management is another problem difficult to solve in this environments, but in our model, the identity information is complemented with trust information.

3.1. Security Support of Pervasive Devices

Operating systems for pervasive devices have a security support very similar. They include both symmetric and asymmetric cryptographic algorithms, simple authentication schemes, secure communication, and secure code installation/execution. These three last are based on X.509 certificates, which are previously configured by the manufacturer (e.g. Verisign, Thawte, Entrust.net, ...). The certificate management is so restricted, therefore, it is no trivial

[1] This is an ongoing research collaborating with Marc Lacoste from France Telecom R&D, in the UBISEC project [19]

Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06)
0-7695-2520-2/06 $20.00 © 2006 IEEE

to add new certificates for "non-expert users". For example, a user can authenticate his bank, when he/she accesses to the secure web, if the bank has a certificate issued by any commercial CA preconfigured. For authorization, some devices support Role Based Access Control (RBAC) as Pocket PC or Access Control Lists (ACLs) as the latest Palm OS version, Palm OS Cobalt.

The common modules supported by such operating systems are shown in Fig. 3, which are white and the caption is in arial format. This figure includes generic security components, which consist of the following modules, as well as a logic repository:

- **Cryptographic Provider**. It implements cryptographic algorithms: symmetric (DES, RC4, RC5), asymmetric (RSA, DSA, DH), hash function (MD5, SHA-1), and key generators. It is responsible to encrypt and decrypt data, codify and decodify ASN.1, support the authentication, signature generation and verification, and secure communication.

- **Communication API**. It supports client-side SSL, which provides authentication, integrity and confidentiality of messages.

- **Credential Manager**. It is responsible to manage the digital certificates: validation, storage, recovery and removal. Credential is used as generic term to represent all digital data structures.

- **Authentication Manager**. It supports authentication user-device, using a PIN number, or even, more recent devices include biometric authentication.

3.2. Enhanced Security Support

The security support mentioned above is extended adding new components for trust management. It provides certain autonomy and dynamism to the relationships that a user can establish in a spontaneous way. We extend the function of the *Credential Manager* with a *Trust Manager*.

The UML diagram of the enhanced security architecture is shown in Fig. 3. This figure depicts the complete security support, which comprises the components that help *Trust Manager* to execute its functions, such as it is described in the model (section 2). These components are: *Recommendation Manager*, *Action Monitor*, and *Context Provider*. The new components are grey colour and have the caption in cursive format.

The *Trust Manager* is responsible for establishing the trust relationship, obtaining an initial degree of trust for new users, recalculating the new degree of trust after the user interaction according to equations in section 2.2, inserting new trusted certificates in the certificate store, managing the

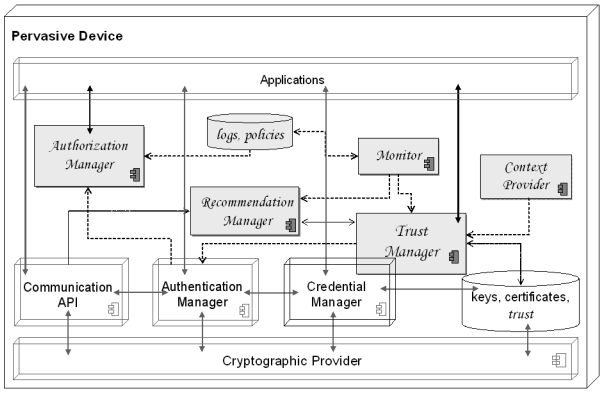

Figure 3. Enhanced Security Architecture

recommendations, and interpreting the trust rules. For that, it relies on the following components:

- *Recommendation Manager* implements the pervasive recommendation protocol (see [2]. This module is responsible for sending recommendation requests and replies in the indirect trust formation phase defined in PTM. It can also send alert messages when malicious action are detected.

- *Action Monitor* is responsible for the accounting of user actions at the device level. It also analyses the user behaviour from a log file and monitors the network interfaces. The analysis is made according to specific patterns defined by the applications, in order to determine the type of interaction. For instance, a user requesting a same service n times in a very short period t is considered a denial of service attack. The log file is generated by *Authorization Manager*. Then, this information is warning the *Trust Manager* to recalculate trust values. If an attack is detected, then it also gives information to the *Recommendation Manager*, in order to send the corresponding message.

- *Context Provider* is responsible for getting available information about external context. It acts as an interface between the context management tools, for instance, embedded in the environment and our device. External context includes location, people, environment, etc. There is another internal context provider which is contained in the *Authorization Manager* to include this information in the access control policies.

The *Trust Manager* resolves queries such as: Is it possible to establish a trust relationship with A? Is A a trustworthy user? Do I have previous experiences with A? Which is the A's degree of trust? Are there closed somebody that knows A?, among others.

Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06)
0-7695-2520-2/06 $20.00 © 2006 IEEE

The repository has been divided in two modules. It is a logical storage space for keys, certificates, information about user's behaviour, trust rules, logs, etc.

On the other hand, Fig. 3 includes an *Authorization Manager*, since trust information can be used to make access control decisions.

Finally, the *Credential Manager* is also enhanced to facilitate the certificate management by "non-expert users" and the *Communication API* to add the server-side SSL.

4. Related Work

There exist some frameworks to manage trust, such as [18], [8], and KeyNote [3]. These frameworks have been designed for servers that are managed by an administrator. For instance, in SECURE the framework is used for protection against spam, establishing whether a mail comes from a trusted user. Moreover, PGP includes trust management for the 'introducers'. Such management specifies deterministic trust values, which are assigned by the user to each one. Deterministic values do not allow granularity for sorting of the users' trustworthiness.

On the other hand, there exist secure communication protocols, such as SSL or IPSec, or the security solution for DNS, DNSSEC [7], that work well in fixed networks, in which the trust relationships are established previously based on the traditional PKI. However, when a server is not available or trust relationships are not preconfigured require a dynamic trust management. Likewise, some management mechanisms for establishing ad hoc networks, for instance, routing, are based on trust to minimize the risk of potential intruders in the network [4].

5. Conclusions

We have proposed a simple pervasive trust management model to enhance the security architecture of devices with restricted capabilities. Our main contribution, in this paper, consists of the mathematical and probabilistic model and the implementation of a prototype to demonstrate its functionality. The trust is the basis to establish relationships in a spontaneous way in infrastructure-less ad hoc networks and peer-to-peer applications. Pervasive devices can interact with closed devices in a secure way, without depend on central server or administrator. Our model minimizes the uncertainty to make suitable decisions, and allows the cooperation among closed devices to be benefited from the common knowledge.

PTM has been developed as a generic prototype, but we are integrating our model in the WCE security architecture [1]. Furthermore, we are analysing the performance and consumption of resources, in order to develop an efficient model computationally [9].

References

[1] F. Almenárez, D. Díaz, and A. Marín. Secure Ad-hoc mBusiness: Enhancing WindowsCE security. In *1st Conference on Trust Digital Business (TrustBus'04)*, 2004.

[2] F. Almenárez, A. Marín, C. Campo, and C. García. PTM: A Pervasive Trust Management Model for Dynamic Open Environments. In *First Workshop on Pervasive Security, Privacy and Trust PSPT'04*, 2004.

[3] R. Ashri, S. D. Ramchurn, J. Sabater, M. Luck, and N. R. Jennings. Trust evaluation through relationship analysis. In *4th Int Joint Conference on Autonomous Agents and Multi-Agent Systems*, July 2005.

[4] S. Buchegger and J.-Y. L. Boudec. Performance analysis of the CONFIDANT protocol (cooperation of nodes: Fairness in dynamic ad-hoc networks). In *ACM MOBIHOC'02*, 2002.

[5] S. Capkun, L. Buttyn, and J.-P. Hubaux. Self-organized public-key management for mobile ad hoc networks. In *IEEE Transactions on Mobile Computing*, volume 2, pages 52–64. IEEE, Jan–March 2003.

[6] R. Chen and W. Yeager. Poblano: A distributed trust model for peer-to-peer networks. Technical report, Sun Microsystems, Inc., 2001.

[7] D. Eastlake. Domain name system security extensions. Technical Report RFC 2535, IETF Network Working Group, March 1999.

[8] T. Grandison. *Trust Management for Internet Applications*. PhD thesis, Imperial College of Science, University of London, Department of Computing, July 2003.

[9] N. Jansma and B. Arredondo. Performance comparison of elliptic curve and rsa digital signatures. Technical report, University of Michigan College of Engineering, April 2004.

[10] A. Jøsang. An algebra for assessing trust in certification chains. In *Network and Distributed Systems Security (NDSS'99) Symposium, The Internet Society*, 1999.

[11] A. Jøsang, M. Daniel, and P. Vannoorenberghe. Strategies for combining conflicting dogmatic beliefs. In *6th International Conference on Information Fusion*, July 2003.

[12] N. Luhmman. *Trust*. MIT Press, Cambridge, MA, USA, 95.

[13] S. Marsh. *Formalising Trust as a Computational Concept*. PhD thesis, Department of Mathematics and Computer Science, University of Stirling, April 1994.

[14] U. Maurer. Modelling a public-key infrastructure. In *European Symposium on Research in Computer Security (ESORICS' 96)*. Springer-Verlag, 1996.

[15] Openssl. http://www.openssl.org/, 1999-2005.

[16] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw Hill, 1984.

[17] K. Ren, T. Li, Z. Wan, F. Bao, R. Deng, and K. Kim. Highly reliable trust establishment scheme in ad hoc networks. Elsevier Preprint, 2004.

[18] Secure environments for collaboration among ubiquitous roaming entities (SECURE), 2001.

[19] Ubiquitous networks with a secure provision of services, access, and content delivery. IST-2002-506926 (Sixth Framework Programme), 2004. http://www.c-lab.de/ubisec/.

[20] P. R. Zimmermann. *The Official PGP User's Guide*. MIT Press, Cambridge, MA, USA, 95.

Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06)
0-7695-2520-2/06 $20.00 © 2006 IEEE