Mobile Devices: Secure Clients or Secure Peers?

Florina Almenárez, Andrés Marín, Celeste Campo, Carlos García R.
Dept. Telematic Engineering, Carlos III University of Madrid
Avda. Universidad 30, 28911 Leganés (Madrid), Spain
http://www.it.uc3m.es/pervasive
{florina, amarin, celeste, cgr}@it.uc3m.es

***Abstract*— The capabilities of mobile devices are continuously increasing in processor speed, storage, and communications. These continuous enhancements enforce their presence in new and traditional applications, specially in ad-hoc and P2P networks. This article shows lacks in the security support of the mobile devices operating systems, which presently only offer encryption capabilities and some authentication schemes to control the physical access, but insufficient protection of device resources and in the secure communication management. To overcome such lacks, we propose an enhanced security architecture for mobile devices, which adds a trust manager in compliance with the pervasive trust management (PTM) model and a authorisation mechanism based on degrees of trust, TrustAC. We aim at enhancing PKI for large communities of end users, because it is the only standardised by ITU, and because it is the only one being used (but some applications for network managers using SPKI). We are also motivated to decrease the complexity of PKI for average users, and the administration required to permit secure P2P interactions, minimising user intervention.**

I. Introduction

Weiser [1] noticed we are living the era of the new paradigm of computation, Pervasive Computing. This new paradigm is becoming a reality with new applications, new services, and new actors, among others. New actors include mobile devices, which form part of the pervasive devices set. So, the proliferation of mobile devices and applications for them are increasingly growing. Although, devices are mobile, small, and limited, they have computation, communications and storage capabilities, which allow them take part in peer-to-peer applications, ad-hoc networks, mobile services, etc., with certain autonomy. For that, many challenges remain still open to overcome as regards network and service management, security, personalisation, etc.

Mobile devices[1] store personal information such as contacts, phone numbers, and calendar information but they are becoming multipurpose devices, in other words, starting to store multimedia information, identity information, or to run health care, e-commerce or business applications that manage private data. For example, people sharing games or multimedia content in a entertainment shop, users roaming several places (e.g. an airport, a shopping mall, etc.), using the services offered by embedded devices in the environment, a customer paying her/his shopping with the mobile phone, or a sales representative holding a corporate application to manage the sales and the customers in her/his Personal Digital Assistant (PDA).

In these scenarios, the mobile devices are the most sensitive point, because they are exposed to many attacks and do not have enough protection mechanisms. Several operating systems have been developed for mobile devices, such as Microsoft Windows CE (WCE), Symbian OS, Palm OS, and Embedded Linux. These operating systems (except Linux) have similar security features that include encryption capabilities and some authentication schemes to control the physical access. Access control generally comprises authentication and authorisation. Unfortunately, there is neither a widely adopted standard for access control services in mobile devices, nor a consensus over standard access control routines in the various mobile device operating systems. Although data security on mobile devices does not have a high priority, manufacturers have spent most of their efforts designing security routines for the communication protocols rather than for the data and applications stored on mobile devices [2].

As mentioned above, the security support in existent operating systems for mobile devices is focused on physical access control to unlock the device, having full access the applications, resources, etc., but, this support has lacks that we can group so:

- **Device Protection**. The resource protection is not considered, as being the access control insufficient. Likewise, the security configuration is not possible or too limited; for instance, a PDA can be a virus carrier, or an authenticated user would have unlimited access to the device's resources.
- **Secure Communication Management**. Most the mobile devices support only the SSL client side, so they might not securely communicate with each other in a peer-to-peer mode. Mobile devices are considered to connect to remote servers that offer services and they act just as a client; therefore, such a remote server must be trusted, that is, its certificate must be stored previously in a trusted certificate list, or have explicit acceptance from the user. The certificate management is based on the hierarchical PKI defined for the Internet [3]. Manufacturers previously set up the trusted certificate list, which can only be updated with the intervention of an "expert" user. PKI's trust model is not suitable for ad-hoc environments, because the certification authorities (CAs) are not always reachable, and the validation of long certification chains is heavy to be executed by mobile devices. In addition, it would be impossible to preconfigure the device with all the certificates with which we will interact in the future.

To overcome such lacks, we propose an enhanced security architecture for mobile devices that provides a wide range of security services, for devices to securely participate in P2P applications, use wireless services, protect their resources/information through an authorisation mechanism, and perform sensitive applications. Such an architecture is based on a new pervasive trust management (PTM) model for PKI that makes use of the existent trust relationships, but it can also establish new trust relationships in a spontaneous way.

The main objective is to enable security protection for either connected and disconnected modes so that: trust relations established in connected mode can be used in disconnected situations where the trust anchors (CAs) are unavailable; and trust relations established in disconnected modes can be validated when the CAs become available. The main idea is to provide mobile devices with the needed tools to manage their own security, so mobile devices would avoid the dependence on a trusted third party. Besides the presence of unknown

[1]Mobile devices refer to handheld devices, we do not consider larger devices; therefore, we will use mobile or handheld devices without distinction.

users, we consider relevant aspects for mobile users such as changes of context, trust evolution, etc.

This paper is organised as follows: Section II gives an overview about related work on security for mobile devices. In section III we present a new trust management model, PTM [4]. Section IV proposes an access control mechanism for protecting resources, Trust-based Access Control (TrustAC). Section V briefly describes the PTM and TrustAC components for implementation. Finally, in section VI we summarise and mention our future research directions.

II. Related Work

According to the security gaps presented in operating systems for mobile devices, the related work on security for mobile devices has two perspectives. The first perspective tries to provide mobile devices with protection mechanisms, and the second one delegates the security management to trusted third parties to guarantee secure communication among handheld devices.

At the same time, the works comprised in the first perspective are divided in three different categories: First, the different security architectures defined by operating systems [5], [6], [7], [8], [9], [10]. Second, works highlighting the importance of the security for handheld devices, which justifies even more this research [11], [12], [13], [14], [15], [16]; mainly, enterprises offering tools to protect the resource's devices. Third, few research works proposed to solve the security lacks in the operating systems [2], [17].

The security architectures defined by the different operating systems (WCE, Symbian OS, and Palm OS) are very similar, they support encryption capabilities and some authentication schemes. Such authentication schemes usually comprise a password and exceptionally biometric measures to control the physical access. WCE and Palm OS support also smart card readers that could form part of the access control. For protecting the resources, Pocket PC (developed from WCE) supports Role-based access control (RBAC) [18], which does not resemble organisation's roles, rather it deals with code-access security [2]. In addition, despite the fact that the Pocket PC supports RBAC, applications do not use any mechanism for controlling access.

In [2], Perselson and Botha identify the features supported by mobile device operating systems for security. They also present the techniques being used for access control. As conclusion, they identify some lacks of security on mobile devices.

Jansen et. al [17] propose to assign and enforce an organisation's security policy on handheld devices. They assume each device to hold a valid policy certificate, obtained through synchronisation with a trusted server or user's desktop. A policy certificate is a structured set of information that conveys the policy assigned to a user, indicating the issuer and the owner, the period of validity, and additional information needed for establishing its authenticity and apply the policy. This information is signed by the issuer. The device also has an enforcement module that ensures fulfilment of the policy. The certificates are bound to a device using the unique device identifiers. Finally, the authors explore the weakness of their implementation, and note that rogue applications can attack the system, exploiting Palm OS's vulnerabilities.

On the other hand, there are also researches on the pervasive computing and ad-hoc network area that delegate the security management to trusted third parties (TTPs), as mentioned above. Thus, the mobile devices perform only minimum client-side functions. So, such approaches assume one of this hypothesis:

- the trusted third party as a server or central device controlling the access. The devices share a secret with a server, then they can use symmetric cryptography to secure the communication; therefore, the devices have to be known. Otherwise, the server uses asymmetric cryptography, issuing credentials for authentication and authorisation. In both cases, this approach is little realistic for ad-hoc networks which cannot depend on a central server. Usually, these networks assume that the users interacting carry their portable devices to establish ad-hoc communities; therefore, they have not a connection to a fixed network infrastructure, a priori knowledge of each other, or a pre-configuration established.
- The TTP as the joint of several devices, which cooperate with each other. They propose threshold cryptographic for sharing the CA functions. This solution increases complexity, relies on unsecured routing protocols for retrieving lists of nodes participating in the CA functions, and do not allow for new trust formation among previously unknown entities.

This research work proposes the adoption of distributed solutions where devices can act autonomously in accordance with their capabilities. For that, we provide mobile devices with a trust manager in compliance with the trust management model in next section.

III. Pervasive Trust Management (PTM) Model

PTM allows mobile devices to manage their own security, similar to Pretty Good Privacy (PGP) [19]. The devices act on behalf of physical bodies such as persons, organisations, departments, etc. Each one has its own key pair, public and private key. 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. These CAs 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.

Nevertheless, our trust management model does not assume the existence of previous trust relationships, since a device might create its own trust relationships in ad-hoc mode. So, each device handles a protected list of trustworthy and untrustworthy users through the degree of associated trust, behavioural information, and available public key certificates. We take into account untrustworthy users because distrust is different from not having any trust. Summarising, the mobile devices do not depend on any previously deployed infrastructure, existing trust relationships, an administrator, or a remote server to manage their own security. They might participate in ad-hoc networks, in disconnected mode and where the users interacting could have either no knowledge or some knowledge about another peer.

Trust relationships are expressed as a continuous function ranging from 0 to 1, these values being the extreme cases of complete distrust and complete trust respectively; in addition, we include intermediate states between the extremes, for instance, 0.5 would be used as ignorance value. We rely on fuzzy logic because it enables us more granularity than boolean logic that is used by PKI. Besides, such relationships fulfil certain properties, such as: reflexive, non-symmetrical, conditionally transitive, and dynamic [4]. These properties indicates that every CA trusts oneself, a CA `x` could trust another CA `y` in spite of the fact that `y` does not have a trust relationship established with `x`, a trusted CA could recommend to another CA allowing maximal two levels (more levels would be not useful), and the trust changes over time according to the user's behaviour.

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'll trust Bob without intervention of third parties, for that, Alice takes into account some available previous knowledge about Bob, otherwise Alice'll use an inference engine to interpret the established rules. Such rules are based on the user's security context, the security level assigned to the environment, and the risky sensitive resources/information.
- On the other hand, the indirect trust relationships are given by recommendations from TTPs. Such recommendations are distributed using a pervasive recommendation protocol (PRP) among close entities or using public key certificates [4]. PRP works in a way similar to an ad-hoc service discovery protocol, whose messages are protected in order to guarantee the communication only between trusted users. 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. This function has been compared with other models for combining beliefs such as Dempster-Shafer and Subjective Logic [4], [20].

Once the trust relationship is established, Alice gets a Bob's degree of trust, which would be our *belief* similar to Jøsang's model [21]. The belief is described as a set of fuzzy propositions that express the ownership degree of a user to the set of trustworthy users. However, this degree of trust is not static, our 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).

The value of such actions (V_a) is calculated in accordance with an associated weight to each kind of action (W_a), which is rewarded or penalised according to the past behaviour, both positive (N_{pact}) and negative (N_{nact}). Thus, when Bob requires a new action to be performed by Alice's device, Alice's device recalculates the degree of trust if Bob ($T(B)_{new}$) with the current degree of trust ($T(B)_{now}$), increasing or decreasing it in accordance with the current behaviour multiplied by a strictness factor (β) which takes into account the security level of the device (m). m values are higher for high security devices. Such calculations are automatically performed as shown the Fig. 1.

$$V_{a_i} = W_{a_i}^{(m)} \frac{e^{\left(\frac{-2N_{nact}-1}{N_{pact}+N_{nact}+1}\right)} N_{pact}}{(N_{pact}+Nnact)}$$

$$T(B)_{new} = \begin{cases} V_{a_i}.\beta + T(B)_{now}.(1-\beta) & V_a > 0 \\ 0 & \text{else} \end{cases}$$

$$\beta = \frac{1}{m+1}$$

Fig. 1. Degree of Trust Evolution

These degrees of trust are used for creating categories that would have assigned permissions, similar to role-based access control. So, we establish our own access control mechanism, called TrustAC.

IV. TrustAC: Trust-Based Access Control

Trust-Based Access Control (TrustAC) protects the device's resources using the trust for granting privileges. We use categories of trustworthiness instead of roles to assign permissions. The categories of trustworthiness are easily set by using numeric values; in addition, the use of numeric values facilitates the mapping between different domains. As well as RBAC, it simplifies the access control management, because it uses groups rather than individuals. Unlike to RBAC, it is not necessary to have an administrator for mapping roles on inter-domain relationships, creation of roles, assigning of users to those roles, or to maintenance the information updated.

TrustAC overcomes the limitations presented by the RBAC in open, dynamic environments, since the RBAC was designed as an access control mechanism for traditional centralized environments, that is, it works well in closed environments, with a definition of roles clearly established, a configuration previously done, and an administrator for management; but, it was not designed for open, dynamic environments and peer-to-peer interactions, since we cannot assume:

- the existence of predefined roles and the relations between them,
- a valid interpretation of roles across all domains,
- that the user sets the association "user-role" when it is required,
- that the user updates the relationships when the conditions change,
- that the user updates the policies when the roles change,
- and, that the user always remains in the same environment and under the same conditions.

TrustAC has been designed for environments without a previously defined organisational hierarchy or assignment of specific roles for each user. It automates the process of assigning users to degree of trust, minimising user intervention. TrustAC takes into account the environment conditions (context) in the access control policies. It is dynamic since the trust evolves, so a user could gets more privileges with her/his good behaviour without explicit updating. Finally, it guarantees the interoperability among several domains, by using fuzzy numeric values.

TrustAC defines a collection of elements and their relations: *users(U), agents(Au), trust degree(TD), resources(R), actions(A),* and *permissions(P)*. A *user* is defined as a human, a organisation, a department, etc. An *agent* is a user process (program) executing on behalf of users within a machine, it can be an autonomous agent. A *trust degree* represents the trustworthiness of the user, it corresponds to an interval of trust values. A *resource* is an entity that contains or receives information; for instance, information containers such as files or directories in an operating system, and/or columns, rows, tables within a database management system; or they can be system resources such as printers, disk space, ports, memory and CPU cycles. An *action* is an executable image of a program, which upon invocation executes some function for the agent; for example, in a file system, actions might include read, write, and execute; in a database management system, actions might include append, update, insert, delete; in a printer, actions might include print, manage printer, manage document; these actions could be classified as positive or negative depending on the user's permissions. A *permission* is an approval to perform an action on one or more protected resources. More precisely, users establish permissions using thresholds of trustworthiness for each action. Thresholds are determined according to the sensitiviness of the protected resource. Nevertheless, traditional system resources would have a by default configuration.

The relation between users and trust degrees (User Assignment, UA) is automatically generated from PTM (section III). The assignment of permissions to trust degrees (Permission Assignment, PA) is done through access control policies. An agent can execute an action only if the action is authorised for the trust degree of the agent (owner's trust degree).

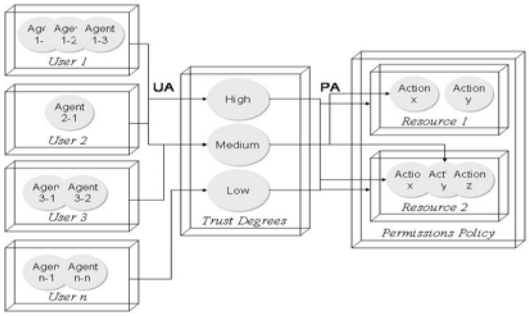

Fig. 2. Trust Based Access Control Reference Model

Such elements and relations are created and maintained using administrative, supporting system, and review functions. Administrative functions comprise creation and maintenance of element sets and their relations (UA, TA). The review functions are used to query and to view the contents of the relations from both the user and degree of trust perspectives. Supporting system functions make the access control decisions through two main modules [2]: the Policy Enforcement Point (PEP), the entry point in the access control system, and the Policy Decision Point (PDP), which finally allows or denies the access. Next section shows some of these functions as well as the trust manager modules.

V. PTM and TrustAC Components

PTM and TrustAC meet the security requirements for mobile devices to act as secure peers, besides being secure clients. PTM is implemented according to a modular architecture, so that the components can be plugged-in and reconfigured[3] according to the requirements. The main components defined in PTM are *Trust Manager (TM)*, *Access Control Service (ACS)*, and *Repository*.

The UML diagram of PTM and TrustAC components is shown in the Fig. 3. Some of these components already form part of the security components in mobile devices, such as the *Repository* and the *Authentication Manager*, as being interaction points with the architecture. The blank components would be added, because the rest belongs to the architecture.

Trust Manager (TM) is responsible for establishing the trust relationship, obtaining an initial degree of trust for new users, recalculating the new degree of trust when a user performs any action, inserting new trusted certificates in the certificate store, managing the recommendations, and interpreting the trust rules. For that, it relies on the following components:

- *Recommendation Manager (RM)* implements the pervasive recommendation protocol (see [4]. 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 Monitoring Service (AMS)* is responsible for the accounting of user actions at the device level. It also calculates the value of actions according to figure 1 and is used by the TM for recalculating trust values.
- *Context Provider Service (CPS)* is responsible for getting available information about the context such as location, time, people, etc. It acts as an interface (CPSI) for the context management tools that are generally embedded in the environment.

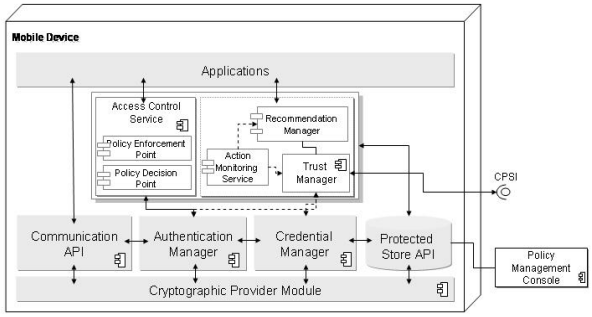

Fig. 3. UML Diagram of PTM and TrustAC Components

Access Control Service (ACS) is responsible for granting or denying access to the resources. The decisions are made based on the information supplied by *Authentication Manager*, *Trust Manager*, and *Context Provider Service*. This module comprises, in turn, the PEP and the PDP; these modules being local and distributed allow a more scalable and reliable system. The policies are specified and managed using *Policy Management Console (PMC)* according to XACML Standard. Currently, we have developed a prototype using Sun implementation [**?**]. We have adapted this implementation for limited devices and extended it to support degrees of trust and security levels as attributes.

The *Repository* is a logical storage space for keys, certificates, information about user's behaviour, access control policies, trust rules, etc. This module interacts with almost all the components. As shown in the Fig. 3, the repository is physically made up of several storage places.

VI. Conclusions

Security in mobile devices is an open topic within the research community. Security services supported by mobile devices need to be enhanced, specially to protect the access to device resources and the management of secure communications. PKI is the only working model for today's security, even IPSec is using X509 certificates, but it has well known problems: scalability, complexity, cost, etc. We propose a new model of trust PTM that helps to overcome some of the complexity and scalability problems of PKI: it needs little administration, it is easily configured, and it is open to establishing new trust relationships. PTM supports for trust evolution is an effort to closely model human's perception of trust. PTM is a decentralised, dynamic and automatic management of trust relationships. PTM contributes to the secure communication management between mobile devices acting as peers. As a mechanism for protecting the device's resources, we have proposed TrustAC, to manage the access control. TrustAC is also decentralized and does not require any infrastructure, and takes into account relevant aspects for mobile users such as the context, and allows the interoperability between different domains.

[2]These modules represent the architecture proposed by the IETF Policy-based Management Framework [22] and the XACML Standard [23], too. Likewise, these modules corresponds to the *AEF, Access control Enforcement Function* and the *ADF, Access control Decision Funcion* defined by the ISO Framework [24].

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

A TrustAC prototype has been implemented by using the XACML standard, besides a graphic tool for helping to generate policy documents. This prototype has been developed in Personal Java and successfully tested on a Pocket PC 2003. Now, we are implementing the PTM prototype to test the functionalities and advantages of our model, but our goal is to integrate our solution with the operating system, which requires more complexity because the source code must be extended. In [26], we have proposed a specific solution to enhance the WCE's security architecture.

ACKNOWLEDGMENTS

This work is being developed within the Pervasive Computing Laboratory Group (PerLab) of the University Carlos III de Madrid. Thanks to UBISEC (IST-STREP 506926) and EVERYWARE (MCyT N°2003-08995-C02-01) projects.

REFERENCES

[1] M. Weiser, "Ubiquitous Computing," March 1997, http://nano.xerox.com/hypertext/. [Online]. Available: http://nano.xerox.com/hypertext/weiser/UbiHome.html

[2] S. Perelsonl and R. Botha, "An investigation into access control for mobile devices," in *Proceedings of the 4th annual ISSA Information Security Conference*, June 2004.

[3] R. Housley, W. Ford, W. Polk, and D. Solo, "Internet X.509 public key infrastructure: Part i: Certificate and CRL profile," IETF, Tech. Rep. RFC 2459, January 1999. [Online]. Available: http://www.ietf.org/rfc/rfc2459.txt

[4] 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 in conjuntion with Mobiquitous 2004*, 2004.

[5] M. Ash and M. Dasgupta, "Security features in windows CE .NET," January 2003, http://msdn.microsoft.com.

[6] "Embedded operating system development," 2002, http://msdn.microsoft.com.

[7] D. Mery, "Symbian os version 7.0 functional description," March 2003, http://www.symbian.com/technology/symbos-v7x-det.html.

[8] Symbian, "Symbian os version 8.0 product sheet," February 2004, http://www.symbian.com/technology/symbos-v8x.html.

[9] PalmSource, "Security and palm OS: A flexible, robust security platform," 2002, http://www.palmsource.com/includes/security.pdf.

[10] G. Wilson, J. Ostrem, R. Levenberg, and R. Lagos, "Security and cryptography, exploring palm OS," June 2004, http://www.palmos.com/dev/support/docs/.

[11] "Pgp mobile," August 2004, http://www.pgp.com/products/mobile/.

[12] D. Friedlander, "Managing and securing mobile devices," Forrester Analysis Report, August 2004, http://www.csoonline.com/analyst/report2794.html.

[13] T. Karygiannis and W. Jansen, "Mobile security: Mobile devices project. unified security framework," Computer Security Resource Center (CSRC) - National Institute of Standards and Technology (NIST), June 2002, http://csrc.nist.gov/mobiledevices/projects.html.

[14] "Integrated antivirus and anti-spam for data-centric wireless mobile devices," January 2005, http://trendmicro.com/en/products/mobile/tmms.

[15] "Mobile and wireless security solutions," 2004, http://bluefiresecurity.com/mobile_firewall_plus.php.

[16] E. Maiwald, A. Robb, J. Bern, and J. Bagby, "Handheld devices audit checklist," August 2003, http://www.sans.org/score/checklists/.

[17] W. Jansen, T. Karygiannis, S. Gravila, and V. Korolev, "Assigning and enforcing security policies on handheld devices," in *Proceedings of the Canadian Information Technology Security Symposium*, May 2002.

[18] D. Ferraiolo, J. Cugini, and D. Jun, "Role based access control (RBAC)." [Online]. Available: http://csrc.nist.gov/rbac/

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

[20] A. Josang, M. Daniel, and P. Vannoorenberghe, "Strategies for combining conflicting dogmatic beliefs," in *Proceedings of the 6th International Conference on Information Fusion*, July 2003.

[21] A. Josang, "An algebra for assessing trust in certification chains," in *Proceedings of the Network and Distributed Systems Security (NDSS'99)*, 99.

[22] R. Yavatkar, D. Pendarakis, and R. Guerin, "A Framework for Policy-based Admission Control," Internet Engineering Task Force (IETF), Internet Request for Comment RFC 2753, January 2000. [Online]. Available: http://www.ietf.org/rfc/rfc2753.txt

[23] OASIS, "extensible access control markup language (XACML)," 2003.

[24] M. Lorch, D. Adams, D. Kafura, M. Koneni, A. Rathi, and S. Shah, "The PRIMA system for privilege management, authorization and enforcement in grid environments," in *4th International Workshop on Grid Computing - Grid 2003*, November 2003.

[25] SUN Microsytem, "(XACML) implementation API," 2003.

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