MADNESS: A Framework Proposal for Securing Work in Ad Hoc Networks

Pierre-François Bonnefoi, Patrick Poulingeas and Damien Sauveron

bonnefoi@unilim.fr, poulingeas@msi.unilim.fr, damien.sauveron@unilim.fr

LMSI, Laboratoire Méthodes et Structures Informatique
EA2632 – Université de Limoges
83 rue d'Isle, 87000 LIMOGES, FRANCE

ABSTRACT

Nowadays, the adoption of ad hoc networks grows very quickly but their approval as working context requires security improvement. Indeed wireless connectivity and mobility are the sources of the main security issues. The existing solutions to cope with these problems often use centralized models (PKI, etc.) and are not easy to deployed. This position paper describes the MADNESS framework (*Mobile AD hoc Network with Embedded Secure Systems)* to secure ad hoc networks thanks to secure processing environment such as smart card.

Keyword: Security, Ad Hoc Network, Smart Card

1. INTRODUCTION

In this position paper, we intend to discuss the possibility to deploy on demand a secure network by taking advantage of mobility and wireless connectivity. This deployment is possible by using mobile code and a trusted and secure processing environment like smart card. In the first section, we will introduce the problematic we want to tackle and our proposals for organizing such a network. In the second section, we will present a trusted framework for ad hoc networks. First, we will justify the need of smart card as solution of some specific security issues related to computation of mobile code; second we will describe shortly our framework requirements; third we will compare them with existing frameworks and fourth, we will present our hardware and software framework. Finally, we will show an example of usage and conclude with further extensions to improve our solution and cope with scalability.

2. PROBLEMATICS

Nowadays, mobility is the new major trend among work habits.
At the beginning, only the data were mobile having been allowed by the ubiquity of personal computer either at work or at home. Worker can complete work at home for the company that employs him by bringing its document at home on a floppy disk and now, on a USB key. The only threat on company security is to expose document to viruses but a pratical solution can be used like ensuring that employer use anti-virus software.
Not a long time ago, not only the data but also the process was mobile, causing severe headaches to IT manager in order to support mobility into the normal workflow of a company: the desktop PC is replaced by a laptop, and this laptop is used everywhere. At work, the solution to ensure security is to enable these mobile computers to connect only in a quarantine network in order to avoid threatening the security policy. No potential malicious software could enter the company network, but the employee could not access all parts of the network and there is no way to bypass the quarantine network.
More mobility has been achieved by using wireless connectivity.
Considering wireless connectivity leads mainly to two points of view:

- the first is to reduce the scope of mobility, being an enjoyable way to make a network which can share Internet connectivity and reaches all rooms at home without disgraceful cables. The person uses its home network mainly to browse the Web, and, sometime to connect to its company network through tunneling;
- the second is to use it as part of the network company. In this case, security is a major problem, and several schemes have been proposed based on strong cryptographic means. Hierarchical authorization, based on certificates, allows employees to gain access to wireless network and through this gate, to full company network.

From different points of view, wireless connectivity and mobility is as much desirable by user as less tolerate by IT manager.
Some new kind of "state of the art" computers blend these two aspects and bring them to the edge of technical excellence for their low weight and good usability (PDA, smartphone, ultra-portable). The tradeoff is to use components that maximize power autonomy: as less powerful processor, small amount of memory, etc.
Our purpose is to use this new kind of computers, so that, it should permit easy and fast deployment of a secure network at working place.
This network should be independent and should not imply support from current company network. The best fitting form of network is an ad hoc wireless network. No infrastructures are provided, nor shared with the company network. This network is purpose driven: being part of the network is being part of the current work. It can bring

together employees from different companies, so it not relies directly on identities delivered by the company network. This network must support real work, so it must also provide users a high level of security and the capability to exchange document for editing and revising for example. This security is hierarchical:

- protect users from the network from being used by other who are not in the network, more precisely for which the users have not allowed this usage;
- protect different groups of users in the network from each other for revising or examining documents.

For example (cf. Figure 1), the Foo company works into the Bar company in order to audit it, Foo employees use resources provided by Bar employees and create a report with other Foo employees. This report should not be accessed by Bar employees but could be exchanged (routed) by Foo employees.

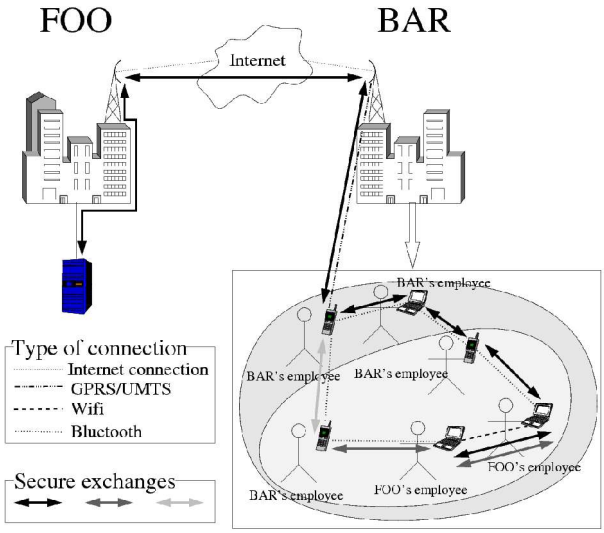

Figure 1: Audit of the Bar company by the Foo company.

The network should allow sharing resources between all users (like computational capabilities, Internet connectivities, amounts of memory) for one user task or for a collaborative task.

The network could not work if each of its node does not cooperate with each other, so, we need to ensure that a node will support the network according to its capabilities.

Our assumption is to use specific secured processor and secure storage that could be embedded in each node like smart card and to use secure communication between these cards.

With smart cards, being part of the network consists in:

- obtaining mobile code from other users, or from the first one in the network;
- executing this mobile code securely into the smart card and accepting control from it for network support: routing messages, sharing resources, enforcing security policy.

These scheme is similar to mobile agent system.

One of the first work of mobile agent is to establish an identity and different authorizations according to several groups attachment (such as in the example: employee from Foo and Bar companies). After some communication between the different nodes, this goal is achieved and the mobile agent switches to:

- network support;
- security gate to access document.

3. A TRUSTED FRAMEWORK FOR AD HOC NETWORKS

3.1 Comparing regular wired network with our proposal

In regular network, a frontier between authorized and not authorized access is enforced by using PKI (Public Key Infrastructure) and certificates: each node of the nework receive a certificate signed by an CA (Certification Authority). This CA must be always accessible for ensuring certificate's validity. More precisely each node trusts the network facilities (Domain Name Server, gateways, links, etc.) and uses the PKI.

In our proposal we want true ad hoc network, so we can not use some of the previous facilities and not trust network anymore. It is not possible to use a dedicated node in order to provide CA and PKI.

The purpose of a certificate is to give an identity to its owner, and this identity is strongly related to the identity of the network, which is related to the identity of the building and the human community that inhabits it (company, laboratory, etc.). In an ad hoc network, network identity can not be related anymore to a specific place, nor could be assigned for a long term such as the company lifetime or more precisely lifetime of the security policy (when all certificates and digital signatures expire).

In our proposal the first step in order to define a secured ad hoc network is to give an identity for the network itself. The validity of this identity must be precisely described, shared and accepted by people who want to gain access to this network for a carefully chosen period. The network is defined by special agreement between all of its nodes (like an organizational policy).

The second step is to provide identity to each node within the identity of the network.

These two steps are obtained by processing some code into the secured process platform provided by the smart card used on each node.

In the next section, we detail why we need a secured processor for this purpose.

This code is located in the smart card; it can be shared between each node (exchanged between nodes or directly provided with the smart card) and its behavior can not be disturbed during its process. In the case of code exchange between cards, we can use some kind of bootstrap code in order to set up secure communication channel securing this code from user's scrutiny.

3.2 Smart Cards for Protecting Agents and Hosts

To work in an ad hoc network, a trusted framework is necessary to ensure the co-operation of the mobile codes (so-called agents) and the security of the execution for the global platform. Indeed, in a multi-agent environment where the hosts are possibly not trusted, there are several

threats against security of the agents and of the hosts.
For the owners of the hosts, the threats are related to the security of the system and the applications already installed. When an agent arrives on a host, the host's owner wishes that it can not jeopardize the security (confidentiality and integrity). The classical solutions proposed to solve these problems are the sandbox approach, the bytecode verification and the code signing.
For the owners of the agents, the threats are related to the confidentiality and integrity of the mobile code and its data. The owners of the agents expect the execution of the mobile code to be correct and trustful. To guarantee the integrity of the data, the Ajanta system [KT01] provides three mechanisms that allow an agent's owner to detect that the agent has been tampered with: read-only state, append-only logs, and selective revealing of states to certain servers. The confidentiality of the data is often achieved by using cryptographic mechanisms. On the other hand, it seems very difficult to protect the agents against the threats on the code [FGS96] but we believe that a tamper-proof resistant hardware (such the smart card) with a secure operating system (such Java Card) should allow preventing them.
Indeed, there exists related work [CGSV03, CGKSV05] led by the *Distributed Objects and Systems* team of the LaBRI to secure the distributed computing thanks to a Java Card Grid. We wish to extend this work performed in a static environment to a dynamic network formed by the mobile ad hoc network.
Even if more formal approaches to protect mobile codes executed on untrusted runtime environments have been developed [LBR02, LM00], they seem difficult to be used in practice because of the strong assumptions they make on the applications.

Obviously the communication channels should be secure everywhere in the framework, as well for the mobile codes migration as for the exchange of data.

3.3 Conceptual Overview of our Trusted Framework

The following table describes our framework MADNESS (*Mobile AD hoc Network with Embedded Secure Systems)*.
The "physical layer" encompasses:

- the processing platform secured (the smart card);
- the processing platform not secured (the PDA or ultra-portable computer);
- and the radio communication technology (Bluetooth)

Upon this physical layer, the network layer requires strongly secured implementation, especially for routing purpose. In this case the secured processing platform is used in order to secure routing protocol (timestamp, identity of nodes and current network).
Next, the middleware relies for its security requirement on cryptographic capabilities provided by smart card and shared identities between nodes of the network.
Application layer could be similar in security requirement to same layer in other wired network (use of certificates, digital signature, etc.) or could use middleware services like secured agent in order to develop micro services: to classify distributed data between network nodes (defining ontologies), to perform searches with hidden pattern within node's knowledge in order to retrieve or compare securely data between a node and another, etc.
Security management occurs at all the levels of the framework. Even if the middleware layer is trusted, there may be some security gaps in the application (for instance, some malicious requests may be given as input, in a similar way as SQL injection attacks).

Application	Collaborative tool for creating a document	
Middleware	Agent (owner's ID, security level, etc.)	Ontology, ACL
	Secure platform for mobile agents: • Basic security mechanisms for agents and hosts (Java or C# libraries for authentication, etc.) • Resources allocation	
Network Layer	Routing	
Physical Layer	Bluetooth Wifi in ad hoc mode	Smart card

Table 1: Some major components and functionalities of our future framework.

3.4 Adequacies of Existing Frameworks

Initially, our purpose was not to implement new framework but to gain a sufficient level of security for mobile code and its process.
They are two interlaced problems: security of mobile code and enforcement of sharing fairly the available resources (an agent must obtain needed resources and must not be too greedy).
Security should be the matter of the framework or should be added afterwards in a none secured framework. Adding security to an existant none secured framework is similar to ensure that processing of mobile code could be protected from examination and alteration (security of mobile code), and that the mobile code could not interfere with none authorized part of the host (host security).
The first answer to the host security problem is the "sandbox" approach provided by vitual machines. The interest of using VM is that security is a design requirement: security policy to access resources is part of the architecture.
Nevertheless, from the state of the available frameworks supporting mobility, we notice that none of these frameworks reaches this level of security to ensure the security of the mobile code: some of these frameworks rely on regular Virtual Machines (Java VM or .Net Common Language runtime), some others on modified VM and finally, on specifically designed frameworks.

3.4.1 Regular VM and the "Write once run anywhere" capability

Using a VM raises the problem that an applet/agent can not audit the security level of the VM that processes it. More complexity is added when user of the VM want to qualify the security level of the downloaded applet/agent. It is a "chicken and egg" problem: the VM could be signed with proper certificate shared by the mobile code. Yet, to verify signature of the VM, applet/agent must be processed by the VM itself! Malicious user can easily

modify VM behavior in order to cheat the applet/agent.
This category contains: GecGo [FGLS04], Xmiddle MCZE02] and Proem [KSPGFS01].

3.4.2 Modified or newly implemented VM

Some frameworks address the problem of accessing resources from the host computer by providing a way to define agreement between this host and the mobile code.
The mobile code could obtain some waranties about its resources requirement, but the problem of security remains similar as regular VM: how induce trust between host and mobile code if processing can not be secured?
This category contains JAMUS [LG02] for instance.

3.4.3 Specifically designed frameworks

Some frameworks are designed from scratch by using dedicated libraries and standard languages. Security concerns are addressed directly by the implementation (use of cryptographic tools, definitions of particular protocol, etc.).
Despite, the good properties and the solutions provided, these frameworks are affected with the same problem: how to protect the processing of mobile code?
This category contains SWAT [SAAKR03] and Ad Hoc Infoware [ADGGH04].

3.4.4 Summary

Table 2 presents a short comparison of cited frameworks.

Framework	*Mobile code*	*Security*
Ad Hoc Infoware	no	poor (use of external PKI)
GecGo	possible	no
JAMUS	yes	some kind
Proem	yes	no
SWAT	yes	yes
XMiddle	no	no

Table 2: Features of several frameworks

The uniqueness of our framework compared with known existing solutions is the use of a secure processor (smart card) to really secure the framework from the hardware level to the software level: the mobile code and the host are protected.

3.4 Real Hardware and Software Platform

Our framework is currently in deployment on 6 PDAs (*MyPal A620BT* from ASUS) with built-in Bluetooth communication capabilities and Compact Flash support. On each PDA in the CF slot, we use a smart cards reader (*SpringCard-CF* from Pro-Active) to exchange with the trusted applications installed on smart cards (note that this reader has 2 smart cards slots: 1 ISO slot and one internal SIM/SAM connector).
The software framework is developed in C# with Visual Studio and based on the .NET Compact Framework. Since our PDAs have a Widcomm Bluetooth stack, we must use their software development kit in order to use the BT capabilities. The communications with the smart cards reader and thus the smart cards inserted are realized through the SpringCard .NET API.
On card side, we use the Java Card technology because of its multi-application and dynamic application loading features. Java Card is most widespread but in the future perhaps we will use Smartcard.NET. Our cards are *JCOP 31 bio* from IBM and *GemXpresso Pro* from Gemplus.

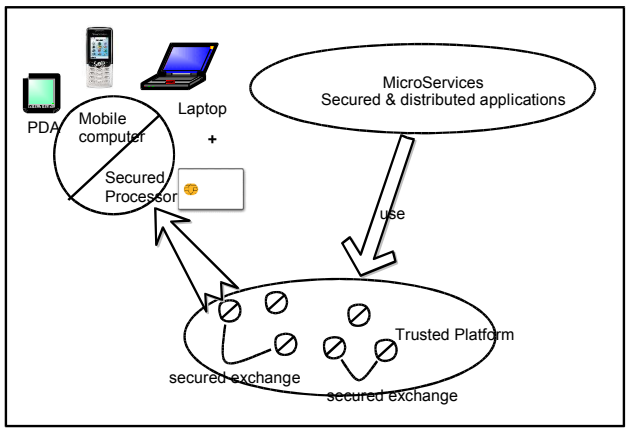

Figure 2: Description of the architecture.

4. APPLICATION

Our proposed framework can support several scenarii that require a high level of security and collaborative work. An example is a secure auction system like [FLS02]. Each person is represented by a mobile code which is a bidder agent. Each agent has a strategy build by its owner. The security mechanisms of our framework avoid a fake bid done by an unauthorized and malicious person. After a certain period, all the involved nodes send the winning agent (its code) of their ad hoc network to an authority which finds the winner. The advantage of this system compared to classical electronic auctions on Internet is the bidder's restricted geographic area. The seller and the buyer can easily meet if the configuration of framework is reduced by examples to a town or a small organization. As stressed in [BKR98], the use of agents can overcome problems due to a poorly connected network environment such as an ad hoc network.

5. CONCLUSION AND FUTURE WORK

A future extension of our secure framework for ad hoc networks is the introduction of welfare economic notions and results in order to improve the reliability and availability of groupware tools. The aim of this extension is to obtain an optimal repartition of the resources (memory, computing power, Internet access, etc.) in maximizing the global welfare of each node in a utilitarian view (or with a Pareto optimality). Each node has a utility function who maps sets of resources to real numbers ([SAN98], [EM04]). Agents representing nodes negotiates then the best repartition of the resources of the ad hoc network be-

tween all the nodes. This component of the middleware layer can be used by any application who wished a good distribution of all the resources of the ad hoc network, preserving the agents' rationality in their negotiation.
The difficulties of this approach are:

- the high-cost of computation and communications before reaching an optimum;
- the volatility of ad hoc components which have to be managed by the algorithm to avoid a full new computation;
- the integration in the algorithm of the routing cost in ad hoc networks.

Finally, we hope to deploy a fully secure framework to make collaborative works in ad hoc networks. To achieve our development, the main promising idea is the use of smart cards to control the workflows.

REFERENCES

[ADGGH04] J. Andersson, O. Drugan, V. Goebel, C. Griwodz, P. Halvorsen, E. Munthe-Kaas, T. Plagemann, M. Puzar, N. Sanderson, K. S. Skjelsvik
Middleware Services for Information Sharing in Mobile ad-hoc Networks - Challenges and Approach..
WCC2004, IFIP WS8.
Toulouse, France. August 27, 2004.

[BKR98] J. Bredin, D. Kotz, D. Rus.
Market-based Resource Control for Mobile Agents. Proceedings of "Autonomous Agents", May 1998.

[CGKSV05] S. Chaumette, P. Grange, A. Karray, D. Sauveron, P. Vignéras.
Secure distributed computing on a Java Card Grid.
Proceedings of 7th International Workshop on Java for Parallel and Distributed Computing.
Denver, Colorado, USA, April 4-8, 2005.

[CGSV03] S. Chaumette, P. Grange, D. Sauveron, P. Vignéras.
Computing with Java Cards.
Proceedings of CCCT'03 and 9th ISAS'03. Orlando, FL, USA, July 31, August 1-2, 2003.

[EM04] U. Endriss, N. Maudet.
On the Communication Complexity of Multilateral Trading.
Proceedings of the 3rd International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS-2004), pages 622-629.
ACM Press, July 2004.

[FGLS04] H. Frey, D. Görgen, J. K. Lehnert and P. Sturm.
Supporting Smart Applications in Multihop Ad-Hoc Networks - The GecGo Middleware
Proceedings of the 6th International Conference on Enterprise Information Systems ICEIS'04.
Porto, Portugal, 2004.

[FGS96] W. M. Farmer, J. D. Guttman, V. Swarup.
Security for mobile agents: Issues and requirements.
Proceedings of 19th National Information Systems Security Conference, National Institute of Standards and Technology.
Baltimore, MD, USA, October 1996.

[FLS02] H. Frey, J.K. Lehnert, P. Sturm.
UbiBay: an Auction System for Mobile Ad-Hoc Networks.
CSCW 2002.
New Orleans, Louisiana, USA. November 16-20, 2002.

[LBR02] S. Loureiro, L. Bussard, Y. Roudier.
Extending Tamper-Proof Hardware Security to Untrusted Execution Environments.
Proceedings of the Fifth Smart Card Research and Advanced Application Conference (CARDIS'02), USENIX.
San Jose, California, November 2002.

[LG02] N. Le Sommer, F. Guidec.
JAMUS: Java Accommodation of Mobile Untrusted Software.
Proceedings of the 4th Nord EurOpen/Usenix Conference (NordU 2002).
Helsinki, Finland, February 2002.

[LM00] S. Loureiro and R. Molva.
Mobile Code Protection with Smarcards. Proceedings of the 6th ECOOP Workshop on Mobile Object Systems.
Cannes, France, June 2000.

[KSPGFS01] G. Kortuem, J. Schneider, D. Preuitt, T. G. Cowan Thompson, S. Fickas and Z. Segall.
When Peer-to-Peer comes Face-to-Face: Collaborative Peer-to-Peer Computing in Mobile Ad-hoc Networks.
International Conference on Peer-to-Peer Computing, Linköping, Sweden, August 27-29, 2001.

[KT01] N. Karnik and A. Tripathi.
Security in the Ajanta Mobile Agent System.
Software - Practice and Experience, 2001.

[MCZE02] C. Mascolo, L. Capra, S. Zachariadis and W. Emmerich.
XMIDDLE: A Data-Sharing Middleware for Mobile Computing.
In Personal and Wireless Communications Journal 21(1). Kluwer. April 2002.

[SAAKR03] E. Sultanik, D. Artz, G. Anderson, M. Kam, W. Regli, M. Peysakhov, J. Sevy, N. Belov, N. Morizio and A. Mroczkowski.
Secure Mobile Agents on Ad Hoc Wireless Networks.
Proceedings of the 5th Innovative Applications of Artificial Intelligence Conference.
Acapulco, Mexico, August 2003.

[SAN98] T. W. Sandholm.
Contract types for satisficing task allocation:I Theoretical results.
Proceedings of the AAAI Spring Symposium: Satisficing Models, 1998.