A Survey of Self-protected Mobile Agents

Salima Hacini
LIRE, computer department,
Constantine2 University
Constantine, Algeria

ABSTRACT

Mobile agents are software which moves autonomously through a computer network with aim to perform some computation or gather information on behalf of its creator or an application. In the last several years, mobile agents have proved their numerous applications including e-commerce, telecommunication systems, information management, on-line auctions or service brokering. In most applications, the security of mobile agents is a burning issue. Indeed, the agent is vulnerable while it is executing on the host's execution platform. Its owner therefore requires some guarantees concerning the protection of the agent against malicious host threats. Thus, the mobile agent has to protect itself from any act aiming at the deterioration, the destruction or the handling of its code, its state or its data. So, mobile agents must be prepared to execute on different hosts with various environmental security conditions. There are plenty of techniques to protect mobile code. This paper presents a survey of existing techniques for achieving a self-protected mobile agent.

General Terms

Mobile Agents Security, Mobile Agents Self-protection.

Keywords

Malicious Host, Mobile agent protection, Security, Trust.

INTRODUCTION

Human activities are increasingly based on the use of distant resources and services, and on the interaction between remotely located parties that may know little about each other. Mobile agents are the most suited technology. These mobile agents are active and autonomous software entities that can suspend their behavior, move to another host of the network, and continue their activity, deciding where to go and what to do along the way [1]. They provide several advantages to design and control distributed applications such as autonomy, dynamic adaptation, data and control distribution, a better use of the network resources and reduction of communication with respect to latency, bandwidth and connection time. The benefits from uses mobile agents are great. However, mobile agents bring a serious security risk when they execute on malevolent hosts. Malicious host may try to attack mobile agent in order to obtain service without providing payment or to remove private information from the agent's memory. Other examples of such attacks are malicious alteration of its code and control of its execution. Thus, an adversary might be able to compromise an agent using one or more of the following approaches:

- Observing the agent's code, data and flow control.
- Manipulating the agent's code, data and flow control.
- Executing the agent's code inappropriately, including replay.
- Returning false values to the agent's system call functions.
- Denying execution of the agent's code, either in part or whole.
- Masquerading as a different host.
- Eavesdropping on agent communications.
- Manipulating agent communications.

A similar and more elaborate classification of security threats for shopping agents (and corresponding security requirements and controls) is given by Schaefer [2]. Further discussions can be found in [3].

This paper deals with the self-protection of the mobile agent behavior from any analysis aiming to divulge it. It also underlines the evolution of the use of the concept of trust as factor enhancing security in mobile agents based systems.

The paper is organized as follows: Section 1 exposes some related work on the most important techniques for providing a mobile agents self-protection and underlines the importance of trust execution environment for a safe execution of mobile agents. Section 2 describes the trust estimation concept. Finally, Section 3 concludes this study.

1. MOBILE AGENT' SELF-PROTECTION

The technology of mobile agents has introduced some serious problems and has emphasized existing security issues. Mobile agents become especially vulnerable when traveling among hosts network. Several works have underlined the importance of protecting host against malicious mobile agents [4], [5], [6], [7]. This problem is out of the scope of this paper. The protection of the mobile agent against malicious hosts remains an open issue and this section presents the most important classical techniques for providing a self-protection of mobile agents.

It highlights the importance of trust execution environments as the single counter-measure that handles all mobile agent threats, gives an overview of this concept and underlines its significance in the protection of mobile agents computing.

1.1 Mobile agent security

The mobile agents are exposed to various threats from hosts they visit. This problem is difficult because the visited environment has a full control on the mobile agent execution. Several approaches have been proposed. They principally aim at making the attacks useless or detectable. The most common approaches that allow a mobile agent to protect itself without any interaction with a trusted third party are:

1.1.1 Cryptographic Approaches

An example of such approaches is hiding function. In hiding function approach, Sander and Tschudin [8], [9] describe an approach to code protection that relies on the execution of encrypted functions. Their approach is based on executing a program embodying an encrypted function on a mobile agent platform. The mobile agent code is a kind of encrypted program which can be executed on encrypted data without decryption of code and data at all. This approach ensures that the platform does not learn anything substantial about the encrypted function f. Thus, it offers the confidentiality of the execution. However, the approach remains theoretical and coding is applied only on a restricted whole of functions (polynomial and rational functions).

It is noted that even if the function f is encrypted, a malicious host might still be able to mount a black-box attack. A host could repeatedly execute P(E(f)), and in certain circumstances might then be able to use the obtained pairs of inputs and (encrypted) outputs to reconstruct (reverse engineer) the function f. However, in recent years a number of possible schemes have been proposed which come close to realizing a general purpose function of the desired type [10], [11]. A more recent work of Gentry [12] is of particular interest, although a truly practical scheme of universal applicability is not yet available.

One solution to this problem proposed somewhat similar technique designed to obfuscate the code so that it becomes difficult to analyze its functionality in real time. These techniques are known as Obfuscation Techniques.

1.1.2 Obfuscation Techniques

Approaches using Obfuscation techniques are explored to protect the code of a mobile agent from reverse engineering for some minimum time. Obfuscation transforms a program into another program that has equivalent behavior but which is harder to understand. Fritz Hohl [13] converts the agent of origin into a black box by using obfuscation algorithms. An expiration date is attached to the black box. This approach prevents any attempt with sophisticated code analysis and any replay. It allows complex functions but guarantees protection only for a certain time interval. However, the scheme has a number of practical limitations arising from the difficulty of successfully obfuscating the code and the problem of choosing an appropriate block box time interval.

Barak et al. [14] studied the theoretical limits of obfuscation techniques and showed that in general achieving completely secure obfuscation is impossible. Larry D'Anna in [15] states that obfuscation can prevent a malicious host from observing or predictably tampering with code and data in the running system, however, it cannot prevent the program from being reverse engineered. Obfuscation techniques are recommended to be used in order to obtain various versions of the same task and thus to provide the mobile agent several equivalent behaviors that generate the same result. A more recent study of this technique can be found in [16].

1.1.3 Environmental Key

Riordan and Schneier [17] use data of current environment to construct a decryption key. When the proper environment information is located, the key is generated, the cipher-text is decrypted, and the resulting plain-text is acted upon. Neither can the mobile agent precisely predict its own execution at the receiver host, nor can the host foresee the incoming agent task. The approach allows the agent owner to specify some constraints on the environment where the mobile agent will execute. Two variants of the scheme are proposed, namely the forward-time approach, which permits key generation only after a specific point in time, and the backward-time approach, which permits key generation only before some point in time. A recent study of this technique can be found in [18].

In spite of that this scheme brings also a limitation which is the access to sensitive mobile code information and will still need to be protected against the entities that might be involved in the key generation process, this solution brought a particular glimmer of hope on a security of a mobile agent execution and some researchers used it to solve some security problems. For example, Filiol in [19] uses the environmental key as a basic technique to forbid a Malware code analysis and Hacini [20] employs it to avoid analysis of mobile agent services. So, the environmental key generation approach can be useful when the receiver is not aware of the conditions required to the execution of the requested service. This case corresponds to the problem of the mobile agent behavior analysis. Hence, this solution is recommended to estimate host's trustworthiness.

1.1.4 The k-out-of-n Threshold Scheme Approach

They are also approaches based on the subdivision of the transmitted secrecy is in several small secrecies. It is the case in the k-out-of-n threshold scheme approach [21] the secrecy of the transaction is distributed between n duplicated mobile agents. The latter are emitted towards different hosts. The confidentiality is dealt with since no mobile agent knows the totality of information. This approach does not take into account the integrity but it is an example of the "secure distributed computing" concept: parts can jointly calculate the result of a particular function without revealing the inputs. This idea appears also in the code-on-demand or the co-operating agent techniques. The Code-On-Demand approach [22] protects the mobile agent integrity by using dynamically upgradeable agent code, in which new agent function modules can be added and redundant ones can be deleted at runtime. This approach enhances code privacy, reduces transport cost and helps the recoverability of agents after malicious attacks; while the co-operating agent technique [23], [24], [25] distributes critical tasks of a single mobile agent between co-operating agents. This technique reduces the possibility of the shared data being pilfered by a single host. This idea deserves to be employed by approaches where the mobile agent service (secret) is subdivided in several tasks and when mobile agent carries out a preliminary investigation of trustworthiness of the visited host which shows the selected execution [20].

1.1.5 Trusted execution environments

Trusted Hardware Approaches are used by researches to guarantee a certain behavior of a system. Herzberg and Pinter [26] describe a device that can be used to protect software against piracy. A more recent approach by Yee and Tygar [27] ensures that the system functions securely. Mobile agents can be protected by executing them on trusted tamper-proof hardware [28] or secure co-processors [27]. The use of software-based tamperproof environments has also been proposed in [29]. More recently, it has been suggested that agents can be securely executed in hosts using trusted computing technologies [30]. These approaches are powerful but they have a too expensive cost. This reason leads to explore the solution used to evaluate the host's trustworthiness.

1.2 Synthesis

Previous study (Subsection 1.1) attests that the approaches based on trusted hardware are powerful. However, they are not scalable and are prohibitively expensive mainly due to the use of the security device.

Table 1. Counter-measures against integrity threats

Threat class	**Threat sub-classes**	**Desirable Counter-measures**
Integrity risk	Integrity interference	- **trusted execution environment** - encrypted function - sliding encryption - reference states - state appraisal
	Integrity modification	- **trusted execution environment** - trusted hardware - mutual itinerary recording - reference states - encrypted function - sliding encryption - code-on-demand - environmental key - path histories - execution tracing

Table 2. Counter-measures against availability threats

Threat class	**Threat sub-classes**	**Desirable Counter-measures**
Availability risk	Denial of service	- **trusted execution environment** - tamper resistant hardware - time sensitive agents - execution tracing - reference states
	Delay of service	- **trusted execution environment** - mutual itinerary recording - reference states secret distribution
	Transmission refusal	- **trusted execution environment** - mutual itinerary recording - reference states secret distribution

Table 3. Counter-measures against confidentiality threats

Threat class	**Threat sub-classes**	**Desirable Counter-measures**
confidentiality risk	Spying	- **trusted execution environment** - secret distribution - sliding encryption - cryptographic function - environmental key
	eavesdropping	- **trusted execution environment** - trusted hardware devices - mutual itinerary recording - sliding encryption - secret distribution
	Reverse engineering	- **trusted execution environment** - sliding encryption - secret distribution - time limited obfuscation - cryptographic function

Table 4. Counter-measures against authentication threats

Threat class	**Threat sub-classes**	**Desirable Counter-measures**
Authentication risk	Masquerade	- **trusted execution environment** - digital signature - execution tracing - reference states - mutual itinerary recording
	cloning	- **trusted execution environment** - hiding function - sliding encryption - time sensitive agents

In addition, the software approaches, less robust, reduce the cost of security and ease maintenance. In addition, various security threats can come from malicious hosts. These threats include integrity attacks, denial of service or confidentiality and authentication risks.

According to the results obtained (see Table 1, Table 2, Table 3, Table 4) based on threats classification given by Bierman in [5], it appears that trusted execution environment is the only measure that can cover all threats. Indeed, creating an environment of trust in which a mobile agent roams freely without being threatened by a malicious host can get rid of most classes of discussed threats. Hence, the mobile agent protection issue focuses on the trust estimation of the visited environment.

In addition, the technique of environmental key [17] seems very interesting. This technique can indeed be used when the host does not know the required conditions for the execution of the service. Environmental key can be used to not only protect the service carried by the mobile agent, but also to assess the credibility of the host visited because it is built on the basis of environmental information. In addition, it is not possible to decide a priori, during the progress of the mobile agent, if a host is reliable or not. Therefore, the security of mobile agent requires dynamic assessment of the credibility of the visited hosts. The idea is to use the concept of dynamic adaptability in order to determine the appropriate behavior of the mobile agent according to the estimated degree of trust. This concept provides the mobile agent the ability to change its behavior making it unpredictable and therefore protects against threats analysis.

2. TRUST ESTIMATION

Trust plays an important role in e-commerce and e-business security. It is a key to the acceptance and general deployment of this type of applications.

The concept of trust has been a subject of large interest in different research areas like economics, game theory and multi-agent systems [31], [32]. Obtaining and maintaining trust estimation is a serious open problem. Emphasis in the literature is mostly on techniques for preventing malicious agents from harming their execution environment. Many general trust models have been proposed to introduce the trust notion in the context of general distributed system applications [33], [34], [35]; however only a small number of these models have addressed the issues of integrating trust with security in mobile agents based systems [36].

Beth [37] proposed one of the earliest trust models for authentication in distributed systems focusing on relationship modeling while Abdul-Rahman et al. [33] provided a general model based on recommendations. But these models did not address the trust dynamics based on behavior.

Wilhelm et al. [38], [28] give one of the more comprehensive discussions on the issue of trust in mobile systems. They identify what they referred as the four foundations of trust, namely: blind trust, trust based on (a good) reputation, trust based on control and punishment and trust based on policy enforcement. Their solution to the trust in mobile agent systems problem is the CryPO protocol, based on tamper-proof hardware to provide tamper-proof environments, which are the foundation for the agent executor. Agents assert which environment manufacturers they trust. The protocol uses certificates and encryption technology to ensure security and is essentially an extension of the certification framework. This solution provides an easier way for a new service provider to establish itself in the market; it also allows an agent owner to protect specific data in a mobile agent. However, the used trust notion is static and is mapped to a particular manufacturer of the hardware; hence this approach does not allow dynamic trust update, and then limits the flexibility in application.

Manchala [39], [40] develops a model based on trust-related variables such as the cost of the transaction and its history, and defines risk-trust decision matrices. The latter are used together with fuzzy logic inference rules to determine whether or not to transact with a particular party.

Tan et al. [41] propose a trust model specifically for mobile agent security using Trusted Third Parties (TTPs) in the form of verification servers, but they do not address how trust can be integrated with security systems.

Braynov et al. [31] give a solution that does not rely on collecting and analyzing information about untrustworthy agents. Instead, they propose an incentive-compatible mechanism in which agents truthfully reveal their trustworthiness at the beginning of every interaction. In this mechanism, agents report their true level of trustworthiness, even if they are untrustworthy.

Cahill et al. propose the SECURE project [42] which investigates the design of security mechanisms for pervasive computing based on the human notion of trust. The central contribution of SECURE is to provide entities with a basis for reasoning about trust and risk embodied in a computational framework that can be adapted to a variety of application scenarios. But, it is not clear how they develop the dynamical adaptation of trust.

Dimitrakos [32] introduces metrics, costs and utility functions as parameters of an algorithm that produces the trust policy for a given trusting decision. Nevertheless, this work lacks a quantitative definition of the various involved measures.

Josang [23], [43] proposes a scheme for propagating trust through computer networks based on public key certificates and trust relationships, and demonstrates how the resulting measures of trust can be used for making decisions about electronic transactions. He also defines a model of trust composed of a reliability trust as the probability of transaction success and a decision trust derived from the decision surface. Trust adaptability with time has not been considered in the model.

Chin Lin et al. [35] suggest a hybrid trust model employing soft trust mechanisms with constructs such as recommendation, direct experiences via interactions and observations. These mechanisms are used to complement hard trust (based on cryptographic mechanisms) for enhancing the mobile agent security in situations where full authentication trust is not available due to absence or unavailability of trusted third parties.

Castelfranchi et al. [44] claim the importance of a cognitive view of Trust. They argue in favor of a cognitive view of trust as a complex structure of beliefs and goals, implying that the trustor must have a “theory of the mind” of the trustee. Such a structure of beliefs determines a degree of trust and an estimation of risk, and then a decision to rely or not on the other. The decision is also based on a personal threshold of risk acceptance/avoidance. They explain rational and irrational components and uses of trust. Their work represents a support of TAMAP approach [20] where trust value is based on trust ingredients (evaluation of the situation and the behavior).

These trust models generate a subjective single value which does not reflect the exact cause of the lack of trust. Thus, the provided decision could be inadequate. Some of them use also the reputation of the host as factor intervening in the trust estimation. It is not necessary for the mobile agent to know the reputation of the visited hosts that can be new in the network. Indeed, the trust estimation is dynamic and it is used to enable the mobile agent to adapt its execution in entrusted environments. Hacini in [20] exploits opportunities offered by the dynamic adaptability mechanism to protect the mobile agent against the visited host. The dynamic adaptability mechanism is used to offer the mobile agent the possibility to modify its behavior. This ability makes it unpredictable and complicates its analysis. The idea is that the mobile agent must verify the host trustworthiness and present to it, according to the trust it inspires, an appropriate behavior.

So, the mobile agent has to be endowed with a perceptual mechanism, enabling it to gather enough information about its virtual surroundings and comprehend current situations [20].

3. CONCLUSION

A malicious host can deduce mobile agent execution strategy by analyzing its behavior. To prevent this kind of attacks, recommendations lie on three levels: the use of artifices for mobile agent protection (like environmental key and code-on-demand), the host trust estimation and the dynamic adaptation of mobile agent process.

The dynamic adaptation process allows the adaptation of the mobile agent behavior based on the host trustworthiness. It is based on mobile agent ability to propose various alternatives to the same service and provides some interests since it offers the mobile agent the ability to react with an unexpected behavior. This aptitude prevents the visited host to deduce the followed strategy. Consequently, any behavior analysis attempt becomes difficult and inefficient. So, trust constitutes a method to build host behavior-aware agent and the agent itself could take the initiative to react after the host trust estimation.

4. REFERENCES

[1] Rouvrais S. : Utilisation d’Agents Mobiles pour la Construction de Services Distribués. Thèse de doctorat de l’université de Rennel, France (2002)

[2] I. Schaefer. Secure Mobile Multiagent Systems In Virtual Marketplaces: A Case Study on Comparison Shopping. Technical Report RR-02-02, Deutsches Forschungszentrum für Künstliche Intelligenz, DFKI GmbH, March 2002.

[3] N. Borselius. Multi-agent system security for mobile communication.PhD thesis, Royal Holloway, University of London, 2003.

[4] Bellavista P., Corradi A., Frederici C., Montanari R., Tibaldi D.: Security for Mobile Agents: Issues and Challenges. Invited Chapter in the Book Handbook of Mobile Computing, I. Mahgoub, M. Ilyas (eds.), CRC Press (2004)

[5] Bierman E., Cloete E.: Classification of Malicious Host Threats in Mobile Agent Computing. Proceedings of SACICSIT2002, (2002) 141-148

[6] Borselius N.: Mobile Agent Security. Electronics & Communication Engineering Journal, vol. 14, No 5, IEEE. London (2002) 211-218

[7] Karnik N.: Security in Mobile Agents Systems. PhD thesis, Department of Computer Sciences and Engineering, University of Minnesota, Minneapolis, USA (1998)

[8] Sander T., Tschudin C.: Toward Mobile Cryptography. IEEE Symposium Security and Privacy, IEEE Computer Soc. Press, Los Alamitos, California (1998) 215-224

[9] Sander T., Tschudin C.: Protecting Mobile Agent against Malicious Hosts. G.Vigna (Ed.), Mobile Agents and Security, Lecture Notes in Computer Science, Vol.1419, ©Springer-Verlag Berlin Heidelberg, Berlin (1998) 44-60

[10] S. G. Choi, A. Elbaz, A. Juels, T. Malkin, and M. Yung. Two-party computing with encrypted data. In Proceedings of the Advances in Cryptology 13th international conference on Theory and application of cryptology and information security (ASIACRYPT ’07), pages 298–314, Berlin, Heidelberg, 2007. Springer-Verlag.

[11] R. Ostrovsky and W. Skeith. Algebraic lower bounds for computing on encrypted data. Cryptology ePrint Archive, Report 2007/064, 2007.

[12] C. Gentry. Computing arbitrary functions of encrypted data. Communications of the ACM, 53(3):97–105, March 2010.

[13] Hohl F.: Time Limited Blackbox Security: Protecting Mobile Agents from Malicious Hosts. G.Vigna (Ed.), Mobile Agents and Security. Lecture Notes in Computer Science, Vol. 1419, Springer-Verlag, (1998) 52-59

[14] Barak B., Goldreich O., Impagliazzo R., Rudich S.,. Sahai A, Vadhan S., Yang K. : On the (Im)possibility of Obfuscating Programs. Advances in Cryptology, Proceedings of Crypto'2001, Lecture Notes in Computer Science, Vol. 2139. (2001) 1-18

[15] D'Anna L., Matt B., Reisse A., Van Vleck T., Schwab S., LeBlanc P. : Self-Protecting Mobile Agents Obfuscation Report. Network Associates Laboratories Report, (2003)

[16] S. W. Shah, P. Nixon, R. I. Ferguson, S. R. Hassnain, M. N. Arbab, and L. Khan. Securing Java-Based Mobile Agents through Byte Code Obfuscation Techniques. In Proceedings of the IEEE Multitopic Conference (INMIC ’06), pages 305–308. IEEE, December 2006.

[17] Riordan J., Schneier B.: Environment key Generation towards Clueless Agents. Lecture Notes in Computer Science, Vol. 1419, (1998) 15-24

[18] L. Weiwei, H. Zhen, and W. Qinglong. An Approach to the Sensitive Information Protection for Mobile Code. In Proceedings of the The First International Symposium on Data, Privacy, and E-Commerce, pages 289–297, Washington DC, USA, 2007. IEEE Computer Society.

[19] Filiol E.: Strong Cryptography Armoured Computer Viruses Forbidding Code Analysis: the Bradley virus. Proceedings of the 14th EICAR Conference, 2005

[20] S. Hacini, Z. Guessoum and Z. Boufaïda, “TAMAP: A New Trust-based Approach for Mobile Agent Protection”, Journal in Computer Virology. Springer Paris. ISSN 1772-9890 (print) 1772-9904 (online), vol.3, n°4/Nov. 2007, pages 267-283. DOI 10.1007/s11416-007-0056-y.

[21] Beimel, A., Burmester, M.: Computing Functions of a Shared Secret. SIAM J. Discrete Math., Vol. 13, No.3, (2000) 324-345

[22] Wang T., Guan S., Khoon Chan T.: Integrity Protection for Code-On-Demand Mobile Agents in E-Commerce. The Journal of Systems and Software 60, (2000) 211-221

[23] Jansen W., Karygiannis T. : Mobile Agent Security. NIST Special Publication 800-19, National Institute of Standard and Technology, (2000)

[24] Roth V. : Secure Recording of Itineraries Through Cooperating Agents. Proceedings of the ECOOP Workshop on Distributed Object Security and 4th Workshop on Mobile Object Systems: Secure Internet Mobile Computations, INRIA, France (1998) 147-154

[25] Roth V. : Mutual Protection of Cooperating Agents. Secure Internet Programming: Security Issues for Mobile and Distributed Objects. J. Vitek and C. Jensen (Eds.), Springer Verlag, (1999)

[26] Herzberg A., Pinter S.S.: Public Protection of Software. Advances in Cryptology: Crypto 85, Springer-Verlag, Berlin (1985) 158-179

[27] Yee B., Tygar D.: Secure Coprocessors in Electronic Commerce Applications. The Proceeding of First Usenix Workshop on Electronic Commerce, Usenix Assoc., Berkeley, California (1995) 155-170

[28] Wilhelm U. G., Staamann S., Buttyan L.: On the Problem of Trust in Mobile Agent Systems. IEEE Symposium on Network and Distributed System Security, San Diego, California (1998)

[29] C.Wang, J. Hill, J. Knight, and J. Davidson. Software tamper resistance: Obstructing static analysis of programs. Technical Report CS-2000-12, Department of Computer Science, University of Virginia, Charlottesville, VA, USA, 2000.

[30] X. He-qun and F. Deng-guo, " Protecting mobile agents' data using trusted computing technology", Journal of Communication and Computer, 4(3):44–57, 2007.

[31] Braynov S., Sandhol T.: Trust Revelation in Multiagent Interaction. Proceedings of CHI'02, Workshop on the Philosophy and Design of Socially Adept Technologies, Minneapolis, (2002)

[32] Dimitrakos T.: A Service-Oriented Trust Management Framework. Falcone R., Barber S., Korba L., and M. Singh, editors, Trust, Reputation, and Security: Theories and Practice, LNAI 2631. Springer, (2003) 53-72

[33] Abdul-Rahman A., Hailes S.: Using Recommendations for Managing Trust in Distributed Systems. In Proceedings of IEEE Malaysia International Conference on Communication'97 (MICC'97), Kuala Lumpur, Malaysia (1997)

[34] Gambetta, D. : Can we Trust Trust? Trust: Making and Breaking Cooperative Relations, Gambetta, D (ed.), Basil Blackwell, Oxford (1990)

[35] Lin C., Varadharajan V.: Modelling and Evaluating Trust Relationships in Mobile Agent Based Systems. In Proceedings of First International Conference on Applied Cryptography and Network Security (ACNS03), Lecture Notes in Computer Science, Vol. 2846, Springer-Verlag, Kunming, China (2003) 176–190

[36] Grandison T.,. Sloman M: A Survey of Trust in Internet Applications. IEEE Communications Surveys and Tutorials, Fourth Quarter (2000)

[37] Yahalom R., Klein B., Beth T.: Trust Relationships in Secure Systems - A Distributed Authentication Perspective. The Proceedings of IEEE Conference on Research in Security and Privacy, (1993)

[38] Wilhelm U. G., Staamann S.M., Buttyán L.: A Pessimistic Approach to Trust in Mobile Agent Platforms. IEEE Internet Computing, Vol. 4, No. 5, ISSN: 1089-7801, (2000)40-48

[39] Manchala D.W.: Trust Metrics, Models and Protocols for Electronic Commerce Transactions. The 18th International Conference on Distributed Computing Systems, (1998)

[40] Manchala D.W.: E-Commerce Trust Metrics and Models. IEEE Internet Computer, (2000) 36-44

[41] Tan H. K., Moreau L.: Trust Relationships in a Mobile Agent System. In G. P. Picco, editor, Fifth IEEE International Conference on Mobile Agents, Lecture Notes in Computer Science, vol. 2240, Springer-Verlag, Atlanta, Georgia (2001).

[42] Cahill V. et al...: Using Trust for Secure Collaboration in Uncertain Environment. IEEE Pervasive Computing, 2(3), (2003) 52–61

[43] Jøsang A.: Trust-Based Decision Making for Electronic Transactions. The 4th Nordic Workshop on Secure ITSystems (NORDSEC'99), Stockholm University Report 99-005, Stockholm (1999)

[44] Castelfranchi C., Falcone R.: Trust is much more than Subjective Probability: Mental Components and Sources of Trust. The 32nd Hawaii International Conference on System Sciences - Mini-Track on Software Agents, Maui, Hawaii (2000)