Keynotes
COMPSAC 2009

Opening Remarks

Susan K. Land
2009 President, IEEE Computer Society

Biography

Kathy Land, an employee of MITRE Corp., has more than 23 years of industry experience in the practical application of software engineering methodologies, the management of information systems, and leadership of software development teams. She is an acknowledged expert in the field of software engineering standardization, process improvement, and engineering management.

Ms. Land is the 2009 President of the IEEE Computer Society (CS) and has served on the CS Board of Governors and in positions as 1st and 2nd vice president. She has also served as vice president for standards and conferences and tutorials. She is a current member of the CS Standards Activities Board (SAB), Software and Systems Engineering Standards Executive Committee (S2ESC). In 2007, she was the recipient of the IEEE Standards Association Standards Medallion.

Ms. Land is the author, or co-author, of a number of texts, papers, podcasts, webinars all supporting sound software engineering principles and practical application of software process methodologies. These include Jumpstart CMM/CMMI Software Process Improvement: Using IEEE Software Engineering Standards (John Wiley & Sons, 2005). She is coauthor of Practical Support for CMMI-SW Software Project Documentation: Using IEEE Software Engineering Standards (John Wiley & Sons, 2005), Practical Support for ISO 9001 Software Project Documentation: Using IEEE Software Engineering Standards (John Wiley & Sons, 2006), and Practical Support for Lean Six Sigma Software Process Documentation: Using IEEE Software Engineering Standards (John Wiley & Sons, 2008).

Program Committee
COMPSAC 2009

Program Chairs
Elisa Bertino, Purdue University, USA, Bertino@cs.purdue.edu
Vladimir Getov, University of Westminster, UK, V.S.Getov@westminster.ac.uk
Lin Liu, Tsinghua University, China, linliu@tsinghua.edu.cn

Best Practice Track Co-chairs
Darren Kerbyson, Los Alamos National Laboratory, USA
Jenny Li, Avaya Labs, USA

Education and Learning Track Co-chairs
Hussein Zedan, De Montfort University, UK
James Cross, Auburn University, USA

Embedded Systems Track Co-chairs
Tiberiu Seceleanu, ABB Corporate Research, Sweden
Bruce McMillin, Missouri University of Science and Technology, USA

Formal Methods Track Co-chairs
Michele Bugliesi, University of Venice, Italy
Cristina Seceleanu, Malardalen University, Sweden

Location Based Services Track Co-chairs
Sumi Helal, University of Florida, USA
Paolo Bellavista, University of degli Studi di Bologna, Italy
Axel Kupper, Ludwig Maximilians University Munich, Germany

Quality Track Co-chairs
Ron Kenett, KPA Ltd., Israel
Joao Cangussu, University of Texas at Dallas, USA

Requirements Track Co-chairs
Eric Dubois, CRP Henry Tudor, Luxembourg
Eric Yu, University of Toronto, Canada

Security Track Co-chairs

Virgil Gligor, University of Maryland at College Park, USA
Jan Camenisch, IBM Research Zurich, Switzerland

Social & Collaborative Networks Track Co-chairs

Murat Kantarcioglu, University of Texas, Dallas, USA
James Joshi, University of Pittsburgh, USA

Software Architecture Track Co-chairs

Jose Moreira, IBM Thomas J. Watson Research Center, USA
Uwe Zdun, Vienna University of Technology, Austria

Software Evolution Track Co-chairs

Atilla Elci, Eastern Mediterranean University, North Cyprus
Hongji Yang, Demontfort University, UK

Testing Track Co-chairs

Fevzi Belli, University of Paderborn, Germany
Hong Zhu, Oxford Brooks University, UK

Mobile & Pervasive Computing Track Co-chairs

Karl Leung, Vocational Training Council, Hong Kong
Zhen Liu, Nokia Research Center, China

Workshop Chair

Sheikh Iqbal Ahamed, Marquette University, USA

Panel Chair

Rajesh Subramanyan, Siemens Corporate Research, USA

Fast Abstract Chair

Katsunori Oyama, Nihon University, Japan

Doctoral Symposium Co-chairs

Jinchun Xia, San Jose State University, USA
Massimo Copolla, CNR, Italy

Technical Program Publicity Co-chairs

Omer F. Rana, Cardiff University, UK
Xianping Tao, Nanjing University, China
Damla Turgut, University of Central Florida, USA

Program Committee Members

Ademar Aguiar, University of Porto, Portugal
Gail-Joon Ahn, Arizona State University, USA
Sabah Al-Fedaghi, Kuwait University, Kuwait
Mikio Aoyama, Nanzan University, Japan
Paris Avgeriou, University of Groningen, Netherland
Doo-Hwan Bae, KAIST, Korea
Xiaoying Bai, Tsinghua University, China
Emanuel Baker, USA
Sergey N. Baranov, Motorola, Russia
Samik Basu, Iowa State University, USA
Paolo Bellavista, University of Bologna, Italy
Fausto Bernardini, IBM Watson Research, USA
Taisuke Boku, Unversity of Tsukuba, Japan
Jan Bosch, Intuit Inc., USA
Roberto Bruni, University of Pisa, Italy
Kai-Yuan Cai, Beijing University of Aeronautics and Astronautics, China
Dave Card, Q-Labs, France
Barbara Carminati, University of Insubria, Italy
Bertrand du Castel, Schlumberger, USA
Sungduk Cha, Korea University, Korea
Melissa Chase, Brown University, USA
W. K. Chan, Hong Kong City University, Hong Kong
T. Y. Chen, Swinburne University, Australia
Y. C. Chen, National Chiao-Tung University, Taiwan
Chi-Hung Chi, Tsinghua University, China
Byoungju Choi, Ewha Woman's University, Korea
William Chu, Tunghai University, Taiwan
Ivica Crnkovic, Malardalen University, Sweden
James Cross, Auburn University, USA
Schahram Dustdar, Technology University of Vienna, Austria
Elena Ferrari, University of Insubria, Italy
Neil Harrison, Utah Valley University, USA
Mohamed Jemni, University of Tunis, Tunis
Jun-jang Jeng, IBM TJ Watson, USA
Hsin-yi Jiang, Iowa State University, USA
T. Ming Jiang, National Chung Cheng University, Taiwan
Yuecel Karabulut, SAP, USA
Jong Kim, POSTECH, Korea
Markulf Kohlweiss, K. U. Leuven, Belgium

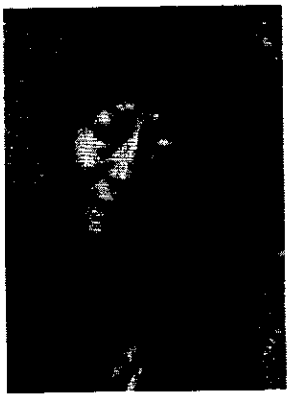
John Koo, Shantou University, China
Sanggoo Lee, Seoul National University, Korea
Joon-Sang Lee, LG Electronics, Korea
Moonkun Lee, Chonbuk National University, Korea
Woojin Lee, Kyungpook National University
Ville Leppanen, University of Turku and TUCS, Finland
Johan Lilius, Abo Akademi University, Finland
Jigang Liu, Kyoto University, Japan
Xiaodong Liu, Napier University, UK
Xiaoqing (Frank) Liu, Missouri University of Science and Technology, USA
John May, University of Bristol, UK
Nancy Mead, Software Engineering Institute, CMU, USA
Jose Miguel - Alonso, UPV/EHU University, Spain
Hua Ming, Iowa State University, USA
P. V. R. Murthy, Siemens Corporate Technology, India
Tien Nguyen, Iowa State University, USA
Mehmet A. Orgun, Macquarie University, Australia
Barbara Paech, University of Heidelberg, Germany
Sooyong Park, Sogang University, Korea
Daniel Paulish, Siemens Inc., USA
Frances Paulisch, Siemens Corporate Technology, Germany
Paul Pettersson, Malardalen University, Sweden
Klaus Pohl, University of Duisburg-Essen, Germany
Shaz Qadeer, Microsoft Research, USA
Colette Rolland, University of PARIS-1 Panth谷on Sorbonne, France
Subhash Saini, NASA Ames, USA
Juha Savolainen, Nokia, Finland
Martin Schulz, Lawrence Livermore National Laboratory, USA
Sahra Sedigh, Missouri University of Science and Technology, USA
Cristina Serban, AT&T, USA
Jawed Siddiqi, Sheffield Hallam University, UK
Christian Skalka, University of Vermont, USA
Pradip Srimani, Clemson University, USA
Michiharu Takemoto, NTT, Japan
Kenji Takahashi, NTT, Japan
Gene Tsudik, University of California, Irvine, USA
Hasan Ural, University of Ottawa, Canada
Feng-Jian Wang, National Chiao Tung University, Taiwan
Qianxiang Wang, Peking University, China
Thomas Weigert, University of Missouri-Rolla, USA
Gerhard Wellein, University of Erlangan, Germany
Will Winsborough, University of Texas at San Antonio, USA
Eric Wong, University Of Texas At Dallas, USA
Dianxiang Xu, North Dakota State University, USA
Jian Yang, Macquarie University, Australia
Weider Yu, San Jose State University, USA

Y. T. Yu, City University of Hong Kong, Hong Kong
Jia Zhang, Northern Illinois University, USA
Peter Zimmerer, Siemens Corporate Technology, Germany
Xukai Zou, IUPUI, USA
Mohammad Zulkernine, Queen's University, Canada

Welcome from the Program Chairs

COMPSAC 2009





Welcome to Seattle and COMPSAC 2009, the 33rd edition of the IEEE Annual International Computer Software and Applications Conference. This year's theme – Harmonizing Humans, Computers, and Software in Services Environments – reflects some of the new challenges that our global society faces at present. As a world-wide forum, COMPSAC can address those challenges by bringing together both researchers and practitioners from various countries and cultures and by offering them unique opportunities to share their ideas, perspectives, and experience with others. The technical program of COMPSAC 2009 continues to feature academic and industrial practice papers covering a relatively wide research area. Topics of interest include but are not limited to: social networks, security, requirements analysis, software architecture, software quality and testing, evolution, formal methods, embedded systems, mobile and pervasive computing. Other new emerging and multidisciplinary research and development work, industry-academia collaborations, and curriculum design, are all part of COMPSAC.

For COMPSAC 2009, we have received 231 submissions covering both the academic and the industrial sectors from different parts of the world. Each paper was evaluated by at least three reviewers for its technical content and suitability to the conference tracks and topics. After a rigorous peer review and selection process which took many hours of work, 46 regular papers and 29 short papers were accepted for presentation and included in the conference proceedings. The regular papers publish mature results. The short papers represent quality work that could spur discussion. Several fast abstracts that discuss promising but preliminary results were also accepted. In addition, the technical program includes 3 keynote addresses, six panel discussions, 14 workshops, and a doctoral symposium with papers presenting work-in-progress by PhD candidates.

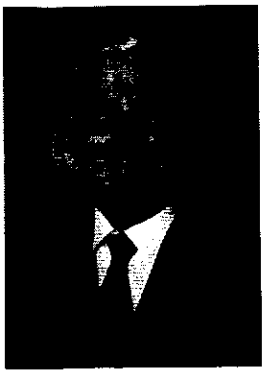
The success of COMPSAC 2009 would not have been possible without the effort and hard work of many volunteers. First, we would like to thank all Track Chairs and the members of the Program Committee for their important service to this community in soliciting quality submissions, reviewing papers, and thus providing invaluable help for the acceptance decisions. Their effort was vital in assuring the high quality of the technical program. We also thank Sheikh Ahamed and Rajesh Subramanyan for playing critical roles in organizing the workshops and the panels. We would like to give our special thanks to Carl Chang, the Steering Committee Chair, who has been providing invaluable guidance and support throughout the entire conference preparation process and to Tony Hey, who has offered important advice about a number of key issues as General Chair. Thanks are also due to all our colleagues – faculty, staff, and students – at Iowa State University for managing the flow of papers and reviews, for operating the corresponding software support systems, for developing the conference website, and for organizing the Program Committee meeting in March 2009. Particular recognitions go to Hua Ming and Laurel Tweed for their tireless work in keeping things running.

Finally and most importantly, our thanks are due to all the keynote speakers, the authors, and the panelists for their high quality research work, results, and papers that we are proud to publish in these proceedings.

We sincerely hope that you will enjoy IEEE COMPSAC 2009!

Elisa Bertino **Vladimir Getov** **Lin Liu**

Greetings from the General Chair
COMPSAC 2009



The rapid development of computing, software, and applications revolutionized the contemporary society and greatly influenced its economic and social progress in the last century. Nowadays, the exponential emergence of innovative computing products and systems is due to the diligent work of both researchers and practitioners and the great synergy in research and implementation across disciplines and geographical boundaries attributed substantially to the breakthrough in computer, telecommunication and Internet. Even so, nothing can replace face-to-face meetings where we meet with old friends, get acquainted with new colleagues, and exchange ideas and share experience under one roof. COMPSAC has been playing this role of a flagship international conference for 33 years thus contributing directly and significantly to the ongoing information revolution. I was therefore very happy to accept the responsibility to serve as the General Chair in 2009 for a meeting with such a long and successful tenure.

Scientific conferences provide the primary meeting ground for researchers and practitioners from both academia and industry in order to define the key path of innovation in the broader space of computer science and engineering. However, not all conferences are created equal and COMPSAC guarantees the superior meeting ground for the key leaders in computer science research and technology. As it has matured into an esteemed venue for publication and discussion of knowledge in the corresponding areas, in recent years COMPSAC has been recognised as the IEEE Computer Society's signature conference on computers, software and applications.

The theme for this year's instance of the conference is Harmonizing Humans, Computers, and Software in Services Environments. Especially considering the fact that the conference is organized in Seattle, the home of Microsoft, the theme is particularly appropriate. The field of computer software and applications has been changing drastically and dramatically over the last decade with the huge growth of the Internet, the proliferation of mobile devices and systems, the emergence of services computing, and the globalization of software engineering. In response to the theme, the conference provides a very exciting technical program, as detailed in the welcome message from the Program Chairs.

In conclusion, I would like to express sincerest thanks to my fellow conference organizers, the Steering Committee and the Program Committee. The Program Committee, led by the three Program Chairs – Elisa Bertino, Vladimir Getov, and Lin Liu, and the reviewers all deserve credits for developing such an excellent technical program that resulted from a thorough and detailed selection process based on all submissions. Special thanks are extended to Carl Chang for his enthusiastic leadership of the entire conference planning and development process and for his serving as the Standing Committee Chair. It is a joy and an honor to work with such committed, talented, and insightful colleagues. We are all indebted to the numerous volunteers for their contributions to making COMPSAC 2009 a big success.

Thank you for your high quality contributions to this unique international meeting. We sincerely hope that you will enjoy the presentations and discussions, use the opportunity to meet interesting researchers, and appreciate the wonderful settings that the Seattle area offers to this conference.

I look forward to your participation and meeting you at the IEEE COMPSAC 2009 in Seattle, July 21-24, 2009.

Tony Hey

Message from the Standing Committee Chair

COMPSAC 2009

This is the 33rd instance of this long-standing international event, historically recognized as a flagship conference of the IEEE Computer Society and most recently designated as its signature conference on computer, software and applications. Thousands of past participants and thousands of technical contributions over the past 32 years from around the globe mark the uniqueness and sustainability of this annual forum. Owing primarily to Professor Stephen Yau's sustained effort in the first 30 years of COMPSAC organization, it has a remarkable track record for bringing in fresh blood, bright and young minds, seasoned industrialists as well as top-notch researchers year after year. At this annual COMPSAC event, thinkers meet doers who like to think, while many doers meet thinkers who actually do.

In 2008, I spoke to the COMPSAC audience of transforming COMPSAC from a software-oriented conference to computing-oriented one with software as its focus. The plan is to expand COMPSAC beginning in 2009 to engage an even broader community where computer, software, and applications are indispensable parts of this community's profession and life. Certainly this ambitious undertaking will take a few years to materialize. The following is the new mission statement of COMPSAC:

COMPSAC is the major international forum for cross-cutting computing research and education. As the signature conference for the IEEE Computer Society, COMPSAC provides computing professionals a unique opportunity for hosting computation as a base science, necessarily applied to applications, employing all aspects of the computing discipline.

Another major undertaking with the COMPSAC organization is the redesign of conference website. By the time we meet in Seattle the mock-up edition will be ready for demonstration. It will be more easily managed by volunteers and staff as we choose to program it on top of a Content Management System. Several community-building concepts will be incorporated including the Wiki and perhaps an interface to Facebook. Please join us to build the COMPSAC community when this new website becomes live, and provide your feedback. COMPSAC is our conference so we will build it, evolve it, and grow it together.

We will again produce electronic proceedings only, in the USB format. In this 21st century we must think green and act green. As having been proven in 2008, the IEEE-CS Conference Publishing Services editorial staff enabled pagination in this USB format as part an integral unit of the IEEE-wide digital library archive. Again, for traditional readers like myself, print-on-demand hardcopy proceedings can be purchased from IEEE Computer Society at a reasonable cost, if necessary.

In 2009 COMPSAC will again co-locate with another important international conference – the International Symposium on Applications and the Internet (SAINT). SAINT represents the major joint program with an agreement signed between IEEE Computer Society and Information Processing Society of Japan (IPSJ). We intend to continue this fruitful collaboration in the coming years. As software is an integral part of Internet technology and applications, both communities are happy with the arrangements.

I would like to take this opportunity to recognize some of the volunteer leaders. General Chair Tony Hey helped to propel program development and even promised to contribute a keynote speech. Program Chairs Elisa Bertino, Vladimir Getov, and Lin Liu provided the leadership and superbly managed to put together an excellent program. We are all deeply indebted to Laurel Tweed's all encompassing support in every aspect of program planning and local arrangements. Hua Ming, Proceedings Chair, and Kai-shin Lu, Registration Chair, worked day and night under the guidance of Finance Chair Simanta Mitra, who is now a veteran of managing these nitty-gritty details and tedious routines. Sheikh Ahamed almost single-handedly led the development of workshops that has become a critical and integral part of COMPSAC. We are also indebted to all track chairs, workshop organizers, and many other volunteers. Finally, we are pleased to recognize the support of the US National Science Foundation and Microsoft Corporation for COMPSAC 09.

Seattle is always a very pleasant place to visit in July. We have also arranged a very best and unique conference banquet – on the Argosy Cruise boat to explore Washington Lake. See you soon.

Carl K. Chang

2009 33rd Annual IEEE International Computer Software and Applications Conference

Agent Protection based on the use of cryptographic hardware*

Antonio Muñoz Antonio Maña Rajesh Harjani Marioli Montenegro
University of Malaga
{amunoz,amg,rajesh,marioli}@lcc.uma.es

ABSTRACT

Mobile agents are processes that can migrate autonomously from new hosts. Despite of the huge number of fields of application of this technology, a lack in the security exists. The main approach of this work is based on the provision of a secure execution environment for mobile agents. Our approach is based on the idea of the trusted migration. This trusted migration is reached by means of the use of cryptographic hardware. Concretely, Trusted Computing Module (TPM). Thus, we have designed and developed a specific protocol, which is the basis to build the solution. In order to build our solution on a robust basis, we have validated this protocol by means of a model checking tool called AVISPA. Finally, we built a library to provide access to TPM (Trusted Platform Module) functionalities. The idea behind of this is based on the easy in using cryptographic hardware in the agent based systems development, disposing to agent developers of the security related tasks of their systems. The most relevant aspects of this library are described along this paper both at development stage of it and while we use it to develop a system based agent.

Index Terms: Agent Protection, Security, Trusted Computing

1 INTRODUCTION

Mobile agent concept is a software entity consisting of code and data that can migrate autonomously from host to host executing its code. A huge number of the current applications are based on this technology. Indeed, several critic applications exist in scenarios like powerplants, air traffic control, etc. However, a lack in the level of security exists. The main motivation of this work is based on the fact that despite its benefits, security issues strongly restrict the use of code mobility. The reason for this happen is that the protection of mobile agents against the attacks of malicious hosts is considered the most difficult security problem to solve in mobile agents systems. Indeed, in [18] is shown how scientific community has put many efforts in this field, many applications exist based on this technology. However, all this efforts are wasted due to the lack of a secure and robust basis to build applications based on agent technology.

In [18] two different solutions are presented to provide a secure framework for agent based applications development. The first of these consists on a standalone software solution, which makes use of the entitled approach 'protected computing' [10]. The second solution presented consists on a cryptographic hardware based solution. Concretely, it is a TPM built-up approach. This solution is further described in the following sections where we present a hardware-based mechanism to protect agent systems. This approach bases its security in the protection of the secure migration process. We focus our work in the design and development of a trusted protocol, which is the core to build the solution. Moreover, to achieve a robust basis, we have validated our protocol by means of a model checking tool called AVISPA, as well as by means of a library to access to TPM (Trusted Platform Module) functionalities from software agents.

*Work partially supported by E.U. through project Serenity (IST-027587) and OKKAM (IST- 215032) and DESEOS project funded by the Regional Government of Andalusia

1.1 State of the Art

A wide variety of techniques for implementing security in agent systems are available. Obviously, not all are compatible with one another, nor are they all suitable for most applications. Indeed, many of these techniques must be implemented within the framework of the agent system, while a number of them can be applied independently within the context of the application. While elementary security techniques should prove adequate for a number of agent-based applications, many applications are expected to require a more comprehensive set of mechanisms. We consider the hardest problem to solve regarding mobile agent security the problem entitled 'the malicious hosts'. There is no proper solution to avoid the attacks of a malicious host while it is running the agent. Thus, malicious hosts could try to get some profit of the agent modifying the data, the communication or even the results due to their complete control on the execution. Therefore the agent cannot host a decryption key because the hosts could read it. Furthermore, it is not sure that the hosts runs the complete code in a correct manner, or simply does not allow the migration to other hosts. Current approaches can be divided in two main categories, attack detection and attack avoidance approaches. Attack detection approaches permit the source host to know if its agent was tampered during the execution due to illegal modifications of code, data or execution flow. The aim of this kind of proposals is dissuading the malicious host because a detection can lead to a punishment. Thus, the harder the punishment, the less attachments will be performed. Nevertheless, detection techniques are not useful for services in which the benefits for tampering a mobile agent are greater than the possible punishment. In such cases, attacks avoidance techniques are more useful. Unfortunately, there is no current approach that avoids attacks completely.

Regarding the attack avoidance approaches, Yee introduced the idea of a closed tamperproof hardware subsystem in [20], where agents can be executed in a secure way. Also Ordille proposed executing the agent only in trusted hosts in [19]. Our solution is based on a closed tamper-proof hardware system where agents can be executed in a secure way, which forces to each host to buy hardware equipment and to consider the hardware provider as trusted. Other alternatives in this line are presented by Roth in [21] with the idea of cooperative agents that share secrets and decisions and have a disjoint itinerary. This fact makes collusion attacks difficult, but not impossible. Several techniques can be applied to an agent in order to verify self-integrity and avoid that the code or the data of the agent is inadvertently manipulated. Anti-tamper techniques, such as encryption, checksumming, anti-debugging, anti-emulation and some others [16] [17] share the same goal, but they are also oriented towards the prevention of the analysis of the function that the agent implements. Additionally, some protection schemes are based on self-modifying code, and code obfuscation [22].

In attack detection related approaches we found that some authors introduce the idea of replication and voting. In each stage, a set of hosts execute the agent in a parallel way and send several

0730-3157/09 $25.00 © 2009 IEEE
DOI 10.1109/COMPSAC.2009.152

IEEE computer society

replicas of the agent to the next stage. This can only be used as an attack detection approach in those scenarios in which the hosts in the same stage are independent, i.e. they must have different interests to attack an agent. In this line, Vigna introduces the idea of cryptographic traces, which are logs of the operations performed by the agent during its lifetime. The operations of the agent can be categorized in white statements, which alter the agent's state due to external variables. These traces contain the changes performed to internal variables as a consequence of black statements. In short, the traces contain all the external input data that altered the agent's state during its execution. Hence, with these traces a re-execution of the agent can be performed. Instead of sending the traces, the host sends a hash of them to avoid repudiation attacks. If the origin host suspects that a host modified the agent and wants to verify the execution, it asks for the traces and executes the agent again. If the new execution does not agree with the traces, the host can be cheated.

This approach not only focused on the detection attacks, but it also proves the malicious behaviours of the host. Nevertheless, this approach in incomplete, among the more relevant drawbacks found we highlight that the verification is only performed in case of suspicion, but the way in which a host becomes suspicious is not detailed; and that for an indefinite period of time, each host must reserve enough capacity to the storage of traces of past transactions because the origin host can request them. Finally some other approaches are based on the use of Vigna's traces but none of them solves the problem of attack detection in a satisfactory way.

2 ACCESSING TPM FUNCTIONALITIES FROM THE JADE FRAMEWORK FOR AGENT SYSTEMS DEVELOPMENT

We defend that it is doubtful that an agent could keep a secret since the information belonging to a mobile agent is completely available to its host system. To the point that we believe that is impossible to prevent agent tampering unless trusted hardware is available in agent platforms. However, we defend the use of cryptographic hardware for highly secure agent systems, but other alternatives are useful whether the requirements in security are more flexible.

Thus, we propose a hardware based mechanism to provide security to agent-based systems. This device provides some mechanisms, such as cryptographic algorithms, secure key storage and remote attestation that are essential to achieve a high level of security, as well as an easy usage of it for agent system developer. The main advantage for this approach is that agent system developers are disposed of security engineering related tasks, due to the underlying security of our approach.

Our main objective is to achieve a high level of security, for this purpose we propose a hardware based mechanism, previously we pointed that the selected element is the TPM. More important reasons for this election are the standardisation and the wide use of this device, which integration in new computers is getting regular, as well as the supporting provided with many important companies specialists in security. To sum up, this element is the cornerstone of our approach and the security of our system is focused on it. We identified two main pillars of agent protection. Firstly we have two protect previously the execution element. The protection of this element is provided with using TPM and the root of trust provided with this. It is based on controlling that only a restricted set of operations can be executed. Secondly the migration procedure must be protected, for this purpose we use remote attestation functionality provided with TPM. In order to facilitate the use of this mechanism we developed a complete library which provides the TPM functionality for JADE based agents.

3 THE ESSENTIAL ROLE OF THE TPM IN THE STARTING STAGES OF THE EXECUTION PROCESS

The use of TPM in these systems is as follows. Each agency takes measures of some system parameters which determine its security, for instance BIOS, keys modules from Operating System, active processes and services in the system. Through these parameters an estimation of the secure state of the agency can be done. Values taken are stored in a secure way inside the trusted device, in such a way that can be neither access nor modified unauthorized. Agency has the ability to report configuration values previously stored to other agencies in such a way that these can determine its security. As aforementioned the agents requests to source agencies, previously to the migration, to determine that destination agencies are secure. By means of this process an agent in a secure agency can extend the limit of its confidence to other agency once the security of destination agency is tested.

The main objective of this paper is to describe a library to provide security to a multiagent system based on JADE, which stands for Java Agents Development Framework. For this purpose we built the security mechanism previously described using a TPM. Development process required some considerations, such as a JADE system is composed for a platform that contains a main container in which agents are deployed. Additional containers can be added to this platform, some of them can be remote containers, and different platform can interacts among them, allowing the migration of the agents among them. The containers play agencies roles in JADE on which agents will be deployed. Taking into account JADE structure, we conclude that two different kinds of migration exists, migration among containers from different platforms and migration from containers from the same platform. In the case that the migration is from containers from different platforms, the agent migrates from a container from source agency to the destination agency main container. In such a case that destination agency is not a JADE built-on platform architecture can be different, depending on the platform. In the other case, the agent migrates from a container to another one but in the same platform.

Both migration processes imply some security concerns. Platform migration is not secure because the main container from source platform can be untrusted. Also containers migration has the same problem, it is, if destination container is not trusted then the migration is not secure. Secure migration library solves arisen problems.

4 INTRODUCING THE SECMILIA: A SECURE MIGRATION LIBRARY FOR AGENTS

In [23] William et al describe what they called "the natural history of an agent", where three crucial types of events exist in the life history of an agent. The Program Creation. The author prepares source code and state appraisal function max for the program. Agent Creation, for prepare a program for sending. Agent Migration. When an agent is ready to migrate from one interpreter to the next. In this section we introduce the Secure Migration Library for agents SecMiLiA based on the protection of the agent migration process. This library makes use of the remote attestation capability provided by TPM. Thus, the migrations are supported by a trusted platform. This feature provides mechanisms to warrant the trustworthy of an agency, by means of taking measures of the values of the agency software and the storage of these values in TPM for a posteriori matching. SecMiLiA provides the mechanisms to perform the secure migration of mobile agents among different agencies. Next figure depicts the underlying basis technology of SecMiLiA. Previously, we mentioned that JADE Agent class provides two 'non-secure' migration methods. We have created a new class inherited from Agent class, which redefines the migration method.

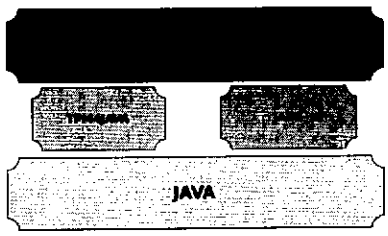


Figure 1: Stack Packages

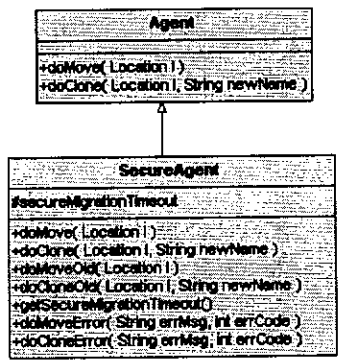


Figure 2: SecureAgent class.

4.1 Some Considerations in the design of SecMiLiA

SecMiLiA is an add-on to the JADE framework that provides a high security level in the migration process. The design principles of SecMiLia are: A secure environment in which agents can be secure executed and migrated, which is obviously the main target of SecMiLiA. The easy integration with JADE, by means of tiny changes to JADE only a new set of added functionalities. A friendly user interface to agent programmers. Compliance with the existing security solutions. Adaptable to use other cryptographic hardware, such as smartcards, with tiny changes. A generic way of use, doing this users can make use of the library to solve a very wide set of problems. Last but not least, it is easily expandable in the future. SecMiLiA provides highly security level in the migration of software agents developed in JADE. We achieve this security level by means of a mechanism that allows the secure migration based on a prior testing of the trustworthy of destination agency in migration step, so that this warrants that the agent execution is in a secure environment. Then, we get a secure environment where agents can run, in such a way that agents are not able to modify the host agency.

We identified a set of requirements in the design of SecMiLiA, which are essential to get a high quality level. These requirements were grouped in two different sets, functional and non-functional. Functional requirements. We consider that the library might to provide a mechanism for secure agent migration in JADE platform. Then, the agent can extend its trustworthy limits by means of adding secure agencies, both from its platform and from remote platforms. It is important to mention the fact that, each agency might provide local functionality to allow an agent to migrate securely to destination platforms. Similarly, agencies might provide the functionality to allow other agencies to take integrity measures to determine whether its configuration is secure. The library might implement a protocol to allow the interchange of the information related to the configuration among agencies. Last but not least, in the functional requirements set, is the library might use trusted hardware. In our case, we used a TPM to measure and store agencies data integrity and to report these data to the agencies. Non-functional requirements. The library might be integrated in JADE platform, in such a way that its use do not imply big changes in the platform. The operation of the library might be transparent to the user. The library might be easily adaptable to current solutions in such a way that the number of modifications is reduced at maximum.

4.2 The core of our approach: The Secure Migration Protocol

Next we describe the core of our approach, the protocol that enable to our library to provide the secure migration. Firstly, we study several interesting features of known attestation protocols and secure migration protocols. We do a balance between their benefits and drawbacks to build our own secure migration protocol. Henceforth, we use the concept of migration both to agent cloning and agent moving, despite of their tiny conceptual difference it is not relevant in our case.

The first approach of secure migration protocol is in [10]. This protocol provides some important ideas to take into account during the design process of the final protocol. We notice the agent trusts in its platform to check the migration security. As well as the necessity of the use of TPM to obtain and report configuration data by a secure way. Other protocol that provides interesting ideas to take into account when we develop a secure migration system is in [16]. The most relevant aspects provided are, a protocol shows how an agent from the agency requests to TPM the signed values from PCRs. As well as the protocol shows how the agent obtain platform credentials. These credentials together with PCRs signed values allow to determine if the destination configuration is secure.

Of particular interest is the Platform Configuration Register (PCR) extension operation. PCRs are initialized at power up and can only be modified by reset or extension. The use of a Certification Authority (CA) that validates the Attestation Identity Key (AIK). The use of configurations to match received results from remote agency. We designed our protocol based on the study of these three previous protocols. Our protocol presents several common features with them. The agency provides to the agent the capability to migrate securely. The agency uses a TPM that takes configuration values measures stored in PCRs. TPM signs PCRs values using a specific AIK for destination agency, in such a way that data receiver knows securely the TPM identity, which signed. A Certification Authority generates needed credentials to correctly verify the AIK identity. Together with signed PCRs values the agency provides AIK credentials in such a way that the requester can correctly verify that data comes from agency TPM. Following we define the 18 steps protocol, used to perform secure migration.

We can observe that the protocol fulfills the five main characteristics we mentioned previously. Then we have a clear idea of the different components of the system as well as the interaction between them to provide the security in the migration.

4.3 Validating the secure migration protocol using the AVISPA toolsuite

The Secure Migration protocol described in the previous section is the core of our research. We want to build a robust solution. Then, the next step in this research was to make a validation of the protocol. Among different alternatives we selected a model checking tool called AVISPA. The main reason to chose this toolsuite is the easy interpretation of the results. No expertise in formal methods is needed to use this set of tools.

Algorithm 1 Secure migration protocol

1: Agent Ag requests to his agency A the migration to Agency B.
2: Agency A (source agency) sends to agency B (destination agency) a request for attestation.
3: Agency B accepts the request for attestation and send a nonce N (this value is composed by random bits used to avoid repetition attacks) and indexes of PCRs values that needs.
4: Agency A requests, from its TPM the signed PCR values needed by agency B together with the nonce N..
5: TPM returns requested data.
6: Agency A obtains AIK credentials from its credentials repository.
7: Agency A requests agency B for signed PCRs values and nonce N all signed. Then it sends AIK credentials, which contains the public key corresponding with the private key used to sign data. Additionally, it sends a nonce N and the indexes of PCRs that wants to know.
8: Agency B validates the authenticity of received key verifying the credentials by means of the CA public key which was used to generate those credentials.
9: Agency B verifies the PCRs values signature and the nonce received using the AIK public key.
10: Agency B verifies that PCRs values received belongs to the set of accepted values and them the agent configuration is valid.
11: Agency B requests to its TPM the PCR values requested by the agency A together with the nonce signed.
12: TPM returns requested data.
13: Agency B obtains AIK credentials from its credential repository.
14: Agency B sends to agency A PCR values requested and the nonce signed. Also it sends AIK credentials, which contains the public key corresponding to the private key used to encrypt the data.
15: Agency A validates the authenticity of received key verifying the credentials by means of CA public key that generated those credentials.
16: Agency A verifies the PCR values signature and the nonce received using the AIK public key.
17: Agency A verifies that PCR values received belongs to the set of accepted values and then the agency B configuration is secure. From this point trustworthy between agencies A & B exists.
18: Then Agency A allows to the agent Ag the migration to agency B.

AVISPA is an automatic push-button formal validation tool for Internet security protocols, developed in a project sponsored by the European Union. It encompasses all security protocols in the first five OSI layers for more than twenty security services and mechanisms. Furthermore this tool covers (that is verifiable by it) more than 85 of IETF security specifications. AVISPA library available on-line has in it verified with code about hundred problems derived from more than two dozen security protocols. AVISPA uses a High Level Protocol Specification Language (HLPSL) to feed a protocol in it; HLPSL is an extremely expressive and intuitive language to model a protocol for AVISPA. Its operational semantic is based on the work of Lamport on Temporal logic of Actions. Communication using HLPSL is always synchronous. Once a protocol is fed in AVISPA and modelled in HLPSL, it is translated into Intermediate Format (IF). IF is an intermediate step where re-write rules are applied in order to further process a given protocol by back-end analyser tools. A protocol, written in IF, is executed over a finite number of iterations, or entirely if no loop is involved. Eventually, either an attack is found, or the protocol is considered safe over the given number of sessions. System behaviour in HLPSL is modelled as a 'state'. Each state has variables which are responsible for the state transitions; that is, when variables change, a state takes a new form. The communicating entities are called 'roles' which own variables. These variables can be local or global. Apart from initiator and receiver, environment and session of protocol execution are also roles in HLPSL. Roles can be basic or composed depending on if they are constituent of one agent or more. Each honest participant or principal has one role. It can be parallel, sequential or composite. All communication between roles and the intruder are synchronous. Communication channels are also represented by the variables carrying different properties of a particular environment. The language used in AVISPA is very expressive allowing great flexibility to express fine details. Further, defining implementation environment of the protocol and user-defined intrusion model may increase the complexity. Results in AVISPA are detailed and explicitly given with reachable number of states. Therefore regarding result interpretation, AVISPA requires no expertise or skills in mathematics to get conclusions of the study.

Of the four available AVISPA Back-Ends we chose the OFMC Model, which is the unique that uses fresh values to generate nonce's. However, this alternative requires a limit value for the search. The results of our research are the following:

```
SUMMARY
SAFE
DETAILS
BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
/testsuite/results/protocolo_2.txt.if
GOAL
as_specified
BACKEND
OFMC
COMMENTS
STATISTICS
parseTime: 0.00s
searchTime: 564.34s
visitedNodes: 18 nodes
depth: 2000 plies
environment()
```

These results shows that the summary of the protocol validation is safe. Also some statistics are shown among them depth line indicates 2000 plies, but this process has been performed for 200, 250, 300, 400, 500 and 1000 of depth values with similar results. We have checked the authentication on pcra(pcrib), pcrb(pcria), aikab and aikba. Because these are the values stored in Agency A TPM and Agency B TPM and therefore the values exchanged by agent A and B. Them, these are the values to check that are securely interchanged to test the protocol.

5 An overview of the services provided by SecMiLiA

In this section we describe main services, classes and methods provided by SecMiLiA. It is out of the scope of this paper a detailed description of every class and method of the library, a further description can be found in [25]. Thus, we make an overview of them describing the most relevant aspects and functionalities.

5.1 The SecureAgentMobility Service

SecureAgentMobility service provides the secure migration functionality between containers in the same platform. The Helper class provides two important methods the secureMove capable to move agents securely to destination, and the secureClone that allows to secure agents to be cloned securely in destination containers.

This diagram shows a case where an agent request for a service to move to a container (c2). Following are steps: This protocol considers the service as a monolithic entity, but in order to clarify several components belonging to that service were missed. Other important concern is that the service invokes doMoveOld method to start the migration. This functionality is similar to doMove from Agent class, moving agent to destination. Previously to this migration service checks that destination is secure. By means of this we achieve a similar behaviour of doMove method from SecureAgent and doMove method from Agent class. Content of these

Algorithm 2 The agent SA is moved to the container C2

1: The agent requests for service (S1) to move to container (C2).
2: S1 sends a remote attestation request to the instance of the service in the destination container, that is (S2) service.
3: S2 service accepts the request.
4: S1 service sends requested information to S2.
5: S2 service responses to S1 sending the attestation result.
6: S1 service starts agent migration to C2 container.

messages is encapsulated using AttestRequest_Interface y AttestData_Interface classes

AttestRequest_Interface class provides access to data from a request attestation message, and AttestData_Interface provides access to attestation information from a concrete container. Both classes encapsulate information from attestation protocol messages. Containers complete needed data using set methods in each message, in such a way that the other container continues the attestation procedure. In order to destination container continues the attestation procedure source container completes the needed data by means of set method. The SecureAgentMobility service uses AttestTool_Implement class to complete data messages. The AttestTool_Implement class manages attestation protocol messages, it is, this generates the messages in their sources and verifies them in destination. AttestTool_Implement class is provided with access to system TPM by means of TPM_Interface as well as to a CA throughout CA_Interface interface. This fact allows using both entity functionalities to complete messages. Also, AttestTool_Implement class can access to system configuration thought AttestConfig_Interface interface. AttestTool_Implement class manage TPM access in such a way that attestation protocol can be performed. AttestTool_Implement class behaviour is similar to Key Cache Manager managing TPM keys. AttestTool_Implement class implements AttestTool_Interface interface where secure migration process states codes are defined. These error codes permit to agents determine the happened when doMoveError and doCloneError methods are called.

In order to generate and verify attestation messages contains is needed the use of a TPM and a CA. More relevant reasons for this are; TPM provides the functionalities to generate attestation data, it is AIK generation, data signature, nonce values generation, etc. CA provides credential generation, these certifies AIKs objectives in the rest of containers trust in AIK signed data.

A relevant aspect in the use of AIK in the protocol is to sign attestation data. TPM generates AIK and this must be certified by a CA. Next protocol details how this key is generated and certified. The AttestTool_Implement requests for generate an AIK key and a credentials request to TPM. TPM generates the AIK key as well as the request and these are delivered to AttestTool. Request is encrypted in such a way that only CA is able to have in clear. The AttestTool_Implement sends a CA request. Then the CA decrypts the request and generates corresponding credentials which are delivered to AttestTool_Implement in such a way that only TPM have in clear. The AttestTool_Implement sends CA response to TPM. The TPM decrypts the requested data and sends the key data to AttestTool_Implement.

Several interfaces interact in this process. The AIKRequestData_Interface, which contains data to allow TPM generates AIK as well as creates credentials generation request. The AIKRequestData_Interface that contains data to allow CA generate credentials to AIK. And the AIKResponse_Interface with the to enable to the TPM to obtain the credentials generated by the Certification Authority.

Finally, we briefly describe some the relevant classes and interfaces of our library. The AIKRequestData_Interface interface that provides the getIdentityLabel() method, which returns AIK identity label. Once the AIK reaches the corresponding destination, the service can generate the configuration attestation data and to produce the signature with these data. Other essential component in this structure are the credentials, the AIKCredentials_Interface interface.

5.2 The SecureInterPlatformMobility Service

The SecureInterPlatformMobility service uses most of used elements in SecureAgentMobility service. Differently to previous service in this case we deal with migration between different platforms, then service messages among source container and destination container are not allowed, for this both containers must belong to the same platform. This fact ought to communicate using ACL messages.

Following, we define the 14 steps protocol, used to SecureInterPlatformMobility service.

Algorithm 3 Secure migration protocol

1: SA agent requests S1 service to move to C2 container.
2: S1 service sends a remote attestation request to S1M service from the main container of its platform.
3: S1M service sends a request for remote attestation to source platform AMS, A1.
4: A1 sends a request for attestation to destination platform AMS, A2.
5: A2 accepts the request and notifies to A1.
6: A1 notifies S1M service acceptation.
7: S1M service notifies S1 service acceptation.
8: S1 service sends request data to S1M service.
9: S1M service sends data to A1 request.
10: A1 sends request data to A2.
11: A2 responses to A1 sending the attestation result.
12: A1 sends result to S1M service.
13: S1M service sends received result to S1 service.
14: S1 service starts agent migration to destination platform the main container.

Thus, the source container service needs the interaction of the main container service to contact with destination platform AMS, in such a way that the only way to access to AMS class implemented from the main container. The communication between source platform AMS and destination platform AMS is done by ACL messages. Destination AMS uses AttestTool_Implement class to deal service messages, means messages sources are dealt by AttestTool_Implement in source container service. The rest of service operation component is similar to SecureAgentMobility previously detailed.

6 Extensions of the TPM4Java technology to our solution

Tpm4java is a java library for accessing a trusted platform module in your java applications. We made use of this library to access to the TPM functionality. Nevertheless, we missed some aspects. For this purpose with library some classes and interfaces from TPM4Java library have been modified to adapt it to our requirements. Following a description of every change is provided.

- TssLowLevel interface. TPM_OwnerReadPubek method is added because this is not included in TPM4Java but TCG 1.2 specification includes that.
- TssHighLevel interface. GenerateAIK method is modified to returns a TPMKeyWrapper class corresponding to generated AIK instead of key handler in TPM. Therefore AIK data are needed to store it in the hard disc.
- TssCoreService class implements TssLowLevel interface. TPM_OwnerReadPubek method has been added.

- TssHighLevelImpl class implements TssHighLevel interface. GenerateAIK method has been modified in such a way that a TPMKeyWrapper class is returned.

- PrivacyCA class is added, this is an altered version of SimplePrivacyCA class provided with TPM4Java.

7 Conclusions & Future Work

We have presented a cryptographic hardware based solution to provide secure migration capability of mobile agents. SecMiLiA is an add-on to the JADE framework that provides these functionalities through a friendly interface. The underlying security of this approach is based on the Secure Migration Protocol, which has been carefully described and validated by AVISPA toolsuite.

Some future work remains. Among them we propose the study of a migration library based on an anonymous direct attestation instead of the Certification Authority based protocol developed. This study might be very interesting due to the advantages provided by the direct anonymous attestation based protocol. Certify issuer entity and verifier entity can not act together to violate the system security, then one unique entity can be verifier and issuer of certificates. Last but not least of the advantages is that certificates only need to be issued once which solves the aforementioned bottleneck. These advantages shows that anonymous attestation protocol is an interesting option for attestation. A possible denial of service attack consists on reboot or even turn off the machine in which is hosted the current agency of the agent itinerary; the system is not working, a blackout, etc. For this purpose, we plan to take some kind of measures to prevent it. The design and implementation of recovery and waiting mechanisms is an interesting open field for future researches. Another improvement consists on the add-on of a reputation based system. Reputation system is focused on found all possible malicious agencies from agents. If a non desirable operation is done this can be monitored and notified. Similarly to the previous section this represents an open field of research due to the fact that the integration of a reputation system adds some advantages and disadvantages of course. In such a case that the reputation system is very strict then every anomaly is monitored and notified to the system which punishes this agency in some way. This method can punish some agencies by external causes, actually not malicious purposes.

Another relevant issue is the security of the reputation system itself. It is worth to ask who controls this system, since who controls this system can benefit by punishing other agencies or even favouring non-trusted agencies. Other field of research consist on the application of a revocation hosts authority (HoRA). This proposal is based on punishing agencies that performed tasks not allowed in the system. This proposal presents several limitations, since a non intended error could happen in the execution of an agent, which can produce the revocation of an agency (which might be an exaggerated punishment). HoRA [24] can apply more flexible punishment policies since can control which to be revoked, and even revoke hosts according to the degree of the attacks or the possible benefits of the malicious hosts. More limitations exist. The internal data base of HoRA grows indefinitely, since revoked hosts are never removed. This practice is required to control the revoked host history. Concretely, authors assume that the storage capability of the system might not be a limitation for the system, since this is a system whose main tasks are storage and management of the data base. The confidentiality of executing data. Despite of HoRA being considered a trusted entity, in the revocation case confidential data might be sent. Among the difficulties that HoRA can find sniff these data depend on the detection mechanism used.

References

[1] Mouratidis H, Kolp M, Faulkner S, Giorgini P, A Secure Architectural Description Language for Agent Systems. AAMAS'05 July 25-29, Utrecht, Netherlands.

[2] Wang, H., and Wang, C. 1997. Intelligent Agents in the Nuclear Industry. IEEE Computer 30(11): 2834.

[3] Schwuttke, U. M., and Quan, A. G. 1993. Enhancing Performance of Cooperating Agents in Real-Time Diagnostic Systems. In Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-93), 332337. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

[4] Clements, P., Papaioannou, T., Edwards, J., Aglets: Enabling the Virtual Enterprise. In the Proc. Of Managing Enterprises - Stakeholders, Engineering, Logistics and Achievement (ME-SELA'97) ISBN 1 86058 066 1, p425.

[5] Online available on: http://cougaar.org/

[6] Shepherdson, D., The JACK Usage Report. In the Proc Of. Autonomous Agents and Multi Agents Systems 2003 (AAMAS 03).

[7] Online available at: http://jade.tilab.com

[8] http://www.irit.fr/recherches/ISPR/IAM/JavAct.html

[9] Alechina, N., Alechina, R., Habner, J., Jago, M., Logan, B., Belief revision for AgentSpeak agents. In the Proc Of Autonomous Agents and MultiAgents Systems 2006. Hakodate, Japan. ISBN:1-59593-303-4, p1288 - 1290.

[10] Maña, A. Proteccin de Software Basada en Tarjetas Inteligentes. PhD Thesis. University of MÃ¡laga. 2003.

[11] Hachez, G. A Comparative Study of Software Protection Tools Suited for E-Commerce with Contributions to Software Watermarking and Smart Cards. PhD Thesis. Universite Catholique de Louvain. 2003.

[12] Efficient Software-Based Fault Isolation. Robert Wahbe, Steven Lucco, Thomas E. Anderson, Susan L. Graham. In Proceedings of the 14th ACM Symposium on Operating Systems Principles. 1993.

[13] Necula G. Proof-Carrying Code. In Proceedings of 24th Annual Symposium on Principles of Programming Languages. 1997.

[14] Gunter Carl A., Homeier Peter, Nettles Scott. Infrastructure for Proof-Referencing Code. In Proceedings, Workshop on Foundations of Secure Mobile Code, March 1997.

[15] Bennet S. Yee,A Sanctuary for Mobile Agents. Secure Internet Programming 1999

[16] Trusted Computing Group: TCG Specifications. 2005. Available online at https://www.trustedcomputinggroup.org/specs/

[17] Stern, J. P., Hachez, G., Koeune, F., Quisquater, J. J. Robust Object Watermarking: Application to Code. In Proceedings of Info Hiding '99, Springer-Verlag. LNCS 1768, pp. 368-378, 1999.

[18] Antonio Maña, Antonio Muñoz, Daniel Serrano. Towards Secure Agent Computing for Ubiquitous Computing and Ambient Intelligence. Fourth International Conference, Ubiquitous Intelligence and Computing, Hong Kong (China) 2007. LNCS.

[19] Ordille J. When agents roam, who can you trust?. Technical report, Computing Science Research Center, Bell Labs, 1996.

[20] Yee B. S. A sanctuary for mobile agents. In DARPA workshop foundations for secure mobile code, 1997.

[21] Roth, V. Mutual protection of cooperating agents. In Secure Internet Programming: Security Issues for Mobile and Distributed Objects, volume 1906 of LNCS. Springer-Verlag, 2001.

[22] B. Barak, et. al., On the (Im)possibility of Obfuscating Programs, Electronic Colloquium on Computational Complexity, Report No. 57, 2001.

[23] William M. Farmer, Joshua D. Guttman, and Vipin Swarup. Security for Mobile Agents: Authentication and State Appraisal. In the proceedings of the European Symposium on Research in Computer Security (ESORICS 96), pp 118-130, September 1996.

[24] Esparza O., Munoz J., Soriano M., Forne J. Punishing Malicious Hosts with the Cryptographic Traces Approach. New Generation Computing Journal. ISSN:0288-3635 vol24,no.4;pp.351-376(2006).

[25] Antonio Muñoz, Antonio Maña, Daniel Serrano. Trusted Computing: The Cornerstone in the Secure Migration Library for Agents. In Proc. of the PAAMS'09. ISSN 1867-5662. ISBN 978-3-642-00486-5. Springer-Verlag.