AUSTRALIAN JOURNAL OF BASIC AND APPLIED SCIENCES

ISSN:1991-8178 EISSN: 2309-8414
Journal home page: www.ajbasweb.com

A Review on Attacks and Security Approaches in Mobile Agent Technology

[1]Adri Jovin J.J. and [2]Marikkannan M.

[1]*Department of Information Technology, Sri Ramakrishna Institute of Technology, Coimbatore, INDIA*
[2]*Department of Computer Science and Engineering, Institute of Road and Transport Technology, Erode, INDIA*

Address For Correspondence:
Adri Jovin J.J., Department of Information Technology, Sri Ramakrishna Institute of Technology, Coimbatore, INDIA.
E-mail: adrijovin.it@srit.org

ARTICLE INFO

Article history:
Received 04 December 2015
Accepted 22 January 2016
Available online 14 February 2016

Keywords:
Mobile Agent Security, Attack detection, Attack prevention, Attack Classification

ABSTRACT

Mobile Agents have become a promising technology in the domain of Distributed Computing. Because of the intelligence it possesses, Mobile Agents are widely used in a number of applications. Exposing the Mobile Agents into the open Distributed System increases the vulnerability towards threats, in spite its intelligence and computational capabilities. It is not an easier task to enforce security at all levels of the itineraries of a Mobile Agent. A number of attacks have been identified recently. This paper classifies the attacks as well makes a detailed study of each attack. We then survey the techniques which detect the attacks and protect the Mobile Agents from the threats. We finally conclude with the suggestion of appropriate countermeasure for a certain type of attack**.**

INTRODUCTION

A Software Agent is a program that assists people and acts on their behalf. Software Agents can be broadly classified as Static Agents and Mobile Agents. A Static Agent is one which resides inside the system whereas, a Mobile Agent is one which is not bound to the system where it begins its execution. It has an ability to transport itself from one system to another. The Mobile Agent Technology comprises of two components namely the Mobile Agent and the Execution Platform. The Mobile Agent could execute only inside the Execution Platform which is located in the host system.

Mobile Agents have a property of hopping from one system to another which is usually termed an itinerary. The main reasons for using Mobile Agents are:

- Reduced network load
- Reduced Network latency
- Protocol Encapsulation
- Asynchronous and Autonomous execution
- Dynamic Adaptation
- Heterogeneity
- Robustness
- Fault tolerant

Due to their extra-ordinary qualities, they are used widely for applications like E-commerce, Personal Assistance, Secure Brokering, Distributed Information Retrieval, Telecommunication Network Services, Parallel Processing, Monitoring and notification etc. (Lange and Oshima, 1998). Mobile Agents are usually written in Machine-independent Languages, so that they could be run in heterogeneous environments (Michael S. Greenberg et al., 1998).

To Cite This Article: Adri Jovin J.J. and Marikkannan M., A Review on Attacks and Security Approaches in Mobile Agent Technology. ***Aust. J. Basic & Appl. Sci.,*** *10(2): 37-43, 2016*

Even though Mobile Agents are used for different applications, they are subjected to a number of security threats when they move within the open Distributed Environment. Literatures are available to categorize the attacks. A detailed categorization is available in the following sections. Numerous security mechanisms and architectures have also been proposed to overcome the security issues faced by Mobile Agents. This paper gives a detailed outline of various threats as well as countermeasures to detect the threats and protect the Mobile Agents.

Attacks:

As discussed earlier, the basic components of the Mobile Agent Technology are the Mobile Agent and the Execution Platform. Based on the Mobile Agent and Execution Platform, attacks can be categorized into four classes (Jansen and Karygiannis, 1999):

- Agent-to-Platform attack
- Agent-to-Agent attack
- Platform-to-Agent attack
- Other-to-Agent attack

Agent-to-Platform attack:

An Agent-to-Platform attack is one in which a Mobile Agent becomes a threat to the platform over which it is about to act. Here, the Mobile Agent is malicious and may launch an attack over the Execution Platform. The attacks such as Masquerading, Denial of Service and Unauthorized Access comes under this trait.

Agent-to-Agent attack:

In case of Agent-to-Agent attack, a Mobile Agent which is malicious tends to attack another Agent. It is a general practice that some components in the platform themselves may be static agents. The malicious Mobile Agent may attack either another Mobile Agent or Static Agents which reside in the host. Attacks such as Masquerade, Denial of Service, Repudiation and Unauthorized access are used to perform Agent-to-Agent attack.

Platform-to-Agent attack:

A Platform-to-Agent attack is one in which the Execution Platform compromises the security of the Mobile Agent. This includes attacks such as Masquerading, Denial of Service, Eavesdropping and Alteration.

Other-to-Agent attack:

The Other-to-Agent attack specifies all attacks which a Mobile Agent may suffer during its travel through the network or visiting a host. This may include masquerading, denial of service, unauthorized access and copy-and-replay.

Apart from the above said attacks, attacks such as colluded truncation (Silei et al., 2008), tailgating (Marikkannu et al., 2011) have also been identified.

Protection Techniques:

Security Architectures:

Various Security Architectures have proposed to resolve the threats to Mobile Agents. A few focus on the Mobile Agent (Garrigues et al., 2010), a few over the Mobile Agent Execution Platform and the others focus over the application as a whole.

MACPL:

The software architecture and the development environment (Garrigues et al., 2010) focus on the development of secure applications using Secure Mobile Agents. The architecture proposed, reduces the complexity in the implementation of Cryptographic Protocols. It also promotes the reuse of the protocols by implementing the agent task and security mechanism as separate entities. The architecture accomplishes the tasks specified by using Mobile Agent Cryptographic Protection Language (MACPL). This facilitates the easy reuse of control codes. The platforms are thereby relieved from providing protection to the agent. However, this mechanism greatly increases the agent size which would affect the performance of real-time systems. This mechanism lacks a looping structure and does not address all possible attacks. Therefore, this architecture is attack specific.

Mobile Trust:

MobileTrust (Lin and Varadharajan, 2010) is Trust enhanced security architecture for Mobile Agent system. This method uses the extension of certain traditional security mechanisms by including trust decisions with the help of certain relationship policies. This provides a trust enhanced security mechanism. A Trust

Management Layer named Codifying Trust Evidence acts to present the security-related trust relationships. The trust evidence is derived, presented, evaluated and decision made, based on the information in the trust relationship database. This model has been worked out in the Aglets platform and is found to be very effective.

Reference Clone:

Reference Based Execution (Benachenhou and Pierre, 2006) is one mode of protecting the Mobile Agent from malicious attacks. This method is based on the verification of code integrity by comparing the Mobile Agent which is under execution and its reference clone. The reference clone is kept in a trusted server, with which the Mobile Agent under execution makes verification for each itinerary. This greatly protects the Mobile Agent from Code, execution and data modification. The reference clone does not help the Mobile Agent if there is a Denial-of-Service and is not suitable to protect the state of the Mobile Agent. This is because; the state of the Mobile Agent gets changed only if it actually moves from one platform to another, which is impossible with the reference clone in the trusted server.

Formal Modeling:

An Extended Elementary Object System(Ma and Tsai, 2008) is a hybridization of Object Orientation and Petri Nets. The Reference Clone is a formal approach which supports weak mobility. The EEOS approach provides support to strong mobility as well provides a secure means of communication. The transactions/executions are based on mutual authentication between the Mobile Agent and the Execution Platform. It also provides a formal approach to detect malicious attacks over the Execution Platform. This model requires more refinement and must involve more security mechanisms which would fit into the generic model.

Code Integrity and Malicious Availability Test:

The eXtended Root Canal Algorithm and Malicious Identification Police (Venkatesan et al., 2010) is a hybrid approach to verify Code Integrity and Maliciousness of a Mobile Agent respectively. The eXtended Root Canal Algorithm used in this method is an extended version of the Root Canal Algorithm which was initially used to check the integrity of Mobile Agent Code. This method exhibits extremely low time complexity compared to that of its competitor techniques such as Code on Demand (Wang et al., 2002) and SeMoA(http://semoa.sourceforge.net). Malicious Identification Police (MIP) is a policy based approach which greatly helps in the identification of malicious activities. The eXtended Root Canal proves to be efficient in terms of time and space complexity and protects the Mobile Agent from almost all attacks except replay attacks.

Dual Check-Point Method:

The Dual Check-Point Analysis (Marikkannu et al., 2011) is a technique which addresses an unidentified attack namely tailgating attack, in which the Malicious Agent injects itself or gets attached to the Mobile Agent to attack either the Mobile Agent of the Execution Platform. It maintains an Authentication Table to check both the authenticity and integrity of the Mobile Agent Code. The technique is named Dual Check-Point Analysis since it checks the mobile agent both in terms of Digital Signature Verification (at outer gate) and Size verification (at inner gate). This technique provides an elementary solution to the tailgating attack.

Artificial Immune System:

The Artificial Immune System (Venkatesan et al., 2012) based model of Mobile Agent Platform Protection gives a clear classification over the separation of duties and clones to handle foreign agents. This improves the computational capability of the system. Since the duties are clearly defined, protection is provided only by those mobile agents which are capable of malicious pattern identification. Moreover, this method greatly reduces the computational cost.

K-reponse Recovery Model:

The K-response Recovery Model (Venkatesan et al., 2009) is an effective mechanism to recover the components of a Mobile Agents namely agent code, itinerary, state and data even after a malicious attack. In this technique, each host dispatching the Mobile Agent takes a clone of the Mobile Agent with its previous state and sends a message to the previous execution platform whether the Mobile Agent is alive or not. This technique may fail if the Mobile Agent is subjected to a host where all execution platforms are malicious and coordinates an attack.

Trust and Reputation Management:

The Trust and Reputation Management (Geetha and Jayakumar, 2015) has been found to be an effective framework against colluded truncation attacks. The major emphasis on security is towards the path which is travelled by the Mobile Agent. A routing table based on trust and reputation has designed to guide the Mobile

Agent with a secure path. Moreover, cryptographic algorithms have been used additionally to ensure the security services.

Code Based Techniques:

Dynamic code:

Dynamically upgradeable code (Wang et al., 2002) which possess the capability of adding new functionality modules to the Mobile Agent and deleting the redundant ones are used to enhance code privacy. It incorporates the techniques of Code Change Authorization Protocols and Double Integrity Verification scheme to ensure the integrity of the Mobile Agent Code. Though the code is dynamically upgradeable, integrity protection is ensured by using an authorized mechanism for the change of agent code and validation of changed code by host. The literature however accepts that the mechanism does not provide a complete solution to the Code Integrity problem.

Proof Carrying Code:

The Proof Carrying Code (Necula and Lee, 1998) is one in which the Mobile Agent code carries the proof required for execution. The code is being checked to verify that it does not violate the safety policies of a system.

Self-Modifying Code:

Self-Modifying Codes (Shan and Emmanuel, 2011) use the code obfuscation algorithm to modify the code by itself to escape attack at functional level. Obfuscation is of two types namely Static Obfuscation and Dynamic Obfuscation. Static Obfuscation is achieved by various techniques. This may be done using Branch Pointer manipulation, Address manipulation, Inheritance relationship modification, branch function obfuscation, variable reconstruction and array reconstruction. The protection provided by static obfuscation is less compared to that of dynamic code obfuscation which is provided by techniques like Jump table spoofing, Inter-process communication, self-modifying code and signal obfuscation. The decision to obfuscate the function call to normal instruction or control flow instruction is done using Liveness analysis. Based on the analysis the further modifications are done over the code. This method is resistant to both static and dynamic attacks over the Mobile Agent code.

Self-Protecting Agents:

Self-protecting Agents (Ametller et al., 2004), unlike the traditional Mobile Agent protection mechanisms which are dependent on the Platform for security, rely on themselves for the protection of the code and the data. This mechanism was found to be novel and independent of the execution platform. An enhanced mechanism of self-protecting agents deploy the technique of fragmentation which involves self-decryption, co-operation and obfuscation (Srivastava and Nandi, 2014).

Communication Based Techniques:

Passport and Visa Model:

The Passport and Visa model (Guan et al., 2003) involves the technique used in Immigration and Emigration services. This model serves up-to-date digital credentials for Agent-Host authentication. This provides effective security mechanism for online groups to control the migration of Mobile Agent. Also, it provides a solution to effectively manage and control the entry and exit of mobile agents in the execution environment. Thus, there would be a continuous monitoring over the movement of the Mobile Agents which reduces the risk of the Execution Environment being affected by a malicious Mobile Agent.

Another approach (Pierre et al., 2007) based on the cooperation between the Mobile Agent and Sedentary Agent greatly helps in addressing various security issues. This technique is completely based on the cooperation between a mobile agent and sedentary agent, reference execution (almost similar to reference clone approach discussed above), cryptographic algorithms and digital signatures. This technique is strictly time constrained and thereby controls the incidence of replay attacks. This technique maintains the integrity and confidentiality of the Mobile Agent code, thereby proving it better than its predecessor.

Security Specific Models:

Sandbox:

Sandboxes (Grandison and Sloman, 2000) for Mobile Agents are implemented using the Java Interpreter inside the Internet Browser. This security Model mainly comprises of three components namely class loader, verifier and security manager.

Code Signing:

Code Signing (Reiser, 2000) is implemented using Microsoft Authenticode which uses ActiveX for the process of signing the code. The system has a particular policy, which if changed by a Mobile Agent or any

other external entity is considered to be a threat. It introduces a concept of trust model to distinguish the trustworthy authors from untrustworthy authors.

Firewall:

Firewall is a general technique used by a host to decide whether to allow the execution of a program or not, based on a certain set of rules or protocols. The same applies for the Mobile Agents also. The firewall decides whether to allow the execution of a Mobile Agent or not.

Key Management Based Techniques:

Hierarchical Key Management:

The Hierarchical Scheme of Key Management (Chen et al., 2010) is one introduced to reduce the complexity of managing the keys by a Mobile Agent, since it has the load of getting transferred from one system to another. The Elliptic Curve Cryptosystem is incorporated with this to enhance the security mechanism. This greatly enhances the performance of the Mobile Agent since the overhead due to key size is reduced. The keys get modified with respect to time. This is very essential in an open system, because a Mobile Agent having access to a resource at a particular time may not be provided the access to the same resource at some other time. Therefore, the resources get accessed only when they are given access. This technique is resistive to reverse attack, collusion attack, and external collective attack and date alteration attack.

The Hierarchical Key Management scheme (Vijayakumar et al., 2012) provides security to the data carried by the Mobile Agent from a hacker who is tending to attack the Mobile Agent. This scheme is different from that of the data protection schemes which use RSA Algorithm or the Elliptic Curve Cryptography in the way of reducing the complexity of the implementation of the security scheme, when used in standalone mode. Moreover, this scheme is applicable in an open distributed environment.

Proxy Signature Protocol:

The threshold proxy signature protocol (Hong, 2009) proves to be an effective technique which employs a proxy signer to sign a digital signature. This digital signature is done on behalf of the owner of the agent. The proxy signing is based on RSA algorithm and the sharing is done using Lagrange Formula. However, this technique suffers from different security pitfalls (Yu et al., 2014). These pitfalls make it vulnerable to security attacks. Case studies reveal that the security algorithm could not ensure secrecy, prone to identifiability, unforgeability, undeniability and is more vulnerable with timing constraints.

Tripmarker:

The usage of agent identifiers and trip markers (Garrigues et al., 2009) have been found to play a better role in preventing replay attacks in Mobile Agents. The tripmarker keeps in track the itinerary of the Mobile Agent. Since the execution platforms have a real track record over the tripmarker, it is possible to avoid replay attacks, provided the execution platform is not compromised. This security mechanism is therefore, dependent over the execution platform.

Self-Reliant Mobile Code:

Self-Reliant Mobile Code (Srivastava and Nandi, 2014) has been a combination of different techniques based on integrity based confidentiality and self-protection approach. An improvised symmetric key algorithm based on Petri net has been formalized. One of the key component is distributed in a secure manner and another key component is derived from the data collected during the run time or during the time of execution.

Table 1: Summary of Attacks and Prevention Mechanism.

Type of Attack	Preventive Technique
Masquerading	MobileTrust (Lin and Varadharajan, 2010), Extended Elementary Object System (Ma and Tsai, 2008), eXtended Root Canal Algorithm and Malicious Identification Police (Venkatesan et. al., 2010), Artificial Immune System (Venkatesan et. al., 2012), Proxy signature protocol (Hong, 2009)
Denial of Service	Reference Clone (Benachenhou and Pierre, 2006)
Unauthorised Access	MACPL (Garrigues et.al. 2010), MobileTrust (Lin and Varadharajan, 2010), Passport and Visa model (Guan et. al., 2003), Extended Elementary Object System (Ma and Tsai, 2008), Code Signing (Reiser, 2000), Firewalling, Proof Carrying Code (Necula and Lee, 1998), eXtended Root Canal Algorithm and Malicious Identification Police (Venkatesan et. al., 2010), Artificial Immune System (Venkatesan et. al., 2012), Proxy signature protocol (Hong, 2009), Self-Reliant Mobile Code (Srivastava and Nandi, 2014), Elliptic Curve Cryptography (Zakerolhosseini and Nikooghadam, 2013)
Repudiation	MobileTrust (Lin and Varadharajan, 2010), Passport and Visa model (Guan et. al., 2003), Proof Carrying Code (Necula and Lee, 1998), Artificial Immune System (Venkatesan et. al., 2012), Elliptic Curve Cryptography along with the Blind Signature (Zakerolhosseini and Nikooghadam, 2013), Trust and Reputation Management (Geetha and Jayakumar, 2015)
Alteration	Dynamic Code (Wang et.al., 2002), Reference Clone (Benachenhou and Pierre, 2006), Extended Elementary Object System (Ma and Tsai, 2008), Self-Modifying Codes (Shan and Emmanuel, 2011), Hierarchical Key Management (Chen et. al., 2010), Hierarchical Key Management scheme (Vijayakumar et. al., 2012), eXtended Root Canal Algorithm and Malicious Identification Police (Venkatesan et. al., 2010), Proxy signature protocol (Hong, 2009), Self- Protecting Mobile Agents (Ametller et al., 2004), Fragmentation based Self-Protecting Mobile Agents (Srivastava and Nandi, 2014), Self-Reliant Mobile Code (Srivastava and Nandi, 2014), Elliptic Curve Cryptography along with the Blind Signature (Zakerolhosseini and Nikooghadam, 2013), Trust and Reputation Management (Geetha and Jayakumar, 2015)
Eavesdropping	MobileTrust (Lin and Varadharajan, 2010), Sandboxes (Grandison and Sloman, 2000), Hierarchical Key Management (Chen et. al., 2010), Elliptic Curve Cryptography along with the Blind Signature (Zakerolhosseini and Nikooghadam, 2013), Trust and Reputation Management (Geetha and Jayakumar, 2015)
Replay	Cooperation Approach (Pierre et.al., 2007), Tripmarker (Garrigues et al., 2009)
Tailgating	Dual Check-Point Analysis (Marikkannu et. al., 2011)

Elliptic Curve Cryptography:

The hybridization of Elliptic Curve Cryptography along with the Blind Signature (Zakerolhosseini and Nikooghadam, 2013) has been found to an effective security mechanism to protect the Mobile Agent as a whole. The base idea was the application of Elliptic Curve Cryptography with a preliminary access control mechanism (Huang et al., 2009). This scheme is evidently resistant towards Reverse Attack, "Man-in-the-Middle" Attack and Conspiracy Attack.

From our detailed review on various known techniques for securing Mobile Agents the following inference can be summarized in Table-1.

From the above inference, it is evident that most of the techniques are attack specific and does not provide complete protection against all types of attacks. Moreover, all types of attacks are not possible in all the applications and hence those techniques which are discussed above can be used if the application designed demands so.

Conclusion:

In this work, various types of attacks over mobile agents are discussed. Also, the techniques which could be used to address each type of attack surveyed. It is found that, not all techniques address all attacks. Some techniques are more effective to encounter certain attacks. Hence, the type of technique to be used can be decided based on the type of application that is about to be designed. Thus it could be concluded that the security mechanism used is dependent on the type and design of application.

REFERENCES

Danny, B., Lange and Mitsuru Oshima, 1998. Introduction to Mobile Agents, Personal Technologies, 2: 49 – 56.

Michael, S., Greenberg, Jennifer C. Byington, Theophany Holding, David G. Harper, 1998. Mobile Agents and Security. IEEE Communications Magazine, 76 – 85.

Wayne Jansen and Tom Karygiannis, 1999. Mobile Agent Security. Computer Security – NIST Special Publication, 800-19.

Lei Silei, Zhang Rui, Liu Jun, Xiao Junmo, 2008. A Novel Security Protocol to Protect Mobile Agent against Colluded Truncation Attack by Cooperation. International Conference on Cyberworlds, 186,191, 22-24. September.

Marikkannu, P., Adri Jovin, T. Purusothaman, 2011. A Secure Mobile Agent System against Tailgating Attacks. Journal of Computer Science, 7: 488-492.

Carles Garrigues, Sergi Robles, Joan Borrell and Guillermo Navarro-Arribas, 2010. Promoting the development of secure mobile agent applications. The Journal of Systems and Software, 83: 959-971.

Ching Lin and Vijay Varadharajan, 2010. MobileTrust: a trust enhanced security architecture for mobile agent system. Journal of Information Security, 9: 153 -178.

Lofti Benachenhou and Samuel Pierre, 2006. Protection of a mobile agent with a reference clone. Computer Communications, 29: 268-278.

Lu Ma and Jeffrey, J.P. Tsai, 2008. Formal Modeling and Analysis of a Secure Mobile-Agent System. IEEE Transactions on Systems, Man and Cybernetics – Part A: Systems and Humans, 38: 180 – 196.

Venkatesan, S., C. Chellappan, T. Vengattaraman, P. Dhavachelvan, Anurika Vaish, 2010. Advanced mobile agent security models for code integrity and malicious availability check. Journal of Network and Computer Applications, 33: 661 – 671.

Venkatesan, S., C. Chellappan, P. Dhavachelvan, 2009. Performance analysis of mobile agent failure recovery in e-service applications. Computer Standards & Interfaces, 32: 38 – 43.

Geetha, G. and C. Jayakumar, 2015. Implementation of Trust and Reputation Managementfor Free-Roaming Mobile Agent Security. IEEE Systems Journal, 9: 556 – 566.

Tianhan Wang, Sheng-Uei Guan and Tai Khoon Chan, 2002. Integrity Protection for Code-on-Demand mobile agents in e-commerce. The Journal of Systems and Software, 60: 211-221.

Secure Mobile Agents Project (SeMoA) http://semoa.sourceforge.net, 2007.

Venkatesan, S., R. Baskaran, C. Chellappan, Anurika Vaish, P. Dhavachelvan, 2013. Artificial Immune System based mobile agent platform protection. Computer Standards and Interfaces, 35: 365 – 373.

Necula, G.C. and P. Lee, 1998. Safe, Untrusted agents using proof-carrying code. Lecture Notes in Computer Science, 1419: 61-69.

Liang Shan, Sabu Emmanuel, 2011. Mobile Agent Protection with Self-Modifying Code. Journal of Signal Processing Systems, 65: 105 – 116.

Ametller, J., S. Robles, J.A. Ortega-Ruiz, 2004. Self-Protected Mobile Agents. Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, 1: 362 – 367.

Shashank Srivastava and G.C. Nandi, 2014. Fragmentation based encryption approach for selfprotected mobile agent. J. King Saud University – Computer and Information Sciences, 26: 131 – 142.

Sheng-Uei Guan, Tianhan Wang, Sim-Heng Ong, 2003. Migration control for mobile agents based on passport and visa. Future Generation Computer Systems, 19: 173 – 186.

Samuel Pierre, Abdelhamid Ouardani, Hanifa Boucheneb, 2007. A Security protocol for mobile agents based upon the cooperation of sedentary agents. J. Network and Computer Applications, 30: 1228 – 1243.

Grandison, T. and M. Sloman, 2000. A survey of trust in internet applications. IEEE Communications Survey Tutorial, Fourth Quarter.

Reiser, H., 2000. Security requirements for management systems using mobile agents. Proceeding of the Fifth IEEE Symposium on Computers and Communications, 160-165.

Tzer-Long Chen, Yu-Fang Chung, Frank Y.S. Lin, 2010. An efficient date-constraint hierarchical key management scheme for mobile agents. Expert Systems with Applications, 37: 7721 – 7728.

Vijayakumar, P., K. Anand, S. Bose, V. Maheswari, R. Kowsalya, A. Kannan, 2012. Hierarchical key management scheme for securing mobile agents with optimal computation time. Proceedia Engineering – International Conference on Modeling, Optimisation and Computing, 38: 1432 – 1443.

Xuan Hong, 2009. Efficient threshold proxy signature protocol for mobile agents. Information Sciences, 179: 4243 – 4248.

Yong Yu, YiMu, Willy Susilo, Man Ho Au, 2014. Security pit falls of an efficient threshold proxy signature scheme for mobile agents. Information Processing Letters, 114: 5 – 8.

Carles Garrigues, Nikos Migas, William Buchanan, Sergi Robles, Joan Borrell, 2009. Protecting mobile agents from external replay attacks. J. Systems and Software, 82: 197 – 206.

Shashank Srivastava, G.C., Nandi, 2014. Self-reliant mobilecode:anewdirectionofagentsecurity. J. Network and Computer Applications, 37: 62 – 75.

Ali Zakerolhosseini and Morteza Nikooghadam, 2013. Secure Transmission of Mobile Agent in DynamicDistributed Environments.Wireless Personal Communications, 70: 641 – 656.

Huang, K.H., Y.F. Chung, C.H. Liu, F. Lai, T.S. Chen, 2009. Efficient migration for mobilecomputing in distributed networks. Computer Standards & Interfaces, 31: 40 – 47.