Security Issues of the Mobile Multiagent Systems

Assistant Barna Iantovics
Petru Maior University of Tg. Mureş

Abstract*. Mobile agents as network computing technology has been applied to solve various parallel and distributed computing problems, including parallel processing, information search and network management. In this paper, we analyze different security issues and solutions related to the mobile multiagent systems. In addition, we analyze the security solutions provided by a novel class of mobile agents. The agents endowed with a novel proposed architecture represent the novel class of mobile agents. The proposed mobile agent architecture represents a combination of mobile and static agent paradigms.*
Keywords*: mobile agent, agent architecture, mobile agent security*

1. Security Issues of the Mobile Agents

Mobile agents exhibit new characteristics such as mobility and autonomy, while retaining the features of processes in conventional parallel and distributed computing systems. Different from the conventional distributed programs, which are bounded to the nodes in the network where they operate, the mobile agents can migrate autonomously from one node to another during their operation. They can choose the migration route according to the runtime condition, providing a model of computation in which the use of the resources and the integration of the services are different from the conventional model based on client/ server architecture.

The paper [11] lists some of the major challenges of the mobile agents like:

- *authentication of the agents.* How do you ensure that an agent is who it says it is, and that it is representing who it claims to be representing?
- *secrecy of the agents*. How do you ensure that an agent maintains they owner's privacy? How do you ensure that someone else does not read your personal agent and use it for his own gains? How do ensure that an agent is not killed?
- *security of the agents*. How do you protect the agents in the network?

The security issues in mobile multiagent systems [3] consist in:

- *protecting the hosts against the mobile agents*. Mobile agents may attack the public services offered by the hosts, may modify the activity of other agents, may transmit viruses, trojan horses and worms;
- *protecting the agents against each other*. Mobile agents may modify other mobile agents;
- *protecting the agents against the executing hosts*. The hosts may modify the mobile agents;
- *protecting the agents against different network sources*. Network sources may modify the mobile agents.

The security issues enumerated before may be caused by errors in the executing environments or by intentional misbehavior. Both of these cases must be prevented, or at least, detected. It must be guaranteed that the code of an agent is executed according to the specification of the programming language and that of the middleware architecture. It is important to ensure that the data carried by an agent is not modified.

2. Security Solutions in Mobile Multiagent Systems

The security solutions in *mobile multiagent systems* must include the protection of the hosts and the protection of the mobile agents. In the following, we will analyze security solutions related to the protection of the hosts against the mobile agents and the protection of the mobile agents against the hosts. The security of a host consists in protecting the host from the visitor mobile agents. A mobile agent at a host uses different resources and services offered by the host. This way the agent may execute different destructive actions. A malicious or erroneous mobile agent can destroy the host on which it is running if the host is inefficiently protected.

For a host's protection techniques such as the following ones can be used [3]:

- *identification of the mobile agents using digital signatures*. Depending on the identity of an agent the host can accept or reject a mobile agent;
- *limitation of the services offered by the host using hierarchies of trust*. Depending on the identity of an agent the host may restrict the use of some resources to the mobile agent;
- *using cryptography techniques*. Some of the private informations of the hosts may be codified using a codifying algorithm unknown to the visitor mobile agents.

A security issue consists in how to make sure that an agent arriving to a host it has not been modified during its operation. Such modifications are independent from the owner of the agent. Such

agents could destroy the host's security, if they go undetected. Before the execution of a mobile agent, the host must authenticate the agent and its owner. Based on this information, different sets of authorizations and resource limitations can be applied to restrict the actions executed by the agent. In this way the access to files, memory or processor usage can also be constrained. Some systems use two classes of resource limitations. One class concerns the amount of given resources that the agent may consume during its whole execution, while the other class consists in restricting resource usage to a given amount per a specified unit of time.

Threats against the mobile agents can be classified as those performed during the agents' migration and as those performed by the executing hosts. The hardest among all security issues raised by mobile agents consists in protecting the agents against the hosts on which they are executing [9, 7, 1]. Once an agent has arrived at a malicious host, little can be done to stop the host from treating the agent as it likes. An efficient protection mechanism of a mobile agent against a host should provide code and execution integrity (code privacy), solutions for computing with secrets (data privacy) and prevention from denial of service attacks against agents. Prevention from an unauthorized data disclosure is made harder by the fact that a set of hosts may collaborate in the fraud.

An example is often used to illustrate how a malicious host can benefit from attacking a mobile agent is the shopping agent [1] specialized in finding the best airfare for a flight with a particular route. Various requirements are given to the agent, such as departure and destination. The agent is sent out to find the cheapest ticket before committing to a particular purchase. The agent will visit some airlines and query their databases before committing to a purchase and reporting back to the owner. A malicious visited host can interfere with the agent execution in several ways.

For example, a malicious host could try to:

- simply terminate the agent to ensure that no other competitor gets the business either;
- erase all information previously collected by the agent;
- change the agent's route so that airlines with more favorable offers are not visited.

Mobile agents can be protected by introducing trusted nodes into the infrastructure to which mobile agents can migrate when required, important information can be prevented from being sent to untrusted hosts, and certain misbehaviors of malicious hosts can be traced [1]. The platform from where a mobile agent is first launched has to be a trusted node.

Neither software system can provide a complete solution in the protection of the mobile agents [3]. The executing host must have deep knowledge about the agent's code and the data it uses. A host has possibility to interpret a visitor mobile agent, to extract the data it uses. Operators of agent platforms

may guarantee, via contractual agreements, to operate their environments securely and not to violate the privacy or the integrity of the visitor mobile agents, their data, and their computation. To prove that such an agreement has been broken might be a non-trivial task [1]. Sanctions can be initiated against the responsible hosts and their owners. It is assumed that it is in the interest of the host's owner to provide its computational service. Proved frauds can result in agents avoiding the host in the future, thus harming the owner's interests. This is expected to have a preventing effect.

Trusted hardware platforms may provide security to the mobile agents [3]. This solution requires the presence of special hardware components whose internal architecture is unknown to the public. An argument against this approach is that it is usually relatively easy to remove the fragment of the code that checks the presence of the special hardware. The main disadvantage of this solution is that it reduces the number of possible execution environments.

Techniques based on *code obfuscation* [3] may secure the mobile agents. A technique based on code obfuscation proposes an extensible set of transformations to be applied to the mobile agent code. These transformations produce code harder to read, but with identical results. This approach usually results in code with lower efficiency.

Encryption proposes that the agent’s code and information are encrypted by a secret key [15]. Only a small window containing the actual point of execution would be decrypted on the fly, making possible its execution. As execution proceeds the passed code fragments are encrypted again. Dynamic decryption and encryption decrease the efficiency of the agents.

Clueless agents are proposed as a solution to prevent code and data disclosure [8]. The main idea is that the observer may have access to the information, but it should not be able to understand for what it is. This idea is implemented through conditional code. The execution of a conditional code is triggered by a special event in the environment. The observer may be aware of what algorithm the agent is intending to perform, but it doesn’t know on what input. The agent does not know the exact condition, it only scans a specific channel codes the received events by a one-way hash function and compares them with a stored constant. If the two results match, the condition is satisfied. As the pattern is only stored in encrypted form, the exact trigger is difficult to deduce. This trick depends on the existence of the one-way hash functions.

Execution tracing [10] has been proposed for detecting unauthorized modifications of an agent through the recording of the agent’s execution at each visited host. Each host is required to create and retain a log of the operations performed by the agent while executing on the host. A drawback of this approach consists in the size of all the created logs by the hosts (each host may execute a large number of mobile agents). Another drawback consists in the difficulty of the logs management. *Partial result*

authentication codes are described in the paper [14]. The idea is to protect the authenticity of an agent state or partial result when the agent is running on a host. Partial result authentication codes can be generated using symmetric cryptographic algorithms. An agent is equipped with a number of encryption keys. Every time when the agent migrates from a host, the agent's state or some other result is processed using one of the keys, producing a *message authentication code.* The key that has been used is then disposed of before the agent migrates. The *partial result authentication codes* can be verified at a later point to identify certain types of tampering.

3. A Novel Mobile Agent Architecture

An *agent architecture* is essentially a map of the internals of an agent, its data structures, the operations that may be performed on these data structures, and the control flow between these data structures [12]. In the paper [5] a novel intelligent mobile agent architecture is proposed. There the proposed mobile agents' knowledge bases and the proposed mobile agents' operation are analyzed. Some introductory elements of the proposed architecture are described in the paper [6]. The paper [4] analyzes different aspects of the multiagent systems formed by the proposed agents. There is also analyzed the possibility to create intelligent mobile multiagent systems with the proposed mobile agents. A mobile agent *MOBILE AGENT* endowed with the proposed architecture is composed of two parts: a static part *STATIC PART* and a mobile part *MOBILE PART*.

MOBILE AGENT = STATIC PART + MOBILE PART.

Figure 1 illustrates the proposed mobile agent architecture. In Figure 1 the following notations are used: *Static* represents the static part of the mobile agent, $Mobile_1$, $Mobile_2$, … $Mobile_n$ represents the mobile part of the mobile agent.

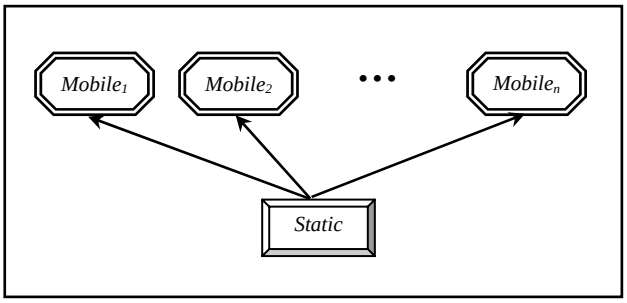

Fig.1. *The proposed mobile agent architecture*

The static part of a mobile agent consists in a static subagent. The notion of *subagent* is defined in the paper [2]. The static subagent is responsible for the overtaking of the problems from the user. The mobile part is composed of a variable number of mobile subagents. The static subagent creates the mobile subagents. The mobile subagents have all the proprieties of the known mobile agents. The mobile subagents are responsible for the problems' solving. The body of a mobile subagent contains the description of the agent's code (the specializations used in the problems' solving) and may contain different data used in the problems' solving. Denote *specialization* the method that describes the solving of a problem. The *algorithm intelligent mobile agent* describes briefly a proposed mobile agent's operation at a problem solving cycle.

Algorithm - Intelligent Mobile Agent

Step 1

The static subagent overtakes the problems transmitted by the user.

Step 2

The static subagent establishes the overtaken problems' solving by mobile subagents. The established mobile subagents are launched for problems solving in the network.

Step 3

The mobile subagents solve the overtaken problems. The obtained results by the mobile subagents are transmitted to the static subagent.

Step 4

The static subagent processes the received results if it is necessary, the solutions obtained are transmitted to the user that has sent the problems.

End.

3.1. Communication Capabilities of the Proposed Mobile Agents

The communication between mobile agents at different hosts is difficult. A mobile agent migrates from host to host the host where there is a mobile agent at a moment of time cannot be estimated. Different difficulties in the communication between the mobile agents are analyzed in the paper [13]. In the following, we will analyze the communication capabilities of the agents endowed with the proposed architecture.

The subagents of the same mobile agent can communicate [4]. We propose the use of the *KQML* communication language [12]. The address of the static subagent remains unchanged during the mobile

agent's life cycle. The mobile subagents' addresses are changing during their operation (the mobile subagents migrate from host to host). Each mobile subagent at a host can communicate with the static subagent.

Formally a message *msg* transmission from the mobile subagent *MS* to the static subagent *SS* can be described as follows:

$$direct\ (msg;\ MS,\ SS).$$

Each mobile subagent transmits a message that contains the current host's address to the static subagent when it arrives to a host. Each mobile subagent communicates when it leaves the current host. In this way, the static subagent can communicate with each mobile subagent that is at a host.

Formally a message *msg* transmission from the static subagent *SS* to the mobile subagent *MS* can be described as follows:

$$direct\ (msg;\ SS,\ MS).$$

When a mobile subagent wants to communicate with another mobile subagent at another host, it transmits the message to the static subagent, the static subagent forwards the message to the target mobile subagent when the target mobile subagent is at a host.

Formally the message *msg* transmission from a mobile subagent MS_i to a mobile subagent MS_j using as mediator the static subagent *SS* can be described as follows:

$$facilitator\ (msg;\ MS_i,\ SS,\ MS_j).$$

A mobile subagent can communicate with another mobile subagent even if the target mobile subagent migrates in the network in the transmission time. The source mobile subagent transmits the message to the static subagent, the static subagent transmits the message to the target mobile subagent when this agent arrives to a host.

3.2. Advantages of the Proposed Mobile Agents

In the following, we analyze the advantages of the proposed mobile agents which illustrate their increased security as opposed to the traditionally used mobile agents.

The static subagent (it is a static agent) of a mobile agent may be endowed with any security solution which can be used by a static agent in the protection against other agents.

A mobile agent may overtake for solving a large number of problems at a problem solving cycle. The overtaken problems are solved by mobile subagents created by the static subagent. The mobile subagents solve the problems at different hosts. The mobile agent's body that migrates in the network consists in the mobile subagents that solve the problems. A network source or a host can access only

some of the mobile subagents (the mobile subagents are distributed in the network), it doesn't have a global view about all the operating mobile subagents (it doesn't know all the data and code of the mobile subagents used in the problems solving). The only one who knows where the mobile subagents are, is the static subagent (each mobile agent announces the static subagent when it arrives to and lives a host). If some mobile subagents are lost the rest of the mobile subagents are also able to solve the overtaken problems. The mobile agent may use different mobile subagents at different problems solving cycles. The distributed operating manner of the proposed mobile agents represents a security solution against the malicious network sources and hosts.

When a mobile subagent does not need a specialization in the following problems solving then the subagent can leave the specialization (it is not necessary for a mobile subagent to return to the static subagent). This way the mobile subagent launched to the next host is smaller, contains less data and code which can be used or modified by different malicious network sources and hosts. The transmitted data and code quantity in the mobile part of a mobile agent decrease in time during a problem solving cycle. This way, hosts and network sources receive less information about the mobile part of the mobile agent in time.

A static subagent can verify if their mobile subagents are modified in they journey. When a mobile subagent arrives at a host it transmits the host's address and its checksum to the static subagent. This way the static subagent can verify if the mobile subagent is modified in the network. When a mobile subagent leaves a host it transmits to the static subagent the host's address and its checksum. This way the static subagent can verify if the mobile subagent is launched to the next host with modifications (the current host is malicious). The static subagent has each mobile subagent's itinerary, this way it can verify if each mobile subagent visits all the hosts specified in its itinerary (each mobile subagent announces when it arrives to a host and when it leaves a host). In the case of a stolen or modified mobile subagent the static subagent can take different measures. As an example of measure, the static subagent can announce the hosts to not accept the mobile subagent and it may retransmit the mobile subagent with a different identifier.

Another security solution of a mobile subagent against the malicious hosts and network sources consist in encrypting each specialization of the mobile subagent using a different encryption algorithm. When a mobile subagent arrives to a host it transmits the host's address to the static subagent. The static subagent transmits to the host how to decrypt the specializations and data that must be used in the problems solving at the host. This way only the specializations and data necessary in the problems solving are understandable for the host.

4. Conclusions

The main disadvantage of the traditionally used mobile agents is their limited security in the network and at the hosts. Mobile agent systems perform quite well on secure networks, but they need more autonomy and intelligence to react in more risky and changing environments. In the paper [5] a novel mobile agent architecture is proposed. The novelty of the proposed architecture consists in the combination of the static and mobile agent paradigms. In the paper [4] the possibility to create intelligent multiagent systems with agents endowed with the proposed mobile agent architecture is analyzed. In this paper we have analyzed the security of the mobile agents endowed with the proposed mobile agent architecture [5, 4]. The mobile agent architecture offers new security solutions in the protection of the mobile agents against the malicious network resources and hosts.

References

[1]. Borselius, N., *Mobile agent security*, Electronics & Communication Engineering Journal, 14(5), 2002, pp. 211-218.

[2]. Ferber, J., *Multi-Agent Systems, An Introduction to Distributed Artificial Intelligence*, Addison Wesley, 1999.

[3]. Gulyás, L., Kovács, L., Micsik, A., Pataki, B., Zsámboki, I., *An Overview of Mobile Software Systems, Department of Distributed Systems*, Computer and Automation Research Institute of the Hungarian Academy of Sciences, MTA SZTAKI Technical Report TR 2000-1, 2001.

[4]. Iantovics, B., *A New Intelligent Mobile Multiagent System*, IEEE International Workshop on Soft Computing Applications (IEEE SOFA 2005), Szeged-Hungary and Arad-Romania, August 2005. (accepted paper).

[5]. Iantovics, B., *A Novel Mobile Agent Architecture*, International Conference on Theory and Application of Mathematics and Informatics (ICTAMI 2005), Alba county, Romania, September 2005. (accepted paper)

[6]. Iantovics, B., *Intelligent Mobile Agents*, Proceedings of the Symposium "Zilele Academice Clujene", Babes-Bolyai University of Cluj-Napoca, 2004, pp. 67-74.

[7]. Nwana, H.S., *Software Agents: An Overview*, *Knowledge Engineering Review*, Cambridge University Press, 11(3), October/November 1996, pp. 205-244.

[8]. Riordan, J., Schneier, B., *Environmental Key Generation Towards Clueless Agents*, G. Vigna (Ed.), Mobile Agents and Security, Springer-Verlag, 1998, pp. 15-24.

[9]. Sander, T., Tschudin, C.F., *Protecting Mobile Agents Against Malicious Hosts*, G. Vigna (Ed.), Mobile Agents and Security, Springer-Verlag, 1998, pp. 44-60.

[10]. Vigna, G., *Protecting mobile agents through tracing*, Proceedings of the Third ECOOP Workshop on Operating System support for Mobile Object Systems, Finland, June 1997, pp. 137-153.

[11]. Wayner, P., *Free Agents*, Byte, March 1995, pp. 105-114.

[12]. Weiss, G., (Ed.), *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, MIT Press, Cambridge Massachusetts London, 2000.

[13]. Xu, H., *A model-based approach for development of multi-agent software systems*, Ph.D thesis, the Graduate College of the University of Illinois at Chicago, 2003.

[14]. Yee, B., *A sanctuary for mobile agents*, J. Vitek and C. Jensen (Eds.), Secure Internet Programming, New York, Springer-Verlag, LNCS 1603, 1999, 261–274.

[15]. Young, A., Yung, M., *Encryption Tools for Mobile Agents: Sliding Encryption*, E. Biham (Ed.): Fast Software Encryption. Proceedings of the 4th International Workshop (FSE 1997), Haifa, Israel, Springer Verlag, LNCS 1267, 1997.