Security Mechanisms for Using Mobile Agents in Electronic Commerce

Paulo Jorge Marques, Luís Moura Silva, João Gabriel Silva
Departamento de Engenharia Informática, Universidade de Coimbra, Portugal

{pmarques, luis, jgabriel}@dei.uc.pt

Abstract

In order for mobile agents to be accepted as a basic technology for enabling electronic commerce, proper security mechanisms must be developed. Hosts must be protected from malicious agents, agents must be protected from other agents and also agents must be protected from malicious hosts. For solving the first three problems, existing technology from operating systems and distributed systems research can be used. The last problem is new and specific to the mobile agent paradigm and it is much harder to solve. Due to this problem, many say that mobile agents are not ready for the e-commerce systems.

In this paper we discuss the security requirements of mobile agents in the context of electronic commerce and analyze how these requirements can be met. We show that, because of the characteristics of e-commerce systems, the security requirements of the agents and their users can be assured in real and open environments as the Internet.

1 Introduction

Mobile agents are one of the most prominent technologies believed to be playing an important role on future electronic commerce (e-commerce) systems. Besides providing a very flexible approach for information gathering on prices and assets available from the several company servers they visit, they can effectively take over the different aspects of the electronic commercial transaction, from price settlement to paying and delivery of the goods purchased.

Adopting the nomenclature of Maes' [1], we identify the following stages where this technology can be especially important:

- Product brokering
- Merchant brokering
- Negotiation
- Payment and Delivery

Product brokering involves gathering information from several merchants about a certain product that the user is looking for acquiring. Mobile agents can be used to collect offers from several hosts representing stores. These offers would be made in response of a query performed by the agent, which refers its owner's whishes ("I want to buy a PC for less than $2000").

Merchant brokering consists in evaluating a set of alternatives, discovered in the previous stage, in order to decide where to make the purchase. Mobile agents can be used to autonomously decide where to make the purchase.

In the negotiation stage, the final terms of the transaction are set. Agents can be used to mediate this part of the transaction.

Finally, in the payment and delivery stage, the goods are delivered against currency (or its electronic equivalent). The mobile agents can be used to actually pay for assets being bought and to collect a receipt as proof.

Mobile agents are especially interesting when considering all the previous activities as a whole. A mobile agent can autonomously take care of all the steps needed for carrying out the deal, without ever bothering its owner. Alternatively the process can be semi-autonomous requiring the user to validate the choices of the agent before the actual commercial transaction is performed.

Although all the advantages that mobile agents can bring to e-commerce frameworks, the success or failure of this paradigm is directly connected to the question on whether proper security mechanisms can be effectively implemented and used.

Security in mobile agent systems can be analyzed in four different perspectives [2]:

- Protecting hosts from access by unauthorized parties.
- Protecting hosts from attacks of malicious agents.
- Protecting agents from attacks of other agents.
- Protecting agents from attacks of malicious hosts.

For the first three items, ideas from the distributed computing and operating systems research can be used [3] [4]. The problems that exist in security for mobile agents are basically the same that exists on these areas. Securing agents against attacks from malicious hosts is a new and difficult problem that is specific to mobile agent systems [3].

At this moment, there is no general solution on the issue of protecting mobile agents from attacks of malicious hosts. At first sight this can be seen as a major limitation on the usefulness of these agents for e-commerce. Agents can be the subject of attacks by hosts which can make them take higher offers than the best available, steal confidential information or tamper with their operation.

We believe that mobile agents are an important technology and that its security requirements for electronic commerce can be met today. Our opinion is that most of the research being done today goes too far by assuming that no host in the network can be trusted and by approaching the problem from a single agent viewpoint. This is not realistic. Security of agents for e-commerce must be viewed in a realistic and pragmatic way:

- Some hosts in the network can be trusted.
- More than one agent can be used in order to build a secure mobile agent environment.

Based on these assumptions we are working on a solution to solve the problem referred before.

Throughout this paper we analyze security of mobile agent systems from the particular perspective of electronic commerce frameworks. The security requirements of hosts and agents will be discussed and approaches on how to guarantee them will be presented. We will show that by exploiting the particular requirements of e-commerce systems, it is possible to guarantee the safety of the agents that are roaming the network.

The rest of this paper is organized as follows. Section 2 analyses security requirements of e-commerce systems based on mobile agents and how these requirements can be guaranteed. Section 3 presents the conclusions of this paper. Finally, Section 4 presents future work to be developed.

2 E-Commerce with Mobile Agents

2.1 Mobile Agent E-Commerce Framework

E-commerce agents normally travel from host to host, inquiring about the cost of a specific product. After visiting several hosts, the agent must make a decision on which one to buy. This decision takes into account a set of requirements defined by its owner (e.g. price, quality, warranty and refund guaranties). After making the decision, the agent travels back to the selected host and then buys the product. This corresponds to the four stages previously identified: product brokering, merchant brokering, negotiation and payment and delivery.

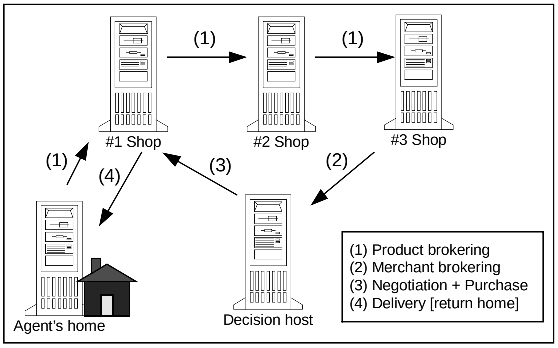

Figure 1 – Agent flow of migration.

Figure 1 shows the previous scenario, identifying the agent's flow of migration with the different phases taking place. During product brokering the agent travels through the network collecting offers from stores. When it decides to make the decision on where to buy, it migrates to a decision host where the merchant brokering stage takes place. Having reached a conclusion on where to buy, it then migrates to the designated host where the negotiation and purchase stages take place. Finally, it migrates back home taking the goods with it, whenever possible. A receipt of the bought made is also issued to the agent. The agent only carries the goods with it if they are in an electronic form. If this does not happen, the goods have to be delivered to the user in some other way.

Some variations on this model are possible. The most important one consists in the agent migrating home after the merchant-brokering phase so that its owner examines the decisions taken. This is perfectly natural since most people prefer to have control over what they buy than to allow a machine to make a decision for them. Although this is still a simplified framework for using mobile agents in e-commerce, it is still useful for analyzing what are the basic security needs that have to be met and how they can be provided.

2.2 Protecting Hosts from Access by Unauthorized Parties

During all the four identified stages it is vital that the hosts are protected from being accessed by unauthorized parties. There are many situations where this can happen. A host may not want an agent from a competing company performing queries about its prices and conditions. Also, a host may only serve certain specific types of clients like gross retailers. A general consumer may try to use the host in order to cut a better deal, although he has no right to do so.

Thus, securing hosts against access from unauthorized parties implies the provision of mechanisms to allow proper identification of agent owners. Some traditional security mechanisms can be used:

- Use of public key cryptography for signing the agent code. This allows the owner of the agent and also its manufacturer to be properly identified.
- Use of public key cryptography for host authentication.
- Use of secure communication channels for agent transmission (e.g., the Secure Socket Layer Protocol – SSL).
- Use of secure hashed and encrypted time stamping to avoid reply and masquerading attacks.

There is one aspect of security in mobile agent systems that is quite different from what happens in the client/server paradigm. There is a very large conceptual difference between who writes the mobile agent code and who uses it. If the code is signed with the private key of the user, making it its principal, then the code writer can plan and execute attacks using the identity of the user. If the programmer signs the code, the user will not be accounted for when the program is being executed. Signing the code with both keys is really necessary.

In order to secure hosts against unauthorized parties and account users for used resources, there must exist a public key distribution directory that hosts can query.

2.3 Protecting Hosts from Attacks of Malicious Agents

Although the user and programmer of an agent can be accounted for, considering the Internet and large anonymous distributed systems, it is quite simple to obtain an untraceable anonymous identity. The Internet is especially propitious for that. Users like to maintain their anonymity unless they effectively have to reveal it. Picturing a global electronic commerce framework, users should probably prefer to make queries about prices and assets anonymously and only reveal their identities at the places where they actually make the acquisitions.

Revealing the user identity to every host queried is not so compelling since everyone would be able to learn what the user is looking for, and could even start to maintain information about the user's shopping patterns.

Even though users with an anonymous identity may have limited executing permissions on hosts, these identities can serve as Trojan horses for agents to get executing inside public hosted platforms.

It is important that hosts can protect themselves from executing agents and that agents are accounted for used resources. Effectively protecting hosts from agents (malicious or not) is a security requirement necessary for all the identified stages of e-commerce model being used. In order to protect hosts from agents, some mechanisms can be used:

- Use of secure languages that isolate system resources and address spaces that agents can access.
- Use of proxies that encapsulate requested resources by agents.
- Use of resource control mechanisms that restrict the agents use of memory, CPU, disk space, threads, network resources and others.
- Use of audit trails for accountability on the actions of agents.
- Use of access control lists (ACLs) mapping what resources each agent can access.
- Use of accounting and contract mechanisms for establishing different accesses in ACLs.

The question of protecting hosts against malicious agents is even larger than this. It is also necessary to protect hosts against denial-of-service attacks. Possible attacks include other servers trying to start innumerous agents on the current machine, or agents trying to clone themselves too many times. These are special cases of resource management that must be properly addressed. The system must limit the resources that each agent is able to use and also monitor how these resources are effectively used.

2.4 Protecting Agents from Attacks of Other Agents

When agents communicate with other agents, sharing objects and resources, it is necessary to prevent malicious agents from attacking others. Also, if several independent agents are executing, each agent must be able to conduct its actions without being harassed by others.

Agents will have to communicate with other agents during all the stages of the transaction. On the presented model of e-commerce, agents will typically communicate with stationary agents that represent the stores. Nevertheless, in other models, agents may communicate with agents from these stores in a neutral ground. Neutral grounds are independent hosts that accept agents from users and also agents representing stores, providing a safe place where they can communicate. Security methods for protecting agents from other agents depend directly on how the coordination between agents is designed.

Coordination models can be viewed in a two dimensional axis, according to their temporal and spatial coupling [5]. Table 1 summarizes the existing agent coordination models.

	Temporally Coupled	Temporally Uncoupled
Spatially Coupled	Direct	Black-board
Spatially Uncoupled	Meeting	Tuple-space

Table 1 – Spatial and temporal coupling in agent coordination models.

In a spatially and temporally coupled model, agents must name each other explicitly and synchronize in order to communicate. Even inside this model, several communication models are possible. Two agents in the same platform can share objects between them, or exchange messages. An explicit protocol must exist between the two, either peer-to-peer or client-server, that allows meaningful communication to take place. Some mechanisms are immediately relevant when considering security on these systems:

- Use of proxies to shield agents from other agents (if object sharing is used).
- Means to stop object sharing, if an agent feels it is necessary.
- Use of "ignore lists" to allow an agent to refuse messages from other agents.
- The possibility of warning the system if an agent attempts an attack.
- Platform mechanisms for monitoring agents and agents' behavior.
- Maintaining lists of agents and agents' principals behavioral patterns (rap sheets).

In spatially coupled and temporally uncoupled models (blackboard based), agents must share a common namespace but do not have to synchronize temporally. Messages are stored by the system in a common named repository. When considering security on those systems, the main issue is properly identifying and protecting messages from a group of communicating agents. Agents must be identified and their permissions to store, read, update and delete messages correctly asserted. Some mechanisms can be used:

- Signing and encryption mechanisms can be used to prevent unauthorized accesses (if all agents can check and try to read all the messages).
- Private conversations between agents can be protected by sparse space tokens and/or the agent's credentials.
- Incremental back off protection techniques can be used in case of successive access failures.

In spatially uncoupled, temporally coupled models, agents join meeting points where interaction can take place. After starting a meeting agents communicate locally. The approach is not spatially coupled because agents do not have to name the involved parties and only join or leave meetings. The security concerns and approaches discussed for the direct model still apply. After joining a meeting, agents still have to have contact in some way (messages, object sharing, etc). Nevertheless, in this model, concerns on how agents join and leave meetings are still an issue:

- Agent authentication and access lists can be used to enforce that agents can only join the meetings they are authorized to.
- Agents in a meeting can vote whether another agent can join the meeting or not.
- Two agents should be able to meet and to keep the meeting private from interaction with other agents.
- Agents should be able to inform the platform of misbehaving agents that should be kept out or expelled from a meeting.
- The platform should monitor agents to detect denial-of-service attacks by flooding other agents in a meeting (in message-oriented meetings).

In spatially and temporally uncoupled models (e.g. tuple-space), agents access a local blackboard using an associative mechanism. Information is organized in tuples and retrieved by a pattern matching. Agents do not have to name each other explicitly or have to synchronize temporally to communicate. Thus the model is fully uncoupled. In typical systems there is a public tuple space which all agents can use to communicate, not requiring any special kind of protecting. Agents may also create private tuple spaces, only accessible by some other agents. Depending on how the private tuple spaces are created and accessed, the coordination model may become blackboard like or meeting-like. Depending on the blackboard and meeting characteristics of the private tuple spaces, the security issues and mechanisms previously discussed for the other models still apply. An alternative to private tuple spaces is to use public tuple spaces with protected tuples: some agents can only access certain tuples. Security in shared tuples-spaces can benefit from some mechanisms:

- Tuples can have access control lists specifying which agents can access, modify or delete the tuples.
- The platforms should monitor accesses to the tuple spaces to prevent denial-of-service attacks by reading or writing too quickly in them.
- Depending on the credentials of agents, they may or may not be able to create private tuple spaces.

E-commerce systems based on mobile agents must provide the means for enabling secure communication and coordination between agents. Securing agents against agent attacks is that it is highly dependent on the coordination model that is used. If several security models are used, different policies must be analyzed and carefully implemented.

2.5 Protecting Agents from Attacks of Malicious Hosts

When a host receives a mobile agent for execution, it is possible that the host spies or interferes with the agent's execution. Let us consider the case where an agent is shopping for flowers. A malicious host may try to read and modify offers from other hosts or even change the flow of execution of the agent in order to force it to take an offer being proposed.

Because agents execute on a target platform and have to expose both their code and state, it does not seem possible to fully protect agents from malicious hosts [3]. Nevertheless, due to the special characteristics of e-commerce systems, we believe it is possible to guarantee some safety for the agents.

When we consider the security requirements in terms of protecting agents from malicious hosts, different needs exist during the different stages of the electronic transaction. Table 2 summarizes the requirements that must be met during each one of the stages of the transaction. In this table, the stages are grouped according to the flow of migration of the agent.

During the brokering stage, the agent must be able to collect individual offers from hosts according to some given criteria defined by its owner. Each host must not be able to tamper with offers made by other hosts or be able to read them. This can be accomplished by using Karjoth's protocols [6]. These protocols enable an agent to collect individual offers from hosts in a secure way. Each offer is encrypted and signed by each host. Also, a secure collision free hash function is used to generate tokens that depend on the previously collect offer and on the next hop of the agent. These tokens are stored along with the offers. Offers are not readable by anyone except the agent's owner, and the tokens can be used to verify if no offers were added or subtracted from the agent.

Stage	Requirements
Product brokering	Offers made by a host should not be readable by other hosts.
	It must not be possible for hosts to modify offers without being detected.
	Hosts must no be able to delete of add false offers in the name of others.
	Hosts must not be able to disclose the decision-making logic of agents on how acquisitions are made.
	Hosts must not be able to read sensitive information maintained by the agent.
Merchant brokering	The state of the agent and data transported by it must not be spied or altered by the host.
	The code of the agent must not be spied or altered by the host.
	The flow of execution of the agent should not be spied or altered while it is executing.
Negotiation & Purchase	The information obtained in the previous stages should not be modifiable without being detected.
	The agent should be able to give selected sensitive information to the host in order to make the purchase, being assured that the information is not disclosed to third parties.
	The owner of the agent actually gets the asset bought and a receipt is issued as proof of the purchase.

Table 2 – Security requirements in each stage of an electronic transaction.

Agents can carry encrypted methods and data for which a private key is needed. These methods correspond to the decision-making logic of the agent and should never be executed on hosts representing stores. Thus, the information concerning the user (e.g., credit-card number and social security number) is not disclosed to untrusted parties. Also, the decision-making logic of the agent is protected.

For the merchant-brokering phase, we take Farmer's perspective [7]: *"An agent's critical decisions should be made on neutral (trusted) hosts"*. Since the agent is making a decision on where to buy, this decision must be made on a known, trusted server to the user.

It is feasible that trusted servers do exist in open network environments. Although most servers are not trusted, the user will typically have a neutral trusted host where to execute code. If mobile agents indeed become ubiquitous, companies will provide (for a certain fee) these sanctuaries where agents can safely execute. This already happens in services like public key distribution directories.

When the agent first migrates to begin the product-brokering phase, another agent carrying the private key of the first one migrates to a pre-arranged trusted server. This is the server where the merchant-brokering phase will take place. When the agent that is performing the product brokering arrives, the offers gathered during that phase can be decrypted. The methods that the agent uses to decide on where to make the purchase are also decrypted. After this step, it can decide on where to make the acquisition.

At this point, the agent can evaluate where it is the best place to make the purchase. After making the decision, it can then migrate to the designated host with the necessary sensitive information needed to make the acquisition. The final terms of the transaction and the actual acquisition can then take place. Also, the agent that was left at the merchant-brokering host can be used for accounting purposes and for reliability of the transaction.

We are currently developing a comprehensive solution to solve the problem of malicious hosts based on the previous discussed ideas. Nevertheless, the complete discussion of the approach is out of the scope of this paper.

3 Conclusion

In this paper we have presented a brief overview on mobile agent security for e-commerce. We analyzed the security requirements of mobile agents and how these requirements can be met. Most importantly, we have discussed as the requirements depend on the phase of the commercial transaction being made, and how these different requirements can be guaranteed.

Currently, the most interesting investigation topics on mobile agent security concerns host-from-agent and agent-from-host protection. The first raises problems on how to implement resource control, management and accounting in platforms. The last is taken to be a hard research topic, with almost no general complete solution in the horizon.

Nevertheless, due to the special characteristics of e-commerce systems, we believe that it is possible to guarantee the security of the agents that roam the network and also the security of the hosts. Hosts can be protected from agents, agents can be protected from other agents, and agents can also be protected from hosts. Thus, e-commerce systems based on mobile agents can be provided with the basic security mechanisms and actually be deployed in real word environments.

4 Future Work

At this moment, extensive experiments with the presented security model must be performed. We intend to implement a secure electronic commerce framework based on the described model. This framework will allow us to further explorer the strengths, weaknesses and usefulness of mobile agents for e-commerce. We also intend to explore the use of fault-tolerance techniques as persistence, voting and replication in creating secure environments for e-commerce based on mobile agents.

Acknowledgements

This investigation was partially supported by the Portuguese Research Agency FCT, through the program PRAXIS XXI (scholarship number DB/18353/98).

References

[1] P. Maes, R. Guttman, and A. Moukas, "Agents that Buy and Sell," in *Communications of the ACM*, vol. 42, 1999, pp. 81-91.

[2] F. Hohl, "A Model of Attacks of Malicious Hosts Against Mobile Agents," presented at 4th Workshop on Mobile Object Systems: Secure Internet Mobile Computations, France, 1998.

[3] W. Farmer, J. Guttman, and V. Swarup, "Security for Mobile Agents: Issues and Requirements," presented at National Information Systems Security Conference, 1996.

[4] M. Greenberg, J. Byington, and D. Harper, "Mobile Agents and Security," *IEEE Communications Magazine*, 1998.

[5] G. Cabri, L. Leonardi, and F. Zamboneli, "Reactive Tuple Spaces for Mobile Agent Coordination," presented at Mobile Agents - MA99, Stuttgart, Germany, 1998.

[6] G. Karjoth, N. Asokan, and C. Gülcü, "Protecting the Computation Results of Free-roaming Agents," presented at Second International Workshop on Mobile Agents (MA' 98), Stuttgart, Germany, 1998.

[7] W. Farmer, J. Guttman, and V. Swarup, "Security for Mobile Agents: Autentication and State Appraisal," presented at Computer Security - ESORICS'96, 1996.