Multi-agent Systems Security in Collaborative E-commerce

Shakeel Iqbal
Ijaz A. Qureshi, Dr.
Khola Ilyas
Robina Yasmin, Dr.
Multi-Agent Systems Security in Collaborative E-Commerce

Dr. Ijaz A. Qureshi
Iqra University Islamabad, Pakistan
ijaz@iqraisb.edu.pk

Khola Ilyas
Iqra University Islamabad, Pakistan
Khola_illy@hotmail.com

Shakeel Iqbal
Iqra University Islamabad, Pakistan
siqbal@iqraisb.edu.pk

Dr. Robina Yasmin
Iqra University Islamabad, Pakistan
robina@iqraisb.edu.pk

ABSTRACT
Multi-agent systems (MAS) have been the focus of researchers because of agents’ habitual, hands-on, and active problem solving behavior. These systems provide and receive useful services and information in areas like e-commerce, healthcare and banking etc. However security of these systems is a major concern which could lead to many operational problems. In this paper security in multi-agent systems in a collaborative environment has been studied with an emphasis on stakeholders of the multi-agent system and the relevant security threats and opportunities.

Key words: Collaborative e-commerce, Multi-agent system, Security, Multi agent stakeholders.

1. INTRODUCTION
Multi-agent systems characterize virtual communities where software agents meet and interact acting in the interest of people or organizations various reasons like exchanging goods, combining services, etc. and in various ways. Taking in to consideration the need to improve online shopping with the current Web growth, support for more collaboration between sales persons and among customers would be an interesting way to enhance the shopping experience of people. The industry has acknowledged the concept of providing e-community with the web shopping experience bringing online shopping closer to the real shopping environment, and is now seriously focusing on collaborative e-commerce.

When deployed in an open e-commerce environments multi-agent systems face testing trust and confidentiality issues at various levels. These have been discussed in this paper along with the possible solutions.

2. COLLABORATIVE E-COMMERCE
Collaborative commerce is realized by the interactions among agents in virtual e-communities. This model provides an ideal customer experience, better understanding of customer behavior to
manufacturer and increased visibility into channel activities and performance. It provides producer with a commercial edge in terms of product arrangement, wide customer base and feedback.

Success of these systems depends on quality of shopping experience for customers. Manufacturer- customer and inter -customer communication is an important aspect of web commerce. This validated by a survey shows that 90% of communication takes place between customers in shopping malls [1]. This has an important bearing on design and application of multi-agent systems.

Online shopping with inter- customer interaction helps in forming online shopping communities. Reference [2] argues for shopping communities as they increase customer loyalty and word of mouth resulting in reduced customer acquisition cost and higher transaction level.

3. MULTI-AGENT SYSTEM

According to most accepted definition [3], "An agent is a software entity, situated in some environment that is capable of flexible, autonomous actions in order to meet its design objectives". Multi- agent systems (MAS) arise when several agents are combined together in a single system [4]. Single agents aim at their own local tasks. Interaction between these agents in a system agents results in a global object.

In MAS, Agents share common characteristics like situatedness, autonomy and flexibility [3, 6, 5]. The agents sense their location based on the sensory information they receive from the environment. They can control their operations with no direct involvement of humans. Furthermore the flexible nature of agents allows them to achieve their goals by adapting to varying situations.

Multi agent systems are exhibiting increased complexity with time. An example is the information society technology project which demonstrates the importance of agents in current commercial environment. These agent based systems involve the interaction of agents to access information and services in order to fulfill their design objectives. Communication between agents in these advanced systems is effected by self-organization and complex behaviors. This can impact most simple individual actions and capabilities.

Unfortunately the free movement of agents in environments like e-commerce makes it very tough to identify fraudulent elements and security threats. Absence of adequate security solutions can result in leakage of sensitive information or system hacking. Particularly, some transactions such as those involving use of credit cards must be properly and securely performed.

The lack of a legal substrate capable of grounding the interaction between agents ultimately means that from legal point of view the developer is solely responsible for the actions of its agents [7]. This situation is then worsened by the impossibility of tracing all actions that agent performs [8]. As a result no law is adequate enough to avoid and punish deceptive agents whether they are human or not.

In order to realize the true benefits of agent technology for a wide range of applications MAS should be secure. To prevent these agents based system against security threats; there is a need to
guarantee confidentiality, integrity, availability and accountability through different mechanisms such as authentication, authorization, trust, management.
4. POSSIBLE SECURITY SOLUTIONS

Recently many security threats to the agent systems have been identified. This can help in devising security techniques for protecting the systems and recovering attacks [11]. In this section, security requirements and solutions are provided to the above mentioned threats.

5.1 Naming and Authentication

The stakeholders in a multi-agent system must be given a unique name before authentication. This name could be static or dynamic but in case of multi agents dynamic names are a logical choice as they can contain a reference to agent’s location. After naming, the agents are authenticated through different methods. Some of these methods are described below.
5.1.1 Public Key Infrastructure
Public Key Infrastructure (PKI) uses asymmetric key encryption also known as public key cryptography [12]. Every principal creates a pair of key, public and private in order to be authenticated. The data encrypted by one key can only be decrypted by the other key. Every stakeholder publishes its public key to the world but keeps its private key confidential. The principal identity can be confirmed by its ability to correctly decrypt a message encrypted by the principal’s public key. The real owner of the public-private key will be able to decrypt the message. So the task of the PKI is to stop an impersonator from getting access to agent’s private information and thus it serves to provide security to the agents.

5.1.2 X.509 Certificates
X.509 [13] is largely used in online business applications for distributed authentication. In this protocol, a certificate authority issues a certificate that attach a public key to principal’s unique ID. The certificate authority can verify the authenticity of a certificate, which itself relies on implicitly trusted root certificates. The main components of a X.509 certificate include the issuer and subject names, the public key information of the subject, and the certificate validity period.

5.1.3 Digital Signature
A digital signature [14] is an electronic signature that can be used to authenticate software agent identity in a particular group. In order to authenticate and gain access to secured resources this certificate must be offered by the entity.

5.2 Communication Security
Communication among agents and between agents and platform needs to be secure. In distributed environments like internet communication occurs over long distances and it can be monitored. Agent systems in which all principals are known in advance are usually considered trustworthy.

5.2.1 Encryption methods
Encryption is used to secure the confidentiality and integrity of communication. Secure Socket Layers (SSL)-based communication [15] and Internet Protocol Security (IPsec) [16] are the two well known encryption methods. In SSL the encryption is performed by the application where as in IPsec encryption is employed at a much lower level by the operating system resulting in secure communication automatically. However, most agent platforms employ their own secure communication via SSL as its implementation is relatively simple as compared to IPsec.

5.3 Malicious Host
The safety of an agent’s code and the data it obtains while traveling through a network is the main security concern. Especially, when agents operate in an insecure e-commerce environment and carry out their functions outside the agent’s owner control. Agents may reside on malicious hosts who may have temporary control over the runtime environment of residing agents.

Protecting a migrating agent if it operates on a host outside the agent’s owner control is almost impossible. Such a malicious host can change the internal and external state of an agent completely. Unfortunately, in practice, it is almost impossible to protect a migrating agent if it runs on hosts that are outside the control of an agent’s owner. Such a malicious host can view, modify or even delete the agent altogether.
However, some software and hardware solutions try to allow detection of agents tampered with by malicious host. In principle, agent’s protection from malicious hosts requires [17]:

- Protecting the migration path integrity of an agent
- Protecting the data and code integrity of an agent.
- Guaranteeing agent’s data confidentiality.
- Guaranteeing agent’s control flow integrity.

A migration path involves series of multiple movement steps of several relocations of agents that recognizes all hosts visited by an agent. As a rule, the migration path integrity is used to detect malicious agents and can be used to stop them from doing any harm. Some possible solutions to protect integrity and confidentiality are briefly mentioned below.

5.3.1 Contractual Agreement Method
This method makes host operators establish and use contractual agreements to guarantee the host security.

5.3.2 Trusted Hardware
Trusted hardware such as Trusted Platform Module offer the most protection as the hardware within the computer cannot be easily tampered.

5.3.3 Execution Tracing Technique
The execution tracing technique [18] keeps a track of the agent’s executions in prior platforms of the host. Its can easily detect any change in a mobile agent which is not permitted. However, one of the drawbacks of this technique is the huge size of the log.

5.3.4 Sliding Encryption Technique
This technique [19] can guarantee confidentiality by reducing overhead in case of small messages exchange.

5.3.5 Obfuscated Code Technique
The obfuscated code technique [20] or blackbox security scrambles the agent’s codes to secure the privacy of an agent. However, this technique is only effective for a short term as there is no universal algorithm for blackbox security.

5.4 Malicious Agent
Just like the agent owners who try to protect their agents against malicious hosts, so do platform administrators try to guard their agents against migrating agents having a potential to be harmful. Malicious agents usually attempt to access unauthorized information and resources such as the private data of the host or other agents. Fortunately, platform owner can apply a number of solutions to decrease the malicious agent’s threat and control their right to use of host’s resources.

5.4.1 Signed Code
The signed code technique [21] allows the owner of an agent and/or a reliable third party to protect the integrity of the agent with digital certificates signed by them.

5.4.2 Sandboxing
The sandboxing method isolates agents into a domain that is limited and enforced by software [22, 23]. For example, sandboxing typically restricts access to network and file system. Through Java system programmers define sandboxes by using a security manager.

5.4.3 Safe Code Interpretation
The safe code interpretation technique [21, 22] is vital when the code of the mobile agent is interpreted. It simply cures or ignores an unsafe command while implementing the interpreted code.

5.4.4 State appraisal
State appraisal, according to [23] prevents any change in the status of the agent and thus minimizes the threat of access to information and resources to harmful agents.

5.4.5 Path history
This technique [17, 18] enables agents to have a record of previously visited hosts. This way they will be able to distinguish trustworthy agents from the malicious agents.

5.4.6 Proof carrying code
It [24] allows generation of a proof by the agent author guaranteeing agent code security and then the agent can be verified by employing the proof provided with the agent. On the other hand, generating a formal proof is not an easy task.

5. CONCLUSION
Agent technology with its advanced characteristics like autonomy, intelligence and active problem solving can play a major role in collaborative e-commerce environments. But guaranteeing the security of agent based environments is critical. This paper identified the security threats and solutions to the multi-agent systems in collaborative e-commerce. The usage of agent based collaborative e-commerce depends upon how secure the environment is provided to the customers. The agent owner and platform administrator can combat security threats of malicious agents and malicious hosts by employing the best suited security solutions. Otherwise the consequences of leakage of secure information or resources to harmful agents can be serious threat to its adoption. There is a need for further research on implementing and operationalizing these security solutions in the collaborative e-commerce architecture to identify solutions to any loop holes that may exist.
6. REFERENCES


