Security Aspects of Software Agents in Pervasive Information Systems

John Page
Arkady Zaslavsky
Maria Indrawan

School of Computer Science and Software Engineering
Monash University
Melbourne, Australia
e-mail: {john.page,arkady.zaslavsky,maria.indrawan}@infotech.monash.edu.au

Abstract

This paper discusses vulnerabilities of mobile agent based information systems and calls for an increased level of security awareness, which can be achieved by employing self-reliant security features. The paper describes an IS realised as a multi-agent architecture. For a secure channel to exist within the IS, the use of indicator digests reduces the risk of exposing sensitive information to malicious objects. These solutions add to the underlying layer of security offered by the execution environment and work as plug-ins. This enables them to seamlessly integrate with the system they are protecting and to be transparent to the application user.

Keywords

Information System (IS), Multi Agent System (MAS), Mobile Agent (MA), Security, Mobility

INTRODUCTION

Over the years different definitions for Information Systems (IS) have been proposed which have attempted to capture the functioning and the goal behind the development of these systems. Alter (1999) has defined an IS as a work system, whose business processes are directed towards capturing, transporting, holding, retrieving, manipulating and presenting the information within the system. While this presents a simplistic definition of a complex system, (Whitten et al 2001) extends this thought to describe an IS as an arrangement of people, data, processes, communications and information technology which provide the necessary support for day to day operations of a business including support for problem solving and decision making. This definition brings to light the various entities of an Information system and relates their common functions to each other. Implementation of an IS has been done using different paradigms. In this paper, we model an IS using a multi-agent system (MAS).

Increased volume of information, the need for disconnected computing, requirements of faster access, improvements in wireless technologies and a reduction in computing costs has set the stage for the advent of Mobile Agents (MA). These agents have been defined as computerized servants, which communicate, cooperate and negotiate with other agents. A MAS is a network of MAs, which travels and operates as a cohesive unit. While each of these MAs may have their individual tasks cut out, they work and coordinate their functions towards a central goal of the community. According to Franklin et al (1996), a MA extends a user's authority into the computing world. It executes on the user's behalf and attempts to accomplish pre-defined objective(s). An important aspect that distinguishes it from other forms of mobile code is that it possesses intelligence. On being presented with a number of options, it analyses and chooses the best course of action, without contacting its parent server. Autonomy is another aspect which makes it an ideal information gathering and retrieval tool and a suitable candidate for developing an IS.

Developing an IS using a MAS, allows it access to a continuous stream of rapidly changing information from a variety of sources which makes it a powerful tool for the organisation employing it. On the other hand, this feature makes it vulnerable to a number of security threats from different external as well as internal entities. This paper examines security threats and proposes solutions for a safer mode of computing with respect to the communication and authentication function of MAs in a non-trusted environment. For communication, it proposes the use of indicator digests, while the use of intelligent context sensitive identifiers are highlighted for the authentication function of MAs. The rest of the paper is organised as follows. Section "Implementing an IS using a MA" describes the realisation of an IS using a MAS. The sections "Communication Function" and "Authentication Function" analyse the security goals in an IS and explain the proposed solutions. The final section concludes the paper with an indication of ongoing work.

IMPLEMENTING AN IS USING A MAS

The concept of using MAs in developing Information Systems is an evolving new paradigm, which is fast catching momentum and gaining the approval of the data centric community. The advantage of this approach has led to the integration of mobility with information. In the m-commerce world the use of MA, has led to the development of a variety of applications, which range from offering matchmaking services to finding the best quote for an airlines ticket. The use of MAs for developing information based services which work on the basis of context has been examined. As a consequence, some commercial applications are available in the market. The development of Cyberguide by Abowd et al (1997), which uses location as a context to guide tourists around the city is one such initiative.

For setting up an IS using agents, different groups are given specific functionality. In one scenario, information searching is handled by a specific group of agents. These agents are mobile and are programmed to wander around the web searching for specific chunks or pieces of information. This information can be stock quotes for a particular share, as in different stock exchanges around the world or it can be news releases on sale by different news agencies around the world. These information-searching agents roam the web, from node to node searching for information, matching its search criteria. On finding such information it either carries a part of that information or the location of the information back to its parent. This data about a successful information search is then passed on to Retriever agents. The job of these agents is to bring back the information from the given location. Once this information has been retrieved and is available at the parent node, data mining agents take over. These agents mine the data searching for particular references and chunks of data. Information is then organised and made available in the format desired by the user. Thus, the use of a MAS can effectively set up and manage an IS for an organisation. The benefits of such a model are dynamic data updation, faster information retrieval and the advantage of employing the flexibility and the usability of agents. However, on the other hand, an open and dynamic system such as the one proposed above has its weaknesses. It can be susceptible to security threats and attacks. To understand this, we need to examine the smallest unit of our IS, the structure and the characteristics of a MA.

Essentially every MA possesses state, behaviour and location. State refers to the data state and the execution state of the entity. (Cabri et al 2000). Behaviour refers to the intent of the agent or rather the functions of the agent, which enable it to meet its objectives. Location implies a unique position relative to a fixed point of reference, which facilitates locating the agent. At a higher-level three components of a MA space can be identified. These are Data, The Mobile Agent (MA) and The Mobile Agent System (MAS). Attacks to the agent space can be directed at any of these three components, either directly or indirectly. Direct attacks as the name suggest target the entity directly. For example, an agent server may attack an agent, to capture the data carried by it. This is can be termed as a direct attack on the agent code and an indirect attack on the data carried by it. These attacks can work in different modes, for example an open mode and a concealed mode. Attacking in an open mode implies that the attacker is not concerned with concealment. Symptoms of an open attack can range from degradation of service to corruption of system files. Concealed attacks are difficult to detect because they do not disturb the equilibrium of the agent space. Examples of such attacks are eavesdropping, tracking agent travel etc. Figure 1 explains the different modes and kinds of attacks that can be launched on a mobile agent space.

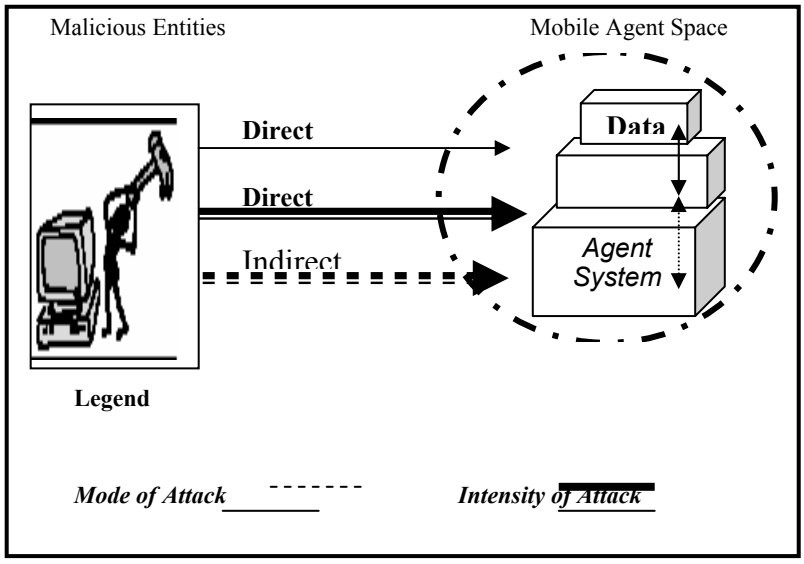

Figure.1. Attacks on an Agent Environment

Security in a MAS is implemented at various levels using different schemes. These schemes constantly monitor incoming and outgoing network traffic. Hence, attacks directed at a MAS usually are concealed and strong. On the other hand, MAs are lightweight processes. Their functionality is specific and they cannot carry elaborate security measures, while in transit. Attacks at MAs, attempt to intercept and subvert. By subverting the MA, attackers can indirectly attack the parent system when the agent returns to it. This paper provides an analysis of the possible vulnerabilities of MAS and some approaches of reducing these vulnerabilities.

COMMUNICATION FUNCTION

Agents communicate and work in a heterogenous and dynamic environment. They are expected to be interoperable and flexible in what they say and in what they can understand. This communication may be between agents of the same group or with agents of different communities. Agent systems use an Agent Communication Language (ACL) to communicate. Although, a lot of effort has gone into the development of ACLs, there still remain a number of issues, which prohibit any one language from becoming the de-facto standard. Singh (1998) proposed a movement in individual representations of ACLs to a more generic approach based on social interaction. The need for this paradigm shift was identified from the fact that agents were not being able to communicate as heterogeneous entities.

The design goals of ACLs, classify them as protocols rather than programming languages because of their relation with "intelligent data". Intelligent data implies information flavoured with the purpose of intent. Take for example: A transaction which states credit a savings account. This can be regarded as data. There is not much information carried with it, save for the fact that a particular account is to be credited. The same transaction if stated as, Credit $50 in Account No 12122 and Debit $50 in Account No 3232 and tag as 'Book Sales' carries more information. It tells us

1. The particular account number to be credited.
2. The particular account number to be debited.
3. The amount involved in transactions 1 & 2.
4. The reason for the transactions i.e. Book Sales.

The ARPA knowledge sharing effort of Patil et al (1992), led to the development of the Knowledge Query and Manipulation Language (KQML) (Labrou 1997) and Knowledge Interchange Format (KIF) (Genesereth et al 1991). Apart from these two ACLs, the Foundation of Independent Physical Agents (FIPA 1998) has come up with its own specification of an ACL These ACLs have certain limitations as pointed out by Dignum et al (2000) and (Mayfield 1995), which restrict their functionality and make them vulnerable to eavesdroppers.

The use of predicate calculus with declarative semantics in KIF (Genesereth et al 1991) allows agents to parse and understand the actual content of messages they may receive. This also allows the agent to compose arbitrary sentences and to understand the meanings of expressions without using an interpreter. While this increases the flexibility of KIF as an ACL, it still does not make a case for a secure framework of communication.

KQML (Labrou 1997) attempted at setting up a formal semantic layer for the efficient exchange of information between agents (Mayfield 1995). The goal of KQML was to provide a standard message format and a robust protocol to handle these messages. While a secure architecture of KQML (Thirunavukkarasu et el 1995) addressing privacy, authentication and non-repudiation (with some considerations) has been proposed, there are limitations. Only agents supporting cryptographic capabilities can send out secured messages. While this is a reasonable assumption, we need to consider that incorporating cryptography into the agent function will only increase the complexity of the agent design. Thus, this approach is to be avoided.

A possible solution to this problem could be the complete removal of transmitting sensitive information in the agent communication function. The agent will carry only an indication of the actual message. This indication is referred to as an 'Indicator Digest'. The indicator digest will be intelligent enough to convey the context and location of the actual message. On receiving the indicator digest, the receiver is alerted to a possible communication session. An analysis of the indicator digest will give the recipient an indication of three things.

1. The sender of the message
2. The context of the message
3. The location of the message

Figure 2 describes this concept. In case 1, the agent is burdened with carrying the entire message. Apart from the additional effort required on the agent's part to transport the entire message correctly, there is also a risk of the agent getting captured and the information being lost. In case 2, a context based compression function is applied to generate an indicator digest. The agent now, carries this indicator digest to the destination server. The idea behind the use of an indicator digest is to reduce the workload of an agent and to make it difficult for an attacker to intercept sensitive information.

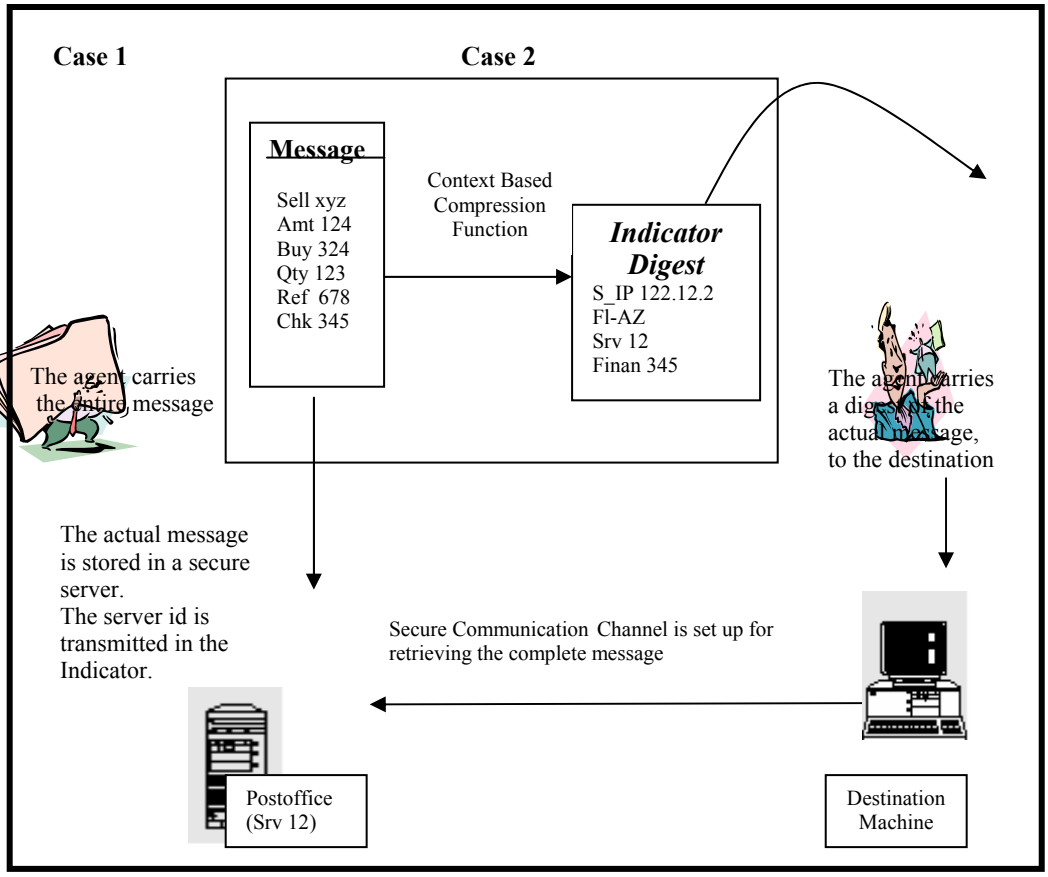

Figure.2. Using an indicator digest for agent communication

The structure of an indicator digest is detailed in figure 3. The indicator digest has to be strong enough to meet the challenges of non-repudiation and forgery. This is only possible if the information carried in the indicator digest is unique and can be reverified easily. Now, what can be considered as unique information? To understand this, consider the case of an aircraft taking off from Frankfurt airport at 1012 hrs. The ATC logs will clearly record the flight no, the time of the flight and the destination of the flight (probably, the flight plan). A number of aircraft may have taken off from the airport during the day, but the entry pertaining to this particular flight will be unique. Similarly, for a server, which sends out messages, there should be server logs, that record details of every message going out. Another good analogy of this concept is the storing of sent messages in the sent items folder in an email client. Each message stored in the sent items folder is a copy of the one sent out and this premise holds good if the software is functioning correctly.

Indicator Id (Maps to a Particular Message)
Sender Id (Sender machine IP)
Location of the Actual Message (Server IP)
Context of the Message (using Key-Words)
Expiry date/time of the Message

Figure.3. Structure of an indicator digest

Consider a scenario in which Bob wants to message Alice. Bob generates his message and stores it in a particular location within a secure server; we refer to this server as the Post Office. Using the context-based compression function, he then generates an indicator digest, a sample of which is shown in figure 4.

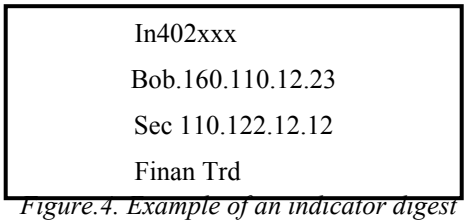

Figure.4. Example of an indicator digest

On receiving the indicator digest, Alice is alerted to the fact that Bob's server 160.110.12.23 has posted a message for her at the server address 110.122.12.12. The message relates to a financial trade (Finan Trd) and will expire on the 5/02/2003 at 6:39:26. To re-confirm whether Bob has really posted a message or there is someone impersonating Bob, Alice can view the message logs of Bob's server at 160.110.12.23. These logs are available in a read mode to entities, which have been sent messages from Bob's server. Thus, Alice is now sure in the knowledge that Bob has posted a message for her at a certain location. All she has to do is set up an encrypted channel to the Post Office using a security protocol such as SSL (SSL2.0 1995) and retrieve the message. The messaging algorithm for the communication will be as follows.

*/*Message Generation and Parsing */*

*/*Pre-processing Steps*/*

1. *Generate the general structure of the message using a pre-proc*
2. *Add the key words of the message*
3. *Parse the message through the parser*

/ Check for completeness of the message */*

4. *Compile the message to check for errors*
5. *Is the message complete in all respects?*
6. *If yes then GOTO step (8) else*
7. *Debug the errors and return to step (3)*
8. *Record message parameters like size, time of generation, addressee /*This will be useful in audit trails*/*

/ Preparation of the message indicators */*

9. *Pass the complete message through the Indicator-generation program*
10. *Check the indicator-digest*
11. *Is the indicator-digest complete in all respects?*
12. *If yes then GOTO step (14) else*
13. *Debug and return to step (9)*
14. *Give the indicator digest a unique identifier*
15. *Record the indicator digest identifier against the actual message*

This approach to messaging has several advantages. Firstly, sensitive data need not be carried over an open network. Secondly, there is significant saving in network bandwidth. Several messages can now be stored in a server and one indicator digest can refer to all of them. Thirdly, the probability of agents getting attacked is now reduced, as the attacker is aware that the agent has nothing of value on it. On intercepting an indicator digest, the attacker may be alerted to the fact that Bob and Alice are brokering a deal, but to find out more he has to approach the secure server where the message is lodged. Thus, his attack will now have to be against a strong, secure server which has been specifically been prepared to meet such attacks rather than against a weak unsuspecting agent.

AUTHENTICATION FUNCTION

The Authentication function is the first line of defence for a MAS. The size of this layer depends mainly upon the functional objectives of the MAS. For example, if the MAS is working on behalf of a bank or a stock broking firm, the authentication layer will be thick with many sub-layers of security involved. On the other hand, if the MAS is engaged in conducting a survey of some kind and has agents from other systems approaching it regularly, it may decide to have a thin layer of authentication.

Having a thick authentication layer for a MAS, which has a high volume of traffic to it, can lead to a general performance degradation of the system. The size of the authentication layer is actually a question of tradeoffs. Having less authentication checks means the turnaround time is less but on the flip side it can become a security hazard too. Thus, the decision on how many authentication layers to have actually hinges on two points:

1. What will be the volume of traffic to the node?

2. How vital is the information to be protected?

The answer to the first question can be approximated pretty fairly but the second question needs careful consideration before an answer is volunteered. Most organisations regard data as their life and blood and very rightly so. Thus, if organisations are not willing to compromise on security for their data but still wish a thin layer there is only one solution. Allow the authentication to happen in customisable layers, which can be added or removed from time to time, depending upon the volume of traffic and the importance of the transaction. In the earlier example of Bob and Alice communicating, consider a scenario in which Bob's agent approaches the server of Alice. Alice is expecting Bob's agent but wants to make sure the agent approaching has actually been sent by Bob. Alice can turn on all layers of authentication, available to her system and validate the identity of the approaching agent. On the other hand, if Alice is not concerned about the identity of the agent sender and the approaching agents are only coming to pick-up information, which is being distributed publicly by Alice, she might switch off some of the authentication layers.

Currently, the authentication process is implemented in a number of ways. A state appraisal approach given by Farmer et al (1996) for authentication is one possible approach. This architecture is based on four different trust relationships, which may exist between principals. A principal is an entity that makes a request to perform operations on objects. Objects can be files, devices and processes. This theory is proved using the distributed authentication theory of Lampson et al (1992). The state appraisal authentication model of Farmer et al (1996) can also be extended to Agent Handoffs and Agent Delegation as a way of transferring or handing over authority to other agents. To make such transfers seem favourable, the authors have proposed the theory of lists and certificates. This theory is built around The Sender Permission List (SPL), The Sender Permission Certificates (SPC), Place Permission Lists and Place Permission Certificates.

Collecting information for generating these lists may involve filling in a migration form with the server's id, agent id, principal id, target server's id, flag and validity period. The agent code and its state appraisal function is created and signed to ensure that there is no tampering with the code. An SPL is also created. This holds the list of users who are permitted to transmit the agent. This information is mentioned in the SPC, which are similar to SPLs but travel as a separate entity with the data. These SPCs indicate the program to which the certificate refers to and its creator.

While the approach outlined by the state appraisal theory may be effective in the preliminary authentication process it has some vulnerabilities. Firstly, it relies on signing the code of the agent after the state appraisal function has been written. This may prove to be a limitation as it is possible to sign only static portions of the code. Agents do not carry private keys and hence cannot sign their code dynamically. Even if they were to carry some cryptographic routines, there is no guarantee of them getting executed correctly as the agent is reliant on the platform for executing its code. Since the agent is moving and functioning in a dynamic environment, it is not possible for the agent to maintain a constant inner environment. Environmental changes may cause the MA to modify certain aspects of its functionality. These changes may render the state appraisal functions dysfunctional. Secondly, the use of certificates and policies is a vulnerable aspect of the model as they are difficult to maintain and expose the system to malicious attacks. Collecting data to generate these lists is always a sensitive issue. The verification and validation of data, which is going to form these lists, requires several questions to be answered. How does once convince oneself, that the data being used is correct and not compromised? It is difficult to do so. Another drawback of a policy-based approach is that data may change. The process of updating the lists further disturbs the equilibrium of the system and exposes it to further threats.

Authentication using agent passports has been proposed by Chess et al (1995). This has been further extended by Schelderup & Ølnes (1999) to include error handling procedures, description of intent and possibly references to authentication certificates and authorising bodies. On one hand, while increasing the amount of information volunteered during an authentication process increases the chances of the agent gaining the trust of the server, it also makes it vulnerable to forgeries or hijacked passports. Furthermore, the system is compromised if a malicious server is able to manufacture similar passports and send out agents. Thus, this approach, which is dependent largely upon cryptography and code signing, has its limitations as discussed by Schelderup & Ølnes (1999).

Since MAs will move and approach many different servers. They will attempt to authenticate themselves to these servers. Thus, the efficiency of an authentication mechanism depends upon two parameters, Information and the correctness of that Information. The authentication mechanism may need to ask the right questions and then correctly analyse the answers received. From the viewpoint of the entity-requesting authentication, a delicate balance in providing the correct amount of information needs to be maintained. Volunteering extra information could lead to a malicious agent server learning enough to use the information against the agent or against its parent server. Providing very less information may not convince the agent platform to grant access rights and the permission to use its resources. An effective authentication mechanism reflects the desired goals of the agent server. Consider for example, an agent server, which is selling news releases to different news

agencies via MAs. What are the parameters of an effective authentication mechanism is this case? What questions should the agent server ask the approaching agent and what information should be provided by the agent?

Case 1

In case 1 of the scenario, the identity of the agent is not a primary concern, because the focus is on selling the information and getting our price. It is similar to visiting a supermarket. When a customer purchases bread or milk, he is asked to show his passports and identify himself.

In this case, to make a successful sale, the agent server needs to tell the agent

1. The nature of the news release i.e. whether it pertains to sport, health, current affairs, business, or weather.
2. The size of the news release in bytes/characters/words/pages.
3. The charges of the news release in dollars (Us/Aus), pounds, rupees, riyals, dirhams.

To make a successful purchase, the agent needs to tell the server

1. Whether it is interested in the purchase of the information available.
2. The medium through it intends to pay for the information i.e. electronic cash, credit card, debit card, barter etc.
3. The medium through which it intends to take delivery of the information.

An analysis of the answers provided by the agent can be sufficient to authenticate it and to move to the next step of the transaction.

Case 2

In case 2 of the scenario, the identity of the agent is of interest because the intention is to build a profile database of the customers and provide them with special offers, when they become regulars. In this case, the agent server needs to ask the agent its personal details and store that information in our database. Thus, the agent server should ask for

1. The identity of the agent. (This may not necessarily imply the identity of the agent's owner. For example, Reuters or PTI may desire to withhold their names. If such a requirement exists, the agent server will have to be satisfied by a unique identifier provided by the agent and leave it at that.
2. The nature of the information the agent is interested in
3. The medium of payment
4. The medium of delivery of information

The agent will provide the requested details. The agent server will verify these details and a unique customer identifier number will be assigned to the agent. This will be linked to secondary identifiers, which could be details of the last five transactions performed by the agent at the server.

Figure 5 explains the authentication mechanism proposed. It brings to light the importance of context and content sensitive information for authentication. Context information may refer to the nature of the transaction, while the content part may concern itself with the details of the transaction. Consider for example, our earlier scenario of Bob and Alice. Bob sells a book to Alice. The context part of the information refers to a sale, while the content part of the transaction are the details of the sale, i.e. the title of the book, the cost of the book etc. In this mechanism, the authentication process is controlled by the context based authentication layer, which acts as a controller. This controller is fed information by a separate database and will attempt to authenticate the approaching entity based on its context. If the answers received are satisfactory, the entity will be allowed to pass through. In all other cases the entity will be diverted through additional levels of authentication, as shown in figure 5. In figure 6, the use of context sensitive information for a news agency, which disseminates information through approaching MAs is described. In our prototype implemented using the Grasshopper (2001) v2.2b MAS, agents approaching the agency for information related to TV programs have got a context sensitive identifier which is recognized by the agency, when the MA docks.

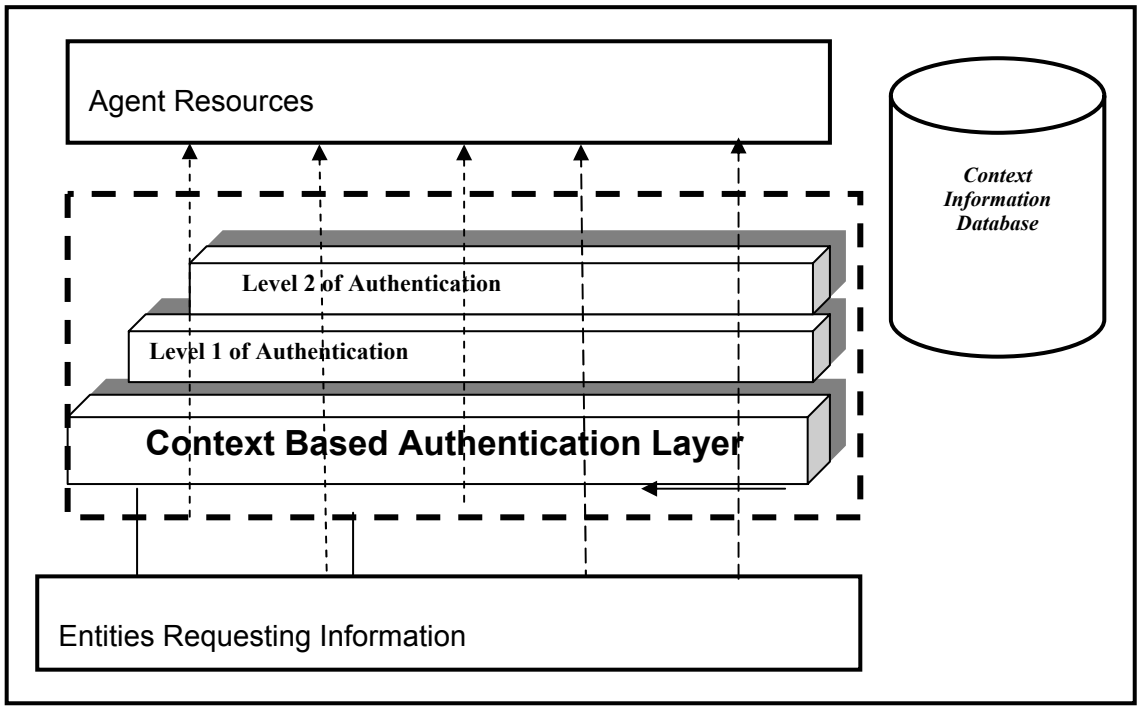

Figure.5. Authentication using a context rich layer

If the context identifier is valid, the agency directs the MA to the place at which this information is held. Once the MA docks at the correct place, its details are authenticated and the information it requests made available.

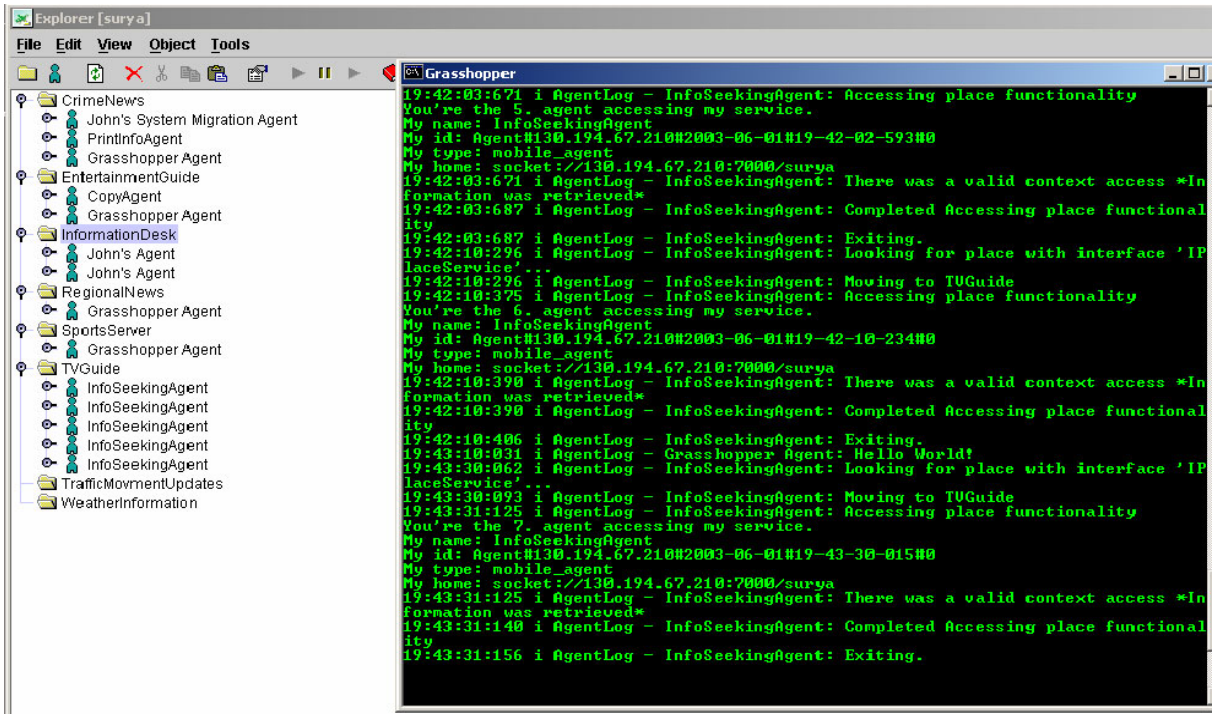

Figure.6. An implementation of a context sensitive layer for authentication

Building an authentication mechanism, which can be customized to meet changing scenarios can greatly improve the efficiency and the turnaround of a pervasive system. It also reduces the confusion and the overhead of verifying details, which may not be of significance to the agent server. For example, if the transaction is going to be a one-time transaction, the agent server does not need to build a profile of the agent and concern it with identifying the agent. This will save storage space and also make the system more intelligent.

CONCLUSION AND ONGOING WORK

Among all decisions that an IS implemented in a pervasive environment has to make; those related to security are the most complex and critical of all. These decisions range from authenticating users to allowing the modification of information stored within the system. Having an intelligent context sensitive based layer, which can understand and recognise context based accesses can significantly improve the turnaround time of the IS. Future work will attempt to make the IS intelligent enough to identify and pick up pieces of data which can be put together to generate a valid context.

While communication within a mobile IS requires a secure and trusted channel, it is very difficult to guarantee the existence and trustworthiness of this channel as it may sometimes require the support of non-trusted servers. In such situations the importance of transmitting only the bare minimum of sensitive information gains significance. A security system designed to recognise and respond to chunks of information, rather than large streams of bytes will make it a difficult target for a potential attacker. The goal is to develop an intelligent information system in which communication detail is minimum but carries maximum information.

The need for a secure and trusted framework to support the essential requirements of authentication and communication are essential for the IS to operate in a pervasive environment. IS developed using MAS, have concentrated only on developing the application layer of the system, but have failed to provide it with its own self-reliant backbone of security. In most systems, this factor is left to the underlying execution framework. For example, Java (Gosling, 1996) based systems rely on the security provided by the JVM. While this may be sufficient to protect the host from malicious agent code, it does nothing to protect the agent code from a malicious server.

Future work will concentrate on further developing and implementing the model of an indicator digest as a plug-in for the MAS. This will allow the agent system to retain it original design, while using this messaging technique. For the authentication function, we will refine the authentication model to allow a layered mode of authentication for approaching entities. These layers will work as switches, which could be turned off and on, depending upon the requirements of the application controller. This will allow the system to manage its security requirements with more intelligence and control.

REFERENCES

Abowd, G.A, Atkeson, C.A., Hong,J.,Long,S., Kooper,R., & Pinkerton.M (1997) Cyberguide: a mobile context-aware tour guide, ACM Wireless Networks, 3, pp. 421-433.

Alter, Steven (1999) Information systems, a management perspective 3rd ed., Reading, Addison Wesley.

Cabri,G.,Leonardi,L., & Zambonelli,F (2000). "Weak and Strong Mobility in Mobile Agent Applications". In Proceedings of the 2nd International Conference and Exhibition on The Practical Application of Java, Manchester (UK).

Chess,D., Grosof,B., Harrison,C., Levine,D., Parris,C., & Tsudik,G. (1995) "Itinerant Agents for Mobile Computing," In IEEE Personal Communications, vol. 2, no. 5, pp. 34-49.

Dignum, F. & Greaves, M. (2000) "Issues in Agent Communication: An Introduction." In F. Dignum and M. Greaves (Eds.) Issues in Agent Communication (LNCS-1916), Springer-Verlag, pp 1-16.

Farmer,W., Guttman,J. & Swarup,V. (1996) "Security for Mobile Agents:Authentication and State Appraisal," In Proceedings of the 4th European Symposium on Research in Computer Security (ESORICS '96), pp. 118-130.

Franklin, S. & Graesser, A. (1996) "Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents". In proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, (LNCS-1193), Springer-Verlag.

FIPA (1998) Agent Management, Specification, part 1, version 2.0, Foundation for Intelligent Physical Agents, October 1998. URL: http://www.fipa.ord/spec/fipa97/fipa97.html Date of Access 05/01/2003.

Genesereth, M. R. (1991) "Knowledge Interchange Format", In Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning, pp 589-600.

Gosling,J., & McGilton,H. (1996) "The Java Language Environment: A White Paper", Sun Microsystems.

Grasshopper. (2001) Release 2.2, Basics and Concepts Revision 1.0,URL http://www.grasshopper.de/download/doc/BasicsAndConcepts2.2.pdf pages 70 Date of Access 03/12/2002.

Labrou,Y. (1997) Semantics for an Agent Communication Language, PhD Thesis, University of Maryland, USA.

Lampson,B., Abadi, M. & Burrows,M. & Wobber,E. (1992) Authentication in distributed systems: Theory and practice. ACM Transactions on Computer Systems, pp 265-310.

Mayfield,J., Labrou, Y. & Finin, T. (1995) " Evaluation of KQML as an Agent Communication Language". In M. Wooldridge, J. P. Muller, M. Tambe, (Eds), Intelligent Agents II, Agent Theories, Architectures, and Languages, IJCAI '95, Workshop (ATAL), (LNCS 1037), Springer-Verlag, pp 347-360

Patil,R.S.,Patel-Schneider,R.E.,McKayF, P. F.,Finin. D., Gruber, & T.,Neches,R.(1992) The ARPA knowledge sharing effort: Progress report, In Rich C.,Swartout W., Nebel B. (Eds), Proceedings of Knowledge representation and Reasoning ,pp. 777-788.

Schelderup, K., & Ølnes,J. (1999) Mobile Agent Security – Issues and Directions in H. Zuidweg et al. (Eds.): IS&N'99, (LNCS 1597), Springer-Verlag ,pp. 155-167.

SSL2.0 (1995) protocol specification. URL http://wp.netscape.com/eng/security/SSL_2.html Date of Access 29/03/2003.

Singh, M.P. (1998) "Agent Communication Languages: Rethinking the Principles." In IEEE Computer, volume 31, number 12, pp 40- 47.

Thirunavukkarasu,C., Finin,T. & MayField, J. (1995) Secret agents: A security architecture for the KQML agent communication language. In Proceedings of the ACM CIKM Intelligent Information Agents Workshop.

Whitten, J. L., Bentley.D.L, & Ditteman, K.C. (2001) Systems analysis and design methods, 5th ed. Boston, Mass., Irwin/McGraw-Hill.

COPYRIGHT