Dealing with Security within DEEPSIA Project

FRANCISCO GOMES MILAGRES
EDSON DOS SANTOS MOREIRA
USP — Universidade de São Paulo
ICMC — Instituto de Ciências Matemáticas e de Computação
Trabalhador Sancarlense, 400 — 13566-590 — São Carlos – SP
BRASIL
francisco@milagres.com , edson@icmc.usp.br — http://www.icmc.usp.br

JOÃO PAULO PIMENTÃO
PEDRO ALEXANDRE DA COSTA SOUSA
ADOLFO STEIGER GARÇÃO
UNINOVA — Centre for Intelligent Robotics
Universidade Nova de Lisboa
Quinta da Torre — 2825-114 — Caparica
PORTUGAL
{pim, pas, asg}@uninova.pt — http://www.uninova.pt

Abstract: - DEEPSIA (*Dynamic on-linE IntErnet Purchasing System based on Intelligent Agents*) aims to develop a system to support companies as purchasers in electronic commerce e-procurement processes. To pursue this task, DEEPSIA is implemented using a Multi-Agent System, which components may, and effectively are, distributed in four countries (Brazil, Poland, Portugal and Spain). Being a system that uses the Internet and deals with critical information, a high security level is required. The objective of this paper is to summarily present the architecture of the DEEPSIA multi-agents system, focusing on the security needs identified. The paper proceeds with the work being developed by the Brazilian and the Portuguese teams towards the enhancement of the security of such systems.

Key-Words: - electronic commerce, mobile agents, Internet, security, privacy, FIPA.

1 Introduction

Since the e-commerce (EC) expression was for the first time defined, new ways were also defined to make business digital networks. E-commerce represents the act of conducting business communication and transactions over networks and through computers. [1]

Traditional approaches to e-commerce reside mostly on individual company's web presences or, more scarcely, on aggregated presences, under the scope of "e-marketplaces".

Whichever of these approaches is followed, the fact remains that, for the purchaser, the solution to find the needed products (in the Internet) is either to search the marketplaces in which the purchaser is a registered client, consult the usual suppliers, or to use available search engines (such as Google [1]) to identify web pages containing a given text (not necessarily selling anything).

Any of these approaches is time consuming and diverts the purchaser from its main task, which is to decide whom to buy to and under which conditions.

To help re-centering the perspective on the buyer, the DEEPSIA project (supported by *Information Society Technologies Programme* from European Community [2] – IST-1999-20483) was created. It aims to address the purchasing business process within Small and Medium Enterprises (SMEs) with a customizable e-commerce application that, by searching the Internet, produces a catalogue of products (catered by a set of suppliers) that meets the needs of the purchaser.

In this era of Information Society, companies are beginning to realize that information is becoming more and more valuable. As Naomi Fine, CEO of Pro-Tec Data puts it, "The steady rise can also be attributed to two additional factors that have been rising exponentially over the same years as the study: increased recognition that information has value and increase in perceived value of information." [2]

[1] http://www.google.com

[2] http://www.cordis.lu/ist

In fact, information relates not only with technical or financial information stored within the company, it relates with everything a company does in order to conduct business, which may give its competitors a competitive edge.

On another angle, producing a system that may aid on some aspect of business decisions, introduces a set of other security requirements that must be considered to allow decisions to be based on reliable information.

It is under this scope that the focus of this paper falls: to ensure, within the DEEPSIA project, reliable and private information to be conveyed to the decision-makers desk.

In short, this paper briefly presents the architecture of the DEEPSIA's Multi-Agent System and describes its vulnerabilities and the threats it faces, focusing on the Research and Development being undertaken to circumvent these problems.

This paper is divided as follows: section 2 presents the DEEPSIA project; section 3 describes basic concepts of software agents; section 4 discusses security requirements regarding to the DEEPSIA project specially; section 5 presents a review of previous multi-agent systems security reports; section 6 describes the first approach for security into DEEPSIA, the S-KQML approach and some considerations about its upgrade, a FIPA complaint one; section 7 describes the second approach for DEEPSIA security — "Split and Merge", based on communication security; finally, section 8 presents this paper conclusions and some end remarks.

2 The DEEPSIA Project

2.1 Scope

The main objective of DEEPSIA (Dynamic on-linE IntErnet Purchasing System based on Intelligent Agents) is to address the purchasing business process within Small and Medium Enterprises with an e-commerce application, helping to perform usual day-to-day purchasing tasks taking advantage of the potential of the WWW.

The system under development is based on a Multi Intelligent Agent System that autonomously generates an electronic catalogue of products. This catalogue gathers products data from multiple vendors so it can be easily compared. In this way, it is expected to achieve cheaper and more time effective purchases using these search results. [3]

The DEEPSIA's main target is to change the traditional e-commerce business model, which generally considers SMEs mainly as suppliers (e.g. in virtual shops or marketplaces).

The Consortium consists of the following University partners: UNINOVA (Institute for the Development of New Technologies – New University of Lisbon), ULB (Université Libre de Bruxelles), University of Sunderland; and the commercial partners: ComArch S.A. (Poland), Atlante (Spain), Zeus Consulting S.A. (Greece). USP (Universidade de São Paulo), from Brazil, is the invited member from outside the European Community.

For further project details, visit its web site at http://www.deepsia.com.

2.2 The Multi-Agent System

The kernel of the DEEPSIA system is a Multi-Agent System (MAS). The functionalities of the most relevant agents are outlined in the reminder of this section.

The MAS — (which general architecture is illustrated in Fig. 1) consists of a set of agents that implement two basic mechanisms to collect information: the autonomous search (which is performed using crawler agents) and the direct search (which is performed using Portal Interface Agents — PIA). The PIA can be installed in commercial web sites. The commercial web sites, in this case, form a community of companies that, by realizing DEEPSIA's potential, provide data directly to the system's catalogue. [4]

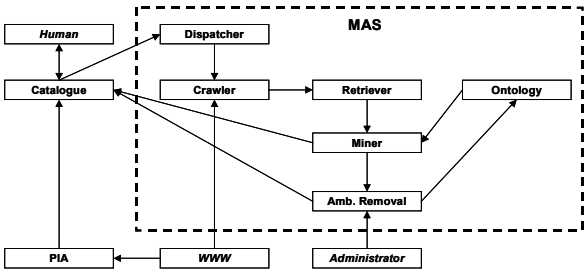

Fig. 1: DEEPSIA's simplified architecture (adapted from [4]).

The dispatcher agent is responsible for the interaction with the catalogue. It has the ability to choose a crawler for the catalogue to use and make it crawl with the selected URL.

The Crawler agent is in charge of navigating through the link structure associated with a given URL given and classifying the selected pages with the user's themes of interest (in the specific case of DEEPSIA, it identifies the pages that are selling products). The process is based on text classification methods, which detailed presentation is out of the scope of this paper. [5, 6]

When the Crawler agent finds interesting pages, it stores these pages to a repository. Then, the Retriever agent obtains the pages from the repository and notifies the Miner agent about new pages found.

The Miner agent has the responsibility of identifying the relevant concepts included in the pages selected by the crawler. In DEEPSIA, it extracts product information from the pages using heuristic methods and a knowledge base.

For every concept identified, the Miner agent collects the information of the concept. If the Miner is able to extract interesting content from the page (in this case, product information such as its description and price, for instance), it queries the Ontology agent for each product found, to determine to which class the product belongs. [7, 8]

When the Miner receives the answers regarding the ontology classification, it sends the product and class information to the assigned catalogue. If the ontology is unable to univocally classify the page, the Miner sends all the information to the Ambiguity Removal agent for user assessment.

The dynamic catalogue is the user interface, and it is responsible for presenting the multi-agents' collected information based on users' preferences. This catalogue holds information about the data selected by the agents on the web, the sites contacted, the ontology in-use and all user requirements. The data of the electronic catalogue is stored in a database and will be made available through a web-browsing interface [4].

3 Software Agents

For the past years, computer have evolved from centralized and monolithic systems supporting centralized applications to networked environments, allowing complex forms of distributed computing, lately evolving towards the concept of *software agents.*

A software agent can also be defined as a program that exercises the authority of an individual or organization, working autonomously towards a goal, to meet and interact with other agents. Possible interactions include contract and service negotiation, auctioning, and exchange. [9]

Software agents may be either stationary or mobile. Stationary agents remain resident at a single platform (computer or other "agent enabled" device), while mobile agents are capable of suspending activity on one platform and moving to another, where they resume execution. The concept of mobile code is not new, dating back to the 1960's when remote job entry systems were used to submit programs to a central computer. Recently, code mobility has been popularized through the use of web browsers to download Java applets from web servers. Mobile agents go one step further, allowing the complete mobility of software among supporting platforms to form large-scale, loosely-coupled distributed systems. For further information about software agents, refer to a specific agents' research group. [3] [10]

The current status of DEEPSIA project, with a prototype system available through DEEPSIA's web site, is a set of software agents, running in fixed platforms spread throughout Europe and Brazil, communicating by KQML messages. [7, 8]

DEEPSIA's architecture has since been analyzed and there is now, a common agreement that, considering communication costs and efficiency, some agents should be localized where their work takes place.

One of such examples is the Web crawler. The major effort on crawling takes place within a specific site and the result of its effort is a set of pages that satisfies a given criterion. Presently, the crawlers are located in Portugal and all the crawling (i.e. fetching the web pages) is done from this location; regardless of the fact that most of the pages fail to meet the criterion. The efficiency would surely improve if the crawling was to be done in the country where the web site is located.

4 Security Requirements

In general, security in electronic communication is based on the implementation of a set of principles. This section outlines those principles analysing with more detail those who are particularly relevant within DEEPSIA's context. The focus is then on how these requirements are met by the system being developed.

4.1 Securing electronic communication

It is a common understanding that in order to achieve secure electronic communication among parties, one should assure: [21]

Confidentiality: guarantee that the message is only read by its intended destination;

Authentication: certify that the message was really sent by the alleged sender;

Integrity: guarantee that the message that reaches its destination has the exact content that was generated at the source;

Non-repudiation: guarantee that the sender cannot deny having sent the message and guarantee,

[3] http://agents.umbc.edu

that once received, the destination gan not deny its reception;

Access control: control access to the information being exchanged;

Availability: guarantee that communication is available between the source and the destination.

4.2 Security Specifications within the DEEPSIA Project

Knowledge about the set of products being inquired by a given company can supply relevant information (namely about the projects being developed or about the set of pending orders) to its competitors or to the financial market.

On the other hand, the possibility of having falsified information injected into the system can totally disrupt its operation.

The need to protect these communication aspects within DEEPSIA system leads to the need to ensure:

Anonymity: the capacity to hide the final client from the queries he/she is performing;

Confidentiality: assure that the contents of the messages being exchanged remain hidden;

Reliability: assure that the messages arrive intact as they left their origin;

Authentication of the sender: assure that the originator was who it was supposed to be.

In order to implement these requirements, two teams (one from Europe and another from South America) are working together. The reminder of this paper will present the current status of research in these topics.

5 Review of MAS Security Published Reports

As the sophistication of mobile software increases, the associated security threats and vulnerabilities also increase.

Threats to the security of mobile agents generally fall into four comprehensive classes: disclosure of information, denial of service, corruption of information, and interference or nuisance. [9]

The components of an agent system are going to be used for further delineate threats by identifying the possible source and target of an attack regarding to elements within that paradigm. It is important to note that many of the threats that are discussed have counterparts in classical client-server systems and have always existed in some form in the past (e.g., executing any code from an unknown source either downloaded from a network or supplied on physical media).

Mobile agents simply offer a greater opportunity for abusing and misusing, broadening the scale of threats significantly. New threats arising from the mobile agent paradigm are due to the fact that against the usual situation in computer security where the owner of the application and the operator of the computer system are the same, the owner of the agentand the system operator may be different.

There are many models for agent systems description; however, for security issues discussions, it is sufficient to use a very simple one, consisting of only two main components: the agent itself and the agent platform. A complete model of threats against MASs is out of the scope of this paper and more information can be obtained on other specific papers. [11, 14, 22]

An agent comprises the code and state information needed to carry out some computation; multiple agents cooperate with one another to carry out some application and mobility allows an agent to move or hop among agent platforms. The agent platform provides the computational environment in which an agent operates. The platform where an agent originates is referred to as the home platform, and normally is the most trusted environment for that agent. Figure 2 depicts the movement of an agent among several agent platforms. [12, 13]

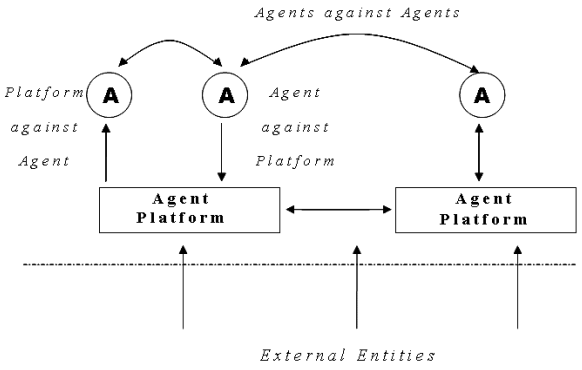

Fig. 2: A model of threats in software agents. [11] *apud* [14]

Four threats' categories can be identified from the simplified model on Fig. 2:

- An agent attacking an agent platform;
- An agent platform attacking an agent;
- An agent attacking another agent;
- External entities attacking the agent system, which is composed by the agents and the agent platforms.

The cases of an agent attacking an agent on an agent platform and of an agent platform attacking another platform are covered within the last category, since these attacks are primarily focused on the communications capabilities of the platform to exploit potential vulnerabilities. The last category also includes more conventional attacks against the underlying operating system of the agent platform. [9, 11]

The research on mobile agent system security has counterparts in non-mobile agents systems such as conventional client-server system security. For example, direct attacks on the code integrity of a mobile agent by an untrusted foreign host environment can be equated to integrity and DoS attacks by an untrusted remote agent, e.g., they can construct messages to cause the receiving agent's message handler to fail. As MASs of communicative agents reaches out more into the untrusted heterogeneous environment of other MASs, communicative agents will likely face similar threats to those threats in mobile agent systems. [26]

There are however, important differences between MASs of communicative agents and mobile agents: the protection of the agent code against code modification, despite the fact that being an obvious concern in mobile agent systems is not a major threat in MASs of communicative agents. Communicative agents are also more prone to communication threats than mobile agents. Multi-agent systems of communicative agents offer a comparable challenge to mobile agent systems, but to an extent, a different opportunity for misuse and abuse.

The next sections are going to present the work being developed by the Brazilian and the Portuguese teams towards the enhancement of the security of DEEPSIA's MAS.

6 Focusing on ACL Security: the S-KQML approach

The basic security requirements that are going to be presented here for KQML — the Agent Communication Language (ACL) in use by the DEEPSIA's Multi-Agents System at the present time — are based on the analysis of the security models for Privacy Enhanced Mail (PEM), Common Object Request Broker Architecture (CORBA), Distributed Computing Environment (DCE) and Secure Infrastructure for ACLs (S-KQML) — a previously defined secure infrastructure for agent communication languages based on the original KQML. [17, 18, 19, 20]

The security capabilities that a model such as this should support include: [20, 21]

Authentication of principals: Agents should be capable of proving their identities to other agents and verifying the identity of other agents;

Preservation of message integrity: Agents should be able to detect intentional or accidental corruption of messages;

Protection of privacy: The security architecture should provide facilities for agents to exchange confidential data;

Detection of Message duplication or replay: A malicious agent may record a legitimate conversation and later play it back to disguise its identity. Agents should be able to detect and prevent such playback security attacks;

Non-repudiation of messages: An agent should be accountable for the messages that they have sent or received (i.e., they should not be able to deny having sent or received a message);

Prevention of message hijacking: A rogue agent should not be able to extract the authentication information from an authenticated message and use it to masquerade as a legitimate agent.

The architecture proposed is a basic security model, which basically supports authentication of sender message integrity and privacy of data. An enhanced security model should additionally provide non-repudiation of origin, proof of sending and protection from message replay attacks, including support to frequent change of encryption keys to protect from cipher attacks. [20, 22, 23]

6.1 Proposed KQML Improvements

In order to implement this security architecture, the S-KQML model proposes several new KQML parameters and some modifications to a proposed standard ontology for agents. [20, 22]

It is assumed that KQML-speaking agents use a basic agent ontology which provides a small set of classes, attributes and relations helpful in talking about agents, their properties and the relationships and events in which they partake.

Assuming this ontology, the S-KQML architecture introduces a new sub-class of agent (*authenticator*) and a new relation (*key/5*) which describes a key used by an agent. [20]

An instance of this sub-class specifies a key that the sending agent will use in secure communication with the receiving agent and a flag into this sub-class is set if this is a master of a session key. If the destination agent was not defined, then the key is used by the sending agent to communicate with all other agents, characterizing the case of asymmetric

keys in this case. It is assumed that agent addresses are represented in this ontology with an *address/3* relation that contains the destination agent identification, the transport protocol and the destination address. [23, 24]

Some addresses are known by special agents, such as *agent name servers* and *authenticator agents*.

6.2 New KQML parameters and *performatives*

Several new KQML parameters are required to implement the proposed security architecture: [20]

The *digest-type* specifies the hashing function used (e.g. MD5) to compute the message's digest. The *encrypted-digest* is the message's digest encrypted using the key specified by the *:auth-key* parameter. This parameter should be present to prevent message hijack and to provide sender authentication and integrity assurance. [23]

The parameter *:auth-key* specifies the key being used to encrypt any *:auth-digest* parameters present. If the first element of the triple is true then the master key is used, otherwise, the session key is used.

The following new KQML performatives — or KQML messages — were also added to standard KQML in order to allow the implementation of this architecture: [8]

`auth-link:` The message sender wishes to authenticate itself to the receiver and set up a session key and message ID for a secure connection using this performative.

`auth-challenge:` The sender challenges the identity of the receiver in response to an *auth-link.* The sender then encrypts a random string using the master key $K_{s,r}$ or K_s and sends it as a *:content.*

`auth-private:` When the sender is posting a confidential message to the receiver, the content parameter contains the encrypted message and the *auth-key* parameter specifies the encryption key. The *:auth-digest* parameter should be present to verify the identity of the sender and the *:auth-msg-id* and *:auth-key* parameters may be present if an enhanced security model is required.

6.3 The Secure-KQML Model

The implementation of S-KQML should support a protocol with authentication, integrity and privacy of data in transit features to conform to the basic security model. If asymmetric keys are used for session and master keys, this model also supports non-repudiation of origin. [20]

When *Agent_A* sends a secure message to *Agent_B*, it would compute a message digest and encrypt it using the master key (as indicated by the value *K* for the *:auth-key* parameter). [22]

```
<performative>
   :sender Agent_A
   :receiver Agent_B
   :auth-key K
   :auth-digest (<digest-type>
   <encrypted-digest>)
```

Alternatively, if *Agent_A* needs to send a confidential message to *Agent_B*, it can encrypt the message and embed it in an *auth-private* performative, like shown below:

```
auth-private
   :sender Agent_A
   :receiver Agent_B
   :auth-key K
   :auth-digest      (<digest-type>
   <encrypted-digest>)
   :content          <encrypted-KQML-
message>
```

This proposed model can be used when the *:sender* does not know the *:receiver* in advance, e.g., for messages to be broadcasted, routed by some other specific agent or if *Agent_A* and *Agent_B* do not require prevention of message replay and can afford the cost of using the master key during all the communication session, for instance.

In the previous message, the *:auth-digest* parameter can be used to verify the integrity of the message, authenticate the sender and ensure non-repudiation of origin (if the master key is asymmetric). If the message has been corrupted, the message digest will not agree with the value of the *:auth-digest* parameter. Since the message digest is encrypted with the master key of *Agent_A*, only itself or the agents with which the sender shares the encryption key could have generated the message.

If the master key is an asymmetric key, only the message sender could have generated the message, as only the sender knows the private key that has been used for encryption. Note that this method can only verify the identity of the generator (i.e. if the message was encrypted by the sender agent of the message). This message can be a replay of a legitimate message previously sent by the generator.

6.4 Limitations of the Secure KQML Model

The secure model that was proposed for KQML has a number of limitations which now are going to be briefly enumerated. [11] *apud* [20]

Credentials: This model does not provide a mechanism to exchange credentials, that is, for one agent to empower another to act on its behalf;

Non-repudiation of receipt: This model does not support non-repudiation of message receipt. This can be a very useful capability, but would be difficult to implement due to the asynchronous nature of KQML and can be done only at the application level;

Messages to unknown receivers: Although one enhanced security model could support message replay detection, the proper use of the *:auth-msg-id* parameter is required. This requires that the recipient is known in advance. One of the essential features of KQML is the use of facilitator class agents (e.g. brokers and proxy agents to automatically route messages which intended recipients are described in general terms by the sending agent);

Stateless: The security architecture requires that agents maintain state information. The agents can choose not to use this feature if they are not concerned with message replay attack and cipher attack;

Crypto-awareness: An agent can send out authenticated messages if and only if it has crypto capabilities;

Constraints on delivery: Messages delivery must be reliable and in order. (A fair limitation considering that KQML itself assumes that);

Use of recommended APIs: The model should be enhanced to support the use of the Crypto APIs recommended by NSA (National Security Agency), especially for the `key-type` and `digest-type` values, due to cryptography export international regulations.

Some of these limitations origin from the basic features of KQML; others, according to other secure KQML implementations and ACLs researches, can be lived with and the rest could be addressed by if required by updating the KQML architecture or substituting this ACL by FIPA ACL (Foundation for Intelligent Physical Agents Communication Language), for example, which is a framework specification for agents' communication and management nowadays. For further details about FIPA and its development, visit its web site at http://www.fipa.org.

The proposed KQML security model addresses privacy, authentication and non-repudiation (if asymmetric key mechanism is used for the master and session keys) in agent communication. It does not fully address the issue of message replay, especially if the recipient of a KQML performative is not known in advance. Ultimately, this security model depends on definitive approval on default ACL by the research and industry communities, which tends to be FIPA, instead of DEEPSIA's KQML suggested implementation. [12, 26]

Nowadays, the current FIPA specifications contain minimal support for agent security and several other research groups have reported adding security to FIPA based MASs, most of them are adopting encryption-based mechanisms to protect their systems. Two key architectural elements are added: a secure channel to provide message privacy and a certification authority (CA) to provide authentication. [15, 16, 27, 28].

For further information about FIPA MAS Security progress, refer to the FIPA Security Workgroup website at the URL http://www.fipa.org/activities/security.html

7 Focusing on Communication Security: the Split and Merge approach

One of the issues of research focus is the way of securing communication within the system.

It has been shown that the traditional approaches to message encryption (who rely strongly on the lack of computational power to perform complex mathematical tasks) are being undermined by the increasing of microprocessor speed (Fig. 3).

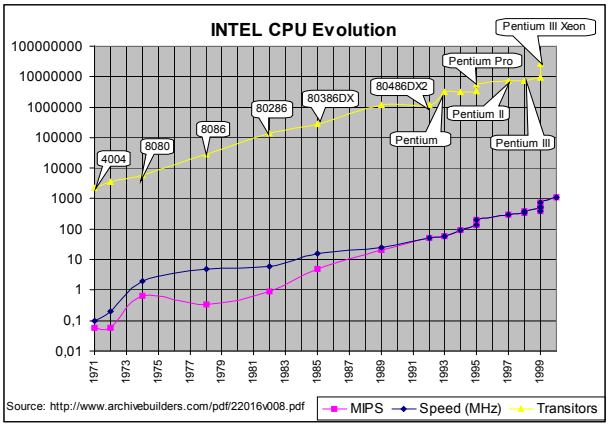

Fig. 3: Evolution of INTEL CPUs

It is believed that with the promised increase of computational capacity (e.g. quantum computing) the solution should not rely on mathematically complex algorithms.

To this end, UNINOVA has been developing a method ("Split and Merge") as an alternative to the methods currently in use. [25]

In the Split and Merge algorithm, the security of the message does not rely on the ability of being able to decipher the message, but on the ability to get the message itself. It does not deal with obscuring the contents of the message, but in splitting the message in parts (as many and as small as whished) so that the possession of a part does not give information about the content of the message. The next key concept has to do with routing different pieces of the message through different paths, while going from the source to the destination. The routing of each piece is done randomly so that the paths followed by each piece are probably different. The purpose of routing the pieces through different paths is to decrease the possibility of a perpetrator compromising all possible nodes on the route of the message. Further details of this algorithm can be found on [25].

The basis for the security of the algorithm is on the small probability an attacker has of being able to secure enough nodes to guarantee catching enough parts of the message to enable its understanding.

The work currently under way is on the definition of a mathematical model for the system, in order to give a network of nodes, determine which (and how many) nodes must the attacker control to have access to a given percentage of a message sent by a given source node.

Once this model is defined, it will be possible to measure, for each different network, the gain of the system in terms of the effort needed to decipher the message. This effort should then be added (if cipher is used) to the effort needed to break the ciphering mechanism used.

It has been noticed that if a Cipher Block Chaining (CBC) method is used and the attacker does not hold the first block of the message, the message is undecipherable even in possession of the key. This may primarily be used for improving efficiency of the algorithm by using Split and Merge only for the first block and sending the rest of the message directly to the destination.

8 Conclusions and Final Remarks

This paper presented the status of the European Commission's IST DEEPSIA project, regarding to its security researches. It describes the system being developed as a tool to assist in *e-procurement* and, due to its dealing with critical information; a high security level is required. It briefly describes the architecture of the system (that uses a multi-agent-system), going into some detail as the components of the system, their functionalities and interactions are described. The focus is then on the underlying technology (Software Agents) and on the security threats such systems suffer.

The paper concludes with a description of the efforts currently under way by the Brazilian and Portuguese project teams with regards to, respectively: a security enhancement of the KQML language currently being used and a future FIPA adoption and a specification of a new method to deal with protection of privacy of the messages being exchanged.

Acknowledgements

We would like to express our gratitude to all members of DEEPSIA Consortium (through IST-1999-20483) and the Brazilian funding agency CNPq (Process n° 680263/01-2), that supports the Brazilian research group.

References

[1] Howe, D. "E-Commerce" The Free On-line Dictionary of Computing, 1993-2002 <http://www.dictionary.com>

[2] Power, R. "2002 CSI/FBI Computer Crime and Security Survey", in Computer Security Issues and Trends, Vol. VIII, Nº 1, Spring 2002.

[3] DEEPSIA Consortium. Technical DEEPSIA Annex 1: Description of Work. July, 2000. 87 p. Report. IST PROJECT-1999-20483.

[4] Garção, A.S.; Sousa, P.A.; Pimentão, J.P.; Santos, B.R.; Blazquéz, V.; Obratanski, L. Annex to DEEPSIA's Deliverable 4 — System Architecture. January, 2001. 135p. Report. IST PROJECT-1999-20483.

[5] Sousa, P., Pimentão, J., Garção, A., "DEEPSIA - From Supply Chains to Supply Webs", in Intelligent Engineering Systems through Artificial Neural Networks, Cihan H. Dagli, Anna L.Buczak, Joydeep Ghosh, Mark J. Embrechts, Okan Ersoy, StephenHercel, Volume 11, ASME PRESS, NEW YORK, ISBN 0-7918-0176-4, 2001, pp. 1019-1024.

[6] Sousa, P., Pimentão, J., Garção, A., "DEEPSIA – Focusing E-Commerce on the Purchaser's Side", in International ICSC Congress on Computational Intelligence: Methods and Applications (CIMA'2001), Ludmila l. Kuncheva, Friedrich Steimann, Christian Haefke, Mayer Aladjem, Vilem Novak, ICSC Academic Press, Canada, ISBN 3-906454-26-6, 2001, pp. 436-442.

[7] Finin, T.; Weber, J. "Specification of the KQML Agent-Communication Language". The DARPA Knowledge Sharing Initiative, 1993.

[8] Finin, T.; Labrou, Y. "A Proposal for a new KQML Specification". University of Maryland Baltimore Count (UMBC). Baltimore, 1997.

[9] Jansen, W.; Karygiannis, T. "Mobile Agent Security". Technical Report, National Institute of Standards of Technology (NIST). Baltimore, 1999.

[10] Fuggetta, A.; Picco, G. P.; Vigna, G. Understanding Code Mobility, IEEE Transactions on Software Engineering, 24(5), May 1998, pp. 342-361.

[11] Uto, N.; Dahab, R. "Segurança de Sistemas de Agentes Móveis", in Anais do III Simpósio de Segurança em Informática, São José dos Campos, 2001.

[12] FIPA Agent Management Specification (Experimental). Foundation for Intelligent Physical Agents, 2000. <http://www.fipa.org/specs/fipa00023/>

[13] White, J.E. "Mobile Agents". In: Software Agents, J.M. Bradshaw (Ed.), Menlo Park, Calif., AAAI Press, 1997.

[14] Jansen, W. "Countermeasures for Mobile Agent Security". In Computer Communications, Special Issue on Advances in Research and Application of Network Security, November 2000.

[15] He Q, Sychara K, Finin T. "Personal Security Agent KQML-based PKI". Proceedings of (AA'98) Autonomous Agents, 1998.

[16] Poggi A, Rimassa G and Tomaiuolo M. "Multi-User and Security Support for Multi-Agent Systems". Proceedings of WOA 2001 Workshop, Modena, September 2001.

[17] RFC 1421: "Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures", 1993. <http://www.ietf.org/rfc/rfc1421.txt>

[18] Common Object Request Broker Architecture Security Suite, 2000. <http://www.omg.org/technology/documents/formal/omg_security.htm>

[19] "Security in the Distributed Computing Environment. Security on the Web Using DCE" Technology, Document Number SG24-4949-00. IBM Corporation, 1997. <http://www-3.ibm.com/software/network/dce/library/redbooks/sg244949/4949c112.htm>

[20] Thirunavukkarasu, C.; Finin, T.; Mayfield, J. "Secret Agents – A Security Architecture for the KQML Agent Communication Language". In Intelligent Information Agents Workshop held in conjuntion with Fourth International Conference on Information and Knowledge Management. Baltimore, 1995.

[21] Spafford, G.; Garfinkel, S. Practical UNIX & Internet Security. O'Reilly & Associates. 2 nd ed. April, 1996.

[22] Milagres, F. G.; Moreira, E. S. "Especificação de Segurança na Comunicação de Agentes" ("Specification of Security on Agent Communication Languages"), *In* Módulo Security News number 251 and on-line at Módulo Security Solutions web site, Academic Research category. Módulo Security Solutions S. A. <http://www.modulo.com.br/pdf/milagres-deepsia_security.pdf>. São Paulo. July 2002.

[23] Schneier, B. Applied Cryptography, Second Edition: Protocols, Algorithms, and Source Code in C. John Wiley & Sons, Inc., 2nd ed. January, 1996.

[24] Diffie, W.; Hellman, M. E. "New Directions in Cryptography", IEEE Transactions on Information Theory 22 (1976), 644-654.

[25] Pimentão J, Sousa P, Garção A, "Split and Merge - An Algorithm to Implement Security on the Internet", in Communications World, N. Mastorakis ed., WSES Press, 2001.

[26] FIPA MAS Security White Paper, Version 1.6. Foundation for Intelligent Physical Agents Security Workgroup, 2002. <http://www.fipa.org/docs/output/f-out-00113/>

[27] Zhang M, Karmouch A and Impey R. "Towards a Secure Agent Platform based on FIPA". Proceedings of MATA 2001. Springer-Verlag. LCNS, (2001), Vol. 2164, 277-289.

[28] Hu Y-J. "Some Thoughts on Agent Trust and Delegation". Proceedings of 5th Int. Conf. on Autonomous Agents (AA2000), Montreal, 2000. 489-496