An Enhanced Authentication System for the JADE-S Platform

Vincenzo Conti[1], Salvatore Vitabile[2], Giovanni Pilato[2], and Filippo Sorbello[1,2]

[1]Dipartimento Ingegneria Informatica - University of Palermo,
Viale delle Scienze, 90128 Palermo, Italy

[2]I.CA.R. - Italian National Research Council
Viale delle Scienze, 90128 Palermo, Italy

Abstract. Multi-agent systems require a set of security mechanism such as confidentiality, privacy, authentication, data integrity, access control. Authentication is the process of deducing which user has made a specific request in a multi-agent system. The goal of authentication is to allow legitimate system entities to process while denying illegitimate ones. The JADE-S platform requires that users must be authenticated, providing a username and password, in order to be able to perform actions on the platform. In this paper an enhanced user authentication system for the JADE-S platform, requiring user fingerprint in addition to the standard items, is outlined. Due to their uniqueness and immutability, fingerprints can be successfully used in personal authentication systems. JADE-S authentication system has been modified integrating the proposed approach. The first security access level verifies username and password supplied by the user. The second security access level, requires user fingerprint and compares it with the related item stored in a fingerprints database. A public/private key pair is used to sign each database fingerprint, so the system is able to check against the authenticity of the signed fingerprint before matching.

Key Words: - Multi-Agent Systems – Security – Fingerprints – Authentication Systems – JADE-S

1 Introduction

Multi-agent systems require a set of security mechanism, policies and trust models [20], [9], [19], [21]. Security mechanisms are a set of techniques used to address concrete threats such as confidentiality, privacy, authentication, data integrity, access control, etc.

Authentication is the process of deducing which user has made a specific request, such as register an agent in a multi-agent system (MAS) or require a service to one or more registered agents. The goal of authentication is to allow legitimate system entities to process while denying illegitimate ones. Closely related to the trust relationship between the human beings that the entities represent, authentication is performed according to recognized protocols.

Several works were proposed in the mobile agent security domain in the last years [23], [24], [25]. In most of these papers, user and agent authentication process has been considered a basic security mechanism against common security treats, such as masquerading, eavesdropping, spoofing, denial of service etc.

On the other hand, due to their uniqueness and immutability, fingerprints have been widely used in personal authentication tasks [10], [11], [12], [13]. Since fingerprint features do not change, many human identification systems have been based on them. A fingerprint is composed by ridges and valley, having a curvature value. Ridges may have bifurcations or endpoints. These meaningfully points, called minutiae, are used to identify a person since they change from person to person. Each minutia has a spatial position, a direction and can belong to a class. The National Institute of Standards and Technology (NIST), for example, has proposed five standard classes: Right Loop, Left Loop, Whorl, Arch and Tented Arch. The 5 classes have been fixed using the existence and the spatial position of two singular points, called Core and Delta [18].

JADE-S, the JADE Secure Agent Platform [16], is one of the most common secure MAS implementation. JADE-S is formed by the combination of the standard version of JADE with the new JADE security plug-in [16], [8]. It includes features such as user/agent authentication, authorization and secure communication between agents into the same platform.

The main features of JADE-S platform are the following [8]:

- *Authentication*: an user must be authenticated, providing a username and password, to be able to own or perform actions on a component of the platform. Only authenticated users can own AMS, DF, containers and other agents;
- *Authorization*: JADE-S uses the concept of Principal as an abstraction for a user account, an agent or a container. A Principal must be authorized by the Java security manager. The security manager allows or denies the action according to the JADE platform's policy;
- *Permissions and Policies*: a permission is an object that describes the possibility of performing an action on a certain resource such as pieces of code, but also who executes that code. A policy specifies which permissions are available for various principals;
- *Certificates and Certification Authority*: the Certification Authority (CA) is the entity that signs all the certificates for the whole platform, using a public/private key pair. When the CA signs a document, it first makes a digest of it, which is a shorter non-reversible version of the document, a kind of a checksum. Then the digest is encrypted with the private key;
- *Delegation*: this mechanism allows the "lending" of permissions to an agent. Besides the identity certificate, an agent can also own other certificates given to it by other agents;
- *Secure Communication*: communication between agents on different containers/hosts, are performed using the Secure Socket Layer (SSL) protocol. This enables a solid protection against malicious attempts of packet sniffing.

Poggi et al. [7] consider agent owner authentication the first step to assign access permission to various agents executing on the system. After the authentication process based on the pair username and password, a user receives back a certification of his/her identity along with granted permissions.

In this paper an enhanced access system for the Jade-S platform is proposed. User identity is fixed after the new authentication process requiring username, password and fingerprint. Initially, the system verifies username and password supplied by the user and successively it requests his fingerprint. The fingerprint, acquired through a sensor, is compared with the related item stored in the fingerprint database.

A public/private key pair is used to sign each fingerprint stored in the database, in order to check against the authenticity of the signed stored fingerprint before the matching phase. Fingerprint matching is performed via a set of algorithms proposed by the authors [1], [2] exploiting both the adaptive Local Energy Threshold (LET) and a new matching operator, based on the Tanimoto distance.

The paper is organized as follows. In the section 2 the fingerprint verification process proposed by the authors is described; in the section 3 the proposed authentication system is described, in section 4 the implementation of the enhanced authentication system in the JADE-S platform is outlined. Finally, in section 5 the conclusions are reported.

2 Fingerprint Verification

Fingerprints are made by a set of lines having endpoints and bifurcations (see Figure 1-a). Fingerprints are used in two different kind of system for establishing person identity: verification and identification. Identification systems must establish the identity of a person comparing a processed fingerprint with each fingerprint image, stored in a database. Verification systems have to recognize an acquired fingerprint image using personal data information to select a fingerprint database item. A fingerprint matching is successively performed between the acquired fingerprint image and the selected database item.

A verification system able to process the NIST 4 fingerprint database with good results was proposed in [1], [2] by the authors. Fingerprint verification was performed in three main phases: the pre-processing phase, the minutiae extraction phase, and the matching phase.

The pre-processing phase aims to obtain a binary image containing a set of ridges whose thickness is only one pixel. The whole phase is based on the automatic computation of the Local Energy Threshold (LET) for fingerprint image binarization. The LET is the average energy of the eight neighbors of each processed pixel. If p_i is the brightness of the central pixel of a 3x3 mask, the new value for p_i is calculated as follows:

$$p_i = \begin{cases} 255 & \frac{1}{8}\sum_{\substack{k=0 \\ k\neq i}}^{8} p_k \geq LET \\ 0 & \text{otherwise} \end{cases} \qquad (1)$$

The minutiae extraction phase aims to obtain a n-dimensional vector, where n is the number of the extracted minutiae and the vector components are the spatial coordinates and the ridge direction of the extracted minutiae.

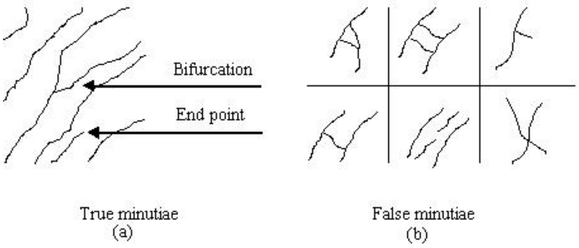


Fig.1: (a) true fingerprint minutiae, such as endpoint and bifurcation, (b) some recurrent false features, such as triangles, structure ladders, spurs, bridges, interrupted ridges and forks.

The considered features are only endpoint and bifurcation, called true minutiae, while other elements as triangles, structure ladders, spurs, bridges, interrupted ridges and forks are considered false minutiae and, consequently, erased. Figure 1 shows both the real and false minutiae.

The matching phase aims to verify if a processed fingerprint pairs belong to the same person. The matching rate is given by an operator that extends the concept of the Tanimoto distance [15]. Let V the set of records *r(X, Y,* θ*)* representing the current image, W_i the set of records *r(X, Y,* θ*)* of the i-th database image:

$$F(W_j) \equiv V \therefore W_j \qquad (2)$$

where $\therefore$ is the intersection operator defined as follows:

$$k_i \in (V \therefore W_j) \\ \Leftrightarrow \\ \left\{ \left|X - X_i\right| \leq T_x ; \left|Y - Y_i\right| \leq T_y \,;\, \left|\theta - \theta_i\right| \leq T_\theta \right\} \qquad (3)$$

with T_x, T_y, T_θ noisy (translation and/or rotation) immunity thresholds and *X*, *Y*, and θ the spatial coordinates and the ridge direction, respectively.

With more details, the main phases of fingerprint verification process are:

1. *the preprocessing phase*
   a. directional image extraction;
   b. fingerprint image segmentation;
   c. fingerprint image binarization, based on LET;
   d. fingerprint image median filtering;
   e. fingerprint image thinning;
2. *the minutiae extraction phase*
   a. false minutiae erasing process;
   b. true minutiae extraction process;
   c. fingerprint image codifying process through an n-dimensional vector;
3. *the matching phase.*

In Figure 2 are depicted the obtained results after the preprocessing phase and the minutiae extraction phase.

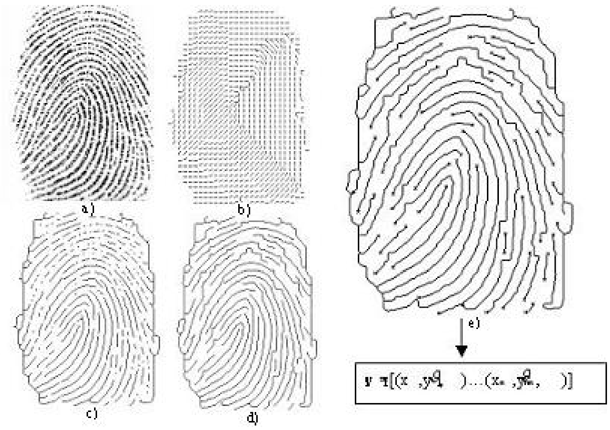


Fig.2 The obtained results after the pre-processing phase and the minutiae extraction phase: (a) the original image, (b) the extracted directional image, (c) the final result of pre-processing phase, (d) and (e) the extracted minutiae and the related vector v. The vector will contain two spatial coordinates and the direction of each extracted minutia.

3 The proposed Fingerprint based Authentication System

Due to their uniqueness and immutability [10][11][12][13], fingerprints can be used in personal authentication systems. A Fingerprint Verification System (FVS) can be added to existing MAS authentication systems to enhance the access security level.

In many common MAS, the users authentication process is only based on username and password. The current version of the JADE-S platform, a platform formed by the combination of the standard version of JADE with the JADE security plug-in, has a user authentication system that is only based on username and password. Only authenticated users can own AMS, DF, containers and other agents.

The proposed FVS is based on the algorithms described in the previous section, adapted to process fingerprint images acquired through a sensor. The system requires two phases: the first one deals with the system setup in which users fingerprint are first signed, via a public/private key pair, and then stored in the related fingerprint file; the second phase deals with fingerprint pair verification to access in the system. The block diagram describing the whole system is depicted in Figure 3.

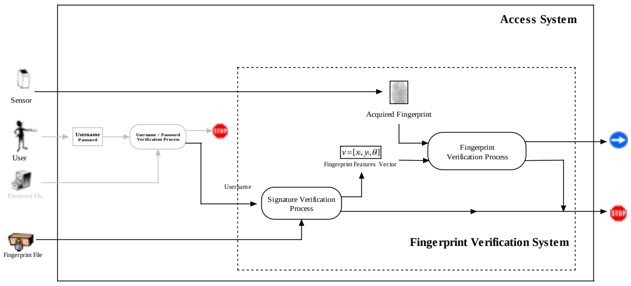


Fig. 3: The proposed Fingerprint based Authentication System

3.1. The Fingerprint Signature Process

Each user enabled to access in the system must be previously registered. In the registration phase, system administrator assigns him/her a username/password pair and stores his/her own fingerprint. So user authentication is performed against two files: the *password file*, containing the username/password pair and the *fingerprint file*, containing the username/fingerprint features pair.

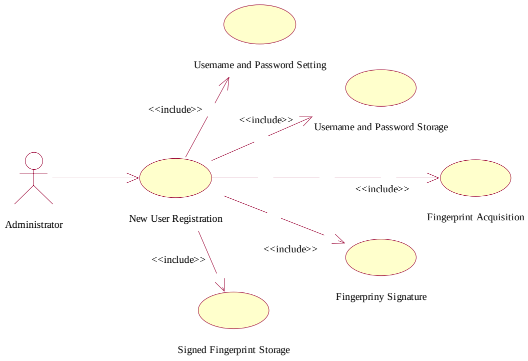


Fig.4: The UML Use Case Diagrams related to a new user registration.

Fingerprint is described by a vector, which elements are minutiae spatial position and minutiae orientation; the vector is signed using a public/private key pair before to store it in the fingerprint file. The solution allows to check fingerprint integrity before the matching process.

Figure 4 shows the UML Use Case Diagrams related to a new user registration: for each user the administrator needs of username, password and fingerprint. User fingerprint is signed before its storage. Figure 5 shows the UML State and Activity Diagram related to the user registration: after the chosen username and password, the user will supply its own fingerprint.

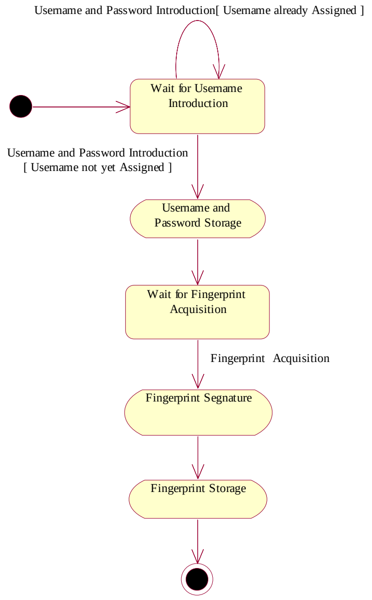

Username and Password Introduction[Username already Assigned]
Wait for Username Introduction
Username and Password Introduction
[Username not yet Assigned]
Username and Password Storage
Wait for Fingerprint Acquisition
Fingerprint Acquisition
Fingerprint Segnature
Fingerprint Storage

Fig.5: The UML State and Activity Diagram related to the user registration

3.2. The Fingerprint Verification System

Each user is identified by his/her *username*, *password*, and *fingerprint* assigned him/her in the registration phase. When a user triggers an access request, the platform verifies username and password supplied by the user and successively it requests his/her fingerprint. The fingerprint, acquired through a sensor, is compared with the related item stored in the fingerprint database. The fingerprint matching process is performed via a set of algorithms proposed by the authors [1], [2] exploiting both the adaptive Local Energy Threshold (LET) and a new matching operator, based on the Tanimoto distance [15]. The matching process phases are shown in Figure 6.

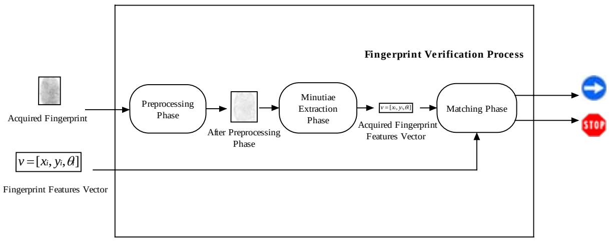

Fingerprint Verification Process
Acquired Fingerprint
Preprocessing Phase
After Preprocessing Phase
Minutiae Extraction Phase
$v=[x_i, y_i, \theta_i]$
Acquired Fingerprint Features Vector
Matching Phase
STOP
$v=[x_i, y_i, \theta_i]$
Fingerprint Features Vector

Fig.6 : This figure shows the phases of the Fingerprint Verification Process.

Figure 7 shows the UML Use Case Diagrams related to an user access request: for each request username and password matching is first performed and then user fingerprint is acquired and matched with the stored ones. Before fingerprint pair matching, the stored fingerprint integrity is verified.

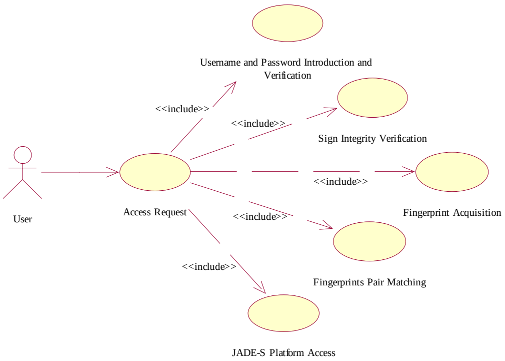


Fig.7: The UML Use Case Diagrams related to an user access request.

Figure 8 shows the UML State and Activity Diagram related to the user access request: the JADE-S platform will run if and only if both the (*username*, *password*) pair and the (*username*, *fingerprint*) pair match with the stored ones.

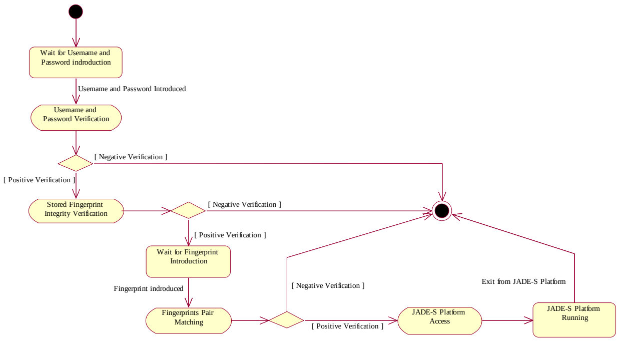

Wait for Username and Password indroduction
Username and Password Introduced
Username and Password Verification
[Negative Verification]
[Positive Verification]
Stored Fingerprint Integrity Verification
[Negative Verification]
[Positive Verification]
Wait for Fingerprint Introduction
Fingerprint indroduced
[Negative Verification]
Exit from JADE-S Platform
Fingerprints Pair Matching
[Positive Verification]
JADE-S Platform Access
JADE-S Platform Running

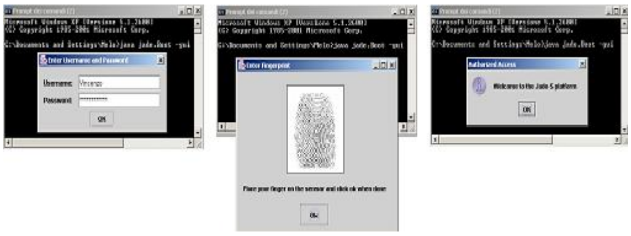
Fig.8: The UML State and Activity Diagram related to the user access request.

4 Experimental Trials

We have modified the platform start-up phase introducing two security level: a first security level in which username/password are requested and a second security level in which user fingerprint are requested. In JADE-S the user who starts-up the platform initially owns only the AMS, the DF and the main container [8]. Using the modified user authentication system, JADE-S platform can run only when the entire verification procedure is correctly performed: the main container will be opened and the RMA, DF and AMS will be activated. The SecuGen Eyed Hamster sensor [9] has been used to acquire users fingerprint.

Specifically, the platform access procedure is the following:

1. when an user is going to run the JADE-S platform, the system requires username/password pair and verify it (first security level in Figure 9-a);
2. if the username/password verification result is positive and the related signed fingerprint has not been modified and replaced, the user fingerprint will be acquired by the sensor;
3. the acquired fingerprint will be compared with the stored ones (second security level in Figure 9-b) using the authors developed algorithms [1], [2];
4. if the fingerprint matching process gives negative result, the access will be denied, otherwise the JADE-S platform will be launched (see Figure 9c).



```
Microsoft Windows XP [Versione 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Melo>java jade.Boot -gui
```

Enter Username and Password

Username: Vincenzo

Password: ************

Enter Fingerprint

Place your finger on the sensor and click ok when done

Authorized Access

Welcome to the Jade-S platform

Fig.9: In the figure is shown the whole access platform procedure. In (a) the first security level with username/password requesting is shown; in (b) the second security level with the user fingerprint requesting is shown; finally in (c) a welcome access window is shown.

The JADE-S platform integrated with the proposed access system was installed on a Intel 1,13 GHz Pentium III processor, with 384 Mb SDRam, under MS Windows XP operating system. Fingerprint image size is 300x260 pixels. The whole user authentication phase requires 2,59 s. The most of the processing time (2,32 s) is required by the fingerprint extraction minutiae process.

5 Conclusions

The JADE-S platform requires that users must be authenticated, providing a username and password, in order to be able to perform actions on the platform. In this paper an enhanced user authentication system for the JADE-S platform, requiring user fingerprint in addition to the standard items, has been outlined. Due to their uniqueness and immutability, fingerprints can be successfully used in personal authentication systems. JADE-S authentication system has been modified integrating the proposed approach. The first security access level verifies username and password supplied by the user. The second security access level, requires user fingerprint and compares it with the related item stored in a fingerprints database. A public/private key pair is used to sign each database fingerprint, so system is able to check against the authenticity of the signed fingerprint before matching.

References

[1] V. Conti, G. Pilato, S. Vitabile, F. Sorbello, "A Robust System for Fingerprints Identification", Proc. of Knowledge-Based Intelligent Information Engineering System & Allied Technologies, September 2002, pp. 1162-1166.

[2] V. Conti, G. Pilato, S. Vitabile, F. Sorbello, "Verification of Ink-on-paper Fingerprints by Using Image Processing Techniques and a New Matching Operator", Proc. of VIII Convegno AI*IA, September 2002.

[3] S. Vitabile, V. Conti, G. Pilato, F. Sorbello (2003). "A Fingerprint Based Authentication System for the JADE-S Platform", Agentcities ID3, Feb. 6-8, 2003, Barcelona, Spain.

[4] G. Pilato S. Vitabile, V. Conti, G. Vassallo, F. Sorbello (2003). "A Concurrent Neural Classifier for HTML Documents Retrieval", Proc. of 14-th Italian Workshop on Neural Nets (WIRN 2003), June 2003 (in press).

[5] G. Pilato S. Vitabile, V. Conti, G. Vassallo, F. Sorbello (2003). "A Neural Multi-Agent Based System for Smart Html Pages Retrieval", Proc. of 2003 IEEE/WIC International Conference on Intelligent Agent Technology (IAT 2003), October 2003 (in press).

[6] F. Bellifemmine , A. Poggi , G. Rimassa. "JADE - A FIPA compliant agent framework", 4th International Conference and Exhibition on the Pratical Application of Intelligent Agents and Multi-Agents, UK, 1999.

[7] A. Poggi, G. Rimassa, M. Tomaiuolo "Multi-User and Security Support for Multi-Agent Systems", AI*IA – TABOO, Modena 4-5 September 2001.

[8] G. Vitaglione, "Jade Tutorial Security Administrator Guide", http://sharon.cselt.it/projects/jade/doc/tutorials/SecurityAdminGuide.pdf.

[9] H. Chi Wong, K. Sycara, "Adding Security and Trust to Multi-Agent System", Proceedings of Autonomous Agents '99.

[10] A. Jain, L. Hong, R. bolle: "On-Line Fingerprint Verification", IEEE Transaction on pattern Analysis and Machine Intelligence, Vol. 19, No. 4, April 1997.

[11] A. Jain, L. Hong, S. Pankanti, R. Bolle: "An Identity-Authentication System Using Fingerprints", Proceedings of the IEEE, Vol. 85, No. 9, September 1997.

[12] Z. M. Kovacs-Vajna: "A Fingerprint Verification System Based on Triangular Matching and Dynamic Time Warping", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 22, No. 11, November 2000.

[13] X. Tan, B. Bhanu: "Robust Fingerprint Identification", IEEE International Conference on Image Processing, 2002.

[14] M. Ballan, F. Ayhan Sakarya, "A Fingerprint Classification Technique Using Directional Images", *IEEE*, 1998.

[15] K. R. Sloan Jr., Steven L. Tanimoto, Progressive Refinement of Raster Images, IEEE Transactions on Computers, Volume 28, Number 11, pp. 871-874, November 1979.

[16] http://sharon.cselt.it/projects/jade

[17] http://www.secugen.com/home.htm

[18] http://www.nist.gov/srd/nistsd4.htm

[19] Sycara, K., Multi-agent Infrastructure, Agent Discovery, Middle Agents for Web Services and Interoperation. In Lecture Notes in Artificial Intelligence: Multi-Agent Systems and Applications, pp. 17-49, Springer-Verlage Berlin Heidelberg New York, ISBN 3-540-42312-5, 2001.

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

[21] Jansen, W., Karygiannis, T.: NIST Special Publication 800-19 - Mobile Agent Security. National Institute of Standards and Technology, 2000.

[22] S. Poslad & M. Calisti, Towards improved trust and security in FIPA agent platforms, Autonomous Agents 2000 Workshop on Deception, Fraud and Trust in Agent Societies, Barcelona, June 2000.

[23] Corradi, A., R., Montanari and C., Stefanelli. Security issues in mobile agent technology. Distributed Computing Systems, 1999. Proc. 7th IEEE Workshop on Future Trends of distributed computing systems, 3 -8, (1999).

[24] L., Korba. Towards Secure Agent Distribution and Communication, Proc. 32nd Hawaii International Conference on System Sciences, 10pp, (1999).

[25] Farmer, W.M., J. D. Guttman, and V Swarup. Security for mobile agents: Authentication and state appraisal. Proc. 4th European Symposium on Research in Computer Security, Springer-Verlag Lecture Notes in Computer Science No. 1146, pages 118-130, (1996).