Annals of the University of Craiova, Mathematics and Computer Science Series
Volume 38(2), 2011, Pages 59–68
ISSN: 1223-6934, Online 2246-9958

Authentication model based on Multi-Agent System

Nicolae Constantinescu and Claudiu Ionut Popirlan

Abstract. In any communication process the most important stage is represented by the authentication. If this stage ends successfully, the authenticated part can perform all operations for which it was given permission. So, along with the authentication, this process also informs the authenticated part what are its rights and its restrictions on the system. We propose an authentication model based on multi-agent system (MAS) structure. There is a master agent (MA) which receives the authentication request from the user, and multiple slave agents(SA) that, after solving a fixed number of tasks, they will send back the result to MA, which will decide if the authentication ends successfully. The structure of such a system and all the agent's tasks are presented in details. We apply the agent concept to facilitate the authentication process in order to work with multi-clients more dynamically and efficiently.

2010 Mathematics Subject Classification. Primary 94A62; Secondary 68T42.
Key words and phrases. authentication, cryptosystem, multi-agent system, master/slave architecture, JADE implementation.

1. Introduction

The safety of all the communication systems relies on the authentication stage. If this stage does not end successfully the process does not begin. Authentication is a process for identifying and verifying who is sending a request [13]. Because of its importance there have been developed a variety of authentication mechanisms. These mechanisms are divided into three categorizes:

(1) authentication based on *Something you are.*
(2) authentication based on *Something you know.*
(3) authentication based on *Something you have.*

The first category includes biometric methods, while the second includes authentication methods based on passwords. The third one consists in challenge-response lists, one-time pads, smart cards, hardware token etc.. In [4] the authors presented a fourth category based on social network of the user, *Somebody you know.* There are authentication systems which combine methods from different categories. The most popular example is the bank card. To unlock the card the user must type his personal identification number (PIN). The PIN is something that the user knows while the card is something he has.

A workable authentication system requires at least two modes of authentication, the primary mode and the emergency one. The second is used only when the first one is not available. To exemplify this, we take an e-mail account. Suppose the user has forgotten his password, which represents the primary authentication mode. So, he must try the emergency mode, which in most cases implies answering to some questions. The user must give the same answers he gave when the account was

Received January 11, 2011. Revision received April 21, 2011.

registered. In corporate environments, the emergency mode consists in telephoning the support personnel to re-establish the access privileges.

An authentication system security relies on the security of its weakest component. As the U.S. Government's National Information Assurance Glossary declares:

> " . . . strong authentication is layered authentication approach relying on two or more authenticators to establish the identity of an originator or receiver of information."

Over the past few years, a new concept has entered the field of computer science. This is the concept of an agent. Some authors have used this concept to unify the field of artificial intelligence [17, 9], while others see it as a new programming concept to extend the object concept [14]. The concept of an agent is strongly associated with localization [23]. Most discussions on agents focus on their autonomy, intelligence, mobility and interaction [18, 19]. In [17], multi-agent system (MAS) is defined as *a loosely coupled network of problem-solver entities that work together to find answers to problems that are beyond the individual capabilities or knowledge of each entity.* Agent-based systems [22] claim to be next generation software capable of adapting dynamically to changing business environment and of solving a wide range of knowledge processing application. Nevertheless, several issues still need to be faced to make the multi-agent technology widely accepted:

- Secure and efficient execution supports;
- Standardization;
- Appropriate programming languages and coordination models.

In this paper we propose an authentication model based on support provided by multi-agent systems. Technically, we apply Multi-Agent System concept as a mediator to:

- perform the authentication control of the relying entities having certificate;
- verify the client role and grant the permission to the legal applications;
- schedule multi clients requests autonomously and dynamically.

In the proposed model the agent concept facilitates the authentication process in order to work with multi-clients more dynamically and efficiently.

2. State of Art

Biometric authentication methods are very popular, the most important being by far fingerprint identification. Fingerprint identification is the most mature biometric method being implemented at an early level since 1960. The recognition of a fingerprint can be done with two methods: "one-to-one" (verification) and "one-to-many" (identification). The first method is applied when we have two fingerprints and we want to verify if they belong to the same person. The second one is used when we have one fingerprint and we search it in a database. The verification is much easier and faster because we have the two fingerprints and we just need to compare them. On the other hand, the identification implies more time for extracting the fingerprint because there are needed much more details. The authentication systems which use fingerprint methods may use identification or verification. For an authentication based on fingerprint identification the user does not have any other information. The fingerprint is scanned and is searched in the database. For an authentication based on fingerprint verification the user must have a username (or even his real name) and the fingerprint will be used as a password. If the fingerprint coincides with the one stored for that username the process ends successfully. If it does not coincide the fingerprint will not be compared with the other fingerprints from the database. The systems based on fingerprint verification can replace the username with "*Something you have*", for example a smart card. Because of its popularity, increasingly more PCs and other devices are released on the market with built-in fingerprint readers. Most of these products are not expensive at all. All these devices allow the user to not remember any password. The encrypted passwords are invoked once the user puts his finger on the reader. For a two-factor authentication there can be, also, used a secret password.

The fingerprint can also be replaced with other biometric data such as face recognition, hand geometry biometrics, retina scan, iris scan, signature dynamics, voice analysis, etc.. The methods for fingerprint identification and verification are the most advanced from the biometric methods. Retina scan and iris scan are also very important, and there is expected that the methods based on them will be increasingly used in the future exceeding even those based on fingerprints. There is no known way to replicate a retina. There is considered that the pattern of the blood vessels at the back of the eye is unique and stays the same for a lifetime. A big disadvantage is that it requires about 15 seconds of careful concentration to take a good scan. But with the rapid development of technology, devices and methods scanning and comparing retina will need less time. Retina scan remains a standard in military and government installations.

Another popular authentication method is the one based on username and password. The method is designed to allow a client program to provide credentials when making a request. First, the username is appended with a colon and concatenated with the password. the colon is used to separate the username and the password. The string resulted is encrypted with an encryption algorithm and transmitted. The receiver decrypts and obtains the username and the password [15]. This authentication system popularity is due to its easy implementation and the fact that it is supported by the most used browsers. A big disadvantage is represented by the fact that the security relies on the assumption that the connection between the client and the server is secure and can be trusted. An alternative to this method is the digest access authentication. The server can use this method to negociate credentials with a user. The user identity is established without sending a plaintext password. The method uses MD5 to hash the password, preventing cryptanalysis [16].

In spite of the fact that their use has been roundly criticized on many occasions, passwords remain the most popular method for user authentication [5]. The main problems are caused by the way they are used by the people that hold them. Users usually choose poor passwords based on private data which can be easily guessed by others. In [8, 7] the author presents some facts that the user must take in consideration when choosing a password:

(1) users must not choose poor passwords (passwords that are too short, passwords that are related to personal information and can be easily guessed by someone who knows them, passwords that are found in a dictionary and can be cracked by dictionary tools);
(2) users must not share their passwords with anyone else;
(3) users must not write passwords down or store them in unsafe places;
(4) users must periodically change their passwords and they must not revert to using the previous one again at a later time.
(5) users must not use the same password for a variety of systems because this may cause vulnerabilities to all the systems when one of them had been broken.

In [1] the authors present some types of authentication attacks which have been applied on systems with username/password authentication:
(1) Brute-force attacks
(2) Dictionary attacks
(3) Credential decryption attacks
(4) Replay attacks.

The classification was made after studying authentication models from papers like [1, 2, 10, 21].

Brute-force attacks means attempting to find valid passwords and usernames through exhaustive search. This attack is possible if the intruder gains access to the database storage (pairs of usernames and passwords). After retrieving this information, the intruder tries all possible combination. Because the information is encrypted will be used different encryption algorithms. Such attack can be done only by programs or machines since the amount of operations is very high. This attack may occur to most of the systems based on passwords and usernames. To avoid it the users must choose strong passwords. Besides that, a very important measure is to provide strong access polices in order to avoid information access for unauthorized persons (processes).

The dictionary attacks are a smarter version of the brute-force attacks. The difference is given by the fact that the search can be done without having access to any information. Such an intruder only needs a valid username. The eavesdropper verifies combinations with possible passwords. The process has a high probability of succeeding mostly because human nature. People tend to use as passwords short characters combination that are very easy to remember. A system which resists to such an attack must have an automatic setting which provides a limit of unsuccessful authentication. A human cannot try more than maximum 50 unsuccessful authentications. Usually, after at most 5 attempts he uses the emergency mode. A process which does a dictionary attack can try at least 200 successive unsuccessful attempts.

Credential decryption attacks try to break the encryption algorithm. These attacks are usually used as supplement for brute-force attacks and dictionary attacks. Vulnerable systems to such attacks are the ones that do not have strong credential polices and strong cryptographic algorithms. The encryption/decryption algorithm must have a high security level and must be correctly implemented.

Replay attack implies an intruder which tries to authenticate himself through valid authentication sequences trapped from the communication channel. The measures against such attacks are the same like the ones for the sniffing attack. All the transmitted data must be encrypted using a very strong cryptographic system. It can also be used a time stamp for all the messages.

Other important authentication methods are the cryptographic ones. One of the most important attacks upon a cryptosystem is the man-in-the-middle attack. Suppose there are two parties (A and B) who want to communicate and an intruder. The intruder will impersonate B, receiving and keeping all the messages from A and sending his own messages back. So A will communicate with the intruder without noticing that he is not communicating with the right person, B. The intruder will use the same method to impersonate A, and to communicate with B. Such an attack can be avoided using an authentication system before starting the communication process. The most popular cryptographic authentication methods are based on digital signatures. The digital signature can, also, be used for verifying the integrity of a document. A digital signature scheme consists in three algorithms:

(1) key generation algorithm
(2) signature generation algorithm
(3) signature verification algorithm

The most popular digital signature schemes are DSA, RSA, ElGamal and Schnorr. Because of the increasing usage of elliptic curve cryptography some of the classic digital signature algorithms have been adapted, using points defined over an elliptic curve instead of numbers [20]. The elliptic curve schemes provide the same security level as the classic ones, but the performance of the algorithms is highly increased due to the fact that shorter keys are used.

To the best of our knowledge, there are very few works dedicated to the integration issue of authentication model and multi-agent systems. The work proposed in [12] describe an authentication broker model that integrates the Single Sign-On and Multi-Agent system to satisfy the security requirements including confidentiality, integrity, and non-repudiation. The system factor and protocol of agent creation are presented with respect to the authority level of the corresponding service. Nevertheless, the clear function of agents used for delegating application to the authorized clients is not provided.

In [6] the idea and implementation of how to apply MAS technique to serve the authentication service in the multi-clients and multi-application environment are proposed. The design of user agent and application agent is introduced to perform the client authentication and multi-application delegation. Nevertheless, the system performance and resource consumption is not provided, and an integration of several types of agents, e.g. mobile agent can be adopted.

3. Our Model

We propose an authentication system based on multiple agents.

The agent system functions when each client requests to sign on and it is responsible for validating a client certificate, granting an access role to the client.

3.1. The Multi-Agent System Architecture. A mobile agents system, intended very general and flexible, to effectively improve the authentication process is proposed. The system architecture that includes the related components is described below and can be visualized in Figure 1, that presents the conceptual view of our proposed model.

The role of the view-related components (defined terms) is as follows:

Definition 3.1 (Master Agent (level 1))**.** *The Master Agent (MA) is the agent that coordinates the authentication process, receives the authentication request from the user, divides the tasks and sends to slave agents (SA^2), depending on the authentication method (biometric method, user/password method and digital signature method) and the degree of similarity of request (this feature shows intelligence of the system).*

Definition 3.2 (Slave Agents(level 2))**.** *The Slave Agent on level 2 (SA^2) is an agent that corresponds to one of authentication methods. Number of these agents is equal to the number of authentication methods to be implemented. In our proposed system we have illustrated three authentication methods: biometric method, user/pass- word and digital signature.*

Definition 3.3 (Slave Agents(level 3))**.** *The Slave Agents on level 3 (SA^3) corresponds to data operation method, and are of three kinds: agent for reading data (RA),*

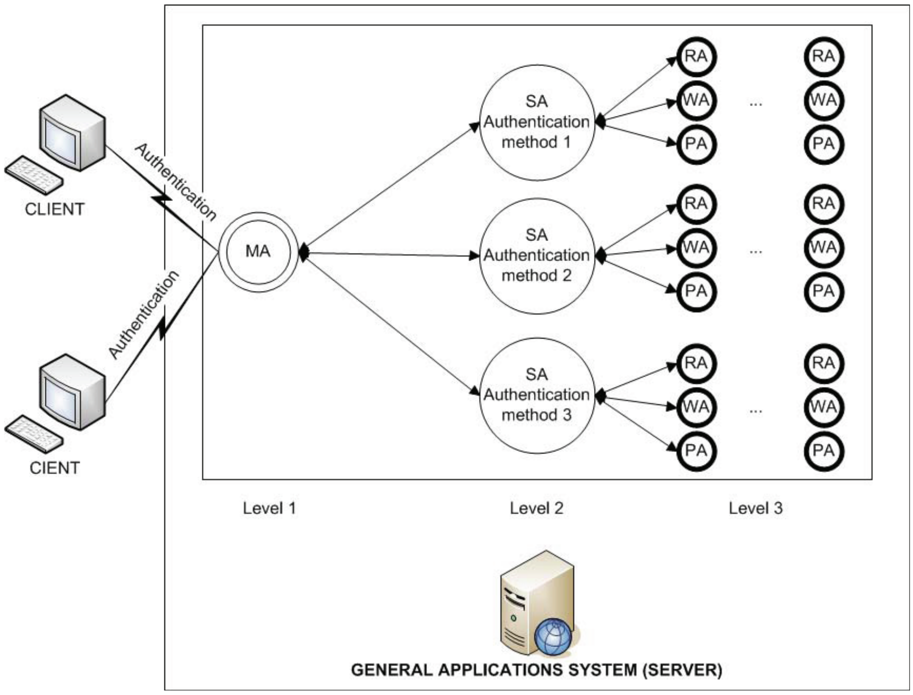


Figure 1. The Multi-Agents System Structure

agent for writing data (WA) and agent for processing data (PA). An SA^2 agent can create more SA^2 agents depending on the tasks received from the MA.

Definition 3.4 (Communication Language)**.** *The Communication Language (CL) is implemented by message passing. An agent that wants to communicate with another agent, first has to create a message object and then send it to the target agent. A message object has a kind and an optional argument object. The receiver agent determines what to do by checking the kind of received message and get parameters as the argument object.*

The system agents are arranged on three levels:

(1) the first level is represented by the master agent (MA)
(2) the second level is represented by agents that are linked to the MA
(3) the third level is represented by agents that are linked to the second level agents but not with the master agent.

3.2. The System Functionality. The system uses various authentication methods: biometric method, user/password method and digital signature method. It may have a larger number of methods, but, for a better understanding, we exemplify a system with only the three methods mentioned above. Each second level agent represents one authentication method. We denote them as A_b^2 (agent for biometric method), A_{up}^2 (agent for user/password method) and A_{ds}^2 (agent for digital signature method). The master agent receives the authentication requests from the users. It analyses the data and sends it to the corresponding second level agent. The second level agent is linked to multiple third level agents. Each third level agent corresponds to data method. So, the agent A_b^2 has the following agents:

- A^3_{brd} for reading the biometric data
- A^3_{bwd} for writing the biometric data
- A^3_{bpd} for processing data (if it is a verification, it compares the information with the existing one from the database; if it is an identification it compares the information with all the data from the database searching for a matching one)

Agent A^2_{up} has the following agents:

- A^3_{uprd} for reading data (username and password)
- A^3_{upwd} for writing data (username and password)
- A^3_{uppd} for processing data (verifying if the username and the password match)

Agent A^3_{ds} has the following agents:

- A^3_{dsrd} for reading data (digital signature)
- A^3_{dswd} for writing data (digital signature)
- A^3_{dspd} for processing data (verifying if the digital signature is valid)

When all the third level agents have ended their processes the information is sent back to the second level agent. It then sends the response to the MA. If the MA receives a positive response, the authentication process ends successfully.

From the server part, in such an authentication scheme, there are two kinds of participants, a signer (the server) and a group of requesters (the users of the systems). Requesters request the blind signatures from the signer, and the signer issues the blind signatures to the requesters. In addition, the scheme can be divided into four phases: (1) the initialization phase, (2) the requesting phase, (3) the signing phase, and (4) the extraction phase.

The Initialization Phase The signer computes $n = pq$, where p, q are two large primes, and $p \equiv q \equiv 3 \ (mod\ 4)$. Furthermore, let H be a one-way hash function. The signer keeps p and q secret, and publishes n and H.

The Requesting Phase To obtain a signature of the message m, the requester randomly chooses two integers u and b, such that

$$\alpha = b^2 H(m)(u^2 + 1) \ mod\ n. \tag{1}$$

Then the requester delivers α to the signer.

The Signing Phase While receiving α, the signer randomly chooses an integer x. Because the signer knows the factors p and q of n, and $\alpha(x^2 + 1) \ mod\ n$ is a QR in $\mathbb{Z}^*_n$, therefore, the signer has the ability to derive t from

$$t^{-2} = \alpha(x^2 + 1) \ mod\ n. \tag{2}$$

Then, the signer delivers the pair (t, x) to the requester.

The Extraction Phase After receiving (t, x), the requester computes

$$c = (ux - 1)(x + u)^{-1} \ mod\ n, \tag{3}$$

and

$$s = bt(x + u) \ mod\ n. \tag{4}$$

The pair (c, s) is a signature of m. To verify the validity of (c, s) of m, the verifier checks whether or not the following equation holds,

$$H(m)s^2(c^2 + 1) = 1 \ mod\ p. \tag{5}$$

In the following, we prove that equation 5 always holds while the signature (c, s) is correct. According to equation 2, we get

$$t^2\alpha(x^2 + 1) = t2t - 2 = 1 \ mod\ n. \tag{6}$$

Hence, we have

$$\begin{aligned}
H(m)s^2(c^2+1) &= (\frac{\alpha}{b^2(u^2+1)})(bt(x+u))^2((\frac{ux-1}{x+u})^2+1) \\
&= (\alpha b^2(u^2+1))(bt)^2((ux-1)2+(x+u)2) \\
&= (\alpha b^2(u^2+1))(bt)^2(x^2+1)(u^2+1) \\
&= t^2\alpha(x^2+1) = 1\ (mod\ n). \qquad (7)
\end{aligned}$$

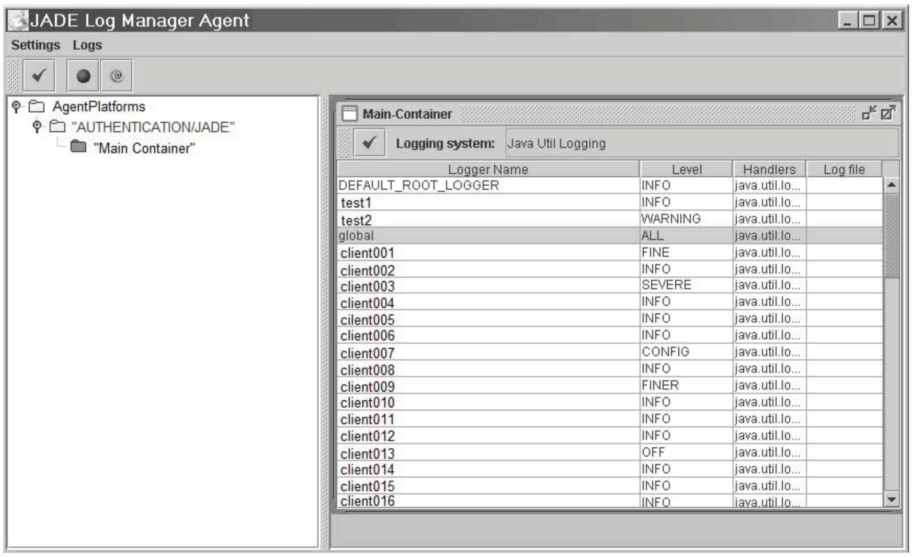
Suppose there is an user which wants to authenticate himself to the system, so he sends a request based on the construction from formula 7 to the master agent (MA). The authentication method used is a biometric one based on fingerprint verification. For our verification the user also has a username. First the master agent receives the request and checks the data to identify the authentication method. Then, it sends the biometric data to A_b^2. The second level agent sends it to the third level agents: $A_{brd}^3, A_{bwd}^3, A_{bpd}^3$. The first reads the information while the second one writes it. The third agent has the task to verify if the user is who he claims he is. So, it will first search the username in the database. If it does not find it he will return a negative response to the second layer agent. If it founds the username it verifies if the received fingerprint matches with the one stored for that username. If it does, it sends a positive response to A_b^2, if it does not it sends a negative one. When A_b^2 receives the response from A_{bpd}^3 he sends it to the master agent. In case of a positive response the master agent allows the user to perform the actions for which he was given permission.

4. Case Study and Implementation

To evaluate the efficiency of our authentication model based on Multi-Agent System(MAS) we consider the following scenario: *One hundred clients are assigned to register for authentication (three authentication methods are available: biometric method, user/password and digital signature) in order to use a web-based application independently.*

The solution is based on our proposed model. The clients will be allowed to get through the web application when their authenticity and application's access right are checked to be valid by the MAS module. We initially prove our proposed idea on how the MAS supports the multi-clients authentication and management. Therefore the primitive goal of our experiment is to verify that the proposed authentication model is functionally correct and feasible to support the authentication of multi-clients. For implementation we used a web server, MySQL database system, and JADE(Java Agent DEvelopment Framework) for MAS module development [3, 11]. To verify the authentication process using multi-agent system we can check all activities of the clients from the event log, using system interface as shown in Fig. 2.

The authentication process is controlled by the MAS module, this was proving transparency. If the clients are successfully authenticated, they will be allowed to any applications available to them. The current implementation uses the certificate management function which co-operate with the certification authority to enable our relying parties to request for issuance, suspension, revocation, and renewal in a more feasible way. According to our experiment, in which the system functioned as was described in section 3.2, the result from the verification shows that MAS functionalities are correct and robust for multi-clients.



Logging system: Java Util Logging

Logger Name	Level	Handlers	Log file
DEFAULT_ROOT_LOGGER	INFO	java.util.lo...	
test1	INFO	java.util.lo...	
test2	WARNING	java.util.lo...	
global	ALL	java.util.lo...	
client001	FINE	java.util.lo...	
client002	INFO	java.util.lo...	
client003	SEVERE	java.util.lo...	
client004	INFO	java.util.lo...	
cilent005	INFO	java.util.lo...	
client006	INFO	java.util.lo...	
client007	CONFIG	java.util.lo...	
client008	INFO	java.util.lo...	
client009	FINER	java.util.lo...	
client010	INFO	java.util.lo...	
client011	INFO	java.util.lo...	
client012	INFO	java.util.lo...	
client013	OFF	java.util.lo...	
client014	INFO	java.util.lo...	
client015	INFO	java.util.lo...	
client016	INFO	java.util.lo...	

Figure 2. Authentication process results in our scenario

5. Conclusion and Future Work

We have presented an authentication model based on multi-agent system technologies to serve the authentication service in the multi-clients environment and implementation of how to apply MAS technique and multi-application. The design of master agent (MA) and slave agents(SA) is introduced to perform the multi-client authentication with a very good computational time. Therefore, client convenience is greatly increased by using our model. Finally, the system implementation has been presented and tested in a scenario, with the focus on features of multi-agent system in authentication process. In our future works, agents collaboration (slave agents each other) is very important for system robustness. Complex administration policies could be extensively applied to multi-agent system. In order to implement in a more complex system, an integration of various authentication types methods and the multiple-authentication can be adopted for future version of the model.

Acknowledgments

N. Constantinescu's works was supported by Grant CNCS IDEI PCCE 08/2010.

C.I. Popirlan has been supported by the strategic grant POSDRU/89/1.5/S/61968, Project ID61968 (2009), co-financed by the European Social Fund within the Sectorial Operational Program Human Resources Development 2007-2013.

References

[1] G. Agnew, Cryptography, Data Security, and Applications to E-commerce, *Electronic Commerce TacTechnology Trends Challenges and Opportunities*, IBM Press (2001), 69–85.

[2] M. Bellare, R. Canetti and H. Krawczyk, *A Modular Approach to the Design and Analysis of Authentication and Key Exchange Protocols*, 1998.

[3] F.L. Bellifemine, G.Caire and D. Greenwood, *Developing Multi-Agent Systems with JADE*, Wiley, 2007.

[4] J. Brainard, A. Juels, R.L. Rivest, M. Szydlo and M. Yung, Fourth Factor Authentication: Somebody You Know, *CCS 2006*, Alexandria, Virginia, USA, (2006).

[5] DTI (2006), Information security breaches survey. Department of Trade and Industry, *URN 06/803.*

[6] S. Fugkeaw, P. Manpanpanich and S. Juntapremjitt, Multi-Application Authentication based on Multi-Agent System. *IAENG International Journal of Computer Science* **3** (2007), No. 2, 37–42.

[7] S. Furnell, *Cybercrime: vandalizing the information society*, Addison Wesley Professional, 2001.

[8] S. Furnell, Authenticating ourselves: will we ever escape the password?. *Network Security* **3** (2005), 8–13.

[9] M.R. Genesereth and S.P. Ketchpel, Software Agents *Communications of the ACM* **37** (1994), No. 7, 48–53.

[10] S. Hawkins, D. Yen and D. Chou, Awareness and Challenges of Internet Security. *Information Management & Computer Security* **8** (2000), No. 3, MCB University Press.

[11] JADE, Java Agent Development Environment, website: http://jade.tilab.com

[12] D.G. Lee, S.I. Kang, D.H. Seo and I.Y. Lee, Authentication for Single/Multi Domain in Ubiquitous Computing Using Attribute Certification, *ICCSA* **4** (2000), 326–335.

[13] NIST (2001), Underlying Technical Models for Information Technology Security. *National Institute of Standards and Technology*, Special Publication 800-33, http://csrc.nist.gov/publications/

[14] H. Parunak, Go to the ant: Engineering principles from natural multi-agent systems, *Artificial Intelligence and Management Science*, 1996.

[15] RFC (1945), Hypertext Transfer Protocol - HTTP/1.0.

[16] M.J.B. Robshaw, On Recent Results for MD2, MD4 and MD5, *RSA Laboratories Bulletin* **4** (1996).

[17] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach* (Third Edition), Prentice Hall, 2009.

[18] G. Stoian and C.I. Popirlan, A proposal for an enhanced mobile agent architecture (EMA), *Annals of the University of Craiova, Mathematics and Computer Science Series* **37** (2010), No. 1, 71–79.

[19] N. Tandareanu and C.I. Popirlan, A Mobile Agents Approach for Knowledge Bases Processing. In *Proceedings of the Twelfth IASTED International Conference on Intelligent Systems and Control (ISC 2009)*, 27–32, Cambridge, Massachusetts, USA (2009).

[20] S. Vaudenay, The Security of DSA and ECDSA Bypassing the Standard Elliptic Curve Certification Scheme, *Springer-Verlag Berlin Heidelberg* (2003), 309–323.

[21] S. Wilson, Certificates and trust in electronic commerce. *Information Management & Computer Security* **5** (1997), No. 5, MCB University Press, 175–181.

[22] M.J. Wooldridge, *Introduction to Multiagent Systems*, John Wiley and Sons, 2002.

[23] M.J. Wooldridge and N.R. Jennings, Intelligent Agents: Theory and Practice, *Knowledge Engineering Review* **10** (1995), 115–152.

(Nicolae Constantinescu, Claudiu Ionut Popirlan) Department of Computer Science,
University of Craiova, 13 A.I. Cuza Street, Craiova, 200585, Romania
E-mail address: nikyc@central.ucv.ro, popirlan@inf.ucv.ro