Distributed Security in Multi-agent Systems

George Dan Moiş, Stelian Flonta, Iulia Ştefan, Szilárd Enyedi, Liviu Cristian Miclea

Technical University of Cluj-Napoca

Abstract: *The use of mobile agents in distributed computing represents an alternative to the conventional client-server model. Although agents are present in many critical applications, the security problem in such systems has not been sufficiently studied yet. This paper presents a new method for assuring security in a multi-agent system by using an extension of the ElGamal encryption algorithm. The messages sent between the system's components can be decrypted only if the agents that hold the parts of the private key work together.*

Keywords: security, multi-agent systems, encryption algorithm, security mechanism, public key, private key.

1. INTRODUCTION

The advantages provided by the use of agents in distributed systems have been intensively studied [1, 2, 3]. Nowadays, we no longer have stand-alone computers, but complex computing and information systems running increasingly complex applications. Processing huge amounts of data, testing and maintaining such large and complex systems require the use of new methods for performing these tasks. It is hard to believe that classical centralized methods for task completion, testing and repairing can be scaled and adapted for being used in large, distributed and interactive digital systems.

It has been proven that mobile agents are the best choice in applications like: e-commerce [4], real-time monitoring and management of communication networks [5, 6], information handling [7], web services [8], etc. The agent paradigm makes the understanding and management of large-scale distributed heterogeneous systems easier and provides a way of viewing and characterizing intelligent systems [2].

Their advantages over the client-server paradigm, mostly based on Remote Procedure Call, are easy to point. RPC consists in sending a request to a server or a host computer and waiting for the result to be returned [9]. The method is extended to distributed systems by allowing a procedure to be executed on another node. The data transferred over the network by using this method can overload the communication channels and can lead to unwanted system behavior. The use of agents confers the system a higher level of flexibility and performance. The agents can move from one location of the system to another, thus avoiding transferring large amounts of data (only their executable code along with some results is transferred). Agents support asynchronous execution and don't need connections over the network for long periods of time.

Although in present times agents are used in many critical applications, the security problem in multi-agent systems has not been sufficiently studied yet.

2. SECURITY IN MULTI-AGENT SYSTEMS

The security of a multi-agent system is a complex issue and several approaches have been presented over the last few years.

A mobile agent is an agent that can move from one node of the system that represents its environment to another for carrying out a specific task. The use of agents in applications like e-commerce and network management, where private data is handled, highlights the need for a secure system. Some of the technologies that protect the agent host against malicious agents are java sandboxes, code signing, proof-carrying code, path histories, type safe languages and software fault isolation [10], but protecting agents against malicious behavior in execution environments is specific to the Mobile Agents technology and represents a new research area. A malicious agent can attack the system where it operates or the other agents within this system and an improper hosting environment can attack the incoming agents. Therefore, the development of a security mechanism needs to consider the protection for both the execution environment and agents present in this environment.

A part of the work in this field aims at providing viable security frameworks for mobile agent systems. The authors in [11] present a framework that supports the specification of various security policies for protecting the components of the multi-agent system. MagicNET [12], Mobile Agent Intelligent Community Network, is an architecture developed for secure mobile agent deployment. It implements a comprehensive integrated security system which includes the creation of trusted mobile agents and their adoption, owning, launching, authorization and execution. The work in [13] presents an authorization framework for mobile agents, which introduces XACML and SAML, two widely accepted standards currently used in Web Services and Grid.

Mobile agents are themselves exposed to security threats and methods for assuring agent security have been studied: co-operating agents, execution tracing, environmental key

This research was funded by the Romanian Government through the research grant PN-II IDEI 93/1.10.2007.

generation, non-interactive computing with encrypted functions, obfuscated code, partial result encapsulation, etc. [10].

Some of the approaches implement cryptographic techniques for conferring the multi-agent system a certain level of security. They include digital signatures, hash functions, attribute and proxy certificates [14]. This paper presents an encryption algorithm that allows data access only for an authorized group within a distributed system. This way, sensitive data can be read/modified only by trusted entities in the system.

3. CRYPTOGRAPHY IN MULTI-AGENT SYSTEMS

Cryptography is the art and science of writing in secret code and its new forms appeared soon after the development of computer communications. Some of the security requirements in communication within applications include: authentication, privacy/confidentiality, integrity and non-repudiation. The main goal of cryptographic algorithms is to protect data from being altered or stolen by unauthorized entities, but can also be used for authentication. The three general types of cryptographic schemes are the secret key or symmetric cryptography, the public key or asymmetric cryptography and the hash functions [15].

A few of the works that provide security solutions in multi-agent systems for communication and data transmission using cryptographic methods are presented in the following section.

A multi-agent matchmaker system that uses symmetric cryptography to secure data is Yenta [16]. It is designed to find and introduce to each other people with similar interests over the Internet. Keys within the system are exchanged using public key infrastructure and data is verified using hash messages.

The authors in [17] propose an agent-based medical system in which every agent that wants to communicate to the main host is checked through a signature by a broker agent using a public key mechanism. This way, malicious agents are prevented from stealing or forging other agents' identities.

The work in [18] presents the architecture of a multi-agent system for the assistance and the supervision of medical protocols in hospital environments. The agents communicate through the Internet and the messages sent between them are encrypted by using the triple DES algorithm for confidentiality purposes. The keys are exchanged using the public key RSA mechanism.

The the AES (Advanced Encryption Standard) encryption algorithm can also be used for assuring security in the mobile agents' communication [19]. The AES algorithm is a symmetric block cipher that can encrypt and decrypt information which uses cryptographic keys of 128, 192, and 256 bits to encrypt and decrypt data in blocks of 128 bits.

4. THE ELGAMAL ENCRYPTION ALGORITHM

The algorithm presented in this paper is based on the ElGamal encryption algorithm which was first presented by Taher Elgamal in 1984.

The ElGamal algorithm can be divided into three phases: key generation, encryption and decryption of a message.

4.1. Key generation

Let q be a prime number and g be a generator for Z_p. The private key is an integer between 1 and q-2. $h=g^x \bmod p$ is then computed and the public key for the ElGamal encryption is the triplet (q,g,h).

4.2. Message encryption

A random integer y prime to p-1 is chosen and the following pair of values is computed:

$c_1=g^y \bmod q$

$c_2=m \cdot h^y \bmod q$, where m encrypted plaintext.

The encrypted message is the pair (c_1, c_2).

4.3. Message decryption

For decrypting the message (c_1, c_2), the public key, (q, g, h), and the private key, (x), are needed. m is obtained by computing:

$m \leftarrow c_2/c_1^x \pmod q$.

The division by c_1^x should be interpreted in the context of modular arithmetic, as the multiplication with the inverse of c_1^x in Z_p, $c_1^x = c_1^{p-1-x}$.

Verification:

$$\begin{aligned} c_2/c_1^x \pmod q &= m \cdot h^y/(g^y)^x \pmod q \\ &= m \cdot g^{xy}/g^{yx} \pmod q \\ &= m. \end{aligned}$$

The system is indefinite because the encryption depends on x and another random value, y, chosen during the encryption phase. Thus, there are more encrypted texts corresponding to a certain clear text.

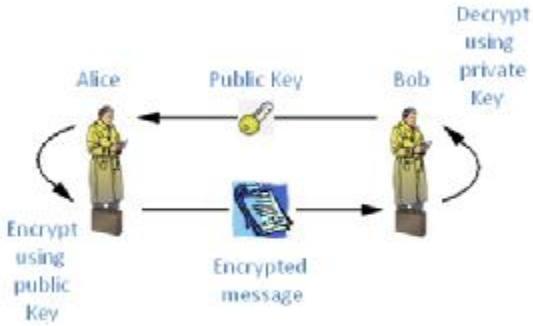

Fig.1. Public Key Cryptography scheme.

4.4. Simple example

Let q=1013, g=610, x=319, and $h=g^x \pmod{1013}=377$.

We have the triplet (1013, 610, 377) as the public Key and (319) as the private Key.

Message encryption is performed by using the public Key (1013,610,377). Let m=560 be the message that has to be encrypted. y=335 is chosen and

$c_1=g^y \pmod{1013}=533$ and

$c_2=m \cdot h^y \pmod{1013}=560 \cdot 46 \pmod{1013}=435$

are computed.

The message is decrypted by computing

$c_2/c_1^x \pmod{1013}=435/533^{319} \pmod{1013}$

$533^{319} \pmod{1013}=533^{1013-1-319} \pmod{1013}=991$ $435/533^{319} \pmod{1013} =435 \cdot 991 \pmod{1013} =560=m$.

5. EXTENSION OF THE ELGAMAL ENCRYPTION ALGORITHM

The present approach extends the ElGamal Encryption scheme. The algorithm was divided so that an encrypted message is sent to more than one receiver, every one of them knowing a different private Key. The Key is divided into (2n+1) segments and the received messages can be decrypted only if the (2n+1) receivers work together.

In the key generation stage, q and g are chosen as in the original scheme. $x_1,x_2,\ldots,x_{2n+1}$ are integers between 1 and q-2. The public Key is generated by computing:

$h_1=g^{x1}, h_2=g^{x2},\ldots, h_{2n+1}=g^{x2n+1}$.

The public Key is $(q,g,h_1,h_2,\ldots,h_{2n+1})$ and the private Key is $(x_1,x_2,\ldots,x_{2n+1})$. The encrypted message can be decrypted only when all the (2n+1) Keys are available.

A message *m* is encrypted by computing:

$c_1=g^y$,

$c_{21}=m \cdot h_1^y$,

$c_{22}=m \cdot h_2^y$,

…,

$c_{22n+1}=m \cdot h_{2n+1}^y$,

$c_2= c_{21} \cdot c_{23} \cdot c_{25} \cdot c_{27}\ldots./ c_{22} \cdot c_{24} \cdot c_{26}\ldots$,

while knowing the public Key and with y randomly chosen between 1 and q-2. The encrypted message is the pair (c_1, c_2). It can be decrypted only by knowing the public Key and the private Keys $(x_1),(x_2),\ldots,(x_{2n+1})$. Because every receiver knows the only its private Key, they have to work together for decrypting the message.

The computations performed in this stage are:

$c_2 \cdot c_1^{x2} \cdot c_1^{x4} \cdot c_1^{x6} \ldots / c_1^{x1} \cdot c_1^{x3} \cdot c_1^{x5} \cdot c_1^{x7}\ldots$

$= (c_{21} \cdot c_{23} \cdot c_{25} \cdot c_{27}\ldots/ c_{22} \cdot c_{24} \cdot c_{26}\ldots)(c_1^{x2} \cdot c_1^{x4} \cdot c_1^{x6}\ldots/ c_1^{x1} \cdot c_1^{x3} \cdot c_1^{x5} \cdot c_1^{x7}\ldots)$

$= (m \cdot h_1^y \cdot m \cdot h_3^y \cdot m \cdot h_5^y \cdot m \cdot h_7^y\ldots/ m \cdot h_2^y \cdot m \cdot h_4^y \cdot m \cdot h_6^y\ldots)(c_1^{x2} \cdot c_1^{x4} \cdot c_1^{x6}\ldots/ c_1^{x1} \cdot c_1^{x3} \cdot c_1^{x5} \cdot c_1^{x7}\ldots)=m$,

knowing that:

- $h_i^y= g^{x_i y}= g^{y x_i}= c_1^{x_i}$, for i=1,…,2n+1,
- the first fraction has n+1 factors as numerators and n factors as denominators,
- the second fraction has n factors an numerators and n+1 factors as denominators.

The algorithm can be easily adapted for protecting messages sent in a multi-agent distributed system. The encrypted message along with the distinct private Keys can be sent to (2n+1) agents, and because each agent knows only its own private Key, the message can be decrypted only by the entire authorized group.

5.1. *Simple example*

Let q=71, g=31, $x_1=15$, $x_2=19$ and $x_3=17$. After computing

$h_1=g^{x1} \pmod{71}=31^{15} \pmod{71}=41$

$h2=g^{x2} \pmod{71}= 31^{19} \pmod{701}=61$

$h3=g^{x3} \pmod{71}= 31^{17} \pmod{701}=67$

we obtain (71,31,41,61,67), the public Key and (15,19,17) as the private Key.

Knowing the previously computed public Key and choosing y=10, the message m=30 is encrypted as follows:

$c_1=g^y \pmod{71}=20$

$c_{21}=m \cdot h_1^y \pmod{71}=30 \cdot 41^{10} \pmod{71}=32$

$c_{22}=m \cdot h_2^y \pmod{71}= 30 \cdot 61^{10} \pmod{71}=48$

$c_{23}=m \cdot h_3^y \pmod{71}= 30 \cdot 67^{10} \pmod{71}=20$

$c_2=c_{21} \cdot c_{23}=32 \cdot 20 /48 \pmod{71}=37$

Therefore, we have the encrypted message $(c_1, c_2) =(20, 37)$.

The message decryption is carried out using the private Keys x_1, x_2 and x_3:

$(c_2 \cdot c_1^{x2}) / (c_1^{x1} \cdot c_1^{x3}) \pmod{71} = (37 \cdot 30) / (20 \cdot 48) \pmod{71} = 45 / 37 \pmod{71} = 45 \cdot 48 \pmod{71} = 30 = m$.

Every agent or node in a multi-agent distributed system that needs to send a message can use the encryption scheme described above. The private key is partitioned in (2n+1) segments, each one being stored by one of (2n+1) agents which have the job of decrypting the message. Depending on the nodes complexity and on the agents' characteristics, the message decryption phase can be performed by the agents owning the segments of the private Key working in parallel (Figure 2) or by one specific agent in the system that previously receives these segments (Figure 3). In the case in

which the agents decrypt the message by working together, they have to move to the message's destination node first, while in the second case only the segments of the private Key are sent to this node.

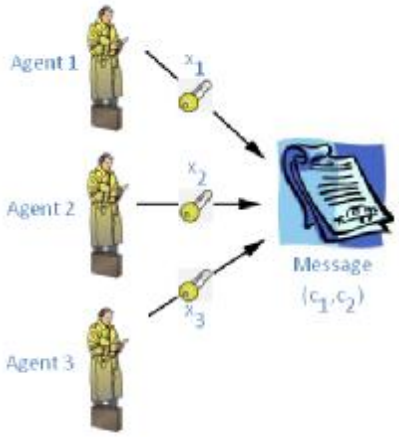

Fig. 2. Agents decrypting together a message

If the message decryption phase is performed by an initiating agent which receives the segments of the private Key through the network, than secured communication channels are required, but the amount of resources needed in the message's destination node, the place where the actual decryption takes place, is smaller than in the first case.

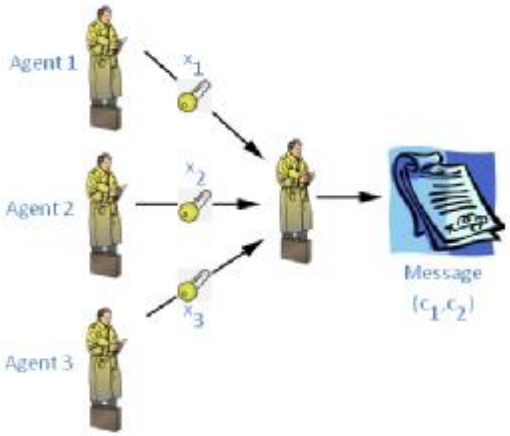

Fig. 3. Agents sending the private Keys for decryption

The proposed encryption algorithm divides the private Key into an uneven number of segments, but decryption can be performed by an even number of agents as well by assigning each one of them two distributed Keys.

6. CONCLUSIONS

Mobile agents are widely used for processing significant amounts of data and for monitoring and maintaining increasingly large and complex distributed systems. Their presence in environments where private data exists leads to the study of mechanisms for assuring certain levels of security in critical applications.

The encryption method presented in this paper confers the multi-agent system in which it is implemented a higher level of security by allowing only certified entities to have access to transmitted messages. These messages can be decrypted only if a previously specified number of trusted agents that own the distributed Key work together. This way, malicious agents or nodes in the system cannot access sensitive data.

REFERENCES

Alfalayleh, M., Brankovic, L., "An overview of security issues and techniques in mobile agents", Proceedings of the Eighth IFIP TC-6 TC-11 Conference on Communications and Multimedia Security (CMS2004), Windermere, UK, **59-78**, 2004.

Alsinet, T., Bejar, R., Fernanadez, C., Manyà, F., "A Multi-agent system architecture for monitoring medical protocols", Proc of the fourth international conference on Autonomous agents, Barcelona, Spain, 499-505, 2000

Awais Shibli, M., Muftic, S., Giambruno, A., Lioy, A., "MagicNET: Security System for Development, Validation and Adoption of Mobile Agents", 2009 Third International Conference on Network and System Security, Gold Coast, Australia, **389-396**, 2009.

Brewington, B., Gray, R., Moizumi. K., Kotz, D., Cybenko, C., Rus, D., "Mobile agents for distributed information retrieval", Intelligent information agents, Springer-Verlag, **355-395**, 1999.

Chung-Ming Ou, C.R. Ou, Yao-Tien Wang, "Security of Mobile Agent-based Web Applications", 2008 IEEE Asia-Pacific Services Computing Conference, Yilan, Taiwan, **107-112**, 2008.

Flonta, S., Miclea, L., "An extension of the El Gamal encryption algorithm", Proceedings of 2008 IEEE-TTTC International Conference on Automation, Quality and Testing, Robotics, Cluj-Napoca, Romania, **444 – 446**, 2008.

Foner, L., "A Security Architecture for Multi-Agent Matchmaking", Proceedings of the Second International Conference on Multiagent Systems, Kyoto, Japan, **80-86**,1996.

Kazi, R., Morreale, P., "Mobile Agents for Active Network Management", Proceedings of IEEE MILCOM '99, Atlantic City, **149-153**, November 1999.

Kessler, G., "An overview of cryptography", http:// www.garykessler.net/ library/crypto.html.

Kleijkers, S., Wisman, F., Roos, N., "A Mobile Multi-Agent System for Distributed Computing", Proceedings of the 1st international conference on Agents and peer-to-peer computing 2003, Bologna, Italy, **158-163**, 2003.

Lin, DeShu, Huang, TingLei, "A Mobile-Agent Security Architecture", 2nd International Conference on e-Business and Information System Security (EBISS), , Wuhan , China, **477-481**, 2010

Loulou, M., Kacem, A.H., Jmaiel, M., Mosbah, M., "A Formal Security Framework for Mobile Agent Systems: Specification and Verification", Proceedings of the 3rd International Conference on Risks and Security of Internet and Systems, Tozeur, Tunisia, **69–76**, 2008.

Moreno, A., Sánchez, D., Isern, D., "Security Measures in a Medical Multi-Agent System", Artificial Intelligence Research and Development (Proc. CCIA 03), vol. 100 of Frontiers in Artificial Intelligence and Applications, I. Aguiló, L. Valverde, and M.T. Escrig, eds., IOS Press, **244–255**, 2003.

Navarro-Arribas, G., Borrell, J., "An XML Standards Based Authorization Framework for Mobile Agents", Secure Mobile Ad-hoc Networks and Sensors, Springer Verlag, vol. 4075 of Lecture Notes in Computer Science, New York, USA, **54-66**, 2006.

Pugazendi, R., Duraiswamy, K., "Mobile Agents - A Solution for Network Monitoring", ARTCom 2009, **579-584**, 2009.

Qi Tang, Fang Xie, "A Multi-Agent System for E-Commerce Automation", IEEE CCECE/CCGEI, Ottawa, **514-517**, May 2006.

Tay, B., Ananda, A., "A Survey of Remote Procedure Calls", Operating Systems Review, **24(3):68-78**, July 1990.

Weiss, G., "Multiagent Systems, A Modern Approach to Distributed Artificial Intelligence", The MIT Press, Cambridge, Massachusetts, 1999.

Wooldridge, M., "An Introduction to MultiAgent Systems", John Wiley & Sons, ISBN: 978-0-471-49691-5, 2001.

Zahreddine, W., Mahmoud, Q., "Blending Web Services and Agents for Mobile Users", Autonomous Decentralized Systems, ISADS 2005 Proceedings, Chengdu, China, **585-590**, 2005,