Securing Mobile Agent Systems in Which the Agents Migrate via Publish/Subscribe Paradigm

Ahmet Ali Karzan and Nadia Erdoğan

***Abstract*—Multi agent systems especially mobile agent systems provide great environment for distributed computing. Forcing the mobile agents to migrate via publish/subscribe paradigm brings the mobile agent systems a very high degree of scalability and flexibility by loosely coupling the communicating entities. However, in order to realize such a scalable and flexible system, security is much more important than ever. In this paper we present a security architecture which is also mobile like the agents in the system. The agents themselves provide the security functionality for their safety. In order to secure whole system hosts visited by the mobile agents also have to be secured against malicious mobile agents. Because we load the security functionality on agents some special agents acting on behalf of their hosts protect the host systems. This architecture also loosely couples the agents with their host platforms and this promotes the inter-platform mobility along with interoperability in the context of security. The architecture itself is flexible. This property provides agility on easy adoption of new security technologies against emerging threats.**

***Index Terms*—Multi agent systems, mobile agents, security, publish/subscribe paradigm, agent migration.**

I. Introduction

The mobile agent paradigm provides a very important feature in developing high performance, decentralized software applications in distributed environments. As agents (executable code) can travel on the network, data transfer between the communicating parties is drastically reduced, which increases communication performance by utilizing network bandwidth. Using topic based publish/subscribe paradigm for the underlying communication path as described in [1] provides a high degree of scalability and flexibility by decoupling the source and target entities. This approach eliminates the necessity of acquaintance of communicating entities. Yet, on the other hand, important issues in the area of security arise because the mobile agent does not know its destination node, and similarly, the host to be visited does not know about the arriving agent. Therefore, as both sides do not know, in advance, the identities and intensions of each other, advanced security measures need to be taken.One way of securing mobile agent systems especially for the agent systems described in [1] is providing security as an agent service in order to preserve the inter-platform mobility for different types of platforms. The security architecture proposed in [2]. However in that work security subsystem of agents can only be used in inter-agent communication. That is, the host platforms must have separate security systems to protect themselves from malicious agents. This results in dependency on host platforms for agents. Furthermore, different agent platforms can offer different security functionality at different levels. Also, [2] does not offer security for agents against malicious hosts. In [3] M. V. Prem *et al.* proposes a three factor security for protecting free roaming mobile agents from other mobile agents against passive attacks, such as eavesdropping. They concentrate on data confidentiality and itinerary protection against attacks such as colluded truncation, which is an important problem for free roaming mobile agents, but they pay little attention on peer entity authentication. They propose just using length check for the integrity control, which can easily be defeated by using compression. In [4], Q. Zhang *et al.* propose a strong security mechanism with designated hosts using tamper-proof policies. However, because the mobile agent has to own the list of hosts to be visited before traveling, this mechanism is not suitable for the agents migrating via topic based publish/subscribe paradigm as described in [1]. In [5] C. Yang proposes an extended role based access control architecture along with establishment of trust between mobile agent and the host visited. The main concern of the work is the introduction of strong access control mechanisms. Trust establishment between the mobile agent and the host takes place after the mobile agent gets executed. However, this can give a chance to a malicious mobile agent to perform some malicious activities before it is marked as malicious or untrusted.

In this paper, we propose a layered security architecture which can be applied to mobile agent systems in general, and specifically for agents which migrates via topic based publish/subscribe paradigm [1]. The approach we propose identifies if the incoming mobile agent is malicious (indeeduntrusted) or not before it starts execution on the host via the use of a specific information which is appended to the mobile agent by the sending entity before the agent is transmitted onto the transmission channel. Thus, a malicious mobile agent can, in no way, perform malicious activities because it is identified and discarded before begins execution.

Organization of the remainder of this paper is as follows. Section II describes the key security requirements to be met for a secure environment. Section III introduces the security architecture proposed to meet the key security requirements described in section II and section IV gives conclusion.

II. Key Security Concepts and Requirements

The key security requirements which must be met by a

Manuscript received June 10, 2013; revised August 20, 2013.

Ahmet Ali Karzan is with the Department of Computer Science, Informatics Institute, Istanbul Technical University, Istanbul, Turkey (e-mail: alikarzan@yahoo.com).

Nadia Erdogan is with the Department of Computer Engineering, Faculty of Computer and Informatics, Istanbul Technical University, Istanbul, Turkey (e-mail: nerdogan@itu.edu.tr).

DOI: 10.7763/LNSE.2014.V2.86

mobile agent system to be secure can be listed as follows.

Integrity: the mobile agent that arrives at a service requesting host must exactly be the same agent sent by its owner. It must not be modified while it is in transit; in other words, agents must be resistant against append entry attacks

Authentication: only the agents whose owners are trusted and verified can run on the hosts. Similarly, mobile agent should also authenticate the host in order to determine that the host has the right to have the service provided by the mobile agent on behalf of its owner.

Confidentiality: two level confidentiality should be provided. First, transport level confidentiality to prevent passive attacks like eavesdropping while the mobile agent is in transit. Second, agent level confidentiality to prevent passive attacks as the mobile agent is at rest or working on the host. Passive attacks on a host can be performed by host itself or other fixed or mobile agents living on the host.

III. Security Architecture

We propose a security architecture embedded into the agent to make the security functionality move along with the agent itself. This, as explained at the beginning of the text, supports and promotes the inter-platform mobility even in the case of different types of platforms in security context.

The agent having the security embedded inside will be called Security Enhanced Agent (SEA). The architecture of SEA is comprised of loosely coupled components namely, Authentication and Integrity Control, Encryption and Decryption and also Credentials and Key Management module. Thehigh level architecture is shown in Fig. 1.

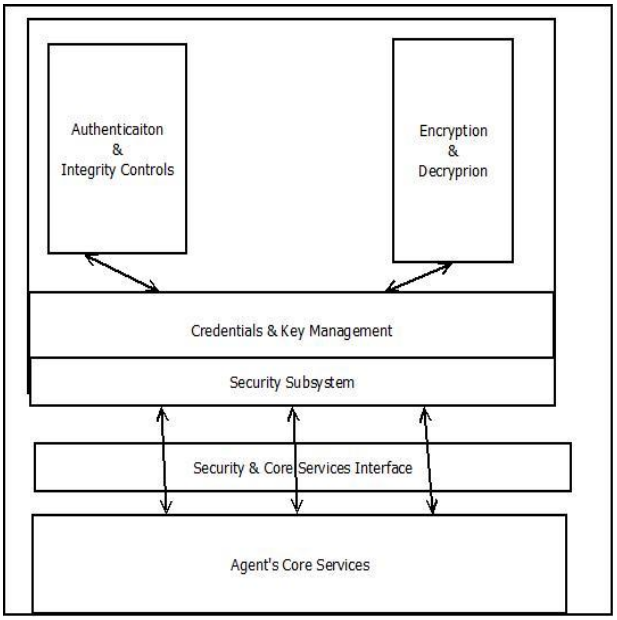

Fig. 1. Security enhanced agent architecture.

A. Authentication and Integrity Controls Module

The module is responsible of generating integrity code and authentication credentials and appending them to the mobile agent. Handler Agent on the receiving host as described in [1] takes the responsibility on behalf of its platform and it first checks the integrity of the incoming mobile agent. The Handler Agent also checks the validity of the credentials presented by the mobile agent. Here we propose to use the digital signatures along with X.509 certificates and PKI for the operations described above. The rationale behind this choice is that the digital signatures are well defined and also have proven to be strong against append entry types of attacks. Beyond integrity feature, digital signatures also provide authentication and non-repudiation services. Thus, using a digital signature meets the three of the key security requirements listed above.

At the mobile agent side, first the Publisher Agent serializes itself as described in [1]. Next, the Publisher Agent (SEA) passes the serialized agent to the Authentication and Integrity Controls module here one-way secure hash function is used to generate digest of its serialized agent. At this point using a secure (cryptographic) hash algorithm is critical because the integrity code generated by a secure hash algorithm can only be unique. In order a hash algorithm to be secure, it must hold the following properties [6], [7]:

- Produce a fixed length output value (digest) regardless of the size of the input (compression property).
- Be relatively easy to compute for any arbitrary size of input (ease of computation property).
- Given a hash h it should be difficult to find any message m such that h = hash(m). This concept isrelated to that of one-way function (pre-image resistance property).
- Given an input m1 it should be difficult to find another input m2 such that $m1 \neq m2$ and hash(m1) = hash(m2) (weak collision resistance or second pre-image resistance property).
- It should be difficult to find two different messages m1 and m2 such that hash(m1) = hash(m2). Such a pairis called acryptographic hash collision. This property is sometimes referred to as strongcollision resistance. It requires a hash value at least twice as long as that required for pre-image-resistance; otherwise collisions may be found by a birthday attack.

As long as the properties listed above are held by the hash function, one and only one input can generate the same output if the output is long enough (today 168 bit hash length seems enough but 256 or 512 bit hash output would be safer). Therefore, if the agent is modified while it is in transit, then it will be easily detected by the Handler Agent because the delivered and the computed hash values will not match and such mobile agents will be discarded by the Handler Agent. This approach protects the host against append entry attacks because the mobile agent is investigated in its serialized form and it never gets executed to perform malicious activities.

SHA-1 which produces 160 bit output is appropriate to be used to generate the digest of the serialized mobile agent. In near future SHA-1 may not be sufficient and, instead, using SHA-2 which produces 256 bit output will be necessary [6].

One important point that needs to be addressed here is that the hash value appended to the serialized mobile agent can also be changed as the mobile agent gets modified. To prevent this case, we use digital signatures to sign the hash value and append it to the serialized mobile agent. Because the signature cannot be imitated, the attacker cannot alter the hash value because if it were altered, then authentication via signature will fail and the mobile agent will be discarded while it is still in its serialized form.

Authentication and integrity control operations for both ends are depicted in Fig. 2.

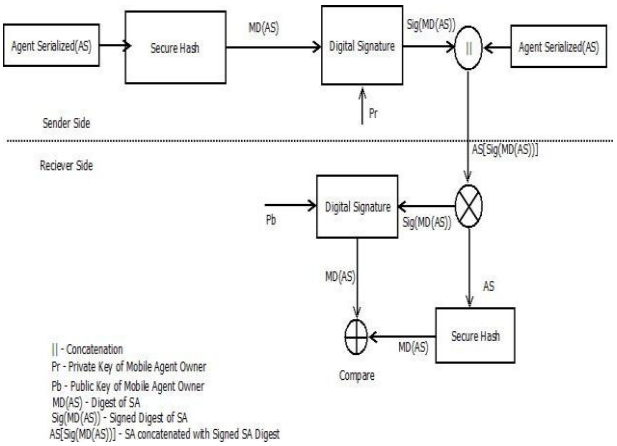

Fig. 2. Authentication and integrity control operations.

In authentication and integrity controls module, first the serialized agent is input to a cryptographically secure hash function such as SHA-1 or SHA-2, and next, this digest is signed by the private key of the owner of the Publisher Agent using RSA asymmetric encryption algorithm. RSA is chosen for the digital signature operations because it is one of the well-defined, security proof digital signature standards. Here, the key length used in RSA must at least be 512 bits long or preferably 1024 bit to let the RSA operation be secure enough against cryptanalytic attacks, such as known plaintext, chosen plaintext, cipher text only, etc., analysis [6]. Next, the signed digest of the serialized agent is appended to the serialized agent itself as shown in the Fig. 2 above. Now, the serialized agent with signed integrity control code is ready to be transmitted to the message oriented middleware (MOM) as explained in [1]. After the mobile agent migrates via publish subscribe paradigm to a host], the Handler Agent on the host meets the serialized agent with its authentication and integrity controls module. Here, the concatenated serialized agent and its signed digest are separated, and first, the digital signature is verified by using the public key of the mobile agent's owner. The identity of the owner is extracted from the serialized part of the agent and the public key associated with this identity is fetched from the Handler Agent's local public key repository if the Handler Agent has previously welcome other mobile agents of the same owner. If this is the first time the Handler agent is hosting a mobile agent from that particular agent owner, then it appoints a trusted third party (CA – Certification Authority) to get the public key (the certificate containing the public key) associated with the identity of the owner agent. Next, RSA algorithm is applied to the signed digest of the serialized agent to verify the signature. If it is valid the digest is obtained, otherwise, the host decides that it is a case of a malicious attempt and discards the agent. If the signature is valid and the digest is obtained then second phase, the integrity control will take place. The serialized agent is input to the cryptographically secure hash algorithm to produce itsdigest and this newly computed digest is compared to the one extracted from the signature to guarantee that the digest itself is not altered while the agent is in transit. If the two are equal then the incoming mobile agent is said to be integral that is, it has not been modified during its trip to the host. If the integrity test fails then the Handler Agent discards the incoming mobile agent on behalf of its host to prevent the modified (probably for malicious activities) mobile agent getting executed.

B. Encryption and Decryption Module

This module is responsible of protecting the sensitive data carried with mobile agent itself or generated while it is in execution on the visited host against passive attacks, such as eavesdropping performed by the (maliciously acting) host or the maliciously acting other mobile or fixed agents co-located on the same host. This privacy protection is at the agent level not at the transport level. Transport level security will be handled later in this sub-section. Symmetric encryption is used due to its ease of computation to enhance the computational performance of the security architecture. Until collaboration with other agents on sensitive data will be needed there is no need to share a secret key which is used in symmetric encryption. This eliminates or at least minimizes the one of the most important problems in symmetric encryption, key sharing. In order to preserve the privacy of confidential data generated during execution, data is encrypted as soon as it is generated not to let it be visible for malicious entities. Here the module uses AES (Advanced Encryption Standard) for encryption algorithm due to its standardized availability, strength and speed over other techniques like DES, etc … 128 bit long key is just enough to secure the operation against cryptanalytic attacks for data having average privacy requirements, for more sensitive data longer key lengths should be used [6]. In the case of collaboration with other agents on sensitive data, agents will have an agreement on temporary session keys.

Here Diffie-Hellman key exchange algorithm should be used in conjunction with PKI based data origin authentication using digital signatures (or any other strong authentication mechanism, we prefer PKI based authentication using digital signatures along with X.509 certificates to provide convenience with authentication module of the proposed security enhanced agent architecture) otherwise the key exchange algorithm is vulnerable to man in the middle attack which impersonates the one or both of the parties involved in the communication. By this way the attacker can just apply eavesdropping on the traffic between the communicating parties or even worse can apply active attacks like modification of the data being sent from one agent to the other.

To perform key sharing Diffie-Hellman key exchange protocol in between the agents occur prior to the communication of the two agents. Diffie-Hellman key exchange protocol message flow can be seen from Fig. 3.

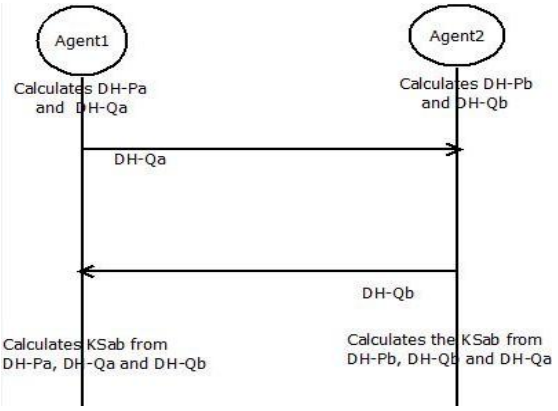

Fig. 3. Diffie-hellman key exchange protocol.

Here, DH-Qa = X = $g^x mod(p)$, DH-Qb = Y = $g^y mod(p)$

and KSab = $Y^x = X^y = g^{xy} mod(p)$. P is big enough (tens to hundreds digits long) [7] prime number and g is a primitive root of p. KSab is the exchanged (calculated) temporary session key.

In this work we propose to use TLS (Transport layer Security) for establishing the security for mobile agents in transit and also for authenticating the peer entities involved in the communication. Rationale behind this choice is, TLS is well defined and strong off the shelf security standard. By this choice, some level of separation of concerns principle with layered security architecture is achieved by separating the transport level security from agent level security. Using TLS will provide peer entity authentication as mentioned before. TLS uses digital certificates and PKI (Public Key Infrastructure) for peer entity authentication as described in [8], [9]. This provides defense against man in the middle attacks, that is, no one can impersonate the source host, message oriented middleware and the target host. Fig. 4 shows the TLS secured form of communication for the architecture proposed in [1].

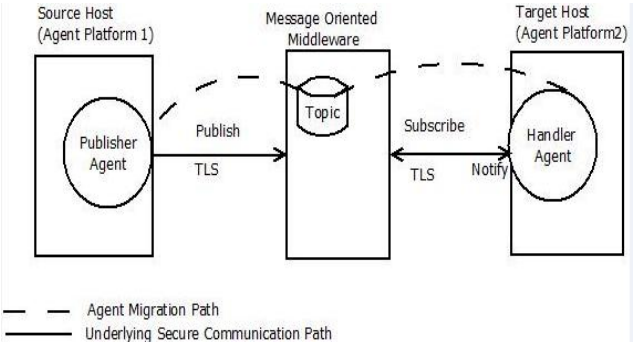

Fig. 4. TLS secured agent migration via publish/subscribe paradigm high level migration architecture.

In TLS protocol after authentication, the peer entities have a session key to use in symmetric encryption to encrypt the data traffic in between them [8], [9]. This encrypted traffic will prevent the eavesdroppers to access mobile agent as it is in transit.

In combination, use of TLS for transport level security and the encryption module of the proposed Security Enhanced Agent (SEA) for agent level security, confidentiality key security requirement for mobileagent systems is met.

C. Credentials and Key Management Module

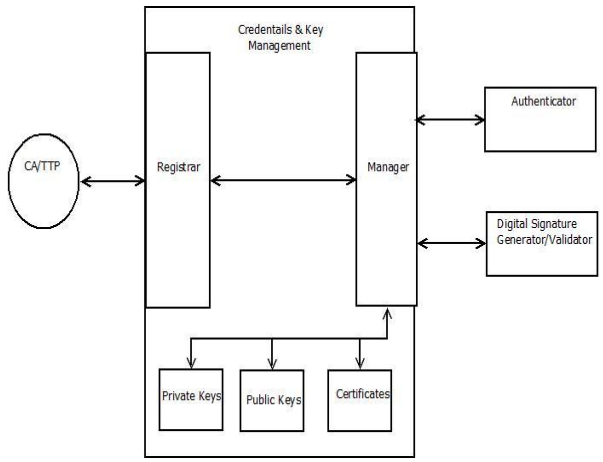

Fig. 5. Credentials and key management module architecture with connections to other modules.

This module by itself does not directly meet any of the key security requirements. However, it takes on an important role by supplying key and credentials management for the other modules which are directly used to meet the key security requirements, namely, authentication and integrity control, encryption and decryption modules. High level architecture of the module is shown in Fig. 5.

The module consists of two major sub modules namely, the manager and the registrar. The registrar is the interface for outside operations. All the interaction with Certification Authority (CA, TTP – Trusted Third Party) is in the responsibility of the registrar. Interaction with CA includes request for a new certificate, renewal request of the already existing certificate, and cancellation of a certificate due to compromise of the private key associated with the certificate. In order to provide security on the communication in between the registrar and the CA again TLS is used to obtain transmission security and peer entity authentication which prevents man in the middle attacks. Because the registrar sub module does not provide any internal service, it delegates the storage operation of the certificates to the manager sub module. The manager is the interface for inside operations. It generates, stores and provides symmetric/asymmetric encryption/decryption keys and digital certificates to encryption and authentication and integrity control modules respectively. Private keys kept within this module are not the private keys of the owner of the travelling mobile agent. Travelling mobile agent only keeps the temporary private keys and their associated X.509 digital certificates along with its owner's certificate holding the public key of the owner of the mobile agent. Owner private keys are only kept within Credentials and Key Management module of the Handler Agent as encrypted and with a critical security level for access control in order to preserve its privacy and prevent it from danger on the remote target hosts from passive and active attacks.

IV. Conclusion

In this paper, we propose a security architecture to protect mobile agent systems in general, and specifically, mobile agents that migrating via publish/subscribe paradigm and hosts that visited by these mobile agents from each other against any malicious activity. The problem of securing the system is reduced to addressing the certain key requirements which are known as confidentiality, integrity, authentication in general. In our approach, we enhance the mobile agents with the security functionality rather than to expect the agent platforms carry out actions to ensure security which removes the dependence on platforms for security. The responsibility of protecting hosts from malicious mobile agents is also given to some specialized agents with special responsibilities on behalf of their hosts (platforms). This design feature enables and eases mobility between different types of agent platforms in the context of security. During the implementation of the proposed security architecture, off the shelf proven standard security technologies are used to provide a high level of interoperability also with a high degree of reliability. The architecture consists of loosely coupled highly cohesive modules by using the separation of concerns principle. This will bring maximum gain with

minimum effort in the case of changes and enhancements in the standard technologies used. That is, changing a technology or method in one module will not affect the others or at most affect minimally. Also, security is layered into two levels namely, transport level security and agent level security. Again, this layering provides separation of duties and both agent level and transport level security functionalities along with the technologies used for them can also be changed or modified independently.These independencies provide flexibility and extendibility to the architecture proposed. As a result, the architecture can easily be strengthened by adopting newly emerged technologiesand algorithms against growing new threats.

REFERENCES

[1] M. A. Karzan and N. Erdogan, "Topic based agent migration scheme via publish/subscribe paradigm," presented at the5th International Conference on Computer Engineering and Technology, Vancouver, BC, April 14-16, 2013.

[2] P. Novak, M. Rollo, J. Hoddik, and T. Vlcek, "Communication security in multi-agent systems," in *Proc. 3rd International Central and Eastern European Conference on Multi-Agent Systems*, Prague, Czech Republic, 2003, pp. 454-463.

[3] M. V. Prem and S. Swamynathan, "Securing mobile agent and its platform from passive attack of malicious mobile agents," presented at the ", IEEE, International Conference on Advances in Engineering, Science and Management (ICAESM), Tamil Nadu, India, March 30-31, 2012.

[4] Q. Zhang, Y. Mu, M. Zhang, and R. H. Deng, "Secure Mobile agents with Designated Hosts," in *Proc. 3rd International Conference on Network and System Security*, Gold Coast, Queensland, Australia, 2009, pp. 286-293.

[5] C. Yang, "Secure Internet Applications Based on Mobile Agents," *International Journal of Network Security*, vol. 2, vo.3, pp. 228–237, 2006.

[6] W. Stallings, *Network Security Essentials: Applications and Standards*, 4th ed., Upper saddle River, NJ: Pearson, 2011.

[7] S. Jacobs, *Engineering Information Security: The Application of Systems Engineering Concepts to Achieve Information Assurance*, Hoboken, NJ, Wiley: IEEE Press, 2011.

[8] T. Dierks and E. Rescorla. (2008). IETF Request for Comment. [Online]. Available: http://tools.ietf.org/html/rfc5246

[9] T. Dierks and C. Allen. (1999). IETF Request for Comment. [Online]. Available: https://ietf.org/rfc/rfc2246.txt

Ahmet Ali Karzan was born in Istanbul, Turkey in 1984. He has BSc. degree in chemical engineering field from Faculty of Engineering in Istanbul University, Istanbul, Turkey since 2008. He has BSc. degree in Computer Engineering field from Faculty of Engineering in Istanbul University, Istanbul, Turkey since 2009. He has MSc. degree in Computer Science field from Informatics Institute in Istanbul Technical University, Istanbul, Turkey since 2013. He works on multi-agent systems and specifically on agent communication and migration and security architectures.

Nadia Erdogan has BSc. degree in Electrical Engineering, Computer branch from Bosphorus University, Istanbul, Turkey since 1978. She has MSc. Degree in Computer Science from Bosphorus University. She has doctoral degree from Institute of Science in Istanbul Technical University. She works on distributed systems, agents systems and parallel programming fields.

She is working as an instructor at Computer Engineering department in Faculty of Computer and Informatics in Istanbul Technical University in Istanbul, Turkey.