World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:2, No:11, 2008

Using a Trust-Based Environment Key for Mobile Agent Code Protection

Salima Hacini, Zahia Guessoum, and Zizette Boufaïda

***Abstract*—**Human activities are increasingly based on the use of remote resources and services, and on the interaction between remotely located parties that may know little about each other. Mobile agents must be prepared to execute on different hosts with various environmental security conditions. The aim of this paper is to propose a trust based mechanism to improve the security of mobile agents and allow their execution in various environments. Thus, an adaptive trust mechanism is proposed. It is based on the dynamic interaction between the agent and the environment. Information collected during the interaction enables generation of an environment key. This key informs on the host's trust degree and permits the mobile agent to adapt its execution. Trust estimation is based on concrete parameters values. Thus, in case of distrust, the source of problem can be located and a mobile agent appropriate behavior can be selected.

***Keywords*—**Internet security, malicious host, mobile agent security, trust management.

I. Introduction

DISTRIBUTED applications (e.g., telecommunication systems, information management, on-line auctions, service brokering) are now increasingly being designed as a set of mobile agents. These are autonomous software entities that can suspend their behavior, move to another host on the network, and continue execution, deciding where to go and what to do along the way [12]. They provide several advantages to design and control distributed applications (e.g., autonomy, dynamic adaptation, software deployment, distributed and heterogeneous computing, a better use of the network resources and reduction of communication with respect to latency, bandwidth and connection time). However, one of the main obstacles to widespread adoption of the mobile agent paradigm is security. The mobile agents must, therefore, be protected from any act aiming at the deterioration, the destruction or the handling of their code, their state or their data.

Salima Hacini is with Lire Laboratory, Computer Science Department, Mentouri University of Constantine, Algeria (phone: 00 213 31 81 88 17; fax: 00 213 31 81 88 17; e-mail: salimahacini@gmail.com).

Zahia Guessoum is with Agent Team, LIP6 Université Pierre et Marie Curie, Paris, France (email: zahia.guessoum@lip6.fr).

Zizette Boufaïda is with Lire Laboratory, Computer Science Department, Mentouri University of Constantine; Algeria (e-mail: boufriche@hotmail.com).

Examples of such attacks are unauthorized access to the mobile agent private data, malicious alteration of its code and control of its execution (e.g. agent is replayed).The protection of mobile agents against malicious behaviors of execution environments represents a challenging research area [8]. Several approaches have therefore been introduced such as tamper proof hardware [12], function hiding [13], black box [6] or clueless agents [11]. These techniques help to enhance the security of code executing in an untrusted environment. Nevertheless, they present different disadvantages. For instance the tamper proof hardware has a prohibitive cost and the function hiding approach is limited to the polynomial and rational functions. So, an important research issue is to introduce a solution which is neither expensive nor limited and is able to reach an acceptable level of security.

This paper deals with the protection of the mobile agent code. It proposes a solution based on the adaptability concept. An adaptable agent behavior is often unexpected; it is therefore protected because one cannot attack an entity whose behavior is unaware. Our mobile agent can perform several services. It comprises a set of modules. Each time, only a subset of them is involved in the execution of a given service and constitutes the mobile agent behavior. The latter depends on the mobile agent context which, in our case, relies on the environment of the visited host. A set of parameters helps to identify the host. This identification leads to the obligation to detect these parameters' values before selecting the modules which will form part of the service [5].

The context of this paper is given by the scenario where the agent is transmitted by the service provider towards the customer to perform a service. Arrived at the level of the receiving host, it ignores what it will do. It tries to detect the various conditions which must be taken into account for the construction of an environment key. The latter informs about the customer's trustworthiness. As soon as the mobile agent succeeds in generating this key, the customer sends it.

The service provider identifies the key, selects the corresponding abstract expression which determines the specific mobile agent execution and emits it towards the customer.

The goal of this article is to define a model to enable mobile agent to establish the customer trustworthiness by the calculation of an environment key.

The paper is organized as follows. Section II points out the context of this work, highlights our contribution which mainly

lies on the definition of a secure environment. Section III describes the mechanism used to perform the trust as a quantified metric. This mechanism generates an environment key. Section IV describes and analyses the related work and shows the advantages of the proposed approach. Finally, Section V summaries our contribution and describes the future work.

II. The Principles of our Approach

Our approach is based on a mobile agent code protection protocol [5] and a control mechanism to improve trust and enhance security. In this section, we first describe the mobile agent code protection protocol. We then discuss the characteristics of a secure environment and give a definition of trust. We first assume that:

- A contract is built beforehand between the customer and the service provider.
- The service provider knows some confidential information concerning the customer (e.g., contract reference).
- The service provider can already have an idea on the former trustworthiness of the host.
- The host knows something about itself or about the environment that the agent does not know.
- The information that the customer knows has an incidence on the agent owner decision to execute or not the requested service.
- The mobile agent has to calculate the environment key without knowing whether the host private information is right or no.

A. Protection Protocol

This protocol aims to protect a mobile agent code against malicious hosts. The environment key occupies the center of our work since it reveals the trust degree of the target host. When the customer needs a service, it sends a request to all the service providers. These providers send their proposals. The proposals are then analyzed by the customer to select the best proposal and inform the corresponding service provider which generates private and public keys, and assigns specific key and the adequate abstract expression to each behavior of the mobile agent. The mobile agent moves then to the customer host. The main steps of this protocol are (see Fig.1):

1) The mobile agent starts interacting with the environment in order to obtain the needed information to generate the environment key (see box 1).

2) The customer encrypts the environment key with the public key of the service provider and sends it to the provider (see box 2).

3) The service provider deciphers the received key (see box 3.1). It identifies the key, selects the corresponding abstract expression, encrypts this expression with the environment key and sends it to the costumer (see box 3.2).

4) The customer tries to decipher the abstract expression with the environment key (see box 4). In case of success, it executes the requested service.

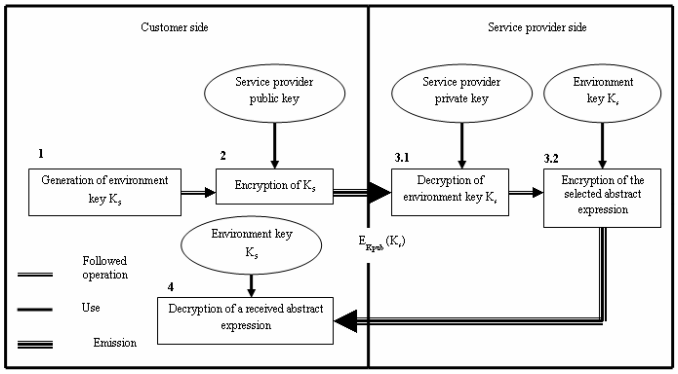

Fig. 1 Mobile agent protection protocol

B. A Secure Environment

Any discussion of computer security necessarily starts from a statement of requirements, i.e. what does really mean an environment of a mobile agent is "secure"?

Castelfranchi *et al.* in [3] claim that the richness of the mental ingredients of trust cannot and should not be compressed simply in the subjective probability estimated by the actor for its decision. The question therefore is: why do we need an explicit account of the mental ingredients of trust?

Observation is required to establish trust and to enhance security. Thus, if a mobile agent does not trust the host, it uses observation to prevent or at least detect its misbehavior. Moreover, if the host knows that it is observed, it tries to be more reliable.

Several definitions of trust have been proposed. For instance, Josang et al. propose the following definition which is suitable for dynamic environment: *trust is the extent to which one party is willing to depend on somebody, or something, in a given situation with a feeling of relative security, even though negative consequences are possible* [7].

Trust of hosts relies thus on several parameters and malicious behaviors. To define trust, we need to answer several questions:

- How can the agent perceive its reception environment so that it can emit a right opinion (on the trust and the profile of the host)?
- How can various perceptions be aggregated to generate the environment key?
- How can this key determine exactly the category of customer?
- How can this key, in case of misbehavior, inform about the origin of the failure?

The next section describes the steps that enable the generation of the environment key and thus the estimation of trust.

III. Establishment of Trust

The idea that trust is quantifiable is common. However, the establishment of trust parameters, their evaluation and thus the trust quantification remain a quite subjective task.

Trust establishment is obtained by a monitoring of the

target host (by observing it, inspecting it and questioning it) and for agent owner reaction (by performing the service, reducing it or stopping it). To evaluate trust, our mobile agent uses:

- observation mechanism that captures all parameters apt to contribute to the perception of the environment and to inform about the host's trust degree,
- interaction mechanism that enables the mobile agent to ask the host questions for which the agent owner already knows the answer. In this case, trust could be based on private information and
- inspection mechanism that allows examining the environment in the search of particular information.

For a good intervention of the mobile agent owner, the analysis of the revealing key of the trust degree must highlight the causes (sources) of this result. So, the key should not be a simple value but an aggregation of a set of significant values.

The whole concept of trust is based on variables. Based on the values of these variables, the mobile agent collects information which will allow the environment's trust estimation.

A. Trust Acquisition

The quantitative dimensions of trust are based on the quantitative dimension of its cognitive constituents [3]. To determine the host trust, we must identify the following parameters:

- Parameters that make a transaction trustworthy.
- Parameters that determine a level of trust of every costumer.
- Parameters that determine a set of costumers to which the host belongs.
- Software and hardware parameters that may affect perception of trust and transaction fulfillment.
- Reputation (may not exist) of hosts provided by the agent's owner or a third party and witch related to the history of host's transactions.

We assume that the values of all these parameters could be aggregated to produce an environment key. The latter informs the agent owner on the customer trustworthiness.

We distinguish between internal and external trust and subdivide parameters on two sets:

1) Internal parameters which define internal trust. They include all information apt to authenticate the relation existing between the customer and the service provider as identity (name, organization…), contracts (type, reference, service acronym, validity date…), certificate, private information (password…), …

2) External parameters which define external trust (circumstances, reputation, situation, environment, infrastructure …).

Basing on the importance given to a parameter compared to another, the trust composition of internal/external parameters produces different evaluations. Moreover, depending on the external or internal attribution of the diagnostic of lack of trust, the strategies of trust establishment will be very different. When opting for the best reaction it will adopt, the agent owner must distinguish between the two parameters (if the host is failing or diffident this does not mean automatically that it is malicious). The analysis of the generated environment key enables to make a meticulous diagnostic of lack of trust.

B. Key Generation

We propose to use simultaneously the asymmetric and the symmetric cryptographic methods.

Let $E = \{E_1, E_2 \ldots E_n\}$ be a set of n abstract expressions that is used to implement the different behaviors of the mobile agent and let $A = \{A_1, A_2 \ldots Ap\}$ be a set of p adaptable modules (including dummy ones) that is included in the different implementations [5]. Each E_j ($j \leq n$) is a sequence of calls of a subset of A and can be viewed as a sequence of bits (each bit indicates a specific module). If we consider p modules, we can have 2^p-2 possible combinations. Each combination is associated to an abstract expression (without considering the empty expression).

The environment key Ks_j is generated for each abstract expression E_j. This key definition uses information collected at the level of the target host with the mobile agent identifier which is unique. It is based on a hash function coupled with a public key cryptography. We suppose the existence of the following couples of public and secret keys:

- Agent Owner keys: (P_O, S_O)
- Host keys: (P_h, S_h)

As soon as the mobile agent arrives at the level of the customer host, it executes some actions which enable trust acquisition (see Algorithm1).

Algorithm 1: The mobile agent behavior

1: Collect data (corresponds to parameters values); let $\{d_1, d_2 \ldots d_k\}$ be the set of collected data.
2: Apply a SHS (Secure Hash Standard) one way function to each data
For i: = 1 to k do M_i: = $H(d_i)$ End For.
3: Concatenate all digests and obtain $M = (M_1, M_2 \ldots M_k)$.
4: Encrypt M with agent owner public key (P_O (M)).
5: Send the Signed Message $SM = S_h(P_O(M))$ to the service provider.
6: Apply hashing to the result of the third step, let D=H(M) be the final digest.
7: Apply D ⊕ id (where id is a unique mobile agent identifier) to generate an environment key Ks_j. Ks_j will be used to decrypt the abstract expression E_j.
8: Receive an abstract expression
9: Try to decrypt the received abstract expression with Ks_j.
10: If the decryption is successful then Execute the selected service.

In order to evaluate the customer trust degree, the service provider must execute some actions (see Algorithm 2). They enable the customer trustworthiness estimation and thus, the

selection of the adapted service.

Algorithm 2: The service provider behavior

1: Receive $SM = S_h(P_O(M))$.
2: Calculate $P_O(M) = P_h(S_h(P_O(M))$.
3: Calculate $M = S_O(P_O(M))$.
4: Obtain k digests $H(d_1)$, $H(d_2)$, …, $H(d_k)$.
5: Check the obtained digests with the digests present in the database (see Table I).
6: Estimate the host trustworthiness by calculating the value of T (see § III.C).
7: Compare the trust values with the intervals values and select the action to be undertaken (see Table II).
8: Select abstract expression of a selected service; let E_j be the selected abstract expression.
9: Apply hashing to the result of the third step, let $D=H(H(d_1), H(d_2), \ldots, H(d_k))$ be the final digest.
10: Apply $D \oplus id$ (where id is a unique mobile agent identifier) to generate a key Ks_j.
11: Encrypt the selected abstract expression E_j with Ks_j.
12: Sign the encrypted abstract expression and emit it to the customer.

In order to protect the environment key and to avoid transmitting it, the key is calculated on the customer side as well as on the service provider side.

At Step 7 of algorithm 2, three cases can occur. They respectively correspond to three ranges of trust value:

a. All received data are in conformity with the recorded data and reflect the profile of an existing customer.
b. Only a subset of the received data is in conformity with the recorded data. The importance of conformed data induces the service provider feedback (reducing or stopping).
c. No data is in conformity with the recorded data. Thus service provider refuses executing any service (stopping).

C. Mobile Agent Owner Feedback

At the level of the service provider, there is a base which comprises for each customer its own parameters values with their corresponding hashing (see Table I). A matching of the received and existent hashing data is performed. We consider a set of possible reactions:

- Stopping the service (in case of the customer trustworthiness is too low)
- Reduction (replace the requested service by another which is less significant)
- Performing the requested service (in case of the customer trustworthiness is high).

TABLE I
EXAMPLE OF CUSTOMER RECORDS

Parameter	Parameter Value	Digest (SHA-1)
Identifier	Sonelgaz	a2eb857fb1aa9f087c7da9d245 c3101a6b64fe42
Acronym	Esprit 4	ed1f3ef005cc44dae336751db3 ec5b8f01d4411f
Validity date	February 26th,2006	82cccf80bb0cfb6fd735c7f84a5 bf5179bfd234a

The trust estimations are based on intervals of values and not on thresholds (see Table II). This provides a greater flexibility of reaction.

TABLE II
EXAMPLE OF ESTIMATION INTERVALS WITH THEIR RELATED FEEDBACK

Interval of trust estimation	Feedback
0-20	Stopping
21-60	Reducing
61-100	Performing

The considered parameters are subdivided into sets (see Table III):

- Parameters with higher importance (3)
- Parameters with medium importance (2)
- Parameters with lower importance (1)

The trust estimation T of the customer host is relative to parameters attributes values and is calculated according to importance I_J of the parameter J, of its weight W_J and the factor S_J which is equal to 1 in the case of success (matching) and 0 in the case of failure (not matching).

$$T = \sum_{j=1}^{k} w_j \; I_j \; s_j$$

With an aim of deciding on an adequate reaction, the value of T is compared with the limits of the various trust estimation intervals (see Table II). If the trust value belongs to the good interval (e.g., [61-100]) the host is trusted and the mobile agent can, after receiving the appropriate abstract expression, perform the complete service. On the other hand, if the obtained trust value is considered to be low, it is possible to find the exact cause of this failure by seeking among the parameters which had a factor S equal to zero.

Given the importance of some parameters, they can (in case of failure) largely influence the choice of the action to be undertaken. The values assigned to the attributes (the weight and the importance) of each parameter define its impact in the final decision (see Table III).

TABLE III
EXAMPLE OF VALUES OF PARAMETERS CHARACTERISTICS

Parameter	Weigh t	Importance value (3, 2, 1)	Reaction in case of failure (relied on the importance of the factor)
identifier	20	3	Stopping
acronym	10	2	Reducing
e-mail	5	1	Performing

IV. RELATED WORK

The concept of trust has been a subject of large interest in different research areas like economics, game theory and multi-agent systems [1]–[4]. Obtaining and maintaining trust estimate is a serious open problem.

Dimitrakos [4] introduces metrics, costs and utility functions as parameters of an algorithm that produces the trust policy for a given trusting decision.

The security project [2] uses the notion of trust that is linked to risk. Risk is evaluated on every possible outcome of a particular action and is represented as a family of cost-PDFs (Probability Density Function). This action is analyzed by a trust engine to compute one cost-PDF. The decision to take the action is made by applying a user-defined policy to select one of the possible outcomes'cost-PDFs.

Braynov *et al.* [1] give a solution that does not rely on collecting and analyzing information about untrustworthy agents. Instead, they propose an incentive-compatible mechanism in which agents truthfully reveal their trustworthiness at the beginning of every interaction. In this mechanism, agents report their true level of trustworthiness, even if they are untrustworthy.

Manchala [9] develops a model based on trust-related variables such as the cost of the transaction and its history, and defines risk-trust decision matrices. The latter are used together with fuzzy logic inference rules to determine whether or not to transact with a particular party.

Chin Lin *et al.* [10] propose a hybrid trust model employing soft trust mechanisms with constructs such as recommendation, direct experiences via interactions and observations to complement hard trust (based on cryptographic mechanisms) for enhancing the mobile agent security in situations where full authentication trust is not available due to absence or unavailability of trusted third parties.

The majority of these trust models generates a subjective single value which does not reflect the exact cause of the lack of trust. Thus, the provided decision could be inadequate. They use also the reputation of the host as factor intervening in the trust estimation. The mobile agent does not need to know the reputation of the visited hosts that can be new in the network.

Compared to the current models of trust, our mechanism provides several advantages. The mechanism does not require the estimation of other agent's trustworthiness. The mechanism eliminates the need to speculate about their intentions or beliefs. Another advantage of this mechanism is that it could simplify many complex and costly infrastructures for risk assessment like reputation databases. The major advantage remains in the fact that our estimation of trust is based on the concrete values and in the case of failure we can locate the exact source of the problem.

V. CONCLUSION

Efforts for promoting trust play an important role in e-commerce and e-business security, which is a key to the acceptance and general deployment of this type of applications.

So we introduced a model to enable mobile agent to establish the host trustworthiness by the calculation of an environment key. We have also proposed a technique which permits a reaction based on realistic parameters values. This model is flexible: the owner of the mobile agent can modify parameters, intervals of values relying on the importance of the service, its time limitation (if it is or not out of date), the risk incurred (or the damage resulted) …

The proposed trust model is based on experiences since it requests a great ability in the configuration of the factors allowing the trust establishment.

Trust constitutes a method to build host behavior-aware agent and we are now considering the fact that the agent itself could take the initiative to react after the host trust estimation (by being attentive to the protection of its code). This would have the advantage of increasing the mobile agent autonomy in our model.

REFERENCES

[1] S. Braynov and T. Sandhol, "Trust revelation in multiagent interaction," in Proceedings of CHI'02 Workshop on the Philosophy and Design of Socially Adept Technologies, Minneapolis, 2002.
[2] V. Cahill et al., "Using trust for secure collaboration in uncertain environment," in IEEE Pervasive Computing, 2(3), pp. 52–61, 2003.
[3] C. Castelfranchi, R. Falcone, "Trust is much more than subjective probability: mental components and sources of trust," 32nd Hawaii International Conference on System Sciences - Mini-Track on Software Agents, Maui, Hawaii (2000).
[4] T. Dimitrakos, "A service-oriented trust management framework," in R. Falcone, S. Barber, L. Korba, and M. Singh, editors, Trust, Reputation, and Security: Theories and Practice, LNAI 2631. Springer, pp. 53-72, 2003.
[5] S. Hacini, "Using adaptability to protect mobile agents code," in IEEE International Conference on Information Technology ITCC 2005, Las Vegas, pp. 49-53, 2005.
[6] F. Hohl, "Time limited blackbox security: protecting mobile agents from malicious hosts," in G.Vigna (Ed.), Mobile Agents and Security. Lecture Notes in Computer Science, Vol. 1419, Springer-Verlag, pp. 52-59, 1998.
[7] A. Josang, S. Lo Presti, "Analyzing the relationship between risk and trust," in T. Dimitrakos (editor) the Proceedings of the Second International Conference on Trust Management, Oxford, 2004.
[8] N. Karnik, "Security in mobile agents systems," PhD thesis, Department of Computer Sciences and Engineering, University of Minnesota, Minneapolis, USA, 1998.
[9] D.W. Manchala, "E-Commerce trust metrics and models," in IEEE Internet Computer, pp. 36-44, 2000.
[10] Y. Mu, C. Lin, V. Varadharajan, Y. Wang, "On the design of a new trust model for mobile agent security, trust and privacy in digital business," Lecture Notes in Computer Science, Vol. 3184. Springer-Verlag, Berlin Heidelberg Germany, pp. 60-69, 2004.

[11] J. Riordan, B. Schneier, "Environment key generation towards clueless agents," Lecture Notes in Computer Science Vol. 1419, pp. 15-24, 1998.
[12] S. Rouvrais, "Utilisation d'agents mobiles pour la construction de services distribués," thèse de doctorat de l'université de Rennel, 2002.
[13] T. Sander, C. Tschudin, "Protecting mobile agent against malicious hosts," in G.Vigna (Ed.), Mobile agents and security, Lecture Notes in Computer Science Vol.1419, ©Springer-Verlag Berlin Heidelberg, pp. 44-60, 1998.