A Reputation Evaluation System for Mobile Agents
Korba, Larry; Song, Ronggong

National Research Council Canada
Institute for Information Technology

Conseil national de recherches Canada
Institut de technologie de l'information

NRC · CNRC

A Reputation Evaluation System for Mobile Agents *

Korba, L., Song, R.
October 2003

* published in Proceedings of the 5th International Workshop on Mobile Agents for Telecommunications Applications. Marrakech, Morocco. October 8-10, 2003. NRC 46500.

Canada

A Reputation Evaluation Framework for Mobile Agents[1]

Larry Korba and Ronggong Song

Institute for Information Technology
National Research Council of Canada
Ottawa, Ontario K1A 0R6, Canada
{Larry.Korba, Ronggong.Song}@nrc.ca

Abstract. Reputation has recently received considerable attention in e-business applications. Having an indication of the level of esteem a person or object is held may be a key factor for decision-making. Within a mobile agent environment, What sort of framework would be required to implement reputation evaluation? In this paper, we attempt to answer this question by describing a reputation evaluation framework for the mobile agent platforms, which could let the client agents choose the reliable services and protects the privacy of the client agents.

1 Introduction

Mobile agent systems have been expected to take an important role in the future information society and especially in e-business applications [3]. However, a major weakness of e-business applications of mobile agent systems is the raised level of risk associated with the loss of notions of reputation and trust. This is loss trust and reputation is due to each mobile agent only having limited information about the reliability of others, or the product and service quality during transaction, especially when the agent moves to a new platform, or uses a new service. A reputation system, which could collect, distribute and aggregate a participant's past experiences with the existing services, would be useful to build a level of trust in the agent society, for instance helping other agents choose the reliable services in a mobile agent systems. Several reputation systems have been proposed [5, 8, 9], but most of them work on the specific services and applications.

In this paper we describe a general reputation evaluation framework to help the agents choose the reliable services available via mobile agent platforms. Our method is straightforward and is described as follows. Our reputation evaluation system consists of several components: a certificate authority (CA), a reputation evaluation agent, a MIX agent, a service provider agent, and a client agent. Each agent involved in this system, must register and get its identification certificate from the CA after it starts. Each service provider agent must register its services to the reputation evaluation agent. During each term (for instance, a term bay be one week, one month, or one year in duration), the client agents evaluate the reputation of the service via their access through the service provider agents according to their past experience. The evaluation results are protected using a nested hybrid encryption algorithm, and sent to a modified MIX cascade consisting of several MIX agents. The final MIX

[1] NRC paper number 46500

agent sends the last layer ciphers to the reputation evaluation agent. After the reputation evaluation agent gets all of the results, it then calculates and publishes the final reputation evaluation results of the service provider agents.

Our system offers several advantages. First, it would protect the privacy of the client agents during evaluation since the modified MIX cascade outputs the evaluation messages in a randomly permuted order. Second, it prevents the same client agents from repeating the evaluation during the same evaluation term since the first MIX agent could authenticate and record the action of the client agents, and since each MIX agent also has a batch signature for the messages it outputs. Additionally, the MIX agent does not know the evaluation results since the evaluation results are encrypted using the nested encryption algorithm.

Our system does not provide a general algorithm or mechanism for the collection and/or calculation of their past experience to determine reputation since different evaluation mechanisms may be used for this purpose, making it difficult or impractical to specify one. However, our system provides a common platform for aggregation and calculation of reputation for different services via the client agents in the mobile agent platforms. For different services, the client agents need to translate their past experiences to a general reputation value in our system. In order to illustrate its operation, we will also give an example for how to translate the past experience on onion routing services to a general reputation value.

The rest of the paper is organized as follows. A MIX technique is briefly introduced in the next section. In Section 3, the reputation evaluation framework for mobile agent platforms is proposed. The overall architecture, the reputation evaluation scheme, and the modified MIX cascade are described. In Section 4, system security and anonymity are analyzed. In Section 5, we present some concluding remarks.

2 MIXes

In order to enable unobservable communication between users of the Internet, David Chaum [4] introduced the MIX technique in 1981. A MIX accepts a number of messages as input, changes their appearanceusing some cryptographic transformation, and outputs a randomly permuted list of function evaluations of the input items, without revealing the relationship between input and output elements. MIXes can be used to prevent traffic analysis in roughly the following manner.

(1) The message will be sent through a series of MIXes, say $i_1, i_2, \ldots, i_d$. The user encrypts the message with an encryption key for MIX i_d, encrypts the result with the key from MIX i_{d-1} and so on with the remaining keys.

(2) The MIXes receive a certain number of these messages, which they decrypt, randomly reorder and send to the next MIX in the routes.

There are different possibilities to organize the cooperation of several MIXes. These are:

(1) MIX network [1]: All MIXes exist independently from each other in the Internet. The routes can be chosen at random, that is, the user chooses $i_1, i_2, \ldots, i_d$ uniformly at random. This type of cooperation is called a MIX network.
(2) MIX cascade [2]: A single valid chain of MIXes is defined for a group of participants. The route can also be constant, that is, it doesn't change. In this setting, the attacker knows the entry, exit and intermediate MIXes. This kind of cooperation is called a MIX cascade.

Since in our system, the purpose of the MIXes is to protect the privacy of the client agents and hide the ownership of the reputation evaluation messages against traffic analysis attacks, an approach that would simplify system design is to use a MIX cascade. Since we also need to prevent repeating evaluation, replay and collusion attacks, we propose a modified MIX cascade to satisfy our system.

3 Reputation Evaluation Framework

Our reputation evaluation framework is designed to provide a common reputation evaluation service for different service provider agents in the mobile agent platforms. Each platform has a modified MIX cascade, which consists of several MIX agents to provide an anonymous reputation evaluation message forwarding service, a Certification Authority (CA), a reputation evaluation agent, and some client agents and service provider agents.

3.1 Terminology and Notations

Notations used in the paper are defined as follows.

- **CA:** The Certification Authority is the entity that signs and issues the certificate for the local agents. The trust model of the CAs could use mutual cross-certification model or PGP trust model.
- **CMA:** The Client Message Agent is an application agent that translates its past experience to a general reputation value. It makes a nested hybrid encryption and sends the cipher to the MIX cascade. The sole purpose of the agent is to test and demonstrate the common reputation system.
- **SPA:** The Service Provider Agent also is an application agent that can provide some special services to CMA. SPA must register its services with the reputation evaluation agent before starting its services. In this paper, we use onion routing agent as SPA to test and demonstrate our system.
- **MIX:** The MIX agent acts as either an intermediate MIX agent or an authentication MIX agent. An intermediate MIX agent verifies whether the message is correct, decrypts one layer of encryption, mixes and forwards the remaining messages to the next MIX agent or the reputation evaluation agent. Except for the functions of the intermediate MIX agent, an authentication MIX agent, which is the first MIX agent of the MIX cascade, also has a function to authenticate the CMAs.

- **REA:** The Reputation Evaluation Agent provides the registration service for SPAs, collects the evaluation results from the CMAs, calculates and publishes the final reputation results of the SPAs.
- $E_{PK_i}(K_i)$: The symmetrical key K_i is encrypted with a public key PK_i, e.g. *RSA*.
- $E_{K_i}(M)$: The message *M* is encrypted with the symmetrical key K_i, *e.g.using the data encryption standared shared key cryptography (DES).*
- *H (M)*: The message *M* is hashed with a hash function*, e.g. MD5.*
- $Sig_{CMA}(M)$: The message *M* is signed with the CMA's private key, *e.g. RSA*.
- $M_1||M_2$: The message M_1 concatenates with the message M_2.
- *Time* : A current time stamp.

3.2 Architecture

The reputation evaluation framework consists of many mobile agent platforms. Each agent platform has a local CA, REA and MIX cascade, some CMAs and SPAs. The CA and REA usually are located in the main container. The MIX cascade is composed of several MIX agents. The MIX, CMA and SPA agents can be located in different containers. All agents communicate to each other via ACL Message [6].

The CA signs all certification for the whole platform in our system. All agents need to register and get their public key certificates from the CA once they start up. Figure 1 depicts a simple mutual cross-certification trust model that could be used for the CAs of the different platforms. So an agent can easily authenticate other mobile agents through this trust model even if they belong to the different platforms or move to other platforms.

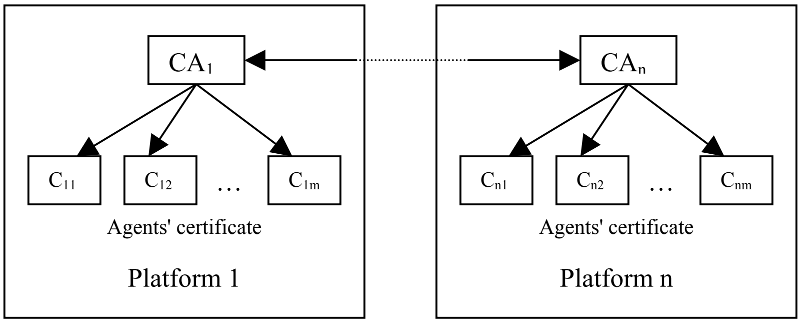

Fig. 1. Mutual cross-certification between two agent platforms trust model.

All SPAs need to register their services with the REA before starting their services. The REA then issues the service certificates to the SPAs and publishes the services including the MIX cascade service. During each evaluation term, A CMA translates its past experience to a reputation value for the services it has used, makes a nested encryption blob, and sends the evaluation message to the authentication MIX agent.

The authentication MIX agent then authenticates the CMAs, verifies whether the messages are correct, mixes and forwards the remaining messages to the next MIX agent. Each intermediate MIX agent verifies whether the messages are correct, decrypts one layer of encryption message, mixes and forwards the remaining messages to the next MIX agent. The final MIX agent then sends the messages to the REA. The REA decrypts the last layer of encryption and gets the evaluation results. After the REA aggregates all evaluation results, it then calculates and publishes the final results. The reputation evaluation framework architecture is illustrated in Figure 2.

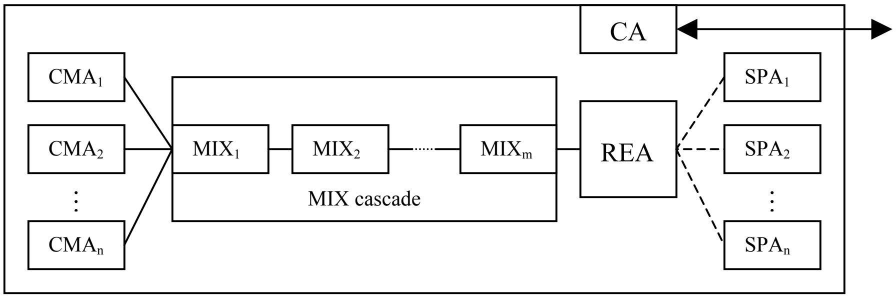

Fig. 2. Reputation evaluation framework architecture.

3.3 Reputation Evaluation Scheme

Our reputation evaluation scheme includes a common reputation measurement means, reputation metrics, reputation translation, reputation collection, calculation and publication. The REA is responsible for the reputation collection, calculation and publication, while the CMA is responsible for the reputation translation.

In the system, we define the range of the common reputation values (meter) from -5 to 5. The positive five is the highest reputation value. The negative five is the lowest reputation value. The zero means that the CMA does not have the experience about the service or the reputation of the service just so-so.

In order to give a better evaluation about the reputation attributes of the services and also make the system simple, we use the following metrics in the REA.

- **Average reputation value (*μ*):**is the arithmetic mean for all past evaluation values ($x_1, x_2, \ldots, x_n$) of the same services from the CMAs. It provides the average reputation of past services provided.

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

- **Total evaluation number (*TN*):** is the total number that the CMAs participated in the evaluation for the same services in the past.

- **Last term average reputation value (μ_L):** is an arithmetic mean for all last term evaluation values ($y_1, y_2, ..., y_m$) of the same services from the CMAs. It describes the current average reputation of the services.

$$\mu_L = \frac{1}{m}\sum_{i=1}^{m} y_i$$

- **Last term standard deviation (σ):**is the standard deviation for all last term evaluation values ($y_1, y_2, ..., y_m$) of the same services from the CMAs. It describes the current debatable degree about the current average reputation value of the services. The standard deviation value ranges from 0 to 5. The value zero means all CMAs' evaluation value is very close the current average reputation value. The value five means all CMAs have the highest debate on the current average reputation value.

$$\sigma = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \mu_L)^2}$$

- **Last term participators' number (*LN*):** is the total number of participant CMAs involved in the evaluation of the same services for the last term.

In the above reputation evaluation metrics, there are two reasons we do not calculate the standard deviation for the whole past. One of them is that we think the current evaluation is more important than the past evaluation. Another reason is that we don't want the REA to store any old (stale) data such that the system requires having the ability for the huge storage. Thus, in the system, the REA just needs to keep the last evaluation metrics and the newest evaluation values from the CMAs. The newest metrics can be deduced from them as follows.

Assume that the tuple (μ, *TN*, μ_L, σ, *LN*) is the last evaluation attributes, and the ($z_1, z_2, ..., z_m$) form all newest evaluation values of the same services which the REA collects from the CMAs. Thus the newest reputation tuple (μ', *TN'*, μ_L', σ', *LN'*) can be calculated as follows.

$$\mu' = \frac{1}{TN + m}[\mu \bullet TN + \sum_{i=1}^{m}(z_i)]$$

$$TN' = TN + m$$

$$\mu_L' = \frac{1}{m}\sum_{i=1}^{m} z_i$$

$$\sigma' = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(z_i - \mu_L')^2}$$

$$LN' = m$$

After the REA calculates this new tuple, it destroys the old reputation attributes (μ, *TN*, μ_L, σ, *LN*) and the data ($z_1, z_2, ..., z_m$), and publishes the new evaluation attributes for the services.

Translation of past experience of the CMAs to the common reputation value is the work of the CMAs. As we mentioned in Section 1, since the different services may use different evaluation mechanisms it is difficult and also impractical to give a common algorithm that would allow the CMAs to translate their past experience to a common reputation value. Here we just give an example for Onion Routing services [7] as follows.

Assume that the SPA_1 registers an onion routing service, and the CMA_1 is a client agent. During each term, the CMA_1 records its experience with this service. It adds 1 score to the service of the SPA_1 when it successfully uses the service, and also records the total number times the service has been used. Consider in one term, if its experience is m successful times out of total n times it has used the service, it can use the following algorithm to translate its experience to the common reputation value R = 10 • (m/n) - 5. For example, if the experience is 100 successful times out of 120 total times to use the service, the common reputation value is 3.3 (≈ 10•100/120-5).

3.4 Modified MIX Cascade

In order to protect the privacy of the client agents and prevent the repeating evaluation, replay and collusion attacks, we propose a modified MIX cascade. The modified MIX cascade consists of three components: CMA, MIX agents and REA.

(a) CMA processing

At the end of the evaluation term, each CMA fills a standard reputation evaluation form made by the REA, and then prepares its evaluation ACL Message as follows.

① The CMA first randomly creates a session key (K), encrypts the evaluation form (M) with the session key and encrypts the session key with the REA's public key (PK_{REA}), and puts them together.

② It then randomly chooses another session key (K_n), encrypts the above message using K_n and encrypts the session key using the last MIX agent's public key (PK_{MIX_n}), puts them together, and so on.

③ Finally, the message is encrypted with a session key (K_l), and the session key is encrypted with the first MIX agent's public key (PK_{MIX_1}). The CMA puts the above message together and attaches its identity (ID) and a time stamp ($Time$), and then hashes the whole message and signs the hashing value with its private key.

$$
\begin{aligned}
&E_{PK_{\text{MIX}_1}}(K_1) \,\|\, E_{K_1}(\\
&\qquad E_{PK_{\text{MIX}_2}}(K_2) \,\|\, E_{K_2}(\;\ldots\ldots \\
&\qquad\qquad E_{PK_{\text{MIX}_n}}(K_n) \,\|\, E_{K_n}(\\
&\qquad\qquad\qquad E_{PK_{REA}}(K) \,\|\, E_K(M))\ldots)) \,\|\, ID_{CMA} \,\|\, Time \,\|\, Sig_{CMA}(H(M_W))
\end{aligned}
$$

Fig. 3. A nested hybrid encryption reputation evaluation message.

The CMA then sends the above ACL Message to the first MIX agent. Figure 3 depicts the final ACL Message, where M_W represents the above whole message except for the signature.

(b) MIX agents processing

When the first MIX agent gets the reputation evaluation ACL Message from a CMA, it first verifies whether the time stamp and the signature are correct, and then checks whether the CMA did the evaluation before. If everything is ok, the MIX agent records the evaluation event for this CMA (this is done to protect against the repeating evaluation attack), and then it decrypts one layer of encryption of the message. When the MIX agent gets enough evaluation messages from the CMAs (e.g. 100 or 1000), it reorders all messages randomly, and then puts a time stamp to the batch messages. Finally, it hashes the whole message and signs the hashing value with its private key. Thus the final messages which the first MIX agent sends to its next MIX agent, has the following appearance:

$$\{\ n \,\|\, M_1 \,\|\, M_2 \,\|\ldots\|\, M_n \,\|\, Time \,\|\, Sig_{\mathrm{MIX}_1}(H(n \,\|\, M_1 \,\|\ldots\|\, M_n \,\|\, Time))\ \}$$

where n is the number of the evaluation messages M_i ($1 \le i \le n$).

When each intermediate MIX agent gets the evaluation messages, it verifies whether the time stamp and the signature are correct. If correct, it records the signature of the batch messages against the repeating evaluation and collusion attack, and then repeats the above processing of the first MIX agent.

Finally, the REA decrypts the last layer of encryption message and gets the reputation evaluation values.

3.5 Reputation Evaluation Example

In this section we provide an example of reputations in action. The example is depicted in Figure 4 where we show just the last term evaluation attributes.

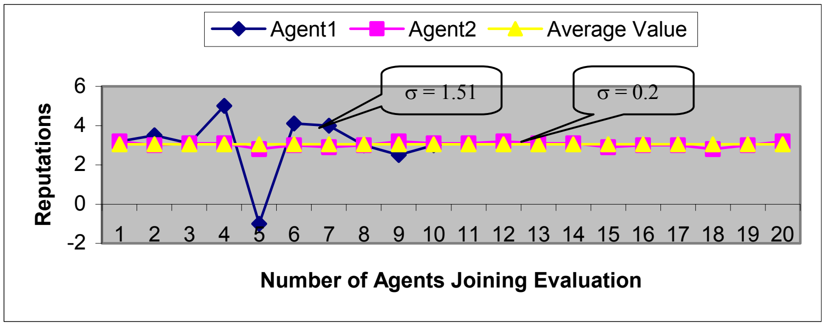

Fig. 4. Evaluations of Two Agents.

In this scenario, we assume that Agent1 and Agent2 have the same services and also have the same average reputation value 3.04. However, for Agent1, it only has evaluation values from 10 client agents and also its standard deviation value is high (σ = 1.51). Agent2, it received evaluation values from 20 client agents with a standard deviation value of just 0.2. From this figure it is clear that Agent2 has a much better reputation than Agent2.

4 Security and Privacy Analysis

In this section we discuss the strengths of the privacy and security provisions of the platform. We show that it has particular strengths for client agent privacy protection and it prevents attacks via repeated evaluation, replay and collusion.

(a) Anonymity Analysis

The nested hybrid encryption method for the reputation evaluation message offers one key advantage. It provides anonymity of the evaluation messages from the client agents, preventing the reversing retrieval attack, i.e. the REA or intermediate MIX agent could link a message it gets with a message the CMA sends to the first MIX agent only using a cryptographic method. This situation would occur when the system only uses public key encryption, for example, if a CMA sends the following message to the first MIX agent,

$$E_{PK_{\mathrm{MIX}_1}}(\ E_{PK_{\mathrm{MIX}_2}}(\ldots E_{PK_{\mathrm{MIX}_\mathrm{n}}}(\ E_{PK_{REA}}(M))\ldots))\ .$$

When the REA gets the message $E_{PK_{REA}}(M)$, it could retrieve the above message using the public keys of the MIX agents, and determine who is the owner of message *M*. In our system this attack does not work since each MIX agent does not know the session key of the other MIX agents.

In addition, since the messages are output in a random order and are also completely changed through each MIX agent, an adversary cannot determine who is the owner of message *M* using the traffic analysis unless **ALL** MIX agents and the REA collude together.

(b) Security Analysis

In the system, since we use a time stamp for freshness, signature for authentication, a hash function for integrity and encryption for confidentiality, it is difficult for an adversary to repeat, replay and/or modify the evaluation messages that the CMAs sends to the first MIX agent.

Another possible attack would occur if the CMA colludes with some intermediate MIX agents to make a repeating evaluation, avoiding the authentication of the first MIX agent, or if the MIX agent itself makes some fake evaluation messages. In order to protect the system against this kind of attack, we let each MIX agent make a batch signature for the messages that it outputs. If there is a debate, we could determine whom made the attack by just letting each MIX agent show the signature of its previous MIX agent.

5 Conclusions

Mobile and multi-agent systems will play important roles in the future information society, especially for e-business applications. Reputation systems could become an important factor influencing the success of these applications. This paper describes a reputation evaluation framework for different services in the mobile and/or multi-agent systems. The approach offers anonymity for client message agents, while offering protection against powerful adversaries who might attempt to subvert the reputation values through repeating evaluations. The straightforward architecture simplifies system implementation. Currently we are exploring the scalability and performance for our system.

Acknowledgements

We would like to thank all members of IIT at the NRC of Canada and the Communications Security Establishment of Canada for their support towards our Security and Privacy R&D program. We are also grateful to the anonymous referees for helpful comments.

References

[1] A. Pfitzmann and M. Waidner. Networks without User Observability - Design Options. In Advances in Cryptology - Eurocrypt '85, LNCS 219, Springer-Verlag, 1985.

[2] A. Pfitzmann, B. Pfitzmann, and M. Waidner. ISDN-MIXes - Untraceable Communication with Very Small Bandwidth Overhead. Proc.Kommunikation in verteilten Systemen, IFB 267, pages 451-463, Springer-Verlag, 1991.

[3] D. B. Lange and M. Oshima. Seven Good Reasons for Mobile Agents. Communications of the ACM, 42(3), 1999.

[4] D. Chaum. Untraceable Electronic Mail, Return Address, and Digital Pseudonyms. Communications of the ACM, vol.24 no.2, pages 84-88, 1981

[5] D. E. Houser and J. Wooders. Reputation in Internet Auctions: Theory and Evidence from eBay. Working paper: http://w3.arizona.edu/~econ/working_papers/Internet_Auctions.pdf, 2001.

[6] JADE -- Java Agent Development Framework. http://sharon.cselt.it/projects/jade/.

[7] L. Korba, R. Song, and G. Yee. Anonymous Communications for Mobile Agents. Proceeding of the 4th International Workshop on Mobile Agents for Telecommunication Applications (MATA'02), LNCS 2521, pp. 171-181, Barcelona, Spain. Oct. 2002. NRC 44948.

[8] R. Dingledine, M. J. Freedman, and D. Molnar. The Free Haven Project: Distributed Anonymous Storage Service. LNCS 2009, Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability, H. Federrath (ed.), July 2000.

[9] R. Dingledine, M. J. Freedman, D. Hopwood, and D. Molnar. A Reputation System to Increase MIX-Net Reliability. LNCS 2137, Information Hiding, 4th International Workshop, I. S. Moskowitz (ed.), April 2001.