Scalability of security technologies on multi-agent applications

Song, Ronggong; Korba, Larry

Publisher's version / Version de l'éditeur:
https://doi.org/10.4224/8913794
Report (National Research Council of Canada. Radio and Electrical Engineering Division. ERB); no. ERB-1106, 2003-11-12

National Research Council Canada
Conseil national de recherches Canada

Institute for Information Technology
Institut de technologie de l'information

NRC · CNRC

*Scalability of Security Technologies on Multi-agent Applications **

Song, R. and Korba, L.
November 2003

* published in NRC/ERB-1106. November 12, 2003. 7 Pages. NRC 46530.

Canada

National Research Council Canada
Conseil national de recherches Canada

ERB-1106

Institute for Information Technology
Institut de technologie de l'information

NRC-CNRC

Scalability of Security Technologies on Multi-agent Applications

Song, R., and Korba, L.
November 2003

Canada

NRC-46530

Scalability of Security Technologies on Multi-agent Applications

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. Multi-agent applications have been expected to take an important role in the future information society. However, security protection for multi-agent applications has become critical issue. The scalability of the security technologies also is an important issue when developing practical agent-based applications. In this paper, we present a simulation of the security technologies under JADE multi-agent platforms to test their scalabilities.

1 Introduction

Multi-agent applications have been more and more applied in a wide range of the information society. However, security protection for multi-agent applications has become critical issue, especially some agent-based e-commerce applications. On the other hand, in order to make the agent-based applications practical and efficiency, the scalability of the security technologies embedded in the applications also is important issue.

In this paper, we first propose a testing model for testing the scalability of some important security technologies such as authentication, IPSec [1]-[4], RSA [5], 3-DES, MD5, etc. Based on the testing model, we then simulate the scalability of the security technologies under the JADE multi-agent platform [6], and present an analysis of its scalability problem.

The rest of the paper is organized as follows. Some security technologies are briefly introduced in the next section. In Section 3, a simulation model is designed for the testing. In Section 4, the simulation and testing metrics are discussed. In Section 5, we show the simulation results and analyze the scalability problems. In Section 6, we present some concluding remarks.

2 Security Technologies

There are many security technologies that could be used to support the security protection for the agent-based applications but we only test some important and basic security technologies in this document. They are described as follows.

- **Entity Authentication:** Entity authentication is the testing process whereby one agent is assured of the identity of a second agent. This ensures that the agent is who it claims it is. There are two kinds of entity authentication methods: one is based on the shared secrets such as password or shared key; another is based on the certificates. The former usually uses the cryptographic algorithms such as hash function (e.g., MD5) or symmetric-key cryptography (e.g., 3-DES). It suits the lightweight applications. The latter usually uses the public-key or signature algorithms such as RSA. It has strong protection for the authentication.
- **Confidentiality:** Confidentiality is a security service used to keep the content of the application data from all but those authorized to have it. Two kinds of cryptographic mechanisms could be used for the confidentiality: one is symmetric-key cryptography such as 3-DES and AES; another is asymmetric-key cryptography such as RSA.
- **Non-Repudiation:** Non-repudiation is a security service to prevent an agent from denying its previous actions. We usually use the signature algorithms for this (e.g., RSA, DSS).
- **Integrity:** Integrity is a security service to address the unauthorized alteration of data. Usually, we use the hash function or MAC for this (e.g., MD5).

- **IPSec:** IPSec is an integrated security service to provide authentication, confidentiality, and integrity protection for the communication data in the Internet IP layer.

3 Simulation Model

In this part, we want to test the scalability of some special security technologies such as entity authentication, IPSec, RSA, 3-DES, MD5 used in the multi-agent applications. These technologies are vital security measures for information systems and would have a general impact on multi-agent system scalability. The testing results will give a good indication of how these technologies might affect the performance of future multi-agent systems. In order to make the testing simple, we use the following client/server model (see Figure 1).

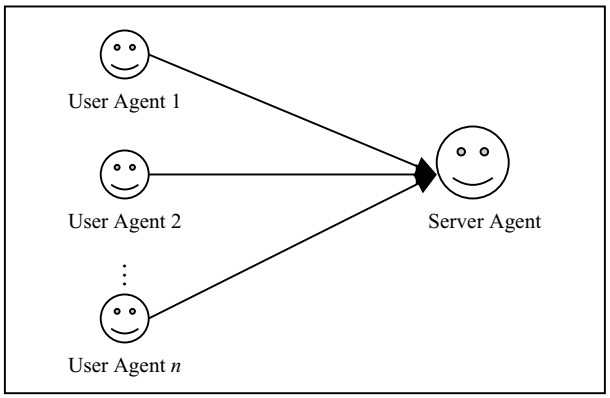

Figure1. Testing Model

4 Simulation Platforms and Metrics

The testing platform includes the hardware and software. The hardware used for the testing includes two computers and local Ethernet. The main testing is on the Intel Pentium 4, the CPU and Memory are 1.50GHz and 256 MB respectively, and the operation system is Windows 2000. The network used for the testing is the local 100Mbps Ethernet.

The software used for the testing includes the operating system and testing software platform. In this paper, all testing is on the Windows 2000 operation system. The testing software platform uses JADE (3.0) multi-agent platform. During testing, we use JAVA as the programming language, and the Java™ 2 Platform, Standard Edition (J2SE™) version 1.4.2 as the essential Java tools and APIs for developing the simulation applications, and IAIK JCE (3.0) as the testing cryptographic package.

In the simulation, the main simulation parameters include user size, total processing time for all messages in the server agent side. The total processing time includes the computing complexity cost for security processing in the above model.

- Message size: the number of the messages sent by the user agents, where each message contains 1Kbits content;
- User agent size: the number of the user agents;
- T_Time: the total processing time that the user agents send all messages to the server agent, and the server agent then processes all messages and sends a reply message to each user agent.

5 Simulation Results

5.1 Entity Authentication Mechanisms

Based on the testing model, we first test the authentication mechanisms under JADE (3.0) multi-agent platform. The server agent is run in the main container, and the user agents are run in the other container, but they all are run in the same computer. The simulation test is described as follows.

The simulation testing was done with the user agents using different entity authentication mechanisms: password-based authentication (using MD5) and certificate-based authentication (using IAIK JCE 1024bit RSA and 2048bit RSA). In this part, we also test the effect of the number of the user agents on the T_Time. During testing, the user agent scalability is from 1 to 2048, where each user agent sends one request-message to the sever agent. Table 1 and Figure 2 depict the total processing time for the request and reply messages under the different authentication processing.

Note: Since as the authentication systems, the workload usually is in the server side, normally, we use the short key as the verification key for the certificate-based authentication in order to make the server agent more efficiency in the testing and real applications.

Table 1. T_Time for the Different Authentication Mechanisms

No. of User agents	1	2	4	8	16	32
MD5 (ms)	80	110	170	320	501	831
RSA 1024 bit (ms)	110	140	200	280	531	901
RSA 2048 bit (ms)	120	150	220	341	551	952

No. of User agents	64	128	256	512	1024	2048
MD5 (ms)	1281	1833	2954	4696	7811	13489
RSA 1024 bit (ms)	1352	2103	3425	5648	9724	17325
RSA 2048 bit (ms)	1772	2804	4646	8062	14220	26479

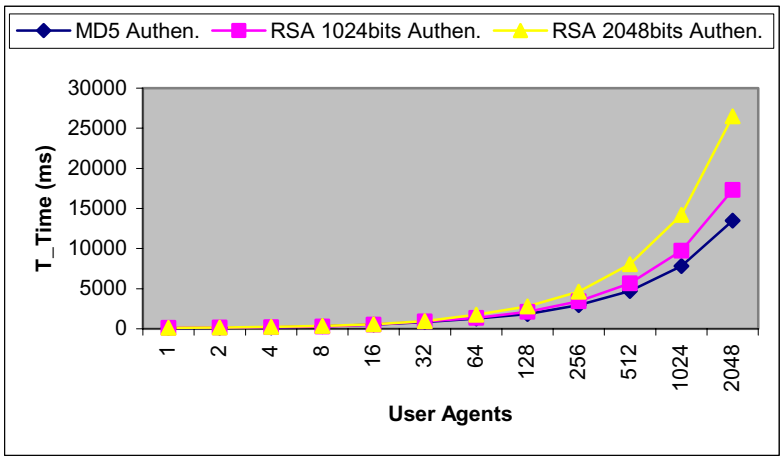

Figure 2. T_Time for the Different Authentication Mechanisms

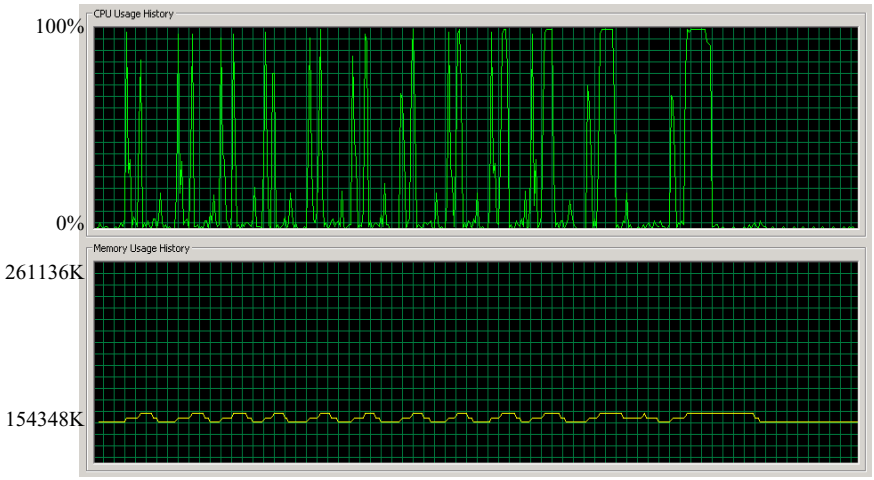

Figure 3. CPU and Memory Usage for the Password-based Authentication

Figure 3 depicts the CPU and Memory usage for the request and reply messages with the password-based authentication technologies. Considering first the CPU usage history, the first pulses are the simulation results for one user agent sending 1 request and reply message, the second pulses are for two user agents and each agent sends 1 request and reply message, ..., the last pulses are for 2048 user agents and each agent sends 1 request and reply message. In addition, each pulse has two peaks in Figure 3, and the first peak is the CPU usage for the JADE platform start-up and the application system setup, and the second peak is the CPU usage for the processing of the authentication messages.

Figure 4 depicts the CPU and Memory usage for the request and reply messages under the certificate-based authentication technologies (RSA 1024bit). Other things are same as the above description.

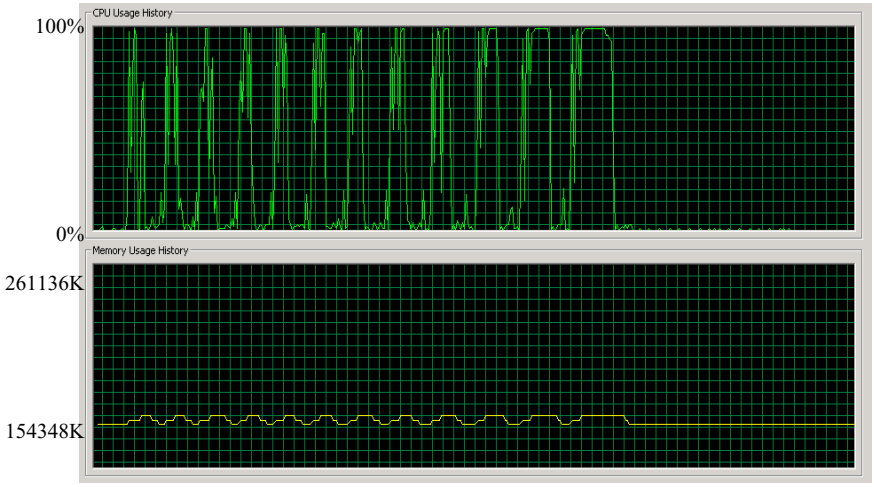

Figure4. CPU and Memory Usage for the RSA (1024bit) Authentication

Figure 5 depicts the CPU and Memory usage for the request and reply messages under the certificate-based authentication technologies (RSA 2048bit). Other things are same as the above description. But in this situation, the CPU usage for the JADE platform start-up and the application system setup is much more than that of the above two situations.

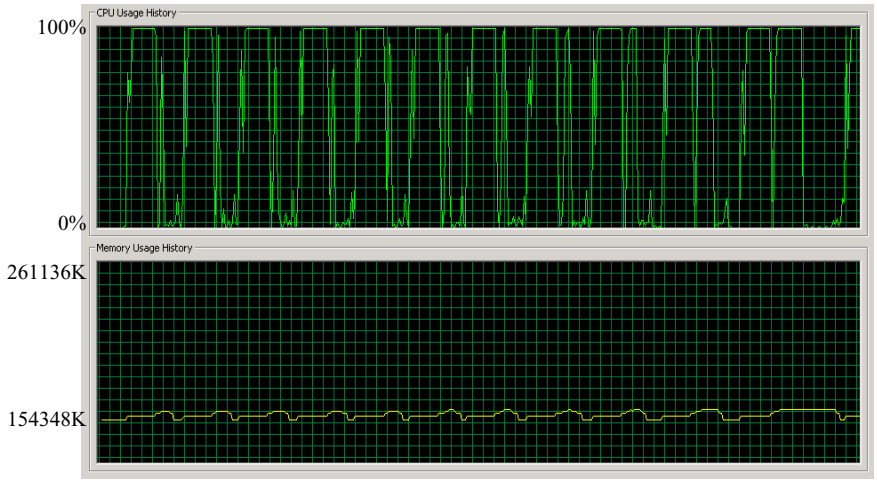

Figure 5. CPU and Memory Usage for the RSA (2048bit) Authentication

From the testing results, we know that the password-based authentication has the best scalability, and the certificate-based authentication (using 1024bit RSA algorithm) also has a good scalability, but the certificate-based authentication (using 2048bit RSA algorithm) has a little bad scalability. On the other hand, as we mentioned, since the verification key (public key) usually is very short (about 14bit) for the certificate-based authentication, its scalability is OK for the real applications. In addition, we also could get a balance between the scalability and the security level for the real applications according to the testing.

5.2 Different Security Technologies

In this part, we test and compare the different security technologies used in the multi-agent applications. During testing, the user agents and server agent are run in the different computers. The simulation test is described as follows.

The simulation testing is done having the user agents adopt different security technologies such as IPSec (AH+ESP), RSA, 3-DES, etc. We simulate the scalability of the user agents from 1 to 128, and each user agent sends 100 request-messages. Table 2, Figure 6, and Figure 7 depict the total processing time for the request and reply messages employing the different security technologies.

Note: In the simulation with RSA algorithm, each message is signed using RSA signature algorithm with a 2048 bit modulus by the user agent and verified by the server agent, but during testing we only collect and calculate the processing time in the server agent side (i.e., the verification time) since in the real applications the user agents usually are distributed on each user's machine. In this testing, we also test the situation that the user agents encrypt the request messages and send them to the server agent. During testing we collect and calculate the decryption processing time in the server agent side.

Table 2. T_Time for the Different Security Technologies

User agents	1	2	4	8	16	32	64	128
Messages	100	200	400	800	1600	3200	6400	12800
RSA Decrypt. (ms)	20960	39296	78823	156325	312049	621223	1242907	2481669
RSA Authen. (ms)	2694	3254	5147	9264	18226	33578	64493	126311
IPSec (ms)	2524	3535	6079	10615	19898	37204	70642	136046
3-DES (ms)	2224	3345	5578	9544	18316	35441	66856	124459
MD5 (ms)	1993	3134	4686	7841	15322	29462	56331	109348
None (ms)	1902	2814	4196	7571	14260	27420	52245	102217

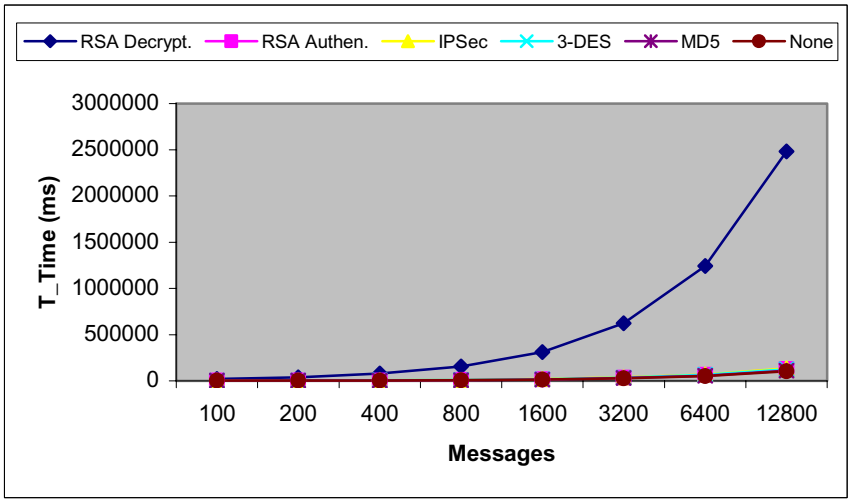

Figure 6. T_Time for the Different Security Technologies

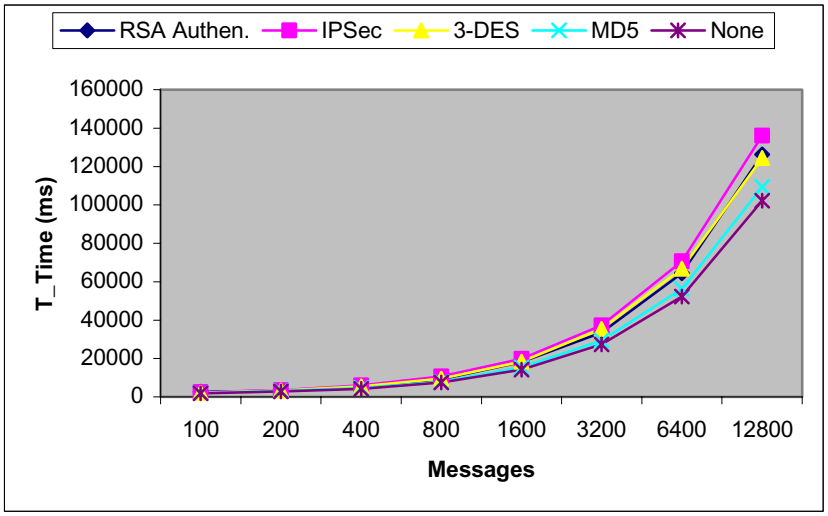

Figure 7. T_Time for the Different Security Technologies

From the testing, we know that most of the basic security technologies such as RSA authentication, IPSec, SSL, 3-DES, MD5, only have a small impact on the multi-agent system scalability. But some public-key cryptography operation like digital signature and decryption would have a huge impact on the multi-agent system scalability since the private key usually is a large key (about 2048bit) for the 2048bit RSA algorithm.

6 Conclusions

Multi-agent systems will play important roles in the future information society, especially for e-business applications, in which security is considered to be the gating factors for their success. Thus security, privacy and trust mechanisms have become the desiderata for multi-agent applications. This paper tests the scalability of the different security technologies used in the multi-agent applications. Simulations show that the most security technologies only have a small impact on the multi-agent systems. But some public-key

algorithms such as signature and decryption would have a large impact on the systems when they are used in the server side.

Acknowledgements

We would like to thank the Communications Security Establishment of Canada for their support towards our Security and Privacy R&D program, and our IST-EU Fifth Framework Project, Privacy Incorporated Software Agent (PISA), partners [7].

References

[1] S. Kent and R. Atkinson. Security Architecture for the Internet Protocol. IETF RFC 2401, November 1998.
[2] S. Kent and R. Atkinson. IP Authentication Header. IETF RFC 2402, November 1998.
[3] S. Kent and R. Atkinson. IP Encapsulating Security Payload (ESP). IETF RFC 2406, November 1998.
[4] D. Harkins and D. Carrel. The Internet Key Exchange (IKE). IETF RFC 2409. November 1998.
[5] R. L. Rivest, A. Shamir and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", *Communications of ACM*, Vol.21, No.2, pp.120-126, Feb 1978.
[6] JADE -- Java Agent Development Framework. http://sharon.cselt.it/projects/jade/.
[7] PISA web site: http://pet-pisa.openspace.nl/.