ISSN 0976 – 1411

Available online at www.internationaleJournals.com

International eJournals

INTERNATIONAL eJOURNAL OF MATHEMATICS AND ENGINEERING

www.InternationaleJournals.com

International eJournal of Mathematics and Engineering 225 (2013) 2201 - 2207

EXPERT SYSTEM FOR REAL-TIME MOBILE AGENT NETWORK BASED ON COMMUNICATION SYSTEM

[1]TARUN DHAR DIWAN, [2], BHOOPENDRA DHAR DIWAN,
[3]JAHANGEER MOHIUDIN LONE
[1, 3] Department of Engineering, [2] Department of Basic Science
Dr. C. V. Raman University, Bilaspur (C.G), India
tarunctech@gmail.com, bddiwan@gmail.com, jahangeer.lone@gmail.com

Abstract—In this paper Many researchers have investigated the development of protection mechanisms in a mobile agent platform. However, the protection mechanisms provided focused on protecting host resources from malicious agents. In these works suspicious cooperating agents have to include protection mechanisms at the application level, which makes the programming of agents a difficult task. In this paper, we proposed the use of semantics for agents to define protection policies using an Interface Definition Language that we have defined. Consequently, the agent application code and the definition of protection policies can be done separately, thus enhancing modularity, and easing programming task. The authentication of agents and the enforcement of their access control policies are done in a transparent way to the agents' applications by our messaging system. as signatures continue to play an important role in financial, commercial and legal transactions, truly secured authentication becomes more and more crucial. to perform verification or identification of a signature, several steps must be performed. Online signature verification has been shown to achieve much higher verification rate than offline verification this paper proposes a novel framework for online signature verification. Different from previous methods, our approach makes use of online handwriting instead of handwritten images for registration. The online registrations enable robust recovery of the writing trajectory from an input online signature and thus allow effective shape matching between registration and verification signatures. In addition, the online registrations enable robust recovery of the writing trajectory from an input online signature and thus allow effective shape matching between registration and verification signatures. in addition, the features have been calculated using 16 bits fixed-point arithmetic and tested with different classifiers, such as hidden markov models, support vector machines, and Euclidean distance classifier. We propose several new techniques to improve the performance of the new signature verification rate system.

Keywords: Verification Rate, markov models, hand writing recognition, training data, testing data. Registration,

1. INTRODUCTION

The goal of research highlighted the benefits of mobile agent technology for large-scale Internet applications, such as distributed electronic commerce frameworks. Mobile agents support resource aware computations; agent migration allows mobile agents to access necessary services locally [1]. Therefore, expensive remote interactions, such as client–server communication over a network, can be minimized. Once a mobile agent has been transferred to a server, it may issue many requests locally at the server. In that way, the use of network bandwidth can be reduced significantly. Other advantages of mobile computations include the support for offline operation, which allows users of mobile computing devices to minimize their connection costs. As a model for distributed computing, mobile agents ease load balancing and help to improve scalability and fault tolerance. Moreover, an agent-oriented programming model facilitates the design and implementation of complex distributed systems [2]. In order to enable agent mobility, dedicated execution environments mobile agent systems have to be developed and to be deployed widely. For the success of a mobile agent platform, a sound security model, portability, and high performance are crucial. Since mobile code may be abused for security attacks (unauthorized disclosure and modification of information, mobile agent platforms must protect the host from malicious agents, as well as each agent from any other agent in the system. In order to support large-scale distributed electronic commerce Applications, mobile agent systems have to be portable and to offer good scalability and Performance [3]. Specifications and mobility of access rights. When an agent moves, the access rights which are used for access control must move with the agent. This gives the agent the possibility to cooperate with other agents on the visiting sites by granting access rights mobile agents, access control lists are used to specify access rights [4]. Two approaches are used to specify and use the lists: in the first approach, as in Aglets and Agents, access control list takes the form of a policy file on a destination. The policy file specifies access policies for mobile agents on the local host resources.

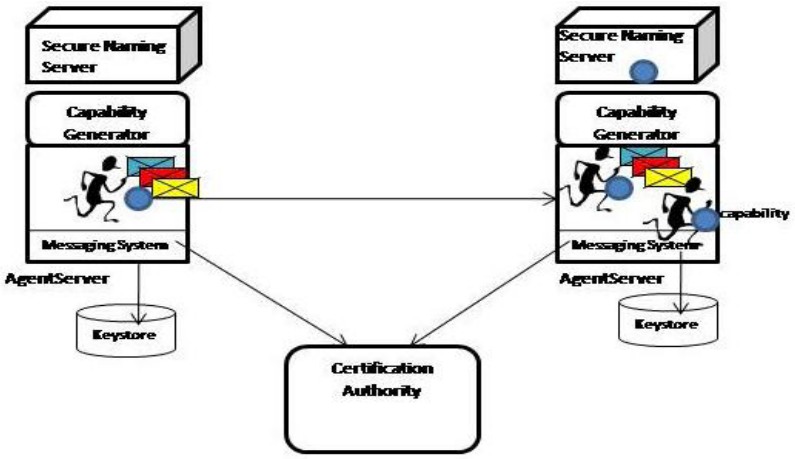

Figure1.Overall Architecture of a Secure Mobile Agent Platform

As stated earlier, this approach is not appropriate for a dynamic environment like the mobile agents [5]. We argue that an agent server cannot predict all the possible cooperation which would take place among cooperating visiting agents.

2. LITERATURE REVIEW

Mobile-agents technology has emerged to build distributed computing over the Internet. A mobile agent is a process with its own code and data that can migrate in the network from one node (called agent server) to another to perform a specific task on behalf of their users. Mobile agents representing different users on a global network can meet and interact with

other agents while migrating in the network [6]. A mobile-agents platform is a distributed middleware that is responsible to create, execute, migrate, send, receive and destroy mobile agents. It also provides communication facilities between mobile agents. As mobile agents are intended to be used over large scale distributed systems [7], security becomes an essential issue to resolve. When received over the network by host servers, a mobile agent must not access resources which it does not have authorization to. Receiving hosts need to have the assurance that a received mobile agent is not malicious. Also, other mobile agents running on the host servers need to have the assurance of whom they are communicating with and consequently give appropriate access rights. Received by host servers, a mobile agent can invoke objects exported either by the servers, or by other agents running on these servers [8]. In this context, protection has become an extremely important issue: nobody will use the mobile-agent paradigm if there are no protection mechanisms which assure the host server and other agents running on this server that the mobile agent will not damage information of the server and of the other agents. Java [9] is probably the best known runtime environment which provides facilities for implementing mobile-code based applications and protection mechanisms. To allow dynamic exchange of access rights in a flexible way between cooperating agents, we have proposed that agents use capabilities for access control [10]. In this initial proposal of protection, the mobility characteristic of an agent was not considered. In particular, when moving from one server host to another, an agent needs to carry its capabilities with it and to export them for other agents running on the destination server.

3.OBJECTIVES

A mobile agent is a particular class of agent with the ability during execution to migrate from one host to another where it can resume its execution. It has been suggested that mobile agent technology, amongst other things, can help to reduce network traffic and to overcome network latencies [11]. The limited processing resources and power supply available to many mobile devices are also arguments for mobile agents. Mobile agents could migrate to hosts with sufficient resources. An agent's ability to move does, however, introduce significant security problems [12]. Today in communication fields research is basically based on the means to provide mobility. If we try to implement mobility then we need modifications in hardware as well as software in existing system. For solving the software problems, a new model is generated, mobile agent based communication system, but this system has still some problems. In mobile agent based communication system there are many problems in networks like low bandwidth, slow data rate and data’s are not secure because signals being available in open.

5. EXPERIMENT & METHODOLOGY

A model of nested protection domains.the kernel maintains a tree hierarchy of agents and service components. Each agent and service executes in a protection domain of its own, called a sealed object or seal for short. Apart from resource limitations, a parent seal may create an arbitrary number of children seals. The root of the tree hierarchy is the Root Seal, which is responsible for starting system services. The Root Seal reads a XML[13] configuration file describing the service infrastructure to be created. Due to a generic configuration format, each service may specify its own set of configuration parameters (nested parameter structures are supported as well). Root Seal executes a well defined service registration protocol, where each service seal registers its interfaces in a local naming service maintained by shows some frequently used service seals: The network service receives mobile agents from other hosts and allows to send them out again. Depending on the application, the network service may implement some protocols to authenticate remote hosts. Furthermore, it is possible to install different network services in the same platform, which may be important if multiple mobile agent applications execute within a single installation.

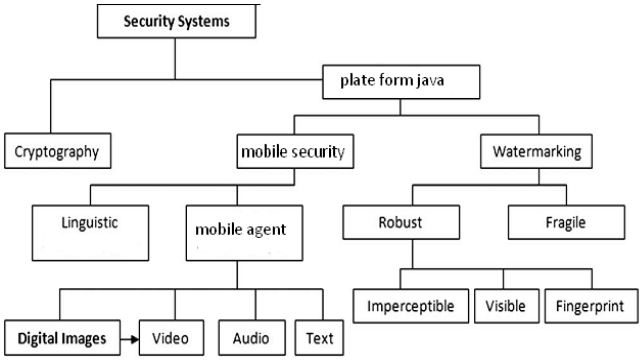

Figure.1mobile Security System Model

The JDBC API is a Java API that used at server side to establish a communication between java and database. With the help of JDBC API user can execute SQL statements in java program, retrieve the fetched results and modify data sources to update database in java JDBC Driver Manager Object class is used to provide connection between java application and JDBC driver [14]. The Driver Manager is very small in size and simple to operate and its very important function in the JDBC architecture JDBC provides these three programming activities.JDBC includes following components The JDBC API: -that provides access method in java by which we access data from relation database management system. The JDBC API can operate in distributed or heterogeneous environment by connecting multiple data source. All API's are either in java.sql or javax.sql.package. we can use Aglet for mobile agent communication and our platform is Tahiti. server is providing GUI (Graphical User Interface) that is used as default user interface to the user. Tahiti presents a main window with a menu bar a list of running agents and toolbar. On the mobile Platform, there is static mobile agent by which we add Security thread in Dynamic mobile agent. So static mobile agent have following tasks[15] . Find out authentication key for dynamic mobile agent Encrypt dynamic mobile agent (Algorithms describe above) Find out MD5 of Dynamic mobile agent Create a GUIplatform by which dynamic mobile agents authenticate each other and check that authentication. Migrate dynamic mobile agent to a given destination And last display result which will get by mobile agent. KDC is just like a central server that provides the authentication passwords for all mobile agents.

Figure.2.The Authentication Passwords For All Mobile Agents.

The agent migration consists of the following steps [1, 2, 15].
a) Serialization of the agent state.
b) Retrieval of the sender agent server's private key and digital certificate, and the digital certificates of owner and writer from the local key store.
c) Creation of an object signature to be used to sign the agent.
d) Initialization of the signature object with the server private key.
e) Updating the signature object using the agent's state for encoding.
f) Production of the signature of the mobile agent using the signature object from step d).
g) Construction of a message which includes the agent's state, code, signature and the (owner, sender, writer)'s digital certificates and sending of the above message to the destination agent server using a TCP/IP connection.
h) Destruction of the agent in the origin agent server.
i) Reception of the message in the destination agent server and creation of a new thread for the execution of the agent.
j) De-serialization of the agent's state.
k) Checking the digital certificates validity
Configuration file has to set where the path of the Input or Output jar/war/class files is mentioned. On running the ProGuard jar, this configuration file will be read to fetch for the input and output.

6. RESULT

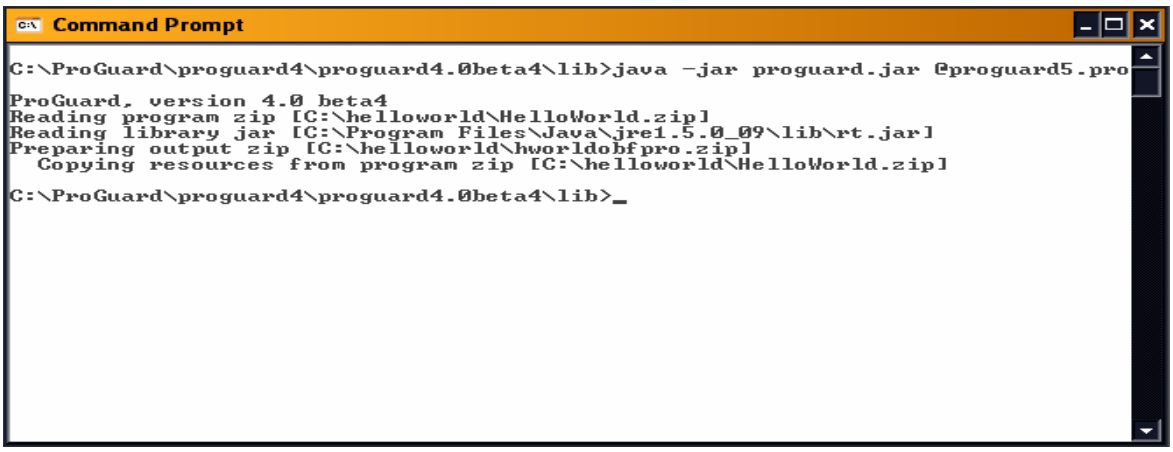

Figure 3.show the working of proguard jar file which is used to obfuscate the Java class files. proguard.pro: is the configuration file which consists of input and output files which is used to obfuscate. This test is taking .zip file as input, this zip file consists of some class file which need to obfuscate, and the rt.jar is a jar(Java Archive) file where the result, the obfuscated class files, are kept.

The measurements in Table1represent the execution time of the Java methods dealing with digital signature manipulation and key store access. We observe that the signature verification is the most expensive operation. The verification algorithm, as implemented by the Java Sun.

Table1 represent the execution time of the Java methods

PROTECTION COMPONENT	*μs*	*μs*
PRIVATE KEY KEY,CERIFICATES RETRIVAL	4.10	5.02
AGENT SIGNATURE CVRU	0.67	.67
AGENT SIGNATURE CERTIFICATES VALIDATION	8.70	8.84
PUBLIC KEYS RETRIEVAL	2.54	3.2
DIGITAL CERTIFICATES VERIFICATION	0.44	0.44
SIGNATURE VERIFICATION CVR	0.87	2.63
SIGNATURE VERIFICATION	50.2	57.3

security provider, uses the signature and the public key of the signer to verify that the signature has been produced by a corresponding private key. the protection overhead is important in case of migration of a minimal agent. We observed an overhead of 182% on a minimal agent migration performance when protection is used. The cost of a minimal agent migration did not compensate the protection overhead which is mainly due to security function used for protection. As the agent size increases, the overhead decreases to 45%. It is obvious that secure mobile agent platform is interesting for mobile agents deployed in large-scaled environments. The protection overhead would compensate the agents migration and local cooperation can take place securely..

7. DISSCUTATION AND CONCLUSATION

We have proposed the framework of security architecture for Mobile agent based communication, along with partial implementation of the system. The design is quite general, so that it can be easily integrated into other mobile agent systems. More over our focus is over an application-layer security for agent based communication to provide end to end authentication and data confidentiality between mobile agents. We have suggested two- way authentication protocol to authenticate mobile agents. This solution can be implemented in aglet (Java based technology) without any changes in underlying protocols and mobile agent communication infrastructure. Transparency enhances the usability issues of nowadays user-level complex security solutions Usability The security mechanisms should not be too much time-consuming or difficult to use. This decreases the productivity of the users who want to use protection and the protection mechanisms end up by being abandoned or used haphazardly introducing security holes to systems Assurance. The authentication mechanisms should meet the non-repudiation security requirements. The sender and the receiver of the mobile agents should not be able to deny sending or receiving agents.

REFERENCES

[1] Domenico Amalfitano, Anna Rita Fasolino, Porfirio Tramontana," A GUI Crawling-based technique for Android Mobile Application Testing", 2011 Fourth International Conference on Software Testing, Verification and Validation Workshops.
[2] Md. Faizan Farooqui, Dr. Md. Rizwan Beg And Dr. Md. Qasim Rafiq," An Extended Model For Effective Migrating Parallel Web Crawling With Domain Specific And Incremental Crawling", International Journal On Web Service Computing (Ijwsc), Vol.3, No.3, September 2012.

[3] Ismet Aktas, Florian Schmidt, Muhammad Hamad Alizai, Tobias Drüner, Klaus Wehrle,"CRAWLER: An Experimentation Platform for System Monitoring and Cross-Layer-Coordination", 978-1-4673-1239-4/12/$31.00 c 2012 IEEE.
[4] Pavalam S. M., S. V. Kasmir Raja, Jawahar M., and Felix K. Akorli," Web Crawler inMobile Systems", International Journal of Machine Learning and Computing, Vol. 2, No.4,August 2012.
[5] Sameendra Samarawickrama, Lakshman Jayaratne," A Survey of Focused Web Crawling Approaches", 11 October 2011, Revised 18 November 2011, Accepted 23 November 2011 -2012 DLINE. All rights reserved.
[6] Jai Balasubramaniyan, Jose Omar Garcia-Fernandez, David Isacoff, E. H. Spafford, and Diego Zamboni, "An Architecture for Intrusion Detection using Autonomous Agents," Department of Computer Sciences, Purdue University; Coast TR98-05,1998.URL: http://www.cs.purdue.edu/coast/coast-library.html
[7] Barrett, Michael, W. Booth, M. Conner, D. Dumas, M. Gaughan, S, Jacobs, M. Little, "Intelligent Agents System Requirements and Architecture," Report to ATIRP,
p. 5, October 1998.
[8] Bauer, David S. and Koblentz, Michael E., "NIDX: An Expert System for Real-Time Network Intrusion Detection," Proceedings of the Computer Networking Symposium, pp. 90-106, April 1988, Washington, DC.
[9] Jeffrey M. Bradshaw,"An Introduction to Software Agents," In Jeffrey M. Bradshaw, editor, Software Agents, chapter 1. AAAI Press/The MIT Press, 1997.
[10] Chess, D., B. Grosof, C. Harrison, D. Levine, C. Parris, G. Tsudik, "Itinerant Agents for Mobile Computing," IBM Research Report, RC 20010, March 1995.
URL:http://www.research.ibm.com/massdist
[11] G.Back and W. Hsieh.Drawing the red line in Java. In Seventh IEEE Workshop on Hot Topics in Operating Systems, Rio Rico, AZ, USA, Mar. 1999.
[12] W.Binder.Design and implementation of the J-SEAL2 mobile agent kernel.In The 2001 Symposium on Applications and the Internet (SAINT-2001), San Diego, CA, USA, Jan. 2001.
[13] W.Binder, J.Hulaas, A. Villaz´on, and R.Vidal. Portable resource control in Java: The J-SEAL2 approach. In ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA'01), Tampa Bay, Florida, USA, Oct. 2001.
[14] G. Bollella, B. Brosgol, P. Dibble, S. Furr, J. Gosling, D. Hardin, and M. Turnbull. TheReal-TimeSpecification for *Java*. Addison-Wesley, Reading, MA, USA, 2000.
[15]Gosling,J., B. Joy, and G.L. Steele. (1996). The Java Language Specification. The Java Series.Reading, MA: Addison-Wesley. Available via WWW at URL: http://java.sun.com/docs/books/jls/html/index.html.

AUTHOR BIOGRAPHIES

TARUN DHAR DIWAN RECEIVED HIS MASTER OF ENGINEERING (COMPUTER TECHNOLOGY AND APPLICATION) DEGREE FROM CHHATTISGARH SWAMI VIVEKANANDA TECHNICAL UNIVERSITY –BHILAI, INDIA, AND MASTER OF PHILOSOPHY (GOLD MEDAL LIST) FROM DR. C.V. RAMAN UNIVERSITY. HE IS CURRENTLY AN HOD & MTECH CO-ORDINATOR DEPTT. OF ENGINEERING AT THE DR.C.V.RAMAN UNIVERSITY-BILASPUR, INDIA. HIS CURRENT RESEARCH WORK ARTIFICIAL INTELLIGENT, IMAGE PROCESSING AND SOFTWARE ENGINEERING.