Security on Mobile Agent Based Crawler (SMABC)

Nisha Pahal
Y.M.C.A. Institute of Engineering, Faridabad
Haryana (India)

Sunil Kumar
Lingayas University, Faridabad
Haryana (India)

Ashu Bhardwaj
Y.M.C.A. Institute of Engineering, Faridabad
Haryana (India)

Naresh Chauhan
Y.M.C.A. Institute of Engineering, Faridabad
Haryana (India)

ABSTRACT
Mobile agents are active objects that can autonomously migrate in a network to perform tasks on behalf of their owners. Current web crawler uses the concept of Mobile Agent to enhance their crawling speed. In mobile crawling, mobile agents are dispatched to remote web severs for local crawling and processing of web documents. After crawling a specific web server, they dispatch themselves either back at the search engine machine, or at the next web server for further crawling. It gives a complete distributed crawling strategy by utilizing the mobile agent's technology but it suffers with the problem of security. In this paper, a security solution has been proposed, which protects both the mobile agent itself and the host resources that encrypt the data before passing it to mobile agent and decrypt it on the visited host sides. The method of "computing with encryption function" has been used. The proposed approach solves the problem of malicious host that can harm mobile agent or the information it contain.

Keywords:
Mobile Agents, crawler

1. INTRODUCTION
Mobile agent [1], namely, is a type of software agent, with the feature of autonomy, social ability, learning, and most importantly, mobility. It is basically a composition of computer software and data, which is able to migrate (move) from one computer to another autonomously and continue its execution on the destination computer. It provides a new abstraction for deploying functionality over the existing Internet infrastructure. Agents [2], [3] are independent pieces of software capable of acting autonomously in response to input from their environment and may be either stationary, always resident at a single platform; or mobile, capable of moving among different platforms at different times. 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. Mobile agent technology therefore offers a new computing paradigm in which a program, in the form of a software agent, can suspend its execution on a host computer, transfer itself to another agent-enabled host on the network, and resume execution on the new host. Mobile agent based crawler is a program that can migrate from machine to machine in a heterogeneous network. It is an

efficient, scalable solution to establishing a specialized search index in the highly distributed, decentralized and dynamic environment of the web. In this, mobile agents [7], [8] are dispatched to remote web severs for local crawling and processing of web documents and after crawling a specific web server, they return either back at the search engine machine, or at the next web server for further crawling. The crawlers [9] based on this approach are called as Mobile Crawlers. It gives a complete distributed crawling strategy by utilizing the mobile agent's technology.

The goals of Mobile Crawling [10] System are: -

- To minimize network utilization,
- To keep up with document changes by performing on-site monitoring,
- To avoid unnecessary overloading of the Web servers by employing time realization,
- To be upgradeable at run time.

2. RELATED WORK
The Anchor Toolkit is a mobile agent system that provides for the secure transmission and management of mobile agents [11]. The toolkit protects the agents being dispatched between hosts through encrypted channels. A mobile agent's host platform is required to sign the agent's persistent state before dispatching the agent to the next platform. The signed persistent state can be used later to detect potential problems with the agent's state.
State Appraisal [12] defines a security mechanism for protection of mobile agents. The goal of State Appraisal is to ensure that an agent has not been somehow subverted due to alterations of its state information. Both the author and owner of an agent produce appraisal functions that become part of an agent's code. Appraisal functions are used to determine what privileges to grant to an agent based both on conditional factors and whether the identified state invariants hold. An agent whose state violates an invariant can be granted no privileges, while an agent whose state fails to meet some conditional factors may be granted a restricted set of privileges. When the author and owner each digitally sign an agent, their respective appraisal functions are protected from undetectable modification. One way of looking at this in comparison with attribute certificates is that state

appraisal conveys both the policy engine and the prescribed policy internal to the agent. An agent platform uses the functions to verify the correct state of an incoming agent and to determine what privileges the agent can possess during execution. Privileges are issued by a platform based on the results of the appraisal function and the platform's security policy.
In reference [13], regarding to the real time issues and also the generic architecture of search engines, a prototype based on a mesh has been presented. Web data is distributed onto various processors of the parallel system. The number of nodes that must be processed in each processor are calculated by, where h is number of levels in the N-ary tree and k is the number of processors in the parallel system. It can be said that this method is presented only in theory, and its implementation has high cost. Reference [14] has presented an adaptive web search system based on a multi-agent reactive architecture, which comes from biological researches on the ant searching behavior.

3. DRAWBACKS OF MOBILE-AGENT BASED CRAWLER

Mobile agent based crawler has some limitations, primarily in the area of security. Recent research efforts in the area of mobile agent security adopt two different points of view. Firstly, from the platform perspective, there is a requirement to protect the host from malicious mobile agents such as viruses and Trojan horses that are visiting it and consuming its resources. Secondly, from the mobile agent point of view, it needs to protect the agent from malicious hosts. Therefore, security [4], [5], [6] is a fundamental precondition for the acceptance of mobile agent applications. The system should have a program that actively protects itself against execution environment that possibly may divert the intended execution towards a malicious goal.
Many approaches aim at protecting mobile agents. There are some problems, which have to be solved before these approaches can be used. The particular attacks regarding malicious host or malicious agent can be summarized as follows.

- **Unauthorized Access:** Malicious mobile agents try to access the services and resources of the platform without adequate permissions. In order to protect this, a mobile agent platform must have a security policy specifying the access rules applicable to various agents, and a mechanism to enforce the policy.
- **Masquerading:** In this, a malicious agent assumes the identity of another agent in order to gain access to platform resources and services, or simply to cause mischief or even serious damage to the platform. Likewise, a platform can claim the identity of another platform in order to gain access to the mobile agent data. In both cases, the malicious agent or platform will not receive any blame for its potentially detrimental actions. Instead, the unsuspecting agent or platform whose identity was misused will be held responsible.
- **Denial of execution:** A malicious platform can cause harm to a visiting mobile agent by ignoring the agent's request for services and resources that are available on the platform, by terminating the agent without notification, or by assigning continuous tasks to the agent so that it will never reach its goal. Likewise, a malicious agent may attempt to consume the resources of the platform, such as disk space or processing time, or delete important files or even the whole hard disk contents, thus causing harm to the platform and launching a denial of service attack against other visiting agents.
- **Annoyance attack:** Examples of this include opening many windows on the platform computer or making the computer beep repeatedly. Such attacks may not represent a very serious problem to the platform, however they still need to be prevented.
- **Eavesdropping:** In this, a malicious platform monitors the behavior of a mobile agent in order to extract sensitive information from it. This is typically used when the mobile agent code and data are encrypted. Monitoring may include the identity of the entities that mobile agent is communicating with, and the types of services requested by the mobile agent.
- **Alteration:** In this, a malicious platform tries to modify mobile agent information, by performing an insertion, deletion and/or alteration to the agent's code, data, and execution state. Modifying the mobile agent execution code and state may result in the agent performing harmful actions to other platforms, including the agent's home platform.
- **Confidentiality:** It is important to ensure that the information carried by a mobile agent or stored on a platform is accessible only to authorized parties. This is also the case for the communication among mobile agent paradigm components.
- **Integrity:** It is essential to protect the mobile agent's code, state, and data from being modified by unauthorized parties. This can be achieved either by preventing or by detecting unauthorized modifications.
- **Availability:** Platforms typically face a huge demand for services and data. In the case that a platform cannot meet mobile agents demands, it should notify them in advance. Additionally, a platform must be able to afford a certain level of fault-tolerance and fault-recovery from unpredicted software and hardware failures.

It is very difficult to protect mobile agent as it visit different node in network so security should be applied on the platform. In this paper a technology called "Computing with Encryption Function" has been used for protection.

4. PROPOSED ARCHITECTURE FOR SECURITY ON MOBILE AGENT BASED CRAWLER

In this paper, a mobile agent-based security model has been proposed which supports flexible and specific security measures required by mobile computers and devices in distributed systems. This approach prevents privacy attacks and integrity attacks to mobile agents from malicious hosts. It is an extension of mobile cryptography and it removes many problems found in the original idea of mobile cryptography while preserving most of the benefits. The mobile agent program is encrypted on the client side before it is transferred to other hosts, and decrypted on the remote hosts side when it gets to the right hosts. The encryption needs not only a private key for the computation with cryptography method but also the private key of every

destination host. After the destination hosts have received the encrypted agent, they decipher or decrypt the cryptograph and recover the original mobile agent code using their corresponding private keys. The proposed SMABC architecture is shown in Fig. 1.

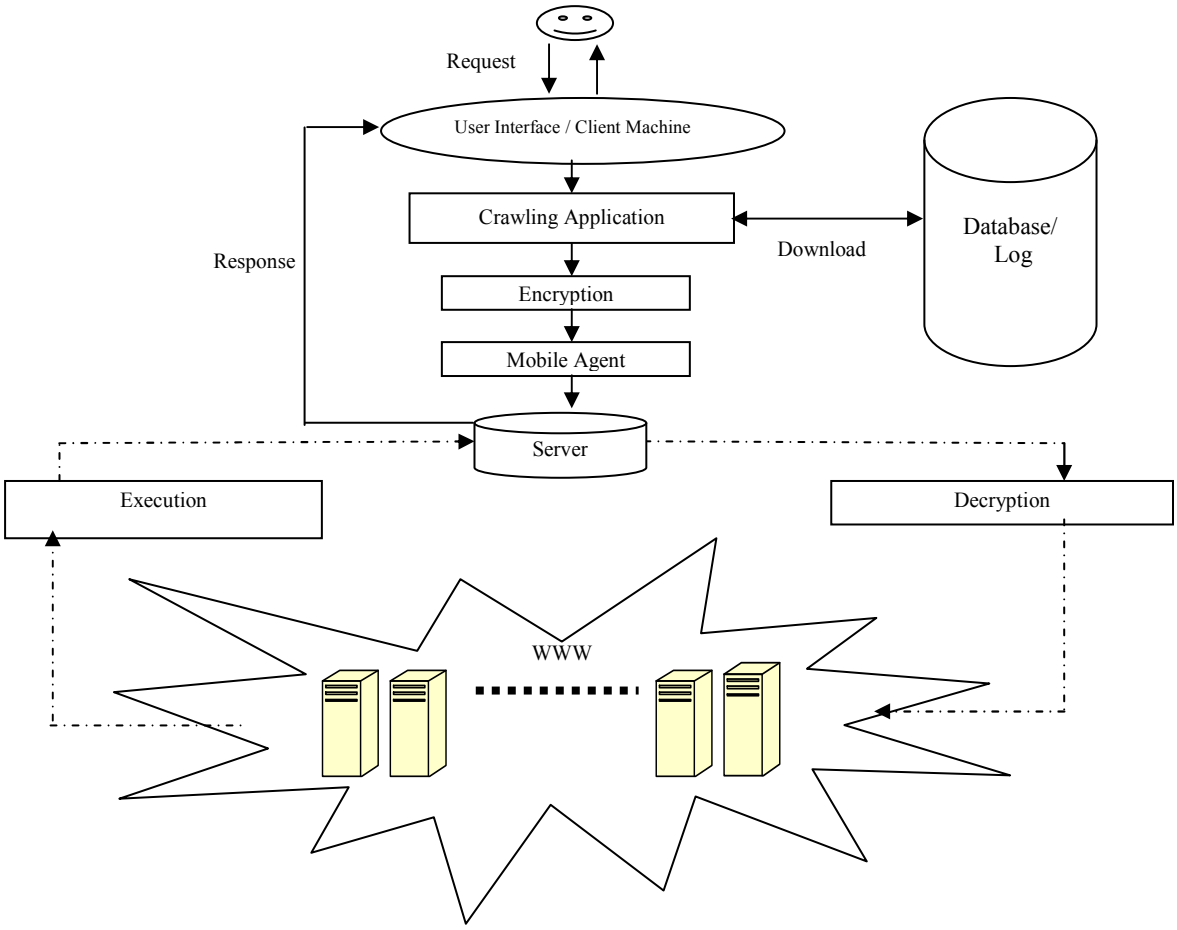


Fig. 1 Security on Mobile Agent Based Crawler (SMABC)

Initially the user sends the request to the user interface/ client machine. It is then passed to the crawling application/search engine which first downloads its related document from the database/log if exists and analyze the downloaded documents and fetches embedded URLs (if present). After that the encrypted URLs are passed to mobile agent. These are then passed to the server where these are decrypted and then downloaded from the web. The downloaded documents are then passed to the server, which are responded back to the user interface. Since the URL given to mobile agent is encrypted, so any host or malicious agent can't retrieve the mobile agent information and it will securely be transferred to web server and the web server can use the URL or such information after decrypting it.

To summarize the above architecture, work has been divided mainly in two parts: First, encryption is done after crawling the web and before submitting to the mobile agent. Second, decryption is performed at the web server to redirect it.

5. FLOW-DIAGRAM OF SECURITY ON MOBILE AGENT BASED CRAWLER

Flow diagram is divided into two parts as shown in Fig. 2:

1. Client Side: Client Side can be referred to as Crawling application, encryption and mobile agent.
2. Server Side: Server Side can be referred to as implementation of decryption and redirecting URL at web server.

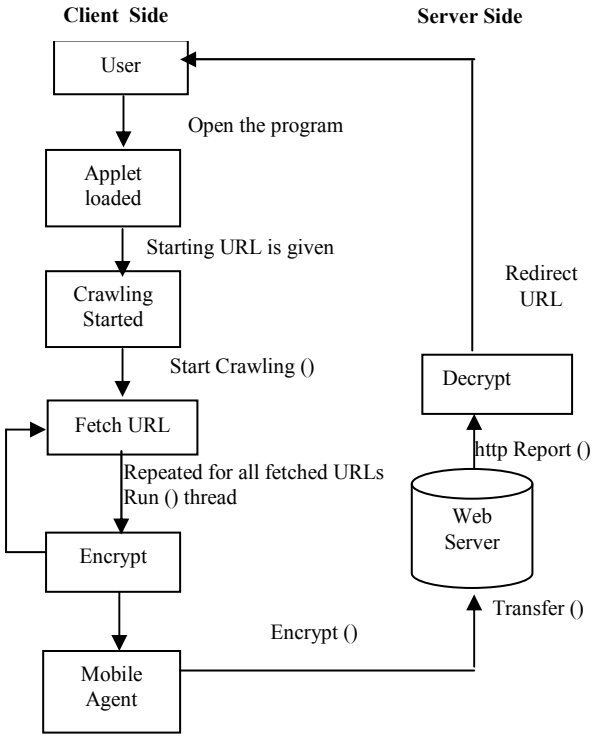


Fig. 2: Flow Diagram

ALGORITHM OF IMPLEMENTATION

The work has mainly divided into two parts: First, encryption after crawling the web and before submitting to the mobile agent and second, decryption at the web server to redirect it. The proposed security algorithm for implementation has been shown below: -

CLIENT SIDE: -

1. Read start URL.
2. Verify it is correct.
3. Search the page recursively from content into list.
4. for (i=0; i<=size (list); i++)
5. Encrypt (URL) and given to mobile agent

Input Parameter; String s, int key

i. ARR[10] ← 10 distinct value
ii. Increase element of ARR by key
iii. l← length (s)
iv. for (i=0; i<=l; i++)
v. ch← char at (s,e)
vi. new ch← ch +ARR[i mod 9]
vii. encrypt str ← encrypt +new ch
viii. end for

SERVER SIDE: -

1. Read URL in encrypted format from mobile agent
2. Decrypt (URL)
3. Redirect URL

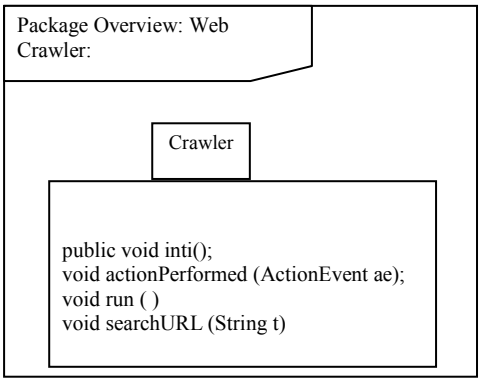
Input Parameter, String s, int key

i. ARR[10] ← 10 distinct value as in encrypt
ii. Increase element of ARR by key
iii. l← length (s)
iv for (i=0; i<=l; i++)
v ch ← char at (s, l)
vi new ch ← ch -ARR [i mod 9]
vii decrypt str <- decrypt str + new ch
viii end for

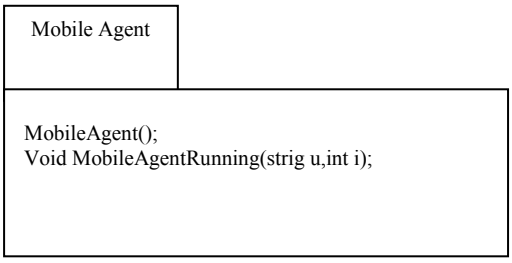
CLASS DIAGRAM

The implementation is mainly divided into 3 parts referred to as package: Crawler, Mobile Agent, and Encrypt-Decrypt.

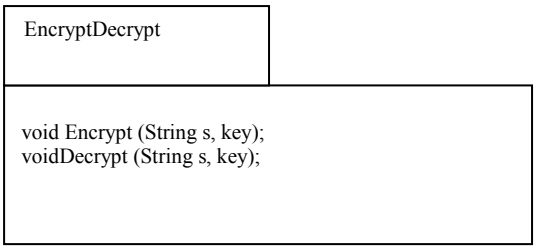
Crawler Class: Responsible to generate specific threads to search URLs from internet. It contains the class Crawler which is the main class of the system. It has mainly threading model implemented for crawler process. The crawler package is responsible for reading the URLs, preprocessing them and directing it to encryption layer.



Mobile agent class: Responsible to take encrypted urls from crawler to web server



EncryptDecrypt class: Responsible to encrypt the urls using private key at client side. and decrypt urls at server side.



IMPLEMENTATION

Implementation of Security based mobile agent crawler is being divided into two phases: -

CLIENT-SIDE
JAVA APPLET:

A program designed to be executed from within another application. Unlike an application, applets cannot be executed directly from the operating system. With the growing popularity of OLE (object linking and embedding), applets are becoming more prevalent. A well-designed applet can be invoked from many different applications. A client-side program is written in Java or JavaScript that downloads and executes on the end user's computer rather than executing on the server. Here, applet is used as user interface. When starting url is given, it then execute the function crawling() which start to collect urls. After collecting urls in database, thread() will execute and encrypted the urls and given to mobile agent.

SERVER SIDE

JAVA SERVLET:

An application program, written in the Java programming language, which is executed on a Web server. A reference to a servlet appears in the markup for a Web page, in the same way that a reference to a graphics file appears. The Web server executes the servlet and sends the results of the execution (if there are any) to the Web browser. It has http requesting () method that is executed when servlet is accessed. Decryption is done here and urls is undirected from there to web server.

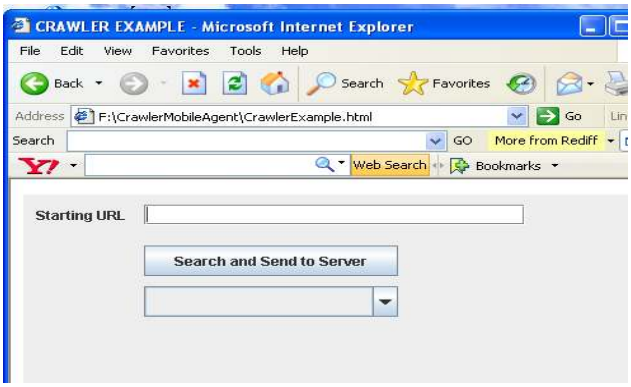
6. ADVANTAGES OF SECURITY ON MOBILE AGENT BASED CRAWLER

The advantage of this solution is that it protects the mobile agent and the data it carries. Though, the data is in the encrypted form, so no malicious host or agent can access the mobile agent code. It provides

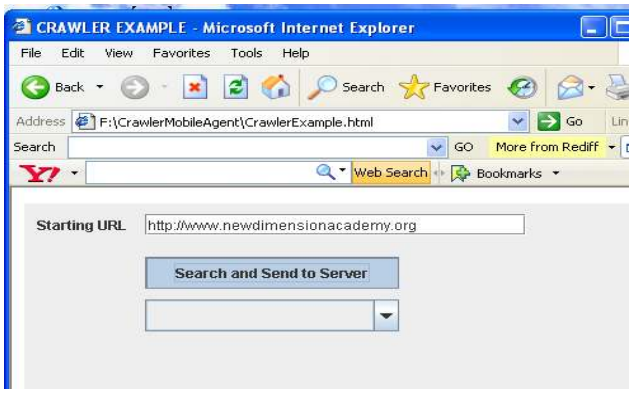
- Potentially better efficiency of the whole system. A client program migrates to a server node, directly communicates with a server program, and returns to an original node with a result. In that way, the number of remote interactions and the amount of data communicated over the network are reduced.
- Greater reliability, because the connection between nodes must not be established all the time.
- Robustness (only a short connection time needed to transfer agent).

7. SNAPSHOTS

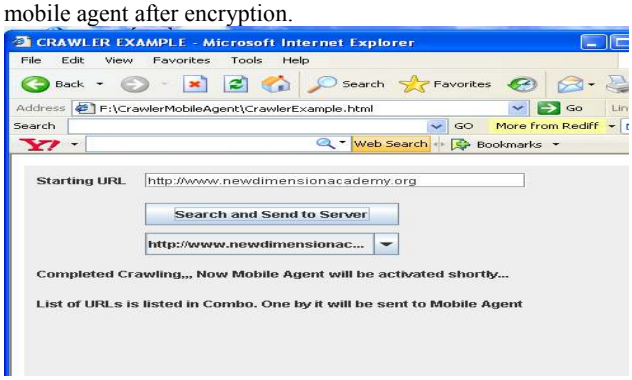
Snapshot 1: The User Interface



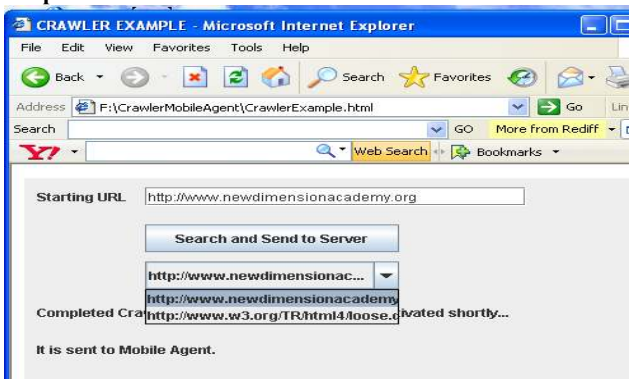
Snapshot 2: User Inputs the Starting URL and press the button "Search and Send to Server". Searching starts and takes little bit of time.



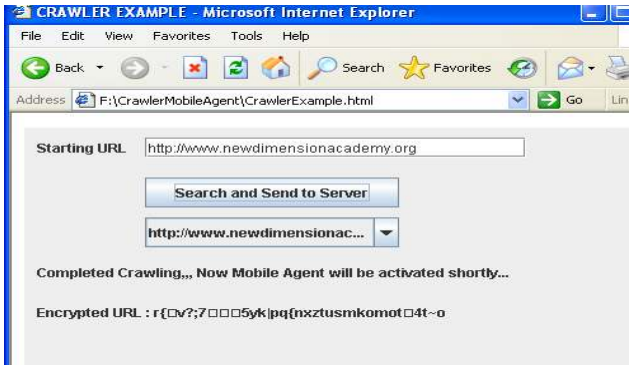
Snapshot 3: After some time, list of all crawled URL is added to the combo box and one by one the URL is transferred to the mobile agent after encryption.



Snapshot 4: Shows the list of URLs

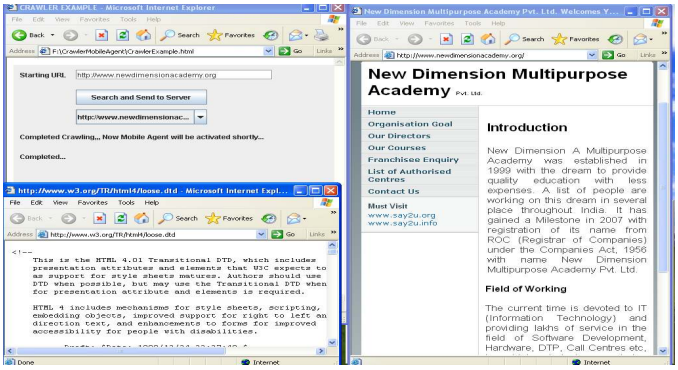


Snapshot 5: Shows encrypted form of URL that is sent to mobile agent.



Snapshot 6: End of the Crawling.

Snapshot 7: Shows all the URLs opened from server after decrypting it.



8. CONCLUSION

Mobile agents have gained a great deal of attention in research and industry in the recent past. Although mobile agents are a promising technology, the large-scale deployment of agents and the existence of hosts running agencies will not happen until proper security mechanisms are well understood and implemented. In this paper, security of mobile agents has been considered from the point of view of both the mobile agent and the agent platform and reconfirmed that it is much more difficult to ensure the security of mobile agents than the security of agent platforms. Proposed Architecture uses "computing with encrypted function method" to protect both the mobile agents themselves and the remote resources they access. Since the URL given to mobile agent is encrypted, so any host can do nothing with the mobile agent information and it will be securely transferred to web server and the web server can use the URL or such information after decrypting it. This solves the problem regarding any malicious host that can harm mobile agent or any information. This solution for mobile agent security is necessary on very important mobile agent applications which have a high security need or for servers where the data integrity is critical. The whole work has been done using Java Applet for User Interface and Servlet for web server. Future Research on this field is envisaged through the definition of new architectures using such mechanism and deployment of new application in which protection is provided on mobile agent by digital signature. Also, the exact specification of the trust algorithm in mathematical terms with analysis of security properties for specific types of security mechanisms and their effect on trust may be considered. Moreover, the mobile agents can be made

self-secure i.e. agents protect their code and data by carrying their own protection mechanisms.

9. REFERENCES

[1] Wayne Jansen and Tom Karygiannis, "Privilege Management Mobile Agents,Twenty-third National Information Systems Security Conference, pp.362-370, October 2006 .

[2] Giacomo Cabri, Letizia Leonardi, Franco Zambonelli, "The Impact of the Coordination Model in the Design ofMobile Agent Applications," Twenty-second Computer Software and Applications Conference (COMPSAC), August 1998.

[3] Huhns, M., Singh, M. 1999, "*Readings in AGENTS* , Morgan Kaufman Publishers, San Francisco, California.

[4] James Riordan and Bruce Schneier. Environmental key generation towards clueless agents. In G. Vigna,editor, *Mobile Agents and Security*, volume 1419 in LNCS, pages 15–24. Springer-Verlag, Berlin, 1998.

[5] C. Kr¨ugel and T. Toth. Sparta - a security policy reinforcement tool for large networks. In *submitted to I-NetSec 01*, 2001.

[6] Niklas Borselius, Namhyun Hur, Marek Kaprynski and Chris J. Mitchell. A security architecture for agent-based mobile systems. In Proceedings of the Third International Conference on Mobile Communications Technologies – 3G2002, London, UK, May 2002, IEE Conference Publication 489, pages 312–318, IEE, London, 2002.

[7] "Building Secure Mobile Agents: The Supervisor-Worker Framework," Diploma Thesis, Technical University of Vienna, Feb. 2004

[8] W. Jansen and T. Karygiannis, "Mobile Agent Security," NIST Special Publication 800-19, National Institute of Standard and Technology, 2000.

[9] Pahal, Nisha. Chauhan, N. Sharma, A.K. "Context-Ontology Driven Focused Crawling of Web Documents". In proceedings of Third International Conference on Wireless Communication and Sensor Networks, 13-15 Dec 2007 IEEE Conference Publication page: 121-124, IIIT Allahabad.

[10] Groot, D.R.A. De, Boonk, M.L., Brazier, F.M.T, Oskamp, A, " Issues in a Mobile Agent-based Multimedia
Retrieval Scenario" , Proc. The Third European Workshop on Multi-Agent Systems (EUMAS'05), pp. 103-113.

[11] Srilekha Mudumbai, Abdeliah Essiari, William Johnston, "Anchor Toolkit: A Secure Mobile Agent System,"
Proceedings of Mobile Agents '99 Conference, October 1999.

[12] William Farmer, Joshua Guttman, Vipin Swarup, "Security for Mobile Agents: Authentication and State Appraisal," Proceedings of the Fourth European Symposium on Research in Computer Security (ESORICS '96), September 1996, pp. 118-130.

[13] K.Satya Sai Prakash, S. V. Raghavan, "DIAPANGSE :Distributed Intelligent Agent based Parallel Architecture for Next Generation Search Engines", Dept. of Computer Science & Engineering, Indian Institute of Technology Madras, India, 2001.

[14] F. Gasparetti, A. Micarelli, "Swarm Intelligence: Agents for Adaptive Web Search", Dept. of Information, University of ROMA TRE, Rome, Italy, 2000.

[15] Potok, T., Elmore, M., Reed, J., and Samatova, N. 2002, "An Ontology-based HTML to XML Conversion Using Intelligent Agents", Proceedings of the Hawaii International Conference On System Sciences.