A Review on Mobile Agent Security

Parul Ahuja, Vivek Sharma

Abstract:- Mobile agents are enjoying a lot of popularity and are destined to influence research in distributed systems for the years to come. Thus far, technology has been instrumental in disseminating new design paradigms where application components are not permanently bound to the hosts where they execute. Mobile agents are gaining in complexity as they evolve and are now widely used in e-commerce. All phases of a business transaction, such as negotiating and signing contracts can be done using mobile agents. In this paper, we provided a brief introduction to the recent researches & developments associated with the field of mobile agents, highlighting various security threats, also touching the weakest hot-spots of the field which need to be nurtured.

Keywords:- Intelligent agent, Mobility, Security, Security Threats.

I. INTRODUCTION

The field of agents has many diverse researchers, approaches and ideas, which has helped to create one of the more dynamic research areas in recent years. The term software agent [1] is widely used in research fraternity and has received widespread acceptance in a number of emerging technologies. Agents are the independent software modules capable of acting autonomously based on the perception of their environment. Agents are an interesting topic of study because of their utility in diverse disciplines of computer science including personal information management, artificial intelligence, electronic commerce, interface design, computer games, distributed processing, distributed algorithms, and many more[1, 2].

Software agents are probably the fastest growing area of Information Technology (IT). Till yet, there is no commonly agreed definition of software agents. Different authors have different views regarding these. A few of the popular definitions are: “a persistent software entity dedicated to a specific purpose” [3], “computer programs that simulate a human relationship by doing something that another person could do for you” [4], “a software entity to which tasks can be delegated” [5]. Based on the above statements, scholars gave a summarized statement for software agents:-"an autonomous software agent is a component that interacts with its environment and with other agents on a user's behalf." An agent exhibiting the feature of mobility (logical mobility/code mobility) is usually termed as a mobile agent. The mobile agent initiates its itinerary (route plan for deciding where to move and when to move) by executing “jump” or “go” instruction which takes as argument, the name (identity) or the address of the destination. The next instruction in the agent’s program is executed on the destination machine. This fact states that a mobile agent is not bound to the system where it began its execution. In other words, it has the unique ability to roam freely in the network by transporting itself from one system to another autonomously. There are various differences among the existing mobile agent systems in the context that what they are allowed to move, and how it is actually moved. One of the distinction can be drawn on the basis of execution state. The systems supporting strong mobility [6] allows the migration of both the code and the execution state along with the executing unit to a different computation environment. On the other hand, the systems supporting weak mobility [6], allows the code migration only. The code contains only some initialization data and execution state is not migrated.

The paper is structured as follows. Section 2 briefly describes the characteristics of agents and multi-agent systems. In section 3 we will discuss some security threats and section 4 describes some recent research aimed at enhancing the security of mobile agent systems.

II. CHARACTERISTICS OF AGENTS

The invasion of various approaches under the banner of ‘agents’ caused a need to classify and define this term. However it quickly became apparent that everyone had their own definition [7] due in part to the historical relationship with the AI community and the vague notion of intelligence. Numerous definitions for the agents have been proposed, but in core most have a set of defining characteristics that every agent must demonstrate. For instance, Woodridge and Jennings’ weak agents [8] should be autonomous, reactive and social. A definition from Franklin & Graesser [9] lists autonomous, reactive, communicative, adaptive, mobile, flexible, goal-oriented, continuous and with some form of character or emotion.

- ***Autonomous*** - An agent should be able to execute without the need for human interaction, although intermittent interaction may be required.
- ***Social / Communicative*** - An agent should have a high level of communication with other agents. The most common protocol for agent communication is the Knowledge Query and Manipulation Language (KQML) [10].
- ***Reactive / Responsive*** - An agent should be able to perceive its environment and react to changes in it.
- ***Proactive*** - Proactive agents do not just react to their environment but can take active steps to change that environment according to their own desires.
- ***Adaptive*** - Adaptive agents have the ability to adjust their behaviour over time in response to internal knowledge or changes in the environment around them.
- ***Goal-oriented / Intentions*** - These agents have an explicit internal plan of action to accomplish a goal or set of objectives.
- ***Persistence / Continuous*** - Persistent agents have an internal state that remains consistent over time.
- ***Mobility*** - Mobile agents can proactively decide to migrate to a different

Manuscript Received on May 24, 2012.
Parul Ahuja, Scholar, CSE, JMIT, Radaur, Haryana, India.
Vivek Sharma, Assistant Professor & HOD CSE, JMIT, Radaur, Haryana, India.

machine or network while maintaining persistence.

- ***Emotion*** - Agents with the ability to express human-like emotion or mood. Such agents might also have some form of anthropomorphic character or appearance.
- ***Intelligence*** - Agents with the ability to reason, learn and adapt over time.
- ***Honesty*** - Agents that believe in the truthful nature of the information they pass on.

These agent characteristics lead to many advantageous features. The very nature of agents as independent, social entities that can respond to and change their environment provides a strong foundation for building reliable, robust, flexible, extensible and scalable systems. Agents can help ease user tasks and adapt to user requirements. Despite of their many benefits, agents are not the solution to every problem. One major disadvantage of building agent systems is that the complexity of agent interactions and dynamic nature of the agents themselves make it difficult to predict agent behaviour. It can cause problems in safety critical environments where outcomes need to be assured. However as the computer world is becoming increasingly networked and distributed, agents are likely to become the next engineering paradigm for system development.

III. SECURITY THREATS

This computing paradigm, which exploits code, data and state mobility, raises many new security issues that are quite different from conventional client/server systems. Agent servers that provide an execution environment for the agents to execute can be attacked by malicious agents. Similarly agents could be carrying sensitive information about their owners and should be protected from tampering by malicious hosts. The data collected by the agent from one host should also be protected from tampering by another host in the itinerary. In this section, various security issues that arise in mobile agents' applications are examined.

A. *Agent-To-Platform*

The agent-to-platform category represents the set of threats in which agents exploit security weaknesses of an agent platform and launch attacks to an agent platform. This set of threats includes masquerading, denial of service and unauthorized access [16], [18].

- Masquerading

When an unauthorized agent claims the identity of another agent it is said to be masquerading. The masquerading agent may pose as an authorized agent in an effort to gain access to services and resources to which it is not entitled. The masquerading agent may also pose as another unauthorized agent in an effort to shift the blame for any actions for which it does not want to be held accountable.

- Denial of Service

Mobile agents can launch denial of service attacks by consuming an excessive amount of the agent platform's computing resources. These denial of service attacks can be launched intentionally by running attack scripts to exploit system vulnerabilities, or unintentionally through programming errors. A rogue agent may carry malicious code that is designed to disrupt the services offered by the agent platform, degrade the performance of the platform, or extract information for which it has no authorization to access. Depending on the level of access, the agent may be able to completely shut down or terminate the agent platform.

- Unauthorized access

An agent that has access to a platform and its services without having the proper authorization can harm other agents and the platform itself. A platform that hosts agents representing various users and organizations must ensure that agents do not have read or write access to data for which they have no authorization, including access to residual data that may be stored in a cache or other temporary storage. Applying the proper access control mechanisms requires the platform or agent to first authenticate a mobile agent's identity before it is instantiated on the platform.

B. *Agent-To-Agent*

The agent-to-agent category represents the set of threats in which agents exploit security weaknesses of other agents or launch attacks against other agents. This set of threats includes masquerading, unauthorized access, denial of service and repudiation [19], [16]. Many agent platform components are also agents themselves.

- Masquerade

Agent-to-agent communication can take place directly between two agents. An agent may attempt to disguise its identity in an effort to deceive the agent with which it is communicating. For example, an agent may pose as a well-known vendor of goods and services and try to convince another unsuspecting agent to provide it with credit card numbers, bank account information, some form of digital cash, or other private information. Masquerading as another agent harms both the agent that is being deceived and the agent whose identity has been assumed, especially in agent societies where reputation is valued and used as a means to establish trust.

- Repudiation

Repudiation occurs when an agent, participating in a transaction or communication, later claims that the transaction or communication never took place. Repudiation can lead to serious disputes that may not be easily resolved unless the proper countermeasures are in place. An agent platform cannot prevent an agent from repudiating a transaction, but platforms can ensure the availability of sufficiently strong evidence to support the resolution of disagreements.

Documents are occasionally forged, documents are often lost, created by someone without authorization, or modified without being properly reviewed. Since an agent may repudiate a transaction as the result of a misunderstanding, it is important that the agents and agent platforms involved in the transaction maintain records to help resolve any dispute.

- Unauthorized access

An agent can directly interfere with another agent by invoking its public methods (e.g., attempt buffer overflow, reset to initial state, etc.), or by accessing and modifying the agent's data or code. Modification of an agent's code is a particularly insidious form of attack, since it can radically change the agent's behaviour (e.g., turning a trusted agent into malicious one). An agent may also gain information about other agents' activities by using platform services to

eavesdrop on their communications.

C. Platform-to-Agent

The platform-to-agent category represents the set of threats in which platforms are not trusted to which an agent is migrated. This set of threats includes masquerading, denial of service, eavesdropping, and alteration [16], [18].

- Masquerade

One agent platform can masquerade as another platform in an effort to deceive a mobile agent as to its true destination. An agent platform masquerading as a trusted third party may be able to lure unsuspecting agents to the platform and extract sensitive information from these agents. The masquerading platform can harm both the visiting agent and the platform whose identity it has assumed. An agent that masquerades as another agent can harm other agents only through the messages they exchange and the actions they take as a result of these messages [20], [21].

- Eavesdropping

The classical eavesdropping threat involves the interception and monitoring of secret communications. However, the threat of eavesdropping, is further extended in mobile agent systems because the agent platform can not only monitor communications, but also can monitor every instruction executed by the agent. The platform can also monitor all the unencrypted or public data, an agent brings to the platform, and all the subsequent data generated on the platform.

- Alteration

This threat involves when an agent suffers from unavailability of the integrity of its data on the new agent platform to which it is migrated. An untrusted platform can modify the contents of the information carried by the agent. When an agent arrives at an agent platform it is exposing its code, state, and data to the platform. A malicious platform must be prevented from modifying an agent's code, state, or data without being detected. Detecting malicious changes to an agent's state during its execution or the data an agent has produced while visiting the compromised platform does not yet have a general solution. For example, the agent platform may be running a modified virtual machine, without the agent's knowledge, and the modified virtual machine may produce erroneous results.

IV. LITERATURE SURVEY

Picco in [6] explored the related research fields by showing evidence of the benefits mobile agents can potentially achieve, illustrated the foundations of architectures and technologies for mobile agents, and discussed some of the open issues still hampering a wider acceptance of this paradigm. The author took the concept of conventional distributed systems' environment and compares the configuration of the system in context of physical mobility & logical mobility. The goal of the work was to introduce the reader to the research field concerned with mobile agents. The paper presented the conceptual foundations that have their grounds in logical mobility at large, and provided the state of the art in an agent technology. Paper also presented the rationale for using mobile agents, hints at why and when mobile agents are preferable over other solutions, reviewed the basic architectural paradigms for code mobility, including mobile agents and lastly presented reflections on the present status and the future scope of the research area. This work studied two case studies, one is mobile agents for database access and secondly, mobile agents for network management were discussed to research in this field. Advantages of agents have also been highlighted. A distinction is also drawn based on whether the execution state is migrated along with the execution unit or not. Strong mobility & weak mobility has been supported by the systems in response to this distinction.

Knoll, Suri, & Bradshaw in [28] introduced a path-based security for mobile agents. The path-based security provided a mechanism that extended the security of the NOMADS mobile agent system in a multiple-hops scenario. NOMADS supported strong mobility and safe agent execution. Strong mobility allowed the execution state of an agent to be captured and moved with the agent from one host to another [6]. A lightweight protocol for tracking agent paths had been developed that was based on chaining IP addresses. A receiving host environment computed a trust level for the agent, which was then used to choose and apply a security policy to the incoming agent. A TTP (Trusted Third Party) was optionally supported to provide more reliable path information. However, this implementation used just three levels of trust: high, medium, and low. So, the system did not appeal much to support a finer granularity of trust levels. Also, the proposed implementation automatically assigned a low level of trust to hosts that were unknown. Hence, a mechanism must be explored that allows the trust levels to be derived through transitive trust relationships. Also, another mechanism must be incorporated, that dynamically vary the trust levels of hosts based on past history information regarding their behaviour.

Gavalas, Tsekouras, & Anagnostopoulos in [11] proposed a mobile agent technology for the management of networks and distributed systems as an answer to the scalability problems of the centralized paradigm. The authors considered the design and implementation of a complete MAP research prototype that sufficiently addressed the issues such as security mechanism, fault tolerance. MAP has been implemented in Java and optimized for network and systems management applications. They introduced the design decisions and implementation aspects of a complete MAP research prototype that sufficiently addresses all the aforementioned concerns. In comparison to existing approaches, this MAP consolidated several novel design features, dictated by their design choices and reflected upon their research implementation: (a) a lightweight code distribution scheme, (b) a class loading mechanism that allows the modification of MA-based NSM tasks at runtime, (c) a tool that supports the user-friendly customization of service-oriented MAs, (d) a component that builds near-optimal network-dependent MA itineraries. In addition, it satisfied other important NSM-related requirements, such as lightweight footprint on systems resources, security (authorization, authentication and encryption), recovery from basic software faults, modularity, incorporation of agent migration optimization

techniques, platform-independence, etc. MAP's performance has been evaluated in realistic management application scenarios.

Gavalas et. al. also presented the evaluation results of prototype in real and simulated networking environments. Besides all this, there raised some issues which must be fixed such as extensive testing and practical evaluation of MAP in real-world monitoring applications by research engineers in order to identify potential deficiencies and derive methods for improving the MAP's performance; extension of the manager platform with the implementation of a web interface that will allow human administrators to view updated management statistics through a typical web browser and remotely control their network. Future research may compare the performance in terms of latency and network overhead between this proposed MAP, general-purpose MAPs and NSM-oriented MAP (when they become publicly available).

Xiaorong, Su, & Mingxuan in [13] introduced the mobile agent technology based on quantitative hierarchical network security situational assessment model. The researchers designed the distributed computing for large-scale network and evaluated the whole network security situation for future prediction. Network security situational assessment quantitative model is a Hierarchical network consisting of Quantification of security situational index. This index has further different levels: Service-level security situational index, Host-level security situational index, Network system-level security situational index. In this model, the various tasks have been accomplished by an agent: The whole network(system-level) agent network security situation assessment task; the subnet(host-level) agent network security situation assessment task; and the device(service-level) agent network security situation assessment task. As the security situational assessment is widely applied to the computer network field, many scholars have designed and implemented a large number of network security situational assessment methods. Network security situational assessment technology took all aspects of security factors into account, reflects the overall dynamic state of network security and predicts the development trend of state, which provides a reliable frame of reference. Therefore, network security situational assessment has become a hotspot field in next generation network security technology. The security model proposed by Xiaorong et. al. is distinct and refined method as compared to other models since most works are based on local area network and single host, which hardly meet the demand of large-scale network security assessment. But the technical realization of the quantitative model for further prediction had not been discussed which degrades the quality aspects of the model.

Moussa & Agha in [14] presented the design of "Bosthan", a multi-agent-based simulation tool that managed resources consumption in multi-inhabitants smart spaces. Bosthan had been built on the top of ActorNet mobile agent platform to simulate different smart space topologies with varying numbers of residents. It allowed strategies for resolution of conflicts between mobile agents, and for preserving inhabitants' anonymity and untraceability inside the smart spaces. The proposed architecture for bosthan agent-based simulator consists of ten modules: bosthan simulator engine, motion patterns generator, floor planner, sensors layout creator, events generator, location predictor, identity hider, actornet generator, conflict resolver, runs analyzer.

Bosthon was designed in the hope that it would help to compare the efficiency of using mobile agents to allow smart spaces to act pro-actively and maintain anonymity and data privacy in multi-inhabitants environments. Bosthan had been used to study how proposed solution affects the performance and efficiency of smart computing environments without revealing their identity. Bosthan has been designed as a tool for agent-based discrete event simulations that would enable the analysis of different smart space scenarios, ranging from small environments such as smart rooms, to large smart spaces such as smart homes, smart offices or even smart buildings. The authors used Bosthan to study how mobile agents can be used to provide a resource consumption management framework that preserves inhabitants' anonymity and untraceability rules through applying non-interactive evaluation of encrypted functions, without inhabitants' intervention, in multiple intersected contexts within a developed smart space. Their goal was to use Bosthan to investigate the effectiveness of strategies to resolve conflicts that may arise between competing agents due to these contexts intersections. This could be applied to simulate environments where residents' privacy is highly required. But all these developments are waiting to be implemented, as bosthon is still under under development in the open system laboratory (OSL), University of Illinois at Urbana-Champaign (UIUC), using Microsoft Visual Studio C# 2005.

Rizvi, Sultana, Sun & Islam in [19] provided a solution for securing mobile agent in an ad hoc network. The paper provided a solution for securing mobile agent in an ad hoc network. The authors used Threshold Cryptography in their model, because it provides solution to the problem of central certificate authority (CA) and trusted third party in PKI, by distributing trust among several network nodes. The model provides prime security services like confidentiality, integrity and authenticity. But mobile agent is not free from threats. Although, the model provides a way to secure not only mobile agent, but also the agent server and the agent platform, still it is tough to provide 100% security in an ad hoc network, to detect and prevent vulnerabilities and intrusions. According to threshold cryptography they have considered the value t as a threshold value for the ad hoc network. It means the system can tolerate up to t compromised servers. However in this paper, they have not discussed about the cryptographic schema used to generate shares of the Key Management Service's private key. Also they have not discussed about the process of combining the partial signatures or the process to generate the private key of Key Management Service by t+1 servers. Besides these, how authorization and access control service will be provided has not been mentioned. Future research is needed to solve these issues.

Singh, Juneja & Sharma in [12] explained about the working of agent community that it works on the core idea of cooperation and delegation of tasks, which in turn should be prevented from any malicious usage. In order to avoid this malicious usage, an instrument for ensuring proficient and secure communication among these collaborating agents is trust. The authors proposed an elliptical curve cryptography based security engine which

extends a novel architecture namely CNTEP which successfully established trust among agents. Encryption of mobile agents and communicated messages is one of the solutions for ensuring security. However traditional encryption algorithm such as DH,DSA [22] and RSA [23], employ key sizes which are very large resulting in high time and space complexity. In contrast to this Elliptical Curve Cryptography (ECC) technique [25, 26] is a public key cryptosystem that besides using much smaller key sizes is able to provide a competitive security edge as that of other strong encryption algorithms. Most attractive feature of ECC is its relatively short operand length compared to that of RSA and also it is based on discrete logarithm in finite fields.

ECC can provide various security services in the form of key exchange, communication privacy through encryption, authentication of sender and digital signatures to ensure message integrity. This paper was an extension of Trust Establishment Protocol (proposed in their earlier paper) and hence an improved version of Contract Net Protocol which was termed as CNTEP [24]. A brief overview of CNTEP architecture as well as elliptical curve cryptosystem was also provided. The authors emphasize that security and trust are two sides of one coin and they must be considered in a way so as to provide a complete security solution for MASs. An attempt to provide such a solution has been made in their proposed model.

The singh et. al. focused on providing both trust among the communicating agents as the first step and after that ensuring the confidentiality and integrity of the messages communicated. Trust Establishment layer (TEL) and Secure communication layer (SCL) were the two components in the model developed. Trust matrix has been maintained to accomplish the process of trust establishment layer. This paper proposed an ECC based security engine for MASs, which provides two dimensional security. In addition to computing trust percentile, the proposed framework makes use of elliptical curve keys for encryption/decryption purpose, which increases message security to a larger extent. Thus this work adds greatly towards improvement of communication security of Multi-Agent Systems. Major advantage of this framework is that existing application will not have to be rewritten to utilize it, instead it can be implemented in the security layer of existing model of wireless communication. The focus of this paper was to propose a security engine for ensuring security of messages communicated in semantic cyberspace. Though, this work adds greatly towards improvement of communication security of Multi- Agent Systems, but the authors remained silent towards the security of agent hosting platforms. So, the need of the hour requires further research in the field of agent technology. Roth & Jalali-Sohi in [27] presented a mobile agent structure which supports authentication, security management and access control for mobile agents. They presented a flexible and extensible structure for the representation of mobile agents which supports hierarchical access control, and proposed an initial interpretation of this structure with respect to the roles in a general mobile agent model. The structure maps well on existing and widely distributed technology. It also facilitated security management and the implementation of security services for both while supporting basic agent tasks and operations – the management of data. A portable and interoperable agent structure which is acceptable to all major agent facilities must be established for improvement purposes. This could be achieved by probably feeding the results to a standardization process. The structure as well as the interpretation should undergo a refinement process while being explored within a security centered mobile agent framework.

Karnouskos in [15] considered the concept of active networks. Active Networks (AN) are a rapid evolving area of research and in parallel an area of great industry interest. However, for this technology to make the step out of the labs and penetrate the market, the security problems have to be tackled effectively. This paper demonstrated why and how agent technology research, can and should be applied to active networks, in order to fulfil the new security challenges this infrastructure poses. First, the researchers identified the key elements of AN, analyzed the nature of active code, specified the role of agents in active networks and presented a multi execution environment active network architecture. Then, the security threats for active code and execution environment were targeted, and the basic as well as the extended security requirements were also stated. Subsequently, the author tried to see how the security solutions can be applied and research can be done for agents to the context of active networks in order to satisfy their requirements.

Threats encountered in an active network infrastructure were also discussed. Misuse of execution environment by the active code can exploit security weakness/attack in the host such as masquerading, denial of service, unauthorized access, and complex attacks. Misuse of AC by other AC involves a set of threats that are very critical because of the variety of victims, including repudiation, masquerading, denial of service, and unauthorized access. Misuse of active code by the execution environment can perform attacks such as masquerading, denial of service, eavesdropping, data and state manipulation, and cloning. Misuse of AC and/or EE by the underlying network infrastructure (external misuse) could perform all kinds of attacks such as masquerade, denial of service, unauthorized access, copy and replay, alteration, etc.

On considering the future scope, Karnouskos et. al. had presented an approach that tried to combine two domains, that of AN and that of agent technology. Both domains shared common ground, especially when it comes to the security issues. The AC that executes within the AN nodes can be seen in its most advanced form as an intelligent mobile agent. Therefore, most of the security techniques developed from the agent technology could be developed and tackled successfully similar issues in ANs. The nature of AC had been analyzed by the researchers and presented the key elements within a multi-EE AN node architecture. Then the author tried to focus on the security threats that existed in such a homogeneous environment in AC to AC, AC to EE and EE to AC relations. In all cases, most of the common security attacks could be accomplished. Subsequently, the most obvious requirements of AN was set and the agent-based solutions that effectively dealt with the threats identified above was explored. The multi-EE AN architecture provided security

within a particular limited domains of mobility. So, it was not proven to be dynamically accepted for all generations. Hence, the architecture should be improved to reinvent the wheel of technology every time in every approach because agent technology advances continuously and had made significant contributions in the area of code mobility and security. Also, the complex security attacks were not resolved and could be tackled by building more sophisticated approaches by integrating already tested solutions in other domains.

V. CONCLUSION

Agent technology has been used in many critical applications such as personal information management, electronic commerce, business process management, artificial intelligence, interface design, distributed processing and distributed algorithms. Besides its bright side, the technology has encountered many security threats. These problems are faced during the itinerary period of an agent traversing from platform to platform in the network. In this paper, we have surveyed various recent developments, researches and proposals related to the field of agents and have thrown some light on the delicate areas that needs to be paid more attention to promote growth in optimistic direction.

REFERENCES

[1] N. Jennings and M. Wooldridge, "Software Agents", IEE Review, January 1996, pp. 17-20.

[2] O.A. Ojesanmi and A. Crowther, "Security Issues in Mobile Agents", International Journal of Agent Technologies and Systems, 2(4), pp. 39-55, October-December 2010, University, Nigeria.

[3] D. C. Smith, A. Cypher and J. Spohrer (1994) "Programming Agents without a programming language" Communications of the ACM 37 (7) pp 55-67.

[4] P. C. Janca (1995) "Pragmatic Application of Information Agents: BIS Strategic Decisions.

[5] T. Selker (1994) "A Teaching Agent that learns" Communications of the ACM 37 (7) pp 92-99. D. C.

[6] G.P. Picco, "Mobile agents: an introduction", Microprocessors and Microsystems 25(2001) pp. 65-74, Dipartimento di Elettronica e Informazione, Politecnico di Milano, Milan, Italy.

[7] H.S. Nwana, "Software Agents: An Overview", Knowledge Engineering Review, 11(3):1- 40, 1996.

[8] M. Woodridge and N. Jennings, "Intelligent Agents: Theory and Practice", The Knowledge Engineering Review, 10(2):114-152, June 1995.

[9] S. Franklin, A. Graesser, "Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents", University of Memphis, Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, Springer-Verlag,1996.

[10]T. Finin, Y. Labrou & J. Mayfield, "KQML as an Agent Communication Language", J. Bradshaw (Eds), MIT Press, 291-316, 1997.

[11] D. Gavalas, G.E. Tsekouras, C. Anagnostopoulos, "A mobile agent platform for distributed network and systems management", In Journal of Systems and Software 82 (2), 355-371, 2009.

[12] A. Singh, D. Juneja, and A.K. Sharma, "Elliptical Curve Cryptography Based Security Engine for Multiagent Systems Operating in Semantic Cyberspace", In International Journal of Research and Review in Computer Science (IJRRCS), Vol. 2, No. 2, April 2011.

[13] C. Xiaorong, L. Su, L. Mingxuan, "Research of Network Security Situational Assessment Quantization Based on Mobile Agent", Volume 25, 2012, Pages 1701–1707, International Conference on Solid State Devices and Materials Science, April 1-2, 2012, Macao.

[14] S. M.. Moussa, G.A. Agha, "Integrating Encrypted Mobile Agents with Smart Spaces in a Multi-agent Simulator for Resource Management", Journal of Software, Vol 5, No 6 (2010), 630-636, Jun 2010.

[15] S. Karnouskos, "Security implications of implementing active network infrastructures using agent technology", Special Issue on Active Networks and Services, In Computer Networks Journal, Elsevier, Vol. 36, Issue 1, pp. 87-100, June 2001.

[16] W. Jansen and T. Karygiannis, "Mobile Agent Security", Nist Special Publication 800-19-, 2000. National Institute of Standards and Technology.

[17] N. Borselius, "Mobile agent security", Electronics & Communication Engineering Journal, IEEE London 14(5), 211-218(October 2002).

[18] W.A. Jansen, "Countermeasures for Mobile Agent Security", National Institute of Standards and Technology, Gaithersburg, MD 20899, USA, wjansen@nist.gov.

[19] S.M.S.I. Rizvi, Z. Sultana, B. Sun, and Md. W. Islam, "Security of Mobile Agent in Ad hoc Network using Threshold Cryptography", World Academy of Science, Engineering and Technology 70- 2010.

[20] P. Dadhich, Dr. K. Dutta, and Prof.(Dr.) M.C. Govil, "Security Issues in Mobile Agents", International Journal of Computer Applications(0975-8887), Volume 11-No.4, December 2010.

[21] D.M. Chess, " Security issues in mobile code systems. In : mobile agents and security", Editor Vigna, vol. LNCS1419. Springer-Verlag 1998.

[22] K. Lauter, "The Advantages of Elliptic Curve Cryptography for Wireless Security", In IEEE Wireless Communications, pp. 62-67. February 2004.

[23] R. Shanmugalakshmi and M. Prabu, "Research Issues on Elliptic Curve Cryptography and its applications", In International Journal of Computer Science and Network Security, Vol. 9, No.6, pp 19- 22, June 2009.

[24] A. Singh, D. Juneja, A.K. Sharma, "Introducing Trust Establishment Protocol in Contract Net Protocol". In Proceedings of IEEE International Conference on Advances in Computer Engineering (ACE'2010), pp. 59-63, June, 2010.

[25] N. Koblitz, "Elliptic Curve Cryptosystems", Mathematics of Computation, Vol. 48, pp. 203-209, 1987.

[26] V.S. Miller, "Use of Elliptic Curves in Cryptography", Advances in Cryptology- CRYPTO'85, LNCS, vol. 218, Springer-Verlag, pp. 417-426,1986.

[27] V. Roth & M. Jalali-Sohi, " Access Control and Key Management for Mobile Agents", Fraunhofer Institute for Computer Graphics, Rundeturmstr. 6, 64283 Darmstadt, Germany, 8 November, 2001.

[28] G. Knoll, N. Suri, and J.M. Bradshaw, "Path-based Security for Mobile Agents", Electronic Notes in Theoretical Computer Science, Vol. 58, No. 2 , pp. 16, (2002).

AUTHOR PROFILE

Parul Ahuja has done B.Tech (CSE) in Honours and is pursuing M.Tech (CSE) Dissertation from Kurukshetra University. She has published a paper proposing a Robust Algorithm for securing an agent hosting platform. Her research work is in the field of Agent Technology. She has delivered seminars on various technologies such as mobile IP, cloud computing, Android technology.

Vivek Sharma has done B.Tech (CSE) & M.Tech (CSE) from Kurukshetra University and achieved gold medallist during his graduation. His publications includes the field of wi-fi protocols, mobile sensor networks in health care. His specialization is in object oriented programming languages. His research area is in mobile networks. He is a member of CSI (Computer Society of India).