RESEARCH ARTICLE OPEN ACCESS

Dynamic Programming He and Elgamal Used For Software Agent Security

Prof. Sachin Upadhye[1], Dr. P.G. Khot[2]
Department of Computer Application[1],
Shri Ramdeobaba College of Engineering and Management
Gittikhadan, Katol Road,
Nagpur-India
P.G.T.D.[2], Department of Statistics,
RTM Nagpur University, Amrawati Road,
Nagpur-India

ABSTRACT

In the last few years, software agents are gaining great attention as a new concept for developing and implementing mobile as well as distributed applications. In this paper we will present the state of the art of securing software agents against malicious hosts and an approach that we suggest. In such state of the art, identify dynamic programming homomorphic encryption and elgamal for security of software agent. The paper is structured as follows. Section 1 describe to Software agent section 2 briefly describe security threats. Section 3 Related works about Security Services. In section 4 discuss Dynamic Programming Homomorphic Encryption and in last section conclude the paper.

Keywords:- software agent security, homomorphic encryption , elgamal public scheme encryption, dynamic programming

I. INTRODUCTION

A software agent is a program that migrates from node to node of a heterogeneous network. Agents may be stationary, always resident at a single platform or mobile, capable of moving among different platforms at different time that agent called as mobile agent. Agents are goal-oriented i.e. work autonomously towards a goal, capable of suspending their execution on one platform and moving to other where they can resume execution using resources of these nodes and they meet and interact with other agents. Researchers, so far, have not been able to agree upon a common definition for software agents. However, different researchers have defined agent according to their own point of view. "Autonomous agents are computational systems that inhabit some complex dynamic environment; sense and act autonomously in this environment and by doing so realize set of goals or task for which they are designed" A software agent is a program that can exercise some work autonomously towards a goal. These may be either stationary or mobile. The former remain resident at a single platform, while latter are capable of suspending activity on one platform and moving to another and resume execution. An agent-based computer system is a distributed computing environment in which mobile autonomous processes called mobile agents operate on behalf of users. The autonomous agent concept has been proposed for a variety of applications on large, heterogeneous, distributed systems (e.g., the Internet). These applications include a specialized search of a large free-text database, middleware services such as an active mail system, electronic malls for shopping, and updated networking devices. Software agent systems are purported to have many advantages over traditional distributed computing environments. They require less network bandwidth, increase asynchrony among clients and servers, dynamically update server interfaces and introduce concurrency. Software agents are autonomous software entities, which can migrate through a network of heterogeneous sites to perform tasks on behalf of their owners. The main difficulty stems from the definition and realization method of software agents security [1, 2].

II. SECURITY THREATS

Threats to security generally fall into three main classes: disclosure of information, denial of service, and corruption of information. There are a variety of ways to examine these classes of threats in greater detail as they apply to agent systems. A number of models exist for describing agent systems [3] however, for discussing security issues it is sufficient to use a very simple one, consisting of only two main components: the agent and the agent platform. Here, an agent is comprised of the code and state information needed to carry out some computation. Mobility allows an agent to move, or hop, among agent platforms. The agent platform provides the computational environment in which an agent operates. The platform from which an agent originates is referred to as the home platform, and normally is the most trusted environment for an agent. Software agents moving around the network are not safe. The Agent-to-Host, Agent-to-Agent, Host-to-Agent, Other-to-Agent Host attacks are the kinds of security attacks that are possible in a Mobile Agent System [4, 5].

III. RELATED WORK

[6] Explored 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. 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.

[7] Introduced a path-based security for mobile agents. 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. Also, another mechanism must be incorporated, that dynamically vary the trust levels of hosts based on past history information regarding their behaviour.

[8] Proposed a mobile agent technology for the management of network 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.

[10] 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.

[11] Provided a solution for securing mobile agent in an ad hoc network. The authors used Threshold Cryptograph 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.

[9] 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. A brief overview of CNTEP architecture as well as elliptical curve cryptosystem was also provided.

[12] Used obfuscation code that generating executable agents which cannot be attacked by reading or manipulating their code. Their technique is based on transforming the agent code in such a way that it is functionally identical to the original one, but it is impossible to understand it. The approach also establishes a time interval during which the agent and its sensitive data are valid. After this time elapses, any attempt to attack the agent becomes worthless. The major drawback of these techniques is the difficulty in establishing the time required by an attacker to understand an obfuscated code.

In [13], introduces the idea of cryptographic traces. The running agent takes traces of instructions that alter the agent's state due to external variables. The host sends a hash of the traces with the results because the complete traces are too large. If the agent owner wants to verify execution, it asks for the traces and executes the agent again. If new execution does not agree with the traces, the host is cheating. The approach not only detects attacks, but it also proves the malicious behavior of the host. This approach has various drawbacks: (1) Verification is only performed in case of suspicion; (2) Each host must store the traces for an indefinite period of time because the origin host can ask for them; and (3) a third trusted party is needed in order to punish malicious behaviors.

In [14], presents the idea of mutual protection. In an open environment like the Internet it can be assumed that trustworthy relationships are limited, so collusion between hosts is difficult. For this reason, the agent's results are saved in a cooperative agent that has a disjoint itinerary. This approach presents two drawbacks: (1) The loss of the cooperative agent implies the loss of the results; (2) The possibility of collusion does not disappear.

IV. DYNAMIC PROGRAMMING HOMOMORPHIC ENCRYPTION

Dynamic programming [15] was developed by R. Bellman during the late 1950's. Dynamic programming is a powerful method that can be applied to various combinatorial optimization problems. Many planning and control problems involve a sequence of decisions that are made over time. The initial decision is followed by a second, the second by a third, and so on. The process continues perhaps infinitely. Because the word dynamic describes situations that occur over time and programming is a synonym for planning, the original definition of dynamic programming was "planning over time." Dynamic programming has been described as the most general of the optimization approaches because conceivably it can solve the broadest class of problems. In many instances, this promise is unfulfilled because of the attending computational requirements. Certain problems, however, are particularly adaptable to the model structure and lend themselves to efficient computational procedures; in cases involving discontinuous functions or discrete variables, dynamic programming may be the only practical solution methodology. An example application of this protocol is the combinatorial auction, where multiple servers can solve a winner determination problem, i.e., they can find the combination of bids so that the sum of the bidding prices is maximized. Although the servers can compute the optimal solution correctly, the information of the bids that are not part of the optimal solution is kept secret even from the servers [16]. DP model represents a sequential decision process rather than an algebraic statement of a problem. The two principal components of the dynamic programming model are the states and decisions. A state is like a snapshot of the situation at some point in time. It describes the developments in sufficient detail so that alternative courses of action starting from the current state can be evaluated. A decision is an action that causes the state to change in some predefined way. Thus a decision causes a movement from one state to another. The state transition equations govern the movement. A sequential decision process starts in some initial state and advances forward, continuing until some final state is reached. The alternating sequence of states and decisions describes a path through the state space.

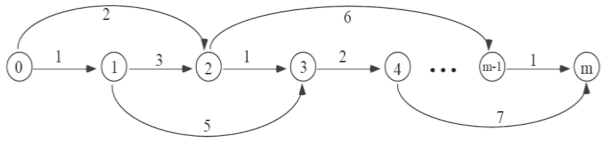

Figure 1: Example of one-dimensional directed graph (I)

The following example illustrates the dynamic programming on one dimensional directed graph. Here describe secure dynamic programming protocol based on this path-finding problem in a one-dimensional directed graph.

This graph consists of nodes 0, 1, 2, m with directed links among them. A link is represented as (j, k), where j < k. For each link (j, k), the weight of the link w (j, k) is defined. The goal is to find the longest path from initial node 0 to terminal node m, i.e., to find a path from 0 to m so that the sum of the weights of links are maximized. For simplicity, we assume for each node j (where 0 ≤ j < m), there exists at least one link that starts from j, i.e., there is no dead-end node except m. One notable characteristic of this problem is as follows. Assume P is the longest path from 0 to m. Then, for any node j which is on P, the last half of P, i.e., the part of P from j to m, is also a longest path from j to m. This characteristic is called the principle of optimality. This fea feature enables us to find the optimal solution of the original problem from the optimal solutions of sub-problems. More specifically, we can obtain the length of the longest path from 0 to m by solving the following recurrence formula from node m − 1 to 0.

f(j) =max(j,k){w(j, k) + f(k)}

In this formula, f(j) represents the length of the longest path from j to m. We call f(j) an evaluation value of node j. For terminal node m, f(m) is defined as 0. For initial node 0, f(0) represents the optimal solution, i.e., the length of the longest path from 0 to m. When calculating this formula, for each node j, we record the link (j, k) that gives the evaluation value f(j), i.e., the link that gives max(j,k){w(j, k) + f(k)}. We can construct the longest path by following these recorded links from 0 to m. generalizing this technique to other types of graphs, such as in two-dimensional multistage networks which are used for multi-unit auctions or in In a general combinatorial auction.

V. HOMOMORPHIC ENCRYPTION

Homomorphic encryption is a form of encryption which allows specific types of computations to be carried out on cipher text and obtain an encrypted result which decrypted matches the result of operations performed on the plaintext.

A public-key encryption scheme E = (KeyGen, Enc, Dec) is homomorphic if for all k and all (pk, sk) output from Key Gen(k), it is possible to define groups M, C so that: The plaintext space M, and all cipher texts output by Encpk are elements of C. For any m1, m2 ∈ M and c1, c2 ∈ C with m1 = Decsk (c1) and m2 = Decsk (c2) it holds that Decsk (c1 * c2) = m1 * m2 where the group operations * are carried out in C and M, respectively [17].

In other words, a homomorphic cryptosystem is a PKS with the additional property that there exists an efficient algorithm (Eval) to compute an encryption of the sum or/and the product, of two messages given the public key and the encryptions of the messages, but not the messages themselves.

The basic idea of our secure dynamic programming protocol is as follows:

• We assume there is a weight publisher P(j,k) for each link (j, k), and an evaluator Ti for each node i. In an auction setting, a weight publisher corresponds to a bidder, and an evaluator corresponds to a part of the multiple auction servers.

• These evaluators cooperatively execute dynamic programming. Evaluator Ti knows only its evaluation value f(i) and does not know any weight of any link.

For this an indistinguishable, homomorphic, and randomizable public key encryption scheme needed. ElGamal encryption [18], which has all of these properties,

Public key encryption: In public key encryption, the key used for encryption is public, so anybody can create ciphertext E(M) from plain text M. On the other hand, the key used for decryption is kept secret and only the one who has the secret key can obtain M from E(M)

ElGamal Cryptosystem: ElGamal encryption: ElGamal encryption is one instance of public key encryption. Let q, p = 2q + 1 be primes and G =< g >⊂ Z∗ p be a cyclic group of order q generated by g, where Zp denotes a set of integers from 0 to p−1 and Z∗ p denotes a set of integers that are in Zp and prime to p. The secret key is x ∈ Zq and the corresponding public key is g, y = gx. ElGamal encryption is based on the assumption of the hardness of discrete logarithm problem (DLP), i.e., to find x from (g, gx) is computationally infeasible. Anyone can encrypt message M ∈ G just using public key g, y = gx, i.e., choose random number r ∈ Zq and create ElGamal cipher text E(M) = (A = gr,B =yrM).One who knows secret key x ∈ Zq can decrypt cipher text E(M) = (A = gr,B = yrM), i.e., compute B/Ax = M.

Indistinguishable encryption: In ElGamal encryption, E(M) is created using random number r. Thus, if the same plaintext is encrypted twice using different random numbers, these two cipher texts looks totally different and we cannot know whether the original plaintexts are the same or not without decrypting them.

Homomorphic encryption: Encryption E is homomorphic if E(M1)E(M2) = E(M1M2) holds. If we define the product of cipher texts E(M1) = (A1,B1) and E(M2) = (A2,B2) by E(M1)E(M2) = (A1A2,B1B2), ElGamal encryption E is homomorphic encryption. By this property, we can take the

Product of two plaintexts by taking the product of two cipher texts without decrypting them.

Randomization: In ElGamal encryption, one can create a new randomized cipher text E (M) = (Ags,Bys) with random value s from the original cipher text E(M) = (A = gr,B = yrM). This is equivalent to making a product of E (1) = (gs, ys) and E (M). If we assume that the DDH problem is infeasible, one cannot determine whether a cipher text is a randomized cipher text of the original cipher text or not.

VI. CONCLUSION

This paper we are discussing the homomorphic encryption property and Elgamal cryptosystem are used to provide the security solution. As the property of Homomorphic we can encrypted add the data without decryption, so that we are achieving the data integrity, confidentiality. Here we are also discussing the required property of Elgamal homomorpic encryption cryptosystem. The area of software agent security is still in somewhat immature state. Both the agent and the agent platform should to be protected by developing techniques and mechanisms.

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.

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

[3] Sachin upadhye, p. g. khot, “Optimize Security solution for mobile agent security: A Review”, International Journal Of Engineering And Computer Science ISSN:2319-7242 Volume 2 Issue 1 Jan 2013 Page No. 322-329

[4] Wayne Jansen, Tom Karygiannis. NIST Special Publication 800-19 - Mobile Agent Security. National Institute of Standards and Technology, Computer Security Division, Gaithersburg, MD 20889. {jansen,karygiannis}@nist.gov.

[5] Fritz Hohl, Time Limited Blackbox Security: Protecting Mobile Agents from Malicious Hosts, in Vigna, Giovanni (Ed.): Mobile Agents and Security, Springer-verlag, 1998.

[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] 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). 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).

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] Tan H.K. and Morean L 2001, Trust Relationships in a Mobile Agent System, In Mobile Agent, Vol., 2240 of Lecture Notes in Computer Science, pp 15-30,Springer Verlag

[13] G. Vigna. Cryptographic traces for mobile agents. In Mobile Agents and Security, volume 1419 of LNCS,Springer-Verlag, 1998.

[14] V. Roth. Mutual protection of cooperating agents. In Secure Internet Programming: Security Issues for Mobile and Distributed Objects, volume 1906 of LNCS. Springer-Verlag, 1999.

[15] E. Rasmusen. *Games and Information*. Blackwell, 1994.

[16] Makoto Yokoo, Koutarou Suzuki. *Secure Multi-agent Dynamic programming based on Homomorphic encryption and its Application to Combinatorial Auctions.*

[17] S. Sobitha,” State Of Art in Homomorphic Encryption Schemes” International Journal of Engineering Research and Applications ISSN : 2248-9622, Vol. 4, Issue 2(Version 6), February 2014, pp.37-43

[18] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans. on Information Theory, IT-31(4):469–472,1985.