行政院國家科學委員會專題研究計畫　期中進度報告

適用於同儕移動服務網格內的確認與授權技術(1/3)

計畫類別：個別型計畫
計畫編號：NSC94-2213-E-009-121-
執行期間：94 年 08 月 01 日至 95 年 07 月 31 日
執行單位：國立交通大學資訊工程學系(所)

計畫主持人：邵家健

計畫參與人員：朱書玄、郭宇軒、龔哲正、吳書修、陳奕興

報告類型：精簡報告
報告附件：出席國際會議研究心得報告及發表論文
處理方式：本計畫可公開查詢

中　華　民　國 95 年 6 月 1 日

行政院國家科學委員會專題研究計畫期中報告

「光環」：適用於同儕移動服務網格內的確認與授權技術

Project Halo: Authentication and Authorization Support for Peer-To-Peer Mobile Service Grid

計畫編號：94-2213-E-009-121

執行期限：94 年 08 月 01 日至 97 年 07 月 31 日

主持人： 邵家健教授 (Prof. John Kar-Kin Zao)

執行機構：國立交通大學資訊工程系

Abstract

Keywords: Peer-to-Peer Overlay Networks, Usage/Access Control Policies, Reputation Management, Application Layer Multicast

Project Halo aims at exploring the *resource selection* and *usage control* issues pertained to persisting multimedia services offered over peer-to-peer overlay networks. In this report, we describe the work done in three areas during the first year of the project: (1) the provision of resilient application layer multicast (ALM) to the peers with intermittent and asymmetric connectivity such as the mobile nodes with wireless links, (2) the inference of peer performance or trustworthiness through a scaleable algorithm for aggregating peer reputation, and (3) the specification of multimedia usage control policies as well as the first attempt to verify these policies using a temporal logic checker, J-Mocha. Among these three efforts, the work on resilient ALM has yielded a robust mechanism for providing quasi-real time audio- visual streaming among cellular phones and PDAs scattered over the world. The mechanism is ready for actual implementation and deployment. The other pieces of work, namely the aggregation of peer reputation and the verification of multimedia usage control policies are in their early stages. More efforts must be spent in order to turn them into useful tools.

Subproject (1) – Resilient Application Layer Multicast for Peers with Asymmetric and Intermittent Connectivity

1. Objective

As multi-receiver multimedia applications, like video conferencing, Internet television, on-line gaming and e-learning become increasingly popular on the Internet, multicast becomes a mechanism much need to be developed. Because of practical issues [1], IP multicast has not been globally deployed. Hence, an alternative approach known as application layer multicast (ALM) has been proposed. The concept of application layer multicast is that data packets are replicated at peers (user's desktops or mobile devices) and send to their topological children in the ALM service. In this effort, we tried to confront two challenges for providing ALM services to household users.

1. Peers with asymmetric connectivity for up and down links, like ADSL, VDSL or Cable modem, often have narrow upstream bandwidth. The capacity of peer's upstream bandwidth will influence on how many topological children it can support.
2. Peers, especially the mobile ones that use wireless links, may be disconnected from the network at any time, and thus cause the ALM services they render with unexpected performance degradation.

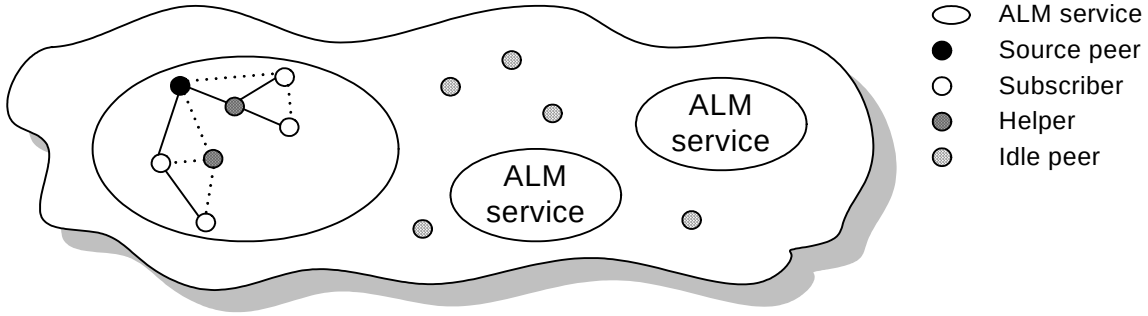

Figure 1: Application Layer Multicast (ALM) System

2. Approach

We devised a resilient application layer multicasting (ALM) mechanism, which has several desirable properties.

1. Peers can tolerate a small number of packet delay or loss and they still can receive complete message in time.
2. A sudden breakdown of some peers or links won't disrupt the reception of downstream peers.
3. Streaming data can be retrieved only by the peers subscribing to the ALM service, but not by the peers helping to provide the service.

We solve the asymmetric connectivity issue by making full use of peers' upstream bandwidth and increasing the total upstream bandwidth of the service.

To reach the purpose we described, we use three approaches, which are described in the following paragraphs.

(A) Information Dispersal Algorithm

Rabin's Information Dispersal Algorithm (IDA) [2] is used to process the message generated from source peer and a message will be divided into several stripes. The peers can't know any content without receiving a constant number of stripes. There are two meanings. First, it means you can't know any content of the message from one or few stripes. Therefore, if some peers have no authority to read the message, we only need to make sure not to let them receive too many different stripes of the message. Second, it means peers don't need to receive whole stripes of a message and they can still know the complete message. By this characteristic, we can improve the fault tolerance and robustness of data transmission.

(B) Multiple stripes

Each message transported from source peer that provide streaming data will be divided into several stripes [3]. In the multicasting tree formation, we not only construct one tree, we construct as many different trees as the number of stripes that one message is divided into. And each tree transmits different stripe of the message. This approach will lead to two advantages. First, each stripe will pass through different path. When one peer leaves or crashed, descendants of the peer will only lose one stripes and they still can receive other stripes. Furthermore, when a hop of the path is congested, it only influences on the reception of one stripe. Second, a stripe is much smaller than a message. The advantage is that a peer with low upstream bandwidth can also support several children because the data it needs to retransmit is small enough. Therefore, every peer can fully utilize their upstream bandwidth.

(C) Helper

Because the lack of upstream bandwidth, even we use multiple stripes approach to fully utilize every peer's upstream bandwidth; it is still possible that new peers can not find the parent in the tree that can provide them smooth streaming data. Therefore, the service will request some peers called helper that still have unnecessary bandwidth to join and share their bandwidth to increase the total upstream bandwidth of the service. With the aid of the helpers, the service can accept more peers and each peer can receive the streaming data

smoothly.

3. Results

We have combined the three approaches we mention above and produced a new application layer multicast system which is showed in Figure 1. In an ALM system, there are many ALM services and each service provides different streaming data. Every ALM service uses information dispersal algorithm, multiple stripes, and helper approach to operate independently. Each user who joins the system is called a peer. The peer who provides streaming data is called source peer. The peer who wants to receive the streaming data of the service and join it is called subscriber in that service. And for other services, it is a helper. The peer who doesn't join any service of the ALM system is called idle peer. And the idle peer is also a helper for all services in the system.

Currently, the simulation experiments are in progress. Because we lack the approach of topology construction, we adopt the existent tree construction approach [3] [4] to do our simulation. In future, we will devise a new distributed topology construction approach. And also, we will implement our mechanism to experiment in real network.

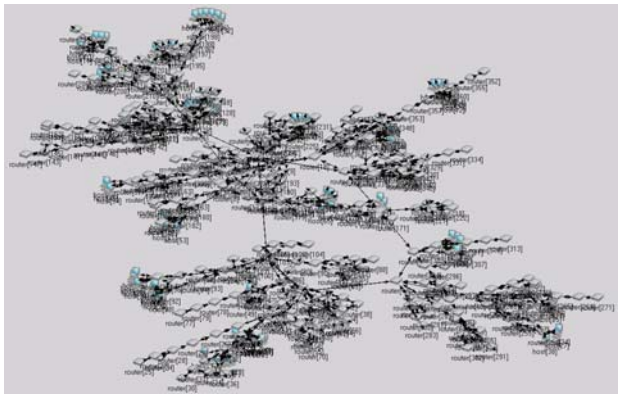

Figure 1. Network topology of Peer-to-Peer ALM Simulation[1]

[1] The ALM simulation network consists of 380 routers and 100 hosts – among them, 10 hosts are subscribers.

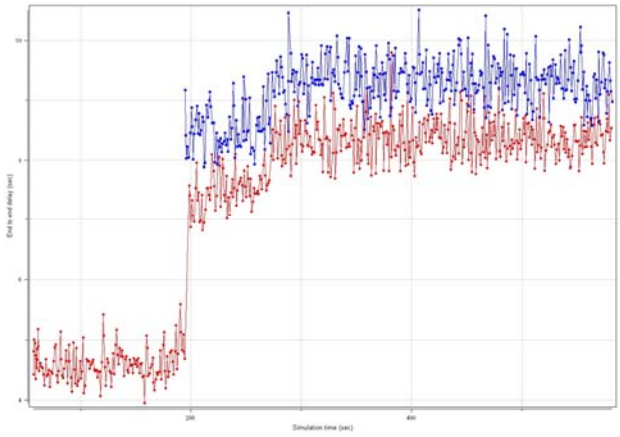

Figure 2. Packet latencies at two dispersed subscribers as more subscribers and helpers were introduced to the ALM trees.

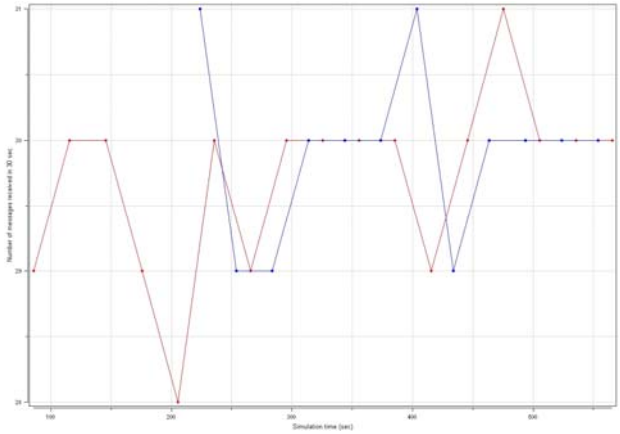

Figure 3. Recovered frame counts at two dispersed subscribers as more subscribers and helpers were added to ALM trees.

Subproject (2) – Scaleable Reputation Management for DHT-based Peer-to-Peer Networks

1. Objective

In a peer-to-peer service grid, when a user needs some resources, he will search for peers which provide the wanted resources and decide to get the resources from one of these peers. But there will be no central servers managing the whole peers in such a peer-to-peer network, and peers always join and leave dynamically. It's difficult for us to know who and where the peers exactly are. So the only way to help us to make decision is asking for other peers' recommendation. By gathering, combining and evaluating others' opinions, we can select the best one from all resource providers and start to

get resources from it.

The main problem is how to select the best provider of a wanted resource in an ad-hoc grid environment, especially in DHT-based peer-to-peer network [5]. We need some decentralized mechanisms to maintain the collection of recommendations from peers. After gathering information, it is necessary to appraise the trustworthiness of the information and evaluate the reputation consisting in the recommendations [6][7][8][9].

2. *Approach*

To reduce the cost of reputation collection, we let users pick up reputations along the search paths when searching for destination nodes. And to increase the number of reputations collected possibly, we make some adjustments in the storage and gathering of reputation scores:

1. In DHT-based architecture, if all nodes only keep their own reputation scores, users can only collect at most $\lceil \log_2 N \rceil$ (where N is the total number of peer nodes in the network) reputation scores along the searching path. To improve the condition, users have to store their own reputation scores additionally in other peer nodes, so that users can get more other peer nodes' reputation scores when they go through the searching path.
2. For a certain destination node, all peer nodes' searching paths are different. So some searching paths are longer and some are shorter. The longer the searching path is, the more reputation scores the user may get. To average the number of reputation scores which peer nodes can gather along their own searching path, we design a scheme about complementary nodes. We assign a complementary node to every peer node. The shorter the searching path of the peer node, the longer the complementary node's searching path. And the sum of the length of the two searching paths (originated from the peer nodes and the complementary nodes) will be $\lceil \log_2 N \rceil$. When a user gathers reputation scores, he can send requests to his complementary node and query for the reputation scores along the searching path of the complementary node. So every node will actually get the

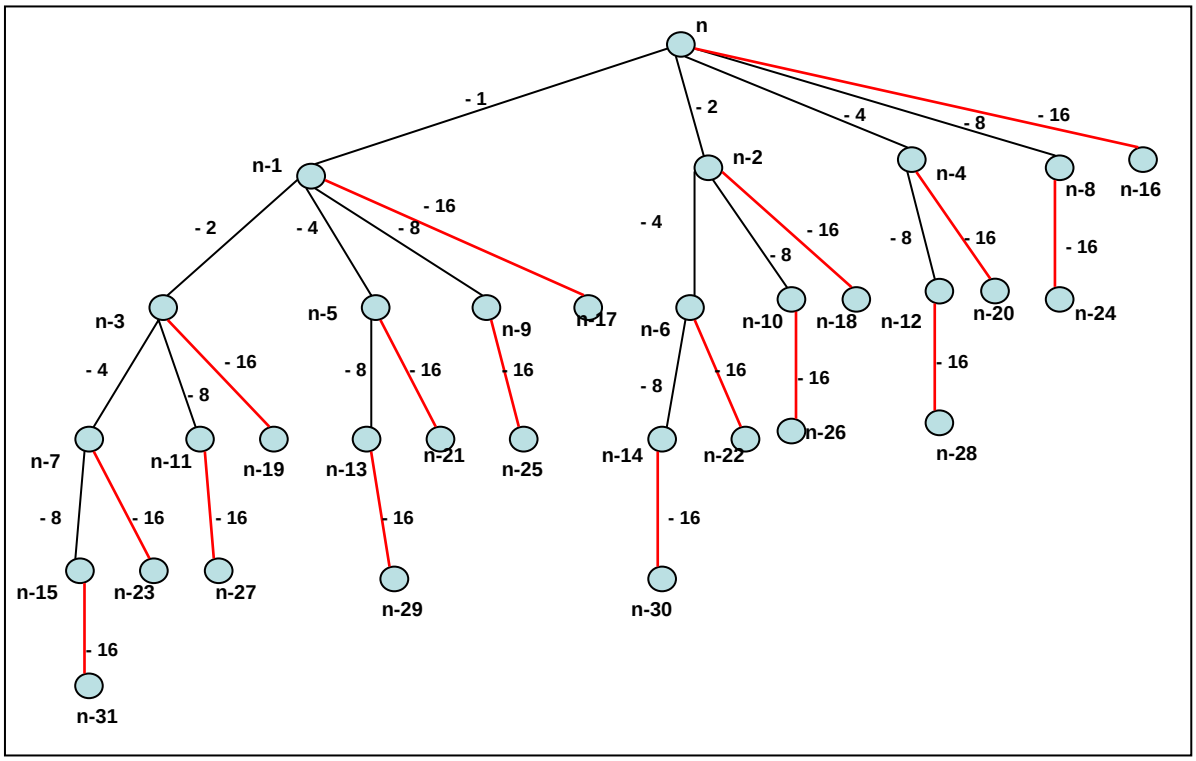

Figure 4. Disposition of aggregated reputation along DHT search paths

reputation scores stored in $\lceil \log_2 N \rceil$ nodes. It is also helpful to increase the number of reputation scores which most peer nodes can get.

After gathering enough reputation scores, users will estimate the trustworthiness of the nodes providing reputation scores and evaluate the information consisting in the reputations. According to the result, users can select the best provider and start to get resources. Moreover, users can assign new reputation scores to the provider after interacting with it and adjust the trustworthiness of the nodes from which users have received reputation scores. It is important to make sure that all information is up to date.

3. Results

Our mechanisms for maintenance of reputation scores are applicable to all DHT-based peer-to-peer networks. And now we can let users get $\theta(N)$ reputation scores by interacting $\lceil \log_2 N \rceil$ nodes averagely (where N is the total number of peer nodes in the network). Compared to the original reputation numbers, $\lceil \log_2 N \rceil$, $\theta(N)$ is obviously better, especially when N is large.

There are also many problems worth researching. The assignment of reputations can be multi-scoring, so the evaluating schemes have to consider that what attributes will influent the decision making most. Additionally, the part of trust management is also improvable. The assignment and adjustment of trustworthiness can be more accurate. At last, the whole mechanism can be more practical and be applied to real peer-to-peer application, such as file sharing system.

Subproject (3) – Verification of Multi-media Usage Control Policies using Temporal Logic Checker

1. Objective

Persisting Internet services such as WebTV, video conferences and on-line games must be monitored and controlled throughout the use of these services as some of the resources, users and/or system components may change their attributes/states, which may in turn alter the eligibility of the services. Notably, the control decisions often refer to the following factors.

- *Participant identities* and *attributes* — the identities may include *local identities* and *pseudonyms*, and the attributes shall include security relevant ones such as *capability* and *reputation*.
- *Resource usage constraints* — they are assigned by resource owners or external administration to enforce traditional discretionary or mandatory access control.
- *Service contexts* — they embody the environmental and situational factors under the influence of which the service may or may not be delivered or used.

These parameters were often considered in traditional access control problems. Yet, the introduction of subject/object anonymity, persistent services and dynamic contexts has created a new set of operational scenarios unforeseen in previous applications. So much so that it prompted experts in Role Based Access Control (RBAC) [10] to expand and rename the field as Usage Control [11]. In 2002, Ravi Sandhu proposed a usage control model called UCON*abc* [12]. The model introduced three components: *authorization*, *obligation* and *condition*. These three components combined with *attribute mutability* are deemed adequate for specifying usage control policies.

In this effort, we would like to construct a concrete usage control model for digital home multimedia applications based upon the UCON*abc* framework and use a temporal logic model checker, J-Mocha, to verify the usage control policies written according to our proposed model.

2. Approach

Our first attempt was to develop a concrete form for the multimedia usage control policies. By analyzing various usage scenarios – for example, a child is only allowed to watch movies in specific time and place specified by his/her parents – we identified the essential components of the multimedia usage control

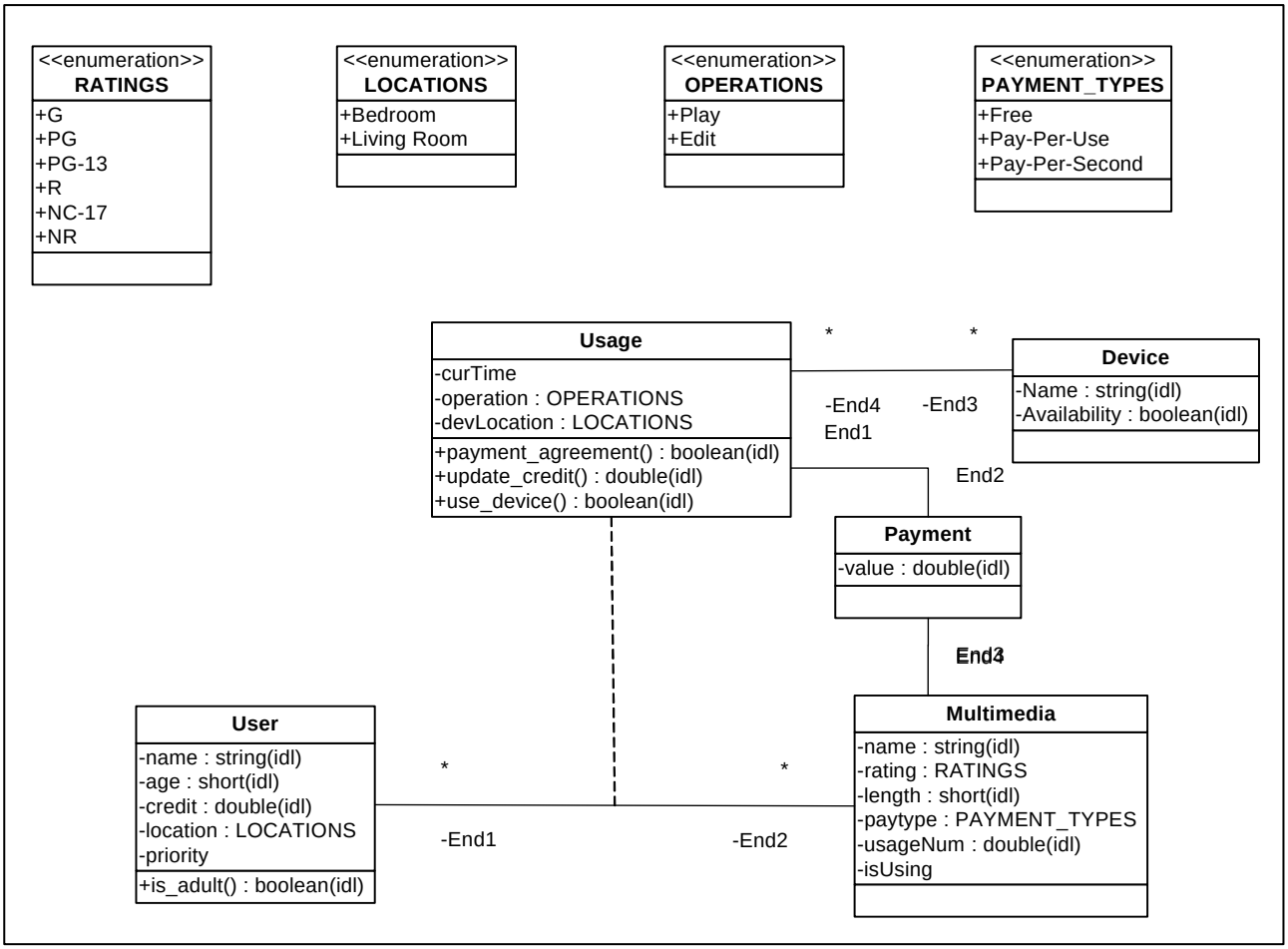

Figure 5. UML Diagram of Multimedia Usage Control Policies

policies. By matching these components with those in UCON*abc* framework, we developed a concrete policy model. Figure 5 shows in UML, the essential components of multimedia usage control policies.

By applying temporal logic checking onto the usage control model, we aim at achieving the following two things:

- *Inconsistency check*: it can discover contradictions in the policies that result in false cases;
- *Usage condition analysis*: it can identify one or more collections of cases that satisfy the set of policies.

3. Results

We chose to use a temporal logic checker J-Mocha [13][14] to conduct policy verification, and use the UNITY program specification language to express our policies. Our preliminary investigation showed that the specification of usage control policies requires sophisticated data structures usually unavailable in BDD-based model checkers. Consequently, we are exploring the use of pre-processing techniques to reduce the complexity of usage control policies.

Self Evaluation

Among the three efforts we conducted since August 2005, the work on resilient ALM has yielded a robust mechanism for providing quasi-real time video streaming among cellular phones and PDAs scattered over the world. The mechanism is ready for actual implementation and deployment.

The other two pieces of work, the aggregation of peer reputation and the verification of multimedia usage control policies are in their early stages. The reputation aggregation method allows peer users acquire the reputation scores of their targets along their search paths. Nevertheless, a viable way to discover the

potential bias of individual reputation scores will need to be devised.

The attempt to verify usage control policies using temporal logic checker proven to be premature as BDD-based verification techniques generally lack the ability to handle complete data structures. A pre-processing techniques is currently searched in order to overcome the problem.

References

[1] C. Diot, B. Levine, J. Lyles, H. Kassem, and D. Balensiefen, "Deployment issues for the IP multicast service and architecture," in *IEEE Network*, Jan. 2000.

[2] M. O. Rabin, "Efficient dispersal of information for security, load balancing, and fault tolerance," in *JACM*, Apr. 1989.

[3] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, "Splitstream: High-bandwidth multicast in cooperative environments," in *SOSP*, Oct. 2003.

[4] M. Castro, P. Druschel, A.-M. Kermarrec, and A. Rowstron, "Scribe: A large-scale and decentralized application-level multicast infrastructure," *IEEE JSAC*, vol. 20, no. 8, October 2002.

[5] E.K.Lua, J. Crowcroft, M. Pias, R. Sharma, S. Lim. "A Survey and comparison of Peer-to-Peer Overlay Network Schemes". IEEE Communications Survey and Tutorial. March 2004.

[6] E. Damiani, etc. "A Reputation-Based Approach for Choosing Reliable Resource in Peer-to-peer Networks" ACM CCS'02. November, 2002.

[7] B. K. Alunkal, etc. "Reputation-Based Grid Resource Selection"

[8] A. Josang, E. Gray, M. Kinateder. "Analyzing Topology of Transitive Trust". Proceeding of the First International Workshop on Formal Aspects in Security & Trust (FAST). September, 2003.

[9] S. D. Kamvar, Mario T. Schlosser, Hector Garcia-Molina. "EigenTrust Algorithm for Reputation Management in P2P Network" WWW2003, May 2003.

[10] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, R. Chandramouli. "Proposed NIST Standard for Role-Based Access Control". *ACM Transactions on Information and Systems Security (TISSEC) 4(3).* August 2001.

[11] R. K. Thomas, R. Sandhu. "Models, Protocols and Architectures for Secure Pervasive Computing: Challenges and Research Directions". *Proceedings of IEEE 2nd Annual Conference on Pervasive Computing and Communications Workshops (PERCOMW).* 2004.

[12] J. Pack, R. Sandhu. "The UCON Usage Control Model". *ACM Transactions on Information & Systems Security (TISSEC) 7(1).* February 2004.

[13] R. Alur and T. A. Henzinger. " Reactive Modules*" 11th IEEE Symposium on Logic Computer Science (LICS) ,1996

[14] R. Alur , H. Anand , R. Gerosu. "MOCHA User Manual, j-mocha v.2.0" 2004.