A Role Based Trust Model for Peer to Peer Systems Using Credential Trees

Chithra Selvaraj and Sheila Anand, *IEEE Senior Member*

Abstract—**This paper deals with a Role based Trust (RT) model for Peer-to-Peer (P2P) networks. P2P networks are essentially decentralized in nature to enhance resource sharing and collaboration. The anonymous and open nature of P2P systems offers an almost ideal environment for unauthorized access of digital content and also for easy distribution of malware. Today's popular P2P systems have to challenge the attacks by anonymous malicious peers. RT is a Trust Management framework for P2P networks where access control decisions are based on policy statements, called credentials, made by different principals and stored in a distributed manner. This paper explains the approach of building credential trees for credential chain construction. Credential trees overcome the cyclic dependency problem of credential graphs which may lead to non-termination. The credential trees are used for evaluating trust relationships.**

Index Terms—**Credential, Credential Chain, RT framework, Credential Tree, P2P Trust, P2P Trust Management.**

I. Introduction

Peer-to-peer networks have been popularized by Napster, an online music file sharing application. P2P computing has since seen tremendous growth with the emergence of other popular file sharing applications like Gnutella, Chord, Freenet and KaZaA. P2P is fast becoming an important technology for use in distributive and collaborative work both in the Web and in other ad-hoc networks. In pure P2P networks, there is no centralized entity or server and computer resources and functions are shared by direct exchange between peer computer systems. Peers can join and leave the network dynamically. As peers are autonomous and depend on each other for computer resources and for getting information, there is a greater security risk as compared with other types of distributed systems.

The open, unrestricted environment of P2P architecture makes it an ideal environment for unauthorized access to resources and information and also for attackers to spread malicious content. P2P systems have to protect themselves from attacks by anonymous malicious peers. Peers must determine that other peers are indeed who they declare they are and should be able to determine whether other peers are authorized to access resources or functionalities. The peers involved must establish trust before their interactions. Trust in P2P systems is the degree of belief about another peer.

Trust Management (TM) is the process which collects the information about a peer which is necessary to establish a trust relationship [9]. Also it monitors and adjusts the existing trust relationship. Access control to the resources may be provided based on the trust relationship with the peer.

Role Based Access Control (RBAC) systems provide access control to resources based on the responsibility of the user within the organization. But this requires centralized administration of users and privileges, which is not supported by distributed systems. Role Based Trust (RT) for P2P systems combines the strengths of RBAC and TM to provide access control. RT systems can be categorized as Policy based systems where peers use credential verification to establish trust relationship.

The demand of Role based trust systems increases rapidly when the resources belong to different security domains and are controlled by different authorities. Also, in P2P systems, the resource owner and the requester are unknown to one another. This makes access control based on identity to be ineffective. Role based Trust Management gives rise to the systems in which the access control decisions are based on the properties about the requester that can be derived from his credentials.

The family of Role based trust (RT) languages among various Trust Management languages [12] is used to represent policies and credentials which help to create a Trust model. The chain of credentials [19] for making access control decisions are created from credentials obtained from the user. A Trust Management credential chain is often a graph, rather than a linear path. In RT language RT_0, credential graphs are the searchable representation of credentials. However, the credential graphs may lead to the problem of cyclic dependency.

This paper proposes a Role based Trust Model which uses Credential Tree to establish the trust relationship between peers in P2P networks. We first review the previous work related to our work in Section II. We discuss the technical overview of RT_0 in Section III. The credential chain construction is illustrated with an example using RT framework Section IV. Credential Tree construction algorithm is discussed in Section V. The implementation details are given in Section VI.

II. Related Work and Our Approach

Several trust-management systems have been proposed in

Manuscript received September 29, 2010

Chithra Selvaraj, Department of Computer Science and Engineering, SSN College of Engineering, Kelambakkam, Chennai, Tamilnadu, India. (e-mail: chithraselvarajphd@gmail.com, chithraselva@yahoo.co.in)

Sheila Anand, Department of Computer Studies, Rajalakshmi Engineering College, Anna University, Chennai, Tamilnadu, India. (e-mail: sanijava@yahoo.com)

recent years. The Trust Management Systems are basically classified as Reputation based Trust systems, Social Network based Trust systems and Policy based Trust systems.

The Reputation based Trust systems like DMRep[21], EigenRep[11], P2Prep[15], XRep[16], NICE[13] in which the trust evaluation is based on measuring reputation. These systems are used to evaluate trust in the peer and trust in the reliability of the resource. Karl Aberer et. al. [20] proposed a Trust Management System which address the problem of reputation based trust management at both the data management and the semantic level. A Gossip based reputation system [2] collects the feedbacks from other peers and computes global reputation scores. PowerTrust [3] is also a reputation based system concentrates more on distributing the feedback about other peers. Instead of considering a peer to be trustworthy or not, various levels of trustworthiness are introduced based on fuzzy logic inferences. [6].

In this paper we present an approach that addresses the problem of reputation-based trust management at both the data management and the semantic level.

Social Network based Trust systems are based on the social relationship between the peers and these systems evaluate trust based on the analysis of the social network. Marsh[25], Regret[17], NodeRanking[18] are some of the Social Network based Trust systems.

The Policy based Trust systems like SPKI/SDSI [14], PolicyMaker [23], KeyNote [24], DelegationLogic [22] use credential verification to establish trust relationship for access control. These systems are based on the notion of delegation, whereby one entity gives some of its authority to other entities.

Role Based Access Control (RBAC) allows users access to resources based on their responsibilities within an organization. RBAC is more challenging in P2P systems, due to the lack of centralized administration. RBAC for P2P networks have been implemented using JXTA in [7]. RBAC deals only with authorization ensuring that a peer has access only to those resources that it should but not with authentication.

The credential is the statement signed by the issuer about a subject containing information about the subject. Policies govern the actions that principals are authorized to perform. Distributed authorization schemes allow enforcement of consistent security policies at end points, without assuming that the end points always have connectivity to a central server.

Role Based Trust (RT) languages are used for representing policies and credentials in distributed authorization [8]. The access control decisions are not necessarily based on the identity of the requester, but on the properties about him that can be derived from his credentials thus allowing anonymous interactions and role based trust models.

The introduction of RT framework [4] enables subject abstraction and supports distributed storage and discovery of credentials. Subject abstraction is the process of expressing the properties of the subject along with the attributes. The Role based Trust Management languages are a family of languages used for the construction of RT framework.

The design of Role Based Trust Framework and its languages are discussed in the work of Ningui Li, John C. Mitchell and William H. Winsborough [8]. RT_0 is the most basic language which introduces the semantics of credentials [10]. RT_1 adds the concept of attributes with fields or parameterized roles to $RT_{0.}$ RT_2 introduces the concept of logical objects in which the roles have values that are set of things other than entities. RT_1 adds the concept of parameterized roles to RT_0 and RT_2 adds the logical object usage to $RT_{1.}$

The language RT^T introduces the concept of Separation of Duties (SoD) and two new operators ⊙ for Multiple-role concurrence and ⊗ for separation of duties. Selective use of role memberships has motivated the introduction of the language RT^D which introduces dynamic delegation credentials.

The syntax and semantics of all these languages are based on that of $RT_{0.}$ The credentials in all these languages are represented based on the semantics of $RT_{0.}$ A credential chain is a chain of one or more credentials that delegates the authority from the source to the requestor. A central problem of Trust Management is to determine whether such a chain exists. This is "*Credential Chain Discovery Problem*".

A simple language RT_0 uses Credential graphs as the searchable representation of credentials. The storage of credentials can be centralized or distributed. Simple search algorithms were introduced for credential search if they are centrally stored. Goal directed algorithms and Heuristic search algorithms were introduced when the credential storage is distributed [1].

The credential graphs are directed graphs, which may lead to the problem of cyclic dependency. The credential search process may terminate due to this problem and trust may not be established. In our approach we have constructed the credential tree using the RT_0 syntax. Credential search can be done using the constructed tree which avoids the cyclic dependency problem.

III. Technical Overview of RT0

This section briefly introduces the terminologies used for the representation of Role Based trust management language RT_0 originally introduced by Li, et al [8].

An *entity*, also known as *principal* is a uniquely identified individual or process. Entities are represented by capital letters or abbreviated words like A, B, University. A *role* defines a set of entities who are members of this role.

Roles are represented by an Entity followed by a role name as University.Student, A.r, B.r1. For example, *University.student* represents a role, where *University* is the owner and *students* denote the set of members of the role. Only *University* has the authority to determine who are the students or members of the role *University.Student.* Roles can also represent permissions as well as other properties (*attributes*) of the relation between members and role. *Credentials* are the statements that are signed by the Issuer about a subject containing the information about the subject. For example, a credential can be represented as:

PConf.Discount ← University.student

In this credential, *PConf* is the Issuer, *University* is the subject and *PConf.Discount* is the body of the credential.

The *role* specifies a job function or a job title within an organization. Some associated semantics specify the authority and responsibility conferred by a role.

The credentials used for representing various roles are of 4 types. This classification can be tabulated in Table I.

TABLE I. REPRESENTATION OF ROLES

Credential	Semantic Expresseion	Role Functionality
Simple Member	A.r ← D	A asserts D is the member of role A.r
Simple Inclusion	A.r ← B.r1	A asserts that A.r includes all members of role B.r1
Linked Inclusion	A.r ← A.r1.r2	A asserts that A.r includes B.r2 for every B that is a member of A.r1 where A.r1.r2 is called linked role
Intersection	A.r ← B.r1 ∩ B2.r2	A asserts that A.r includes every principal who is a member of both B1.r1 and B2.r2

A role identifier is denoted as an entity assigned with a local role. For example, *University.student* represents the student role assigned by the entity "University".

A *principal* may transfer limited authority over one or more resources to other principals using credentials. These credentials are passed from one principal to another and are used to establish the sending principal's access rights.

The credentials can be stored with the Issuer or the subject. If all the credentials are stored only with the subject, a subject must have all the credentials authorizing the subjects for any resource. This may create a bottleneck. Otherwise, all the credentials can be stored only with the Issuer as in the TM Systems like QCM (Query Certificate Manager), SD3. This was again proven to be impractical for practical applications [5].

The *credential chain, which is the chain of,* credentials that delegates the authority from the source to the requestor.

Distributed Chain Discovery algorithm does not assume that the credentials are stored in one place. In Internet, the credentials are stored in a distributed manner, and the goal directed algorithm, issues a request for credentials, collects and then evaluates.

IV. CREDENTIAL CHAIN EXAMPLE

We use an example to illustrate policy, credentials and credential chains.

Let us consider a scenario where a student applies for admission to a Post Graduate programme in a University (*Univ*). The eligibility criteria for admission to the PG programme are:

- The student must have completed the Undergraduate progamme from a reputed University (RUniv) and obtained the degree
- RUniv is a university that is accredited by the Accreditation Board (ABU)
- The student should submit a recommendation letter (RLet) from a faculty of Univ

The Administration Committee (*AdmC*) of *Univ* is responsible for verification of the University credentials of the student, while the Selection Team (*SelT*) is responsible for verifying the recommendation given by the faculty (*Facrec)*

The policies defined by the University can be represented as

PG.Admission←Univ.PGAdm
Univ.PGAdm←AdmC.Preferred∩SelT.Preferred

The credentials for this example scenario can be represented as follows. The credentials for Alice who has submitted the application for PG Course are:

AdmC.preferred←AdmC.RUniv.Degree
AdmC.RUniv←ABU.Accredited
ABU.Accredited←RUniv
RUniv.Student←RUniv.RegId
RUniv.RegId←Alice
AdmC.Degree←Alice

The credentials for Bob, a faculty in RUniv who has given the recommendation letter for Alice can be represented as:

SelT.Preferred←SelT.RUniv.Facrec
SelT.University←ABU.Accredited
ABU.Accredited←RUniv
RUniv.Faculty←RUniv.FacId
RUniv.FacId←Bob
Bob.RLet←Alice
SelT.Facrec←Alice

In the example, Alice is the student who has received the degree from the University which is accredited by ABU and Bob is the faculty who has given the recommendation to Alice.

The basic idea of role-based trust management systems is to establish a credential chain. Effective trust chaining will eliminate forged credentials and secure P2P operations. RT framework uses directed graphs as the searchable representation of credential chains. The graph construction is well explained in the work of Winsborough et al [5] and [8].

A part of the credential graph which would be generated after reduction of credentials for the given example, is shown in Fig 1 [27].

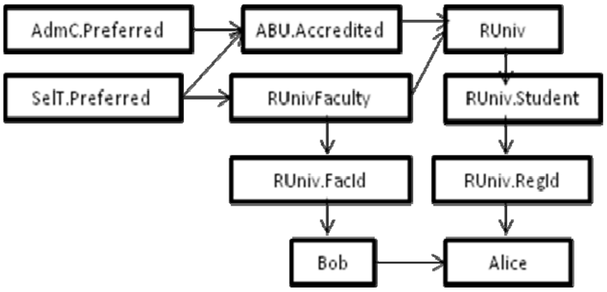


Fig.1 Credential Graph for Example

In the graph, the nodes AdmC.Preferred and SelT.Preferred, refer to the peer node RUniv for two purposes. The node AdmC.Preferred requests the RUniv node, to verify the credentials of the Student and SelT.Preferred refers to the RUniv peer to verify the credentials of Faculty. This results in a cycle in the graph, which would lead to non-termination in the search process of credentials.

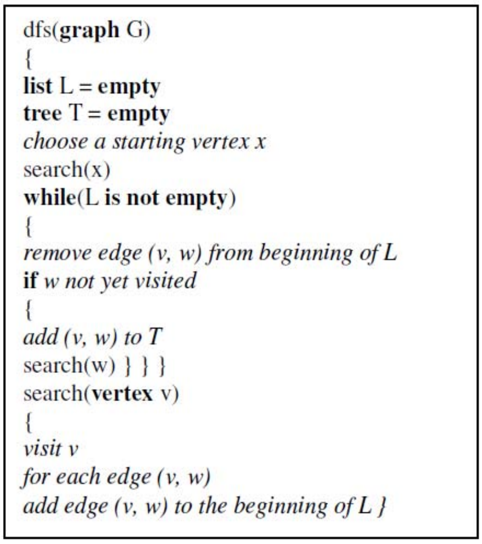
V. Credential Tree Construction

In our approach, we have used Credential tree to construct the credential chain to avoid the cyclic dependency problem.

The algorithm used for credential tree construction is given below for the four types of Role Representations. Create a root node and add the first credential to the root.

1. Node Insertion for Simple Member and Simple Inclusion:
   a. If a new node is to be inserted, search the available nodes for the role of the new node using backtrack approach. Insert the new node linking to the node to which the role matches.
   b. If the role of the new node does not match with any of the existing nodes, create a new node and add it to the root node.
2. Insertion of Linked Role – The linked role of the form A.r1.r2 can be represented as two branches with node values A.r1 and A.r2.
3. Insertion of Intersection Inclusion – The Intersection Inclusion of the form $A \cdot r \leftarrow B \cdot r1 \cap B2 \cdot r2$ can be represented as two branches with node values B1.r1 and B2.r2.

While building the credential tree, we have used the Depth First Search (DFS) algorithm represented in Fig 2 to search for the presence of a node.



```
dfs(graph G)
{
list L = empty
tree T = empty
choose a starting vertex x
search(x)
while(L is not empty)
{
remove edge (v, w) from beginning of L
if w not yet visited
{
add (v, w) to T
search(w) } } }
search(vertex v)
{
visit v
for each edge (v, w)
add edge (v, w) to the beginning of L }
```

Fig.2 Depth First Search algorithm

DFS is a uniformed search that starts from the first child node and progresses deeper and deeper till the goal node is found. If it hits a node that has no children, the search backtracks to the most recent node it has not finished exploring and proceeds with the search. The time taken by DFS to search [26] for an available node is expressed as :

$$O(|V|+|E|)$$

where V is the number of vertices (nodes) and E is the number of edges.

For the example given in Section IV, the first credential PG.Admission is added to the root node. When the node for PG.Admission is created as a new node, the node Univ.PGAdm is added as a child node to it. With the next credential, the intersection role AdmC.Preferred ∩ SelT.Preferred is added to Univ.PGAdm. Since this node is an intersection node, by Step 4 of Credential Tree construction algorithm, it is branched into 2 nodes AdmC.Preferred and SelT.Preferred. The linked role AdmC.RUniv.Degree branches into 2 nodes say AdmC.RUniv and AdmC.Degree. In the case of Linked role, the entity remains the same. This procedure is followed for adding all other nodes to the tree. The constructed credential tree is shown in Fig 3.

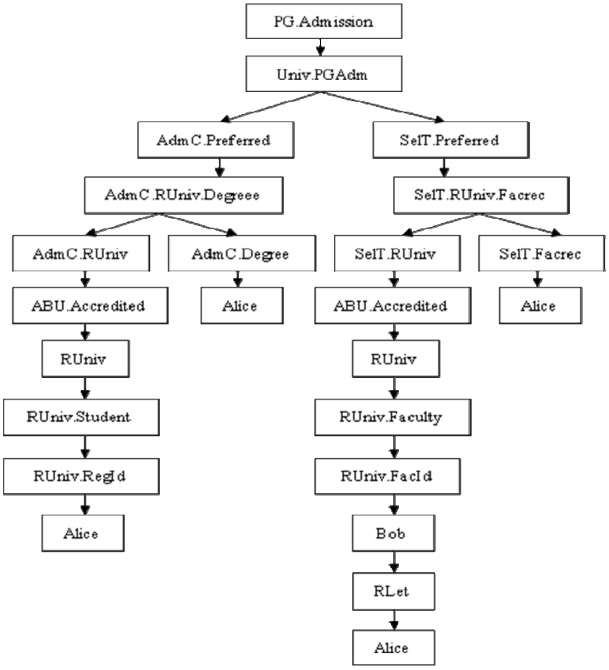


Fig.3 Credential Tree for example

VI. Implementation

We have implemented the Role Based Trust model for peer-to-peer systems using Java.

The credentials were stored in a text file. Each peer maintains a set of credentials relating to peers within its domain. The validating peer collects the credentials for various involved peers and builds the credential tree. The requested peers supply the credentials available with it. The credential tree is used to verify the credentials and trustworthiness of the requester. The trustworthy peer thereby gains access to the requested resource.

To illustrate with the given example, the peer AdmC would build the credential tree for validating the trustworthiness of Alice. It would request peer RUniv for credentials and would be supplied with credentials relating to Alice. Likewise, peer SelT would request credentials relating to Alice from other related peers.

Once the credential tree is built, DFS algorithm is used for searching the credential tree for verification of the trustworthiness of the requester. The time taken for searching the credential tree can be expressed as

$$MAX * O(|V|+|E|)$$

where MAX is the number of child nodes of the root node.

A sample set of credentials for various example scenarios were considered and the experimental results are tabulated in Table II.

TABLE II. EXPERIMENTAL OBSERVATIONS OF CREDENTIAL TREE ALGORITHM

No. of credentials	No. of Nodes	No. of Linked / Intersection Roles	Execution time in milliseconds
4	5	0	47
8	15	2	78
9	15	2	79
15	24	3	109
20	30	3	110

As shown in Table II, the number of nodes generated increases proportionate to the number of linked and intersection roles. The varying number of linked and intersection roles in the input credentials are plotted against the number of nodes generated. This is illustrated in the graph shown in Fig 4.

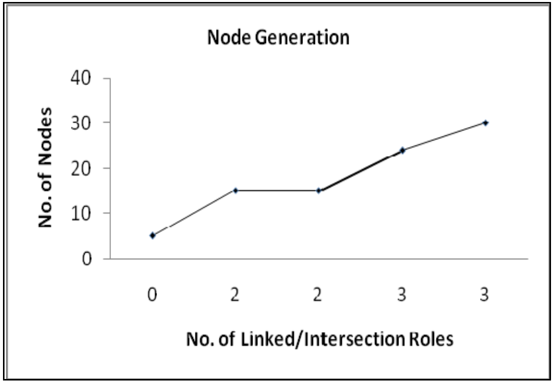


Fig.4 Number of nodes versus Number of linked and intersection roles

The execution time increases with the number of credentials that have to be processed. This is illustrated in the graph given in Fig 5.

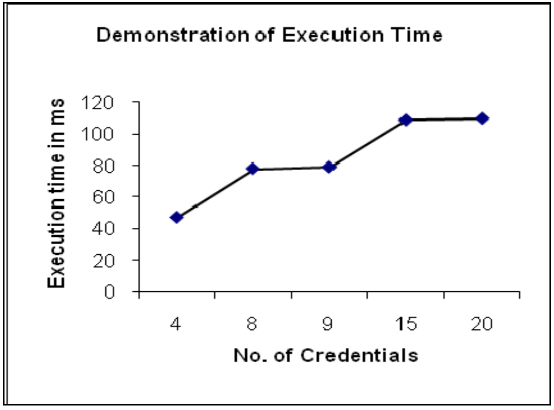


Fig.5 Comparison of number of input credentials and the Execution time of Credential tree algorithm

The credential tree construction time would be further reduced depending on the degree of duplication of the credentials available with the peers. For instance, the credential of Alice is available with the peers RUniv, Bob and Alice. The validating peer may first get the credentials from RUniv. The credentials of Alice and Bob would be included in the credentials supplied by RUniv; which are used by the validating peer to build the credential tree. When the validating peer requests the credentials from Alice from Bob, the credentials would already have been represented in the tree.

VII. CONCLUSION AND FUTURE WORK

P2P computing is emerging as a viable technology and computing model for business as well as for self-organized and self-managed online user communities. Secure exchanges of information and sharing of resources between peers become mandatory requirements for the successful deployment and use of P2P.

In this paper, we have proposed a Role Based Trust Model, a distributed trust architecture which uses credential trees for evaluating trust among peers in P2P networks. Peers can determine whom they can trust based on policies and credentials made by principals. Our work on credential tree construction is for RT_0 framework. This work can be extended for other RT languages.

As future work, we plan to apply Distributed Depth First Search (DDFS) algorithm for building and searching the credential tree. More detailed simulation and testing of the Trust Model network in live environments may also be undertaken for a more pragmatic evaluation of the proposed system.

REFERENCES

[1] Ke Chan, Kai Hwang and Ge Chan, "Heurisitic Discovery of Role Based Trust Chains in Peer to Peer Networks" IEEE Trans. on Parallel and Distributed Systems, TPDS--2007-08-0299, Finalized April 5, 2008

[2] R. Zhou, K. Hwang, and M. Cai, "GossipTrust for Fast Reputation Aggregation in Peer-to-Peer Networks", IEEE Transaction on Knowledge and Data Engineering, 2008.

[3] R. Zhou and K. Hwang, "PowerTrust: A Robust and Scalable Reputation System for Trusted Peer-to-Peer Computing", IEEE Trans. on Parallel and Distributed Systems, Vol. 18, No. 4, April 2007, pp.460-473.

[4] Marcin Czenko1, Sandro Etalle1;2, Dongyi Li3, and William H. Winsborough, "An Introduction to the Role Based Trust Management" Foundations of Security Analysis and Design IV – Volume 4677/2007

[5] Z. Mao, N. H. Li, and W. H. Winsborough, "Distributed Credential Chain Discovery in Trust Management with Parameterized Roles and Constraints", International Conf. on Information and Comm. Security (ICICS), Dec. 2006.

[6] S. Song, K. Hwang, R. Zhou, and Y. K. Kwok, "Trusted P2P Transactions with Fuzzy Reputation Aggregation", IEEE Internet Computing, Nov/Dec. 2005, pp.18-28.

[7] Amit Mathur, Suneuy Kim, Mark Stamp, "Role Based Access Control and the JXTA Peer-to-Peer Framework" www.truststc.org, 2005

[8] Ningui Li, John C. Mitchell and William H. Winsborough, "Design of a Role Based Trust Management Framework " Proc. of LMW02, 2004

[9] E. Bertino, E. Ferrari, A.C. Squicciarini, "Trust-X: A Peerto-Peer framework for Trust Establishment", IEEE Trans. on Knowledge and Data Engineering, July 2004, pp: 827-842.

[10] N. H. Li, W. H. Winsborough, and J. C. Mitchell, "Distributed Credential Chain Discovery in Trust Management", Journal of Computer Security, Vol.11(1), Feb. 2003, pp. 35-86.

[11] D.kamvar, Mario T. Schlosser, "Eigen Trust Algorihm for Reputatation Management in P2P networks", May 2003, ACM 1-58113-680-3/03/0005

[12] N. H. Li, J. C. Mitchell, "Datalog with Constraints: A Foundation for Trust Management Languages", Proc. of the 5th Int'l Symposium on Practical Aspects of Declarative Languages. LNCS 2562, Springer-Verlag, 2003, pp.58-73.

[13] S. Lee, R. Sherwood, "Cooperative peer groups in NICE", IEEE Infocom, San Francisco, USA, 2003.

[14] D. Clarke, J. Elien, C. Ellison, M. Fredette, A. Morcos, and R. L. Rivest, "Certificate Chain Discovery in SPKI/SDSI", Journal of Computer Security, Vol.9, No.4, 2001, pp.285-322. Proc. of the IEEE Symposium on Security and Privacy, May 2002, pp. 114–130.

[15] F. Cornelli, E. Damiani, S.C. Vimercati, S. Paraboschi, and P. Samarati, "A reputation-based approach for choosing reliable resources in peer-to-peer networks," In CCS'02, Washington DC, USA, 2002.

[16] F. Cornelli, E. Damiani, S.C. Vimercati, S. Paraboschi, and P. Samarati, “Choosing reputable servents in a P2P network” In proceedings of the eleventh international conference on World Wide Web, Honolulu, Hawaii, USA, 2002.

[17] J. Sabater, C. Sierra, “Reputation and social network analysis in multi-agent systems”, First International Joint Conference on Autonomous Agents and Multi-Agent Systems, Bologna, Italy, 2002.

[18] J. Pujol, R. Sanguesa, “Extracting reputation in multi agent systems by means of social network topology”, First International Joint Conference on Autonomous Agents and Multi-Agent Systems, Bologna, Italy, 2002.

[19] Ningui Li, John C. Mitchell and William H. Winsborough, “Distributed Credential Chain Discovery in Trust Management” Procs. of the 8th ACM Conference on Computer and Communications Security (CCS-8), pages 156–165, ACM Press, November 2001.

[20] K. Aberer and Z. Despotovic, “Managing Trust in a Peer-2-Peer Information System”, Tenth Int’l Conf. on Information and Knowledge Management, New York, 2001.

[21] DMRep - K. Aberer, Z. Despotovic, “Managing Trust in a Peer-2-Peer Information System”, In Proc. of the IX International Conference on Information and Knowledge Management, Atlanta, Georgia, 2001.

[22] Ninghui Li, Benjamin N. Grosof, and Joan Feigenbaum, “A Practically Implementable and Tractable Delegation Logic,” In Proceedings of the 2000 IEEE Symposium on Security and Privacy, pages 27-42. IEEE Computer Society Press, 2000.

[23] M. Blaze, J. Feigenbaum, and M. Strauss, “Compliance- Checking in the PolicyMaker Trust Management System”, Proc. of 2nd Int’l Conf. on Financial Cryptography (FC'98), LNCS1465, Springer, 1998, pp. 254-274.

[24] M. Blaze, J. Feigenbaum, and J. Lacy, “Decentralized Trust Management”, IEEE Symp. on Security and Privacy, May 1996, pp.164-173.

[25] S. Marsh “Formalising Trust as a Computational Concept”, Ph.D. Thesis, University of Stirling, 1994.

[26] “Introduction to Design and Analysis of Algorithms” -Anany Levitin

[27] Chithra Selvaraj, Sheila Anand, “A Role Based Trust Model for Peer-to-Peer Systems”, proceeding of 2009 International Conference on Computer and Network Technology (ICCNT 2009), World Scientific Publications, Dec 2009.

Chithra Selvaraj is Assistant Professor in SSN College of Engineering, Kelambakkam, Chennai. She has completed her Bachelor of Engineering degree in Computer Science and Engineering from Bharathiar University, Coimbatore. She has ranked as Sixth in the University ranking. She has completed her Master of Engineering in Computer Science and Engineering from Sathybama University, Chennai. She has registered her Ph.d course in Anna University, Chennai and indulged in research work in the field of Security for Peer to Peer Systems. Her work presented in a conference ICCNT ‘09 was awarded as the “Best Paper”. On part of her research work, she has worked in the area of trusting a peer based on the role played by a person. This work has been tested and the results are recorded.

Dr. Sheila Anand is a Senior Member of IEEE and IEEE computer Society. She is an Engineer by qualification, having done her Doctorate in the area of Information security from Anna University, Chennai and Masters in Computer Science Engineering and Bachelors in Electronics & Communication Engineering from Madras University. Has the distinction of being placed first in the Madras University ranking in M.E. She holds many professional qualification and certifications including CISA, CSQA and CISM. She has also completed the ITIL foundation course in IT Service Management. She is presently the Dean (Research) Computer Studies at Rajalakshmi Engineering College, Chennai. She is presently guiding a number of research scholars registered for research at Anna University, Chennai and Tiruchirapalli. She is a member of the Board of Studies of the Faculty of Information and Communication Engineering, Anna University, Chennai.