Secure and Effective P2P Reputation System using Trust Management and Self Certified Cryptographic Exchanges

M. Srikanth[1] and K.B. Madhuri [2]

**[1] Dept. of IT, G.V.P College of engineering(A),
Madhurawada, Visakhapatnam-530048, Andhra Pradesh, India**

**[2] Dept. of IT, G.V.P College of engineering(A),
Madhurawada, Visakhapatnam-530048, Andhra Pradesh, India**

Abstract

The main reason behind the success of Peer-to-Peer (P2P) networks is the anonymity and the utility offered by them to the users. The Client-server security policies implemented in centralized distributed systems do not provide a desirable solution for P2P networks to store peer reputation information. Though a centralized system does exist, it is prone to Sybil attack that can significantly reduce network performance. One feasible way to minimize attacks is to establish the reputation-based trust model. Most of current trust mechanisms are unable to restrain effectively such malicious behavior as collusive attacks, but also take no consideration for the security of the trust management. In this paper, a secure and effective reputation based distributed P2P trust management model has been proposed which has advantages in combating various malicious behaviors and uses Self Certified Cryptographic Exchanges between the peers. This can effectively track each peer's contribution in the system and allows peers to store their reputations locally and exchange that information with other peers via a two-party cryptographic protocol. During network initiations instead of ostracizing a selfish requester completely, it offers services at low bandwidth and its presence can be used to boost overall performance of the network.

Keywords: *P2P, trust management, reputation, security, reputations*

1. Introduction

Peer to Peer is an approach to computer networking where all computers share equivalent responsibility for processing data. Peer-to-Peer networking (also known as peer networking) differs from client-server networking, where certain devices have responsibility for providing or "serving" data and other devices consume or otherwise act as "clients" of those servers. A p2p network is decentralized, self-organized, and dynamic in its pure sense, and offers an alternative to the traditional client-server model of computing. Client-Server architecture enables individuals to connect to a server, although servers are scalable, there is a limit to what they can do. P2P networks are almost unlimited in their scalability. Different applications of P2P networks enable users to share the computation power (distributed systems), data (file-sharing), and bandwidth (using many nodes for transferring data). P2P uses an individual's computer power and resources, instead of powerful centralized servers. The shared resources guarantee high availability among peers. P2P is a really important area to research, because it has a huge potential in distributed computing. It is also important for the industry as well, as new business models are being created around P2P. The peers in the P2P network have to be discouraged from leeching on the network. It has been shown in Tragedy of Commons [1] that a system where peers work only for selfish interests while breaking the rules decays to death. Policing these networks are extremely difficult due to the decentralized and adhoc nature of these networks.

Reliable peer reputations [2] could be used in a variety of ways. They can help to find out the peers who have good reputations and hence help them in making decisions about who to serve content to and who to request content from [13]. During the bootstrapping process for joining the P2P network, peers can potentially use reputations to decide who to directly connect to in the overlay topology. Since there is no centralized node to provide as an authority to monitor and punish the peers that behave badly, malicious peers have an encouragement to provide poor quality services for their benefit because they can get away. Some traditional security techniques, such as service providers requiring access authorization, or consumers requiring server authentication, are used as protection from known malicious peers. However, they cannot prevent from peers providing variable-quality service, or peers that are unknown.

Most of the existing reputation-based trust models [3][4] compute the trusted rank of one peer based on its transaction histories with others, and it is very likely that the peer with the highest trust value is looked on as the service provider. To a certain degree, this approach has some effects on the simple malicious behavior patterns, but shows little effect in dealing with the complex attacks and disturbance activities on reputation systems, such as collusions. Besides, most of current researches concentrate on the design and implementation of the trust system, and pay less attention to the security problem confronted by its reputation management. In fact, security of reputation management is the key element assuring the normal running of the trust management system (TMS), and is as important as any other element of the reputation management.

In this paper, Reputation Systems have been investigated for P2P networks which are resistant to Sybil attacks and used to find out the malicious peers in the network. A more ambitious approach is proposed to protect the P2P network without using any central component, and thereby harnessing the full benefits of the P2P network. The Proposed system encapsulates the reputation of the requester and reputation of the provider for providing efficient reputation. The reputations of the peers are used to determine whether a peer is a malicious peer or a good peer. If the malicious peer is found in the network instead of ostracizing malicious peer completely, it can be offered services at low bandwidth and its presence can be used to boost overall performance of the network. And this paper proposes a reputation based distributed P2P trust model integrated with the security mechanism for the reputation information management for P2P networks. All peers in the P2P network are identified by identity certificates (aka identity). The reputation of a given peer is attached to its identity. The identity certificates are generated using self-certification, and all peers maintain their own (and hence trusted) certificate authority which issues the identity certificate(s) to the peer.

2. Related Work

In 2003 Kamwar S. D et al., proposed a reputation management system, Eigen Rep [5] ,for P2P networks that describes an algorithm to decrease the number of downloads of inauthentic files appearing in a peer to peer file sharing network. In a normal peer to peer network each peer is assigned a unique global trust value, based on the peer's history of uploads. A distributed and secure method is presented to compute global trust values to choose source peer for downloading and the network should effectively identify malicious peer and isolate them from the network.

Aberer et al., [6] have proposed completely distributed solution for trust management over the P-Grid peer-to-peer network. A binary tree is chosen to store reputation data and this information is sent over the network. If an agent looks for recommendation data of another agent, he has to search the P2P network and has to compute the reputation from the recommendations received. Chen and Singh [3] and Schein et al. [17] have also provided trust models, similar to those mentioned above.

In the year 2000, Dellarocas [7] specified the design challenges in the online reporting systems. Dellarocas has extensively surveyed online reputation, reporting mechanisms, and the corresponding issues. In addition to the design challenges, he has provided a good overview of recommendation repositories, professional rating sites, collaborative filtering systems [8], and regression approaches. Dellarocas also emphasized on the reputation systems attacks and techniques for foiling those attacks. Ballot stuffing and bad mouthing are two kinds of attacks that can be inflicted on the reputation systems. In ballot stuffing, a peer receives a large number of (false) positive recommendations from its friends, to raise its own reputation. In Bad mouthing a large number of negative recommendations are issued for a specific peer. The author suggests that by maintaining anonymity of requesters the problems of negative and positive discrimination can be solved.

In 1996 R.L. Rivest et al., proposed a Simple Distributed Security Infrastructure (SDSI) [9], that simplifies the X.509 certificates design and provides the means for self-certification, local name spaces; secure formation of groups, and simple access control mechanisms. Methods are provided to incorporate global name spaces and globally trusted identities within the SDSI infrastructure. The basic underlying principle of P2P networks is distribution of authority among all members of the networks. This is followed in SDSI.

Dynamic Trust Management in dynamic distributed environments provides a concealed trust management, where the members of the system play multiple roles which are frequently changing. In addition, the members themselves are temporary. Stanford University has developed methods for identification of components, their authentication, secure group communication protocols, and dynamic trust management under the project Agile Management of Dynamic Collaborations [10],

Prashant Dewan et al., [11] proposed following rules for a good P2P reputation management:

1. A self-certification-based identity system protected by cryptographically blind identity mechanisms.
2. A light weight and simple reputation model.

3. An attack resistant cryptographic protocol for generation of authentic global reputation information of a peer.

Good peers and malicious peers are identified by their reputation. Good peers cannot perform any transactions with malicious peers as they are ostracized from the network. The malicious activities like false recommendations are reduced once the malicious peers get disconnected from the network. In p2p reputation management systems developed till now, the focus is only on using the reputation of the provider and the reputation of the requester is ignored. A secure and effective p2p reputation system is proposed which focuses on using the reputation of both provider and requester.

3. Proposed Work

3.1 Reputation Based Trust Model

First we categorize the peers in P2P into four classes: Honest Peer, Selfish Peer, Malicious Peer, and Evil Peer.

Honest Peer:

These peers initiate's only good transactions. These peer ratings are always correct i.e. good transactions are rated good and bad transactions are rated bad by them.

Selfish Peer:

These peers are called free-riders. These Peers blocks all inquiries by other agents and refuses to rate his transaction partners. He just initiates neutral to good transactions by himself.

Malicious Peer:

These type peers initiate's good, neutral and bad transactions by chance. These Peers tries to damage the system with his rating behavior and rates every transaction negative.

Evil Peer:

These type peers try to gather a high reputation by building a group in which they know each other. If an evil agent finds another evil agent to trade with, they always give each other a good rating. If an evil agent does not find another evil agent, after seeking for a while, he transacts neutral and rates neutral.

3.2 Reputation Model:

After transacting with each other, one peer i (the service consumer) will submit its ratings to the other peer j. i.e. once a peer has obtained its identity; it joins the P2P network using the standard Join method of the particular P2P network. The peer (requester) searches for one or more files using the Search method provided by the network. On the basis of the responses received, as a result of its search request, the application at the requester side generates a list of peers who have the requested file(s). The requester selects the peer (provider) with the highest reputation from the list and initiates the cryptographic protocol. The cryptographic protocol is presented in detail in the next section. In this protocol, the requester uses the Download method of the network, to download the file from the provider. Subsequently, it verifies the integrity, authenticity, and the quality of the file. Depending on its verification results, it generates secure and effective reputation by the application itself based on their upload ratio & download ratio to the provider and stores it locally that is beyond their control. If any system provides reputation manually to other peers it does not guarantee that reputation is correct feedback or not. That's why the proposed system is providing secure reputation by the system itself based on their upload & downloads ratios.

The recommendations are constrained to boundaries in order to make sure that one recommendation does not completely nullify or drastically improve the reputation of a provider. Once the provider receives the recommendation, it averages the previous recommendations received by it and the recent recommendation to calculate its reputation automatically based on their upload and download speed ratios. The above mentioned steps are repeated for every transaction, and it is necessary to normalize them in some manner, ensuring that all values will be either positive or negative.
The reputations of the peers are used to determine whether a peer is a malicious peer or a good peer. Once detected, the malicious peers are not ostracized from the network .Instead of Ostracizing the self– fish / malicious peer completely the proposed system can offer services at low bandwidth and their presence can boost up the overall performance of the network i.e. indirectly the proposed system increasing the total amount of time required to download the particular file to malicious peer, as a result the decreased speed spread to the network and boost overall performance of the network. In that meanwhile if any peer wants a service or file from malicious peer, they can directly download/take the service from the malicious peer, i.e. indirectly the malicious peer giving its contribution to the network.

3.3 Self-Certification

A trusted authority issues identity certificates in a centralized system. P. Dewan proposed a self-certification mechanism [12] that splits the trusted entity among the peers and enables them to generate their own identities in a decentralized reputation system. Certified Authority (CA) is run by each peer so as to issue an identity certificate(s) for itself. These self certified certificates are similar to SDSI certificates [9]. Each peer has its own reputation and the reputations of all peers collectively form the reputation of a CA.

In Self-certification mechanism there is no need for centralized trusted entity which issues identities in a centralized system. There is no way to map the identity of a peer in the system to its real-life identity when they use self-certified identities. They remain pseudonymous in the system. The idea of making peers anonymous or pseudonymous is desirable in P2P networks, but it can also backfire sometimes.

In Self-certification mechanism the anonymity of the peers is preserved by grouping of peers. The combination of self certification and anonymity limits the possibility of a Sybil attack. In contrast to the traditional CA-based approach, neither the group authority nor the transacting peers can establish the identity of the peer. In addition, certificate revocations are not needed in the group-based approach as the group authority only vouches for the real-life existence of the peer, unlike the traditional certificate-based approaches where various certificate attributes are attested by the authority and necessitate revocation if any of those attributes mutate in time. If a highly reputed identity is compromised, its misuse would be self-destructive as its reputation will go down if misused.

3.4 Reputation Exchange Protocol

As opposed to centralized systems, self-certification distributes the trusted entity among the peers to enable them to generate their own identities. Certified Authority (CA) is run by each peer so as to generate the identity certificate(s) for themselves. The provider with the highest reputation is selected by the Requester; After selecting the provider it uses the reputation exchange protocol [11] with the provider. In the reputation exchange protocol, the requester is denoted by R while the provider is denoted by P. The symbol PK2 represents the private key of the peer P and PK1 represents the public key of the peer P. This protocol only assumes that insert & search functions are available and are not resilient to peers that may not follow the recommended join & leave protocol of the network. The proposed work has extended the existing protocol by including the reputations of the requester and provider. The steps in the reputation exchange protocol are as follows:

The following steps are modified version of Reputation Exchange protocol.

Step 1: Requester send request for transaction and its own certificate identity to provider.

R → P: RTS & IDR

Step 2: The provider sends its own certificate identity, the current transaction id (TID) and the signed TID.

P→ R: IDP & TID & EPK2 (H (TID ‖ RTS))

Step 3: Requester get last transaction id that was used by provider from the network.

R: LTID = Max (Search (PK1‖ TID))

Step 4: If the value of the LTID found by the requester from the network is greater than or same as the TID offered by the provider, it implies that the provider has used the TID in some other transaction.

R: IF (LTID >= TID) GO TO Step 12

Step 5: If the check in Step 4 succeeds, i.e., the requester is sure that the provider is not using the same transaction number.

R→P: Past Recommendation Request & r

Step 6: The provider sends its past recommendations. The provider signs the CHAIN so that the requester can hold the provider accountable for the chain.

P→R: CHAIN, EPK2 (CHAIN)

Step 7: The requester verifies the CHAIN by simple public key cryptography. If it has the certificates of all the peers with whom the provider has interacted in the past, the verification is simple.

R: Result = Verify (RECN1; RECN2 . . . RECNr
If Result! = Verified GO TO STEP 12

Step 8: The provider provides the service or the file as per the requirement mentioned during the search performed for the providers.

P→R: File or Service

Step 9: The provider receives the blinded recommendation from the requester.

R →P: B1 =EBKa (REC ‖ TID ‖ ERK2 {H (REC‖ kTID)})

Step 10: The provider cannot see the recommendation but it signs the recommendation and sends the recommendation of provider, NONCE and the signed recommendation back to the requester.

a.) P → R: B1|| EPK2 (H (B1|| REC (P) || kTID), nonce), nonce
b.) R→ P: Ka

Step 11: The requester signs the recommendation that was given to the provider (REC), the transaction id (TID), and its own identity certificate and stores it in the network.

Insert (IDR,{ REC || TID || ERK2{ H(REC)| | H(TID) }})

Step 12: the steps a requester executes when it expects foul play:

ABORT PROTOCOL
R: Insert(IDR ,{ CHAIN (TID (ERK2{ H(CHAIN)|| H(TID)}})

Peers who enroll can enhance their scores by being good citizens of the P2P network. The solution utilizes a reputation computation agent (RCA) for fair periodic updates to each enrolled peer's reputation, still ensuring that the reputation points for each peer are kept locally for fast retrieval. Several factors affect the accuracy of reputation scores. Some of these are: state maintenance at the RCA, network overheads, and loss of data while communicating with the RCA.

4. Performance

The Proposed Protocol generates Less Network Traffic than that of the previous work. In both, P2P Reputation protocol and the proposed protocol, the requester maintains the list of possible providers when the discovery process completes. P2P Reputation is highly communication intensive; the initiator polls the peers in the network for a vote on the provider. Subsequently each peer sends a message containing a vote to the initiator. As a result, the amount of traffic increases linearly as the number of peers who reply to the initiator increase.

The Security of the system is inversely proportional to the amount of communication between the requester and provider. This is because the reputation values can be discovered with increase in communication. In P2P Reputation, a requester polls peers, which are topologically closer to the provider. This strategy may not return high percent of relevant recommendations because only the reputation of the provider is considered and reputation of the requester is ignored. In the proposed protocol, it is recommended to encapsulate the reputation of both provider and requester. And for the requester to search the network for LTID but it is not mandatory. In P2P Reputation, the vote polling cannot be bypasses because the provider also receives the poll request. Hence, if the provider does not receive any poll request, it will know that the requester has not performed any security checks. Hence, the probability of the provider cheating is high.

5. Experiment Results

This section presents the latency measurements obtained from a prototype implementation of Secure and Effective Reputation Management system deployed on the network. The Protocol was evaluated by creating a physical network consists of 72 peers participating in 500 – 800 transactions. The range was set to 5 i.e. it was assumed that each file was available with 5 possible providers. The requester didn't have any prior knowledge of malicious nodes. The malicious node performed like a selfish node with a probability of 1/2 i.e. malicious nodes cheated in one out of two transactions. The software developed has been installed in each system besides configuring the system as a peer. The Proposed software runs on windows, uses 160-bit keys obtained from the SHA-1 cryptographic hash function, and uses TCP to communicate between nodes. The Proposed system runs in the iterative style. This is different from running virtual nodes at each simulator to provide good load balance; rather, the intention is to measure how well our implementation scales even though we do not have more than a small number of deployed nodes. The main challenging that the proposed system achieved to create memberships and groups that are dynamic, Unpredictable network latencies, Self – certification, No central services, Security. The search process is very fast when compared to the previous work. Used reputations are considered to build trustiness relationships. Here the proposed model mainly focused on security that was achieved successfully by the use of JXTA project application.

5.1. Advantages of using Proposed Reputation System

The change in the number of transactions in the network is measured. The number of malicious nodes was set to constant at 50 percent and the number of malicious transactions was incrementally raised from 50 to 300 as visible in fig. 5.1, the total number of malicious transactions decreased considerably with an increase in the number of transactions in the proposed model. While making number of increasing transactions constant (800), the rate of increase in the number of malicious transactions was much less when reputations were used (fig: 5.1).

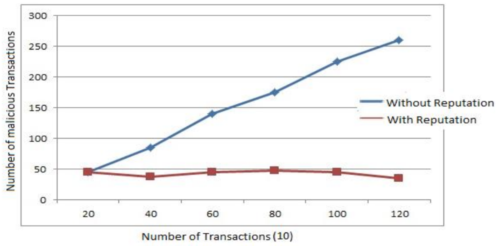

Figure.5.1: Variation in total number of malicious transactions.

Subsequently, the experience of each peer when the reputations are not used is analyzed as shown in fig.5.2; from these experiments it is shown that the proposed model reduces the number of malicious transactions from the perspective of the network and from the perspective of each peer.

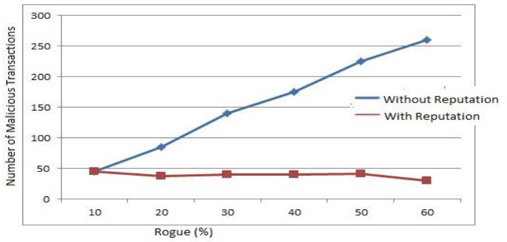

Figure.5.2: Variation in number of rouges

6. Conclusion

Enabling peers to develop trust and reputation among themselves is important in a peer-to-peer system where resources (either computational, or files) of different quality are offered. It will become increasingly important in systems for peer-to-peer computation, where trust and reputation mechanisms can provide a way for protection of unreliable, buggy, infected or malicious peers. The authenticity of the reputation information is the basis of assuring the normal running of Trust Management System (TMS). After analyzing the security risks existing in the current TMS, this paper proposes a secure and effective reputation based distributed trust management model which uses Self-certification, an identity management mechanism, and a cryptographic protocol that facilitates generation of secure reputation data in a P2P network, in order to expedite detection of rogues. This paper discusses the reputations managed in the network, the corresponding reputation information given to peers and identification of malicious nodes. Once the malicious nodes are identified based on their download's ratios and activities in the network, instead of ostracizing the selfish peer completely, the proposed system provides services at lower bandwidth and its presence can boost up network performance. The proposed model is more secure, robust and effective on attacks from various malicious peers, including peers with malicious behaviors and peers with security threats, and shows more improvements in the security feature of the trust management.

References

[1] H. Garett, "Tragedy of Commons," Science, vol. 162, pp. 1243-1248, 1968.

[2] P. Resnick, R. Zeckhauser, and E. Friedman, "Reputation Systems," Comm. ACM, vol. 43, pp. 45-48, Dec. 2000.

[3] M. Chen and J.P. Singh, "Computing and Using Reputations for Internet ratings," Proc. Third ACM Conf. Electronic Commerce, pp. 154-162, 2001.

[4] K. Aberer and Z. Despotovic, "Managing trust in a peer-2-peer information system," Proc. 10th Int'l Conf. Information and Knowledge Management (CIKM '01), pp. 310-317, Nov. 2001.

[5] Kamwar S. D, Schlosser M. T, Hector Garcia-Molina. "The Eigen Trust algorithm for Reputation management in P2P networks". Proc. 12th Int'l World Wide Web Conf., pp. 640-651, 2003.

[6] K. Aberer and Z. Despotovic, "Managing Trust in a Peer-2-Peer Information System," Proc. 10th Int'l Conf. Information and Knowledge Management (CIKM '01), pp. 310-317, Nov. 2001.

[7] C. Dellarocas, "Immunizing Online Reputation Reporting Systems against Unfair Ratings and Discriminatory Behavior," Proc. ACM Conf. Electronic Commerce, pp. 150-157, Oct. 2000.

[8] C. Dellarocas, Building Trust On-Line: "The Design of Reliable Reputation Mechanism for Online Trading Communities". MIT Sloan School of Management, 2001.

[9] R.L. Rivest and B. Lampson, "SDSI: A Simple Distributed Security Infrastructure," Proc. Crypto '96, pp. 104-109, Aug. 1996.

[10] N. Li and J.C. Mitchell, "RT: A Role-Based Trust-Management Framework," Proc. Third DARPA Information Survivability Conf. and Exposition (DISCEX III), Apr. 2003.

[11] Devan and Dasgupta: "P2P Reputation Management using distributed identities and decentralized recommendation chains" IEEE Transactions on Data and Knowledge Engineering, vol. 22, no. 7, July 2010

[12] P. Dewan, "Injecting Trust in Peer-to-Peer Systems," technical report, Arizona State Univ., 2002.

[13] E. Damiani, D. di Vimercati, S. Paraboschi, P. Samarati, and F.Violante, "A Reputation-Based Approach for Choosing Reliable Resources in Peer-to-Peer Networks," Proc. Conf. Computer and Comm. Security (CCS '02). pp. 207-216, 2002.

[14] I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, M.F. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," Proc. ACM SIGCOMM, pp. 149-160, Aug. 2002.

[15] Zhang Q, Sun Y, Liu Z, Zhang X, Wen XZ. "Design of a distributed P2P-based grid content management architecture". In: Ilow J, ed. Proc. of the 3rd Communication Networks and Services Research Conf. New York: IEEE, pp.339−344, 2005.

K.B.Madhuri received M.Tech. degree in Computer Science and Technology from Andhra University in 1999. She obtained Ph.D from JNTU, Hyderabad in 2009. Presently she is working as Professor in department of Information Technology at Gayatri Vidya Parishad, College of Engineering, Visakhapatnam, Andhra Pradesh, India. Her research interests include Data Mining and Pattern Recognition. She published research papers in National and International Journals. She is a member of IEEE and associate member of Institute of Engineers (India).

M. Srikanth received M.Tech degree in Software Engineering from Gayatri Vidya Parishad College of Engineering, Visakhapatnam, Andhra Pradesh, India in 2012. Presently he is working as Project Engineer in Wipro Technologies, Bhubaneswar, Orissa, India. His research interests include Data Mining and computer networks.