ISSN:2320-0790

COMPUSOFT
An International Journal of Advanced Computer Technology

EFFECTIVE RECOMMENDATION CHAINS FOR LARGE SCALE DISTRIBUTED DECENTRALIZED P2P SYSTEMS

Yannam Bharath Bhushan

Email: bharath.yannam@gmail.com

ABSTRACT: The security model used for centralized systems is not suitable for P2P networks as it is centralized in nature. The security challenges in the P2P networks are secure reputation data management, availability of reputation data, Sybil attacks and identity management of peers. In this paper we present a cryptographic protocol for ensuring secure and timely availability of the reputation data of a peer extremely at low cost. We also investigate Reputation Systems for P2P networks more ambitious approach to protect the P2P network without using any central component, and thereby harnessing the full benefits of the P2P network. The results in the form of simulations reveal that the new cryptographic protocol is secure and efficient in a decentralized peer – to – peer network.

1. INTRODUCTION

Peer-to-peer (P2P) computing or networking is a distributed application architecture that partitions tasks or workloads among peers. Peers are equally privileged, Equipotent participants in the application. They are said to form a peer-to-peer network of Nodes. P2P networks are more vulnerable to dissemination of Malicious or spurious content, malicious code, viruses, worms, and Trojans than the traditional client-server networks, due to their unregulated and unmanaged nature. For example, the infamous VBS.Gnutella worm that infected the Gnutella network, stored Trojans in the host machine. Due to ad hoc and decentralized nature of P2P networks, it is extremely difficult to provide security to the network. More over they are spread geographically and they are subject to different laws. The conventional mechanisms used to secure C/S systems are in vain in case of P2P networks for the valid reason specified earlier. Peers make a portion of their resources, such as processing power, disk storage or network bandwidth, directly available to other network participants, without the need for central coordination by servers or stable hosts. Peers are both suppliers and consumers of resources, in contrast to the traditional client–server model where only servers supply, and clients consume.

P2P systems have two main key characteristics:

Scalability: There is no algorithmic or technical limitation of the size of the system, e.g. the complexity of the system should be somewhat constant regardless of number of nodes in the system.

Reliability: The malfunction on any given node will not affect the whole system (or maybe even any other nodes).

Peer-to-peer network are classified into two types

- ✓ Structured Systems
- ✓ Unstructured Systems

Structured P2P networks employ a globally consistent protocol to ensure that any node can efficiently route a search to some peer that has the desired file, even if the file is extremely rare. Such a guarantee necessitates a more structured pattern of overlay links. By far the most common type of structured P2P network is the distributed hash table, in which a variant of consistent hashing is used to assign ownership of each file to a particular peer, in a way analogous to a traditional hash table's assignment of each key to a particular array slot

An unstructured P2P network is formed when the overlay links are established arbitrarily. Such networks can be easily constructed as a new peer that wants to join the network can copy existing links of another node and then form its own links over time. In an unstructured P2P network, if a peer wants to find a desired piece of data in the network, the query has to be flooded through the network to find as many peers as possible that share the data. The main disadvantage with such networks is that the queries may not always be resolved. Popular content is likely to be available at several peers and any peer searching for it is likely to find the same thing. But if a peer is looking for rare data shared by only a few other peers, then it is highly unlikely that search will be successful. Since there is no correlation between a peer and the content managed by it, there is no guarantee that flooding will find a peer that has the desired data. Flooding also causes a high amount of signaling traffic in the network and hence such networks typically have very poor search efficiency. Many of the popular P2P networks are unstructured.

The difficulty of securing P2P networks can be greatly mitigating by utilizing services of a CA (Certificate Authority) which is centralized again. The drawback of a centralized authority is that if the authority is compromised, it itself can spoil the whole P2P network. At the same time without its presents, no magic wand is present to ensure security to P2P networks.

In this paper, investigation is made on P2P networks and their reputation systems. A new approach is invented without making use of a centralized authority besides enjoying all the benefits of a P2P network. Peers are estimated whether they are good or malicious based on their reputations. The malicious peers are separated from good peers soon after detecting them. Malicious activities are significantly reduced by eliminating malicious nodes peers from the network. Identity certificates are used to identify all peers in the network. Such certificates are self certified and all peers are like certificate authorities as they have their own CA which issues certificates. Each and every node has its history pertaining to reputation management. When a transaction takes place between two peers, the two-party cryptographic protocol helps in secure exchange of reputation information between peers. The experiments resulted in providing evidence that the proposed infrastructure for reputation management can greatly reduce the percentage of malicious transactions over P2P networks. The significant contributions of this paper are:

- ✓ A simple and light weight reputation model.
- ✓ Cryptographically blind identity mechanisms are used to arrive at a self-certification based identity system.
- ✓ Generation of an authentic global reputation information of a peer with the help of an attach resistant cryptographic protocol

2. RELATED WORK

This section briefly reviews some of the existing P2P reputation systems, focusing particularly on the storage and integrity issues. We start by giving an overview of the reputation systems. Kevin A. Burton designed the open privacy distributed reputation system on p2p, which is derived from the distributed trust model. It proposed the concept of reputation network, which is composed by identities and certificates. Therefore, the trustworthiness of the identities can be estimated from a visible sub-graph of the reputation network. P2P REP et.al. Is a reputation sharing protocol proposed for Gnutella, where each peer keeps track and shares with others the reputation of their peers. Reputation sharing is based on a distributed polling protocol. Service requesters can access the reliability by polling peers. Karl Aberer et.al. Proposed a trust managing system on the P2P system. It integrates the trust management and data management schemes to build a full-fledged P2P architecture for information systems. The reputations in this system are expressed as complaints; the more complaints a peer gets, the less trustworthy it could be. After each transaction, and only if there is dissatisfaction, a peer will file a complaint about the unhappy experience. To evaluate the reputation of a peer involves searching for complaints about the peer.

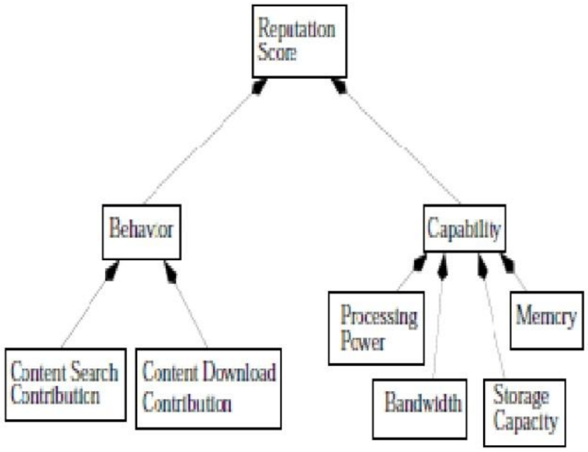
Kamvar et.al proposed a reputation management system, for P2P file sharing systems such as Gnutella to combat the spread of inauthentic file. In their system, each peer is given a global reputation that reflects the experiences of other peers with it. Sit and Morris present a framework for performing security analyses of p2p networks. Their adversarial model allows for nodes to generate packets with arbitrary contents, but assumes that nodes cannot intercept arbitrary traffic. They then present taxonomy of possible attacks. At the routing layer, they identify node lookup, routing table maintenance and network partitioning / virtualization as security risks. They also discuss issues in higher-level protocols, such as file storage, where nodes may not necessarily maintain the necessary invariants, such as storage replication. Finally, they discuss various classes of denial-of-service attacks, including rapidly joining and leaving the network, or arranging for other nodes to send bulk volumes of data to overload a victim's network connection (i.e., distributed denial of service attacks).

Dingle dine et al. and Douceur discuss address spoofing attacks. With a large number of potentially malicious nodes in the system and without a trusted central authority to certify node identities, it becomes very difficult to know whether you can trust the claimed identity of somebody to whom you have never before communicated. Bellovin identifies a number of issues with Napster and Gnutella. He discusses how difficult it might be to limit Napster and Gnutella use via firewalls, and how they can leak information those users might consider private, such as the search queries they issue to the network. Bellovin also expresses concern over Gnutella's "push" feature, intended to work around firewalls, which might be useful for distributed denial of service attacks. He considers Napster's centralized architecture to be more secure against such attacks, although it requires all users to trust the central server. It is worthwhile mentioning a very elegant alternative solution for secure routing table maintenance and forwarding that we rejected. This solution replaces each node by a group of diverse replicas as suggested by Lynch et al. The replicas are coordinated using a state machine replication algorithm like BFT [14] that can tolerate Byzantine faults. BFT can replicate arbitrary state machines and, therefore, it can replicate Pastry's routing table maintenance and forwarding protocols.

Cornelli et al. propose a reputation-based approach for P2P file sharing systems (called P2PRep). In P2PRep, a peer pools other peers by broadcasting a request about the opinion of the select peer. Damiani et al. present a similar approach, called XRep, which considers the reputations of both peers and resources. P2PRep and XRep do not give any metrics to quantify the credibility's of voters. Also, they only can find malicious peers within a given horizon. Our approach involves an adaptive process of neighbor selection, which may help to detect malicious peers who are originally beyond the horizon.

Indexing and resource discovery

Older peer-to-peer networks duplicate resources across each node in the network configured to carry that type of information. This allows local searching, but requires much traffic. Modern networks use central coordinating servers and directed search requests. Central servers are typically used for listing potential peers, coordinating their activities and searching. Decentralized searching was first done by flooding search requests out across peers. More efficient directed search strategies, including super nodes and distributed hash tables are now used. Many P2P systems use stronger peers (super-peers, super-nodes) as servers and client-peers are connected in a star-like fashion to a single super-peer.



R-chain

It is lightweight reputation management system R-Chain where each peer maintains its own transaction history as the reputation. Each transaction in R-Chain involves two equal parties and use file downloading as the example. Each transaction will result in a transaction record (TR) as the proof of its existence R-chain minimizes the maintenance and retrieval cost by maintaining the transaction history on the owner node.

Sybil attack

If a single faulty entity in a P2P system can present in multiple identities it can control a substantial fraction of the system thereby undermining this redundancy. Sybil attacks are always possible except under extreme and unrealistic assumptions of resource parity and coordination among entities.

Reputation system

In decentralized unstructured P2P networks like gnutella content retrieval involves a content search phase and content download phase. To search the desired content a peer generates query appropriate keywords and sends it to all peers that it is directly connected to in the gnutella overlay topology. The peers who process this query reply back if they have the content in their shared directory and forward the request to the peers they are directly connected to depending on the TTL (time-to-live) of the query. This forwarding continues until the TTL specified by the querying peer is exhausted. Once the querying peer receives all the replies it selects a peer to download the content from.

Trust

Trust is a social phenomenon. Any artificial model of trust must be based on how trust works between people in society.1) Assists users in identifying trustworthy entities and 2) Gives artificial autonomous agents the ability to reason about trust.

Dynamic trust management

Dynamic trust management encapsulates trust management in dynamic distributed environments, where the members of the system assume frequently changing multiple roles. In addition the members themselves are transitory.

In this paper, we investigate Reputation Systems for P2P networks—a more ambitious approach to protect the P2P network without using any central component, and thereby harnessing the full benefits of the P2P network

3. REPUTATION MANAGEMENT

In P2P systems peers form ratings of others that they interact with. To evaluate the trustworthiness of a given party, especially prior to any frequent direct interactions, the peers must rely on incorporating the knowledge of other peers—termed witnesses—who have interacted with the same party using reputation mechanisms. In our framework, each peer has a set of acquaintances, a subset of which are identified as its neighbors. The neighbors are the peers that the given peer would contact and the peers that it would refer others to. A peer maintains a model of each acquaintance. This model includes the acquaintance's reliability to provide high-quality services and credibility to provide trustworthy ratings to other peers. More importantly, peers can adaptively choose their neighbors based on the average of local ratings, which they do every so often from among their current acquaintances, e.g., every 5 queries for a peer.

3.1Threat Model

In P2P systems, peers connect and leave with insecure communication channels. Peers may have conflicting interests and malicious intentions as well. Malware can also be spread by rogue peers. Peers should be in a position to judge the genuineness of content before involving in transactions. To achieve this goal a perfect mechanism and reputation system is needed. Ballot stuffing and bad mouthing are results of an imperfect reputation system. It all depends on building reputation system to improve reputation mechanism and prevent peers from indulging malicious activities.

3.2 Self Certification

Every peer should have a handle or identity for participating in reputation system. Based on the recommendations received by a peer to participate in transactions, its reputation is calculated. In a decentralized P2P network, as there is not central authority to issue certificates, each node can generate a certificate and thus act as CA (Certificate Authority). Reputations are associated with identities and in turn the combined reputation of all identities comprises the reputation of CA. An attack by name Sybil can cause a peer to misuse self certification by generating so many identities and thus increasing its reputation. This problem can be prevented by restricting a peer to have only one identity or mapping all identities generated by it to its real life identifier.

There is another problem with CA. A malicious peer can generate multiple CAs and then multiple groups of identities. This can be countered by keeping peers divided into groups. Each peer attaches its group certificate and associates it with its CA.

When a group authority receives blinded credentials of a peer, the authority signs the group certificate after verifying the credentials. However, the authority keeps track of information that can be correlated to certificates of peers. The reputation system is developed in such a way that any peer that involves in malicious practice to improve its reputation will be self destructive as its reputation will really go down. Mathematically P is used to denote peer and A is used to denote authority while Pk2 represents the peer's private key and Pk1 represents the public key of the peer P. Ek(T) represents encryption of the phrase (T) key k. The blinding phrase X with key K is represented by EBk(X).

1. $P \rightarrow A$: $B1 = \{ EB_{Ka}(I_{Alice\ r})\}, I_{Alice}$

The peer Alice generates a BLINDING KEY, K_a and another identity for herself ($I_{Alice\ r}$). Alice cannot be identified from her identity ($I_{Alice\ r}$). Subsequently, she blinds her identity ($I_{Alice\ r}$) with the blinding key K_a. B1 represents the blinded identity. Alice sends B1 to the authority with her real identity that proves her membership to a group.

2. $A \rightarrow P$: $B2 = E_{PAuthority\ k2}\ \{ B1 = EB_{Ka}(I_{Alice\ r})\}$

The authority signs the blinded identity, B1 and sends it (B2) back to the peer.

3. P: $E_{PAuthority\ K2}\{I_{Aclice\ r}\} = \{ EB_{Ka}\{B2\}\}$

The peer unbinds the signed identity and extracts the identity authorized by the authority $E_{PAuthority\ K2}\{ I_{Alice\ r}\}$.

In this approach peers are interested in ranks of the prospective providers. This concept was inspired by Google page rank. When genuine recommendations come from peers in the network, this approach can be argued to be unfair as our experiments revealed that minimal change is there with the ranks of providers.

3.3 Reputation Model

A peer joins the P2P network once it gets identity and then it searches using search method for one or more files. It generates a list of peers who have requested files based on the response to the search. RANGE denotes such peers (providers). A cryptographic protocol (explained in the next section) is initiated by the requester with a peer who has highest reputation. The file is downloaded by requester from provider and its quality, authenticity and integrity are verified. Based on the results, recommendations are sent to the provider. It will be between MIN_RECOMMENDATION and MAX_RECOMMENDATION. Then the provider's overall reputation is recalculated. This process is repeated for every transaction

3.4 Reputation exchange protocol

Once the requester has selected the provider with the highest reputation, it initiates the reputation exchange protocol with the provider. In the reputation exchange protocol, the requester is denoted by R while the provider is denoted by P. Here R→P: X denotes that the requester (R) sends a message X to the provider (P). The symbol P_{k2} represents the private key of the peer P and P_{k1} represents the public key of the peer P. E_K ( ) represents the encryption of the phrase ( ) with key K, H(λ) denotes one way of hash of the value of the λ. This protocol assumes only insert & search methods are available and they are resilient to peers that may not follow the recommended join & leave protocol of the network.

Step 1: R →P: RTS & IDR

Then requester sends a REQUEST FOR TRANSACTION (RTS) and its own IDENTITY CERTIFICATE (IDR) to the provider. Provider needs this identity to show the future requesters.

Step 2: P→R: IDP & TID & EP_{k2} (H (TID‖RTS)

The provider sends its own IDENTITY CERTIFICATE (IDP), the current TRANSACTION ID (TID) and signed TID, EP_{k2} (H (TID‖RTS). The signed TID is needed to ensure that the provider does not use the same transaction id again. End of this protocol same TID will be signed by the requester also and stored in network.

Step 3: R: LTID=Max (Search (P_{k1} ‖TID))

The requester obtains the value of the LAST TRANSACTION ID (LTID) that was used by the provider from the network. The requester concatenates the providers' public key with the string TID and performs the search. Peers having TID for the provider replies back with the TID and requester selects the highest TID out all the received TIDs. The highest TID becomes LTID. LTID and related information will be signed by the requester so that provider cannot play foul.

Step 4: R: IF (LTID≥TID) GO TO step 12

If the value of LTID found by the requester from the network is greater than or same as the the TID offered by the provider, it implies that the provider has used the TID in some other transaction. Hence it is trying to get transaction number (TID). The requester founds foul play and jumps to step 12.

Step 5: R→P: Past Recommendation Request & r

If the check in step 4 succeeds, requester is sure that the provider is not using the same transaction number, it request the provider for the next recommendations. In other words , if the current transaction is the Nth transaction for the provider , the requester makes the request for N-1th, N_2th and so on recommendations till N-rth recommendation where r is less than N. the value of the r is decided by the requester and it is directly proportional to the requesters stake in the transaction.

Step 6: P→P: CHAIN, EPk2 (CHAIN)

CHAIN= ({REC_{N-1}‖E_{ZN-1K2} (H (REC_{N-1}))}‖
{REC_{N-2}‖E_{ZN-2K2} (H (REC_{N-2}, REC_{N-1}))}‖
{REC_{N-3}‖E_{ZN-3K2} (H (REC_{N-3}, $REC_{N-2)}$)}‖…..
{REC_{N-4}‖E_{ZN-4K2} (H (REC_{N-r}, $REC_{N-r-1)}$))})

The provider sends its past recommendations (REC $_{N-1,}$ REC $_{N-2}$,… REC $_{N-3}$) which were provided by the peers (Z $_{N-1,}$ Z $_{N-2}$,….Z $_{N-3}$). The provider signs the chain so that the requester can hold the provider accountable for the chain. The Provider could not have maliciously changed because recommendations have been signed by previous requesters. There is no way the provider can modify the CHAIN.

Step 7: Result=Verify (REC_{N-1}, $RECN_{-2}$ ….REC_{N-r})

If Result not verified GO TO STEP 12

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. Incase if it does not have the required certificates, it obtains the certificate from the provider itself. The provider obtained its requester's certificate in step 1. The requester checks for false recommendation. If verification fails the requester jumps to step 12.

Step 8: P→R: File or Service

The provider provides the service or the files as per the requirement mentioned during the search performed for the providers.

Step 9:

$R \rightarrow P := EB_{Ka}(REC \| TID \| E_{RK2}\{REC \| TID)\})$

If the requester receives a service, it generates a BLINDING KEY, Ka. The requester concatenates the RECOMMANDATION (REC) and the TRANSACTION ID (TID) it had received in the step 2 and signs it. Subsequently, it blinds the signed recommendation with blinding key Ka. The provider receives the blinded recommendation from the requester. The blinded recommendation also signed by the requester. The blinded recommendation contains the Chain that the provider can subsequently use to validate its reputation to another requester.

Step 10:

a) $P \rightarrow R$: $B1 \| E_{PK2}(H(B1), nonce), nonce$

b) $R \rightarrow P$: Ka

The provider cannot see the recommendation but it signs the recommendation and sends the NONCE and the signed recommendation back to the requester. The requester verifies the signature and sends blinding key Ka to the provider which can unbind the string received in step 10a and checks its recommendation.

Step 11:

Insert $(IDR, \{REC \| E_{RK2}\{H(REC) \| H(TID)\}\})$

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 using the insert method of the P2P network. This completes the transaction.

Step 12:

R:Insert

$(IDR, \{CHAIN \| TID \| E_{RK2}\{H(CHAIN) \| H(TID)\}\})$

It explains when it expects foul play ABORT PROTOCOL. If the verification in step 7 fails, the requester takes the CHAIN that was signed by the provider and the transaction id (TID), signs it and uses the INSERT method of the network. As a result any subsequent requester will be able to see failed verification attempt and will assume aMIN_RECOMMANDATION recommendation for the TID of the provider. The requester cannot insert fake recommendations into the network because it has to include the TID signed by the provider. If the requester reaches step 12 from step 4 .it will request for the Chain from the provider subsequently will perform R: Insert $(IDR, \{CHAIN \| TID \| E_{N-K2}\{H(TID \| RTS))\}\})$.

3.5 Features of the Protocol

The main features of the protocol are as follows:

The genuine global reputation information with respect to a provider is obtainable to all peers at one place. The provider will not start several search requests in the network with the purpose of gathering the suggestion got by the supplier in the previous. It has to concern one search appeal to regain the last operation information of the provider it also confirm every proposals of the supplier. It decrease the turnaround time of the transaction but also keep significant volume of resource.

The provider is liable to every older transaction. It cannot spitefully interfere with transaction records by addition or deletion of proposal because the proposals are attached in a series and noticed by the earlier supplicant. The provider can't modify proposals because they are digitally signed by the requesters.

The total information of the provider is saved by the provider itself. The protocol will not have an effect by unreliable accessibility of previous recommenders or other peer in the network. The transaction can be finished productively as long as the requester and the provider connected to the network.

4. EXPECTED RESULTS

This paper presents self-certification, an identity management and mechanism, reputation model, cryptographic protocol that facilitates generation of global reputation data in a P2P network in order to expedite the detection of rogues. The self-certification based identity generation mechanism reduces the threat of liar farms by binding the network identity of a peer to his real-life identity while still providing him anonymity. The global reputation data are protected against any malicious modification by the third party peer and are immune to any malicious modifications by their owner. The proposed protocol reduces the number of malicious transactions and consumes less bandwidth per transaction.

Also this paper presents a queuing model to evaluate the latency associated with file transfers or replications in peer-to- peer (P2P) network. The main contribution is a modeling framework for the peers that accounts for the file size distribution, the search time, load distribution at peers, and number of concurrent downloads allowed by a peer.

5. CONCLUSIONS AND FUTURE WORK

The file searching is efficient and easy in unstructured peer to peer networks based on Reputation Management. During the transfer of files there is possible of distribution of viruses, worms and Trojan horses and malicious peers to over come this the self certification (RSA and DSS) is used, it provides authentication and

Authorization. It easily finds the malicious peers and aborts the transaction. Therefore the proposed method provides the efficient and secure communication between the peers. Bandwidth per Transaction and the number of malicious transactions are reduced by the proposed protocol. The highly probable erratic availability of peers problem is also handled by the protocol. The present system considers the reputation of provider while ignoring the reputation of requester. It can be improved further in order to consider reputations of both requester and provider. More over the reputation values can be updated as per the context of the reputation

6. REFERENCES

[1] J. Douceure, “The Sybil Attack”, Proc, IPTPS ’02 Workshop, 2002.

[2] A. Abdul-Rahman and S. Hailes, “Supporting Trust in Virtual Communities”, Proc. Hawaii Intl Conf. System Sciences, Jan 2000.

[3] M. Gupta, P. Judge, and M. Ammar, “A Reputation System for Peer-to-Peer Networks,”Proc. 13th Int’l Workshop Network and Operating systems Support for Digital Audio and Video (NOSSDAV) , 2003.

[4] L. Liu, S. Zhang, K.D. Ryu, and P. Dasgupta, “R-Chain: A self Maintained Reputation Management System in p2p Networks,” Broc. 17th Int’l Conf. Parallel and Distributed Computing Systems (PDCS), Nov. 2004.

[5] Prashant Dewan and Partha Dasgupta,” P2P reputation Exchange Protocol Using Distributed and Decentralized Recommendation Chains, IEEE Transaction on Knowledge and Data Engineering vol 22, No.7, July 2010.

[6] H.Garett, “Tragedy of Commons,” Science, vol.162, pp.1243-1248, 1968.