IMPLEMENTATION OF REPUTATION EXCHANGE PROTOCOL IN PEER-TO-PEER SYSTEM

B.Udhaya
Department of Information Technology
SRM University

J.Jeysree
Department of Information Technology
SRM University

ABSTRACT

The motivation behind basing applications on peer-to-peer architectures derives to a large extent from their ability to function, scale and self-organize in the presence of a highly transient population of nodes, network and computer failures, without the need of a central server and the overhead of its administration. P2P networks are vulnerable to peers, who cheat, propagate malicious codes, or peers who do not cooperate. Traditional client-server security models are not sufficient to P2P networks because of their centralized nature. Absence of central authority in P2P poses unique challenges like identity management of the peers, secure reputation data management and Sybil attack for reputation management in the network. 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. The past behavior of the peer is encapsulated in its digital reputation and is subsequently used to predict the future actions. The cryptographic protocol is coupled with self-certification and cryptographic mechanisms for identity management and countering Sybil attacks. The latency associated with a file replication in a P2P system consists of two components: the query search time and the time required by the peers to transmit the file. In order to model the peer level latency, we develop a queuing model to evaluate the time required at each peer to serve its replication requests.

INTRODUCTION

Peer-to-Peer systems are self-configuring networks with minimal or no central control comparing to traditional client- server networks P2P networks are vulnerable to viruses ,worms and spurious contents. Because there is no regular management of activities in each peer.eg: Gnutella worm. Tragedy of commons [1] shows that peers in the P2P networks are not encouraged from leeching on the network. Methods like generating trust and protecting client-server networks cannot be used for pure P2P networks. This is because trusted central authority is not used in the P2P networks.

Introducing Certificate Authority (CA) will reduce the difficulty of securing P2P networks. We investigate reputation system for P2P networks to protect the P2P network without using any central component. Reputations of each peer used to determine whether the peer is malicious peer or good peer. Malicious peers are ostracized from the network so that good peers do not perform any transactions with the malicious peers. Excluding malicious peers from the network will reduce the amount of malicious activities.

All peers in the P2P network are identified by identity certificates. Reputation of the peer is attached to its identity. Identity certificates are generated using self-certification method, and all peers maintain their own identity certificate authority which issues the identity certificate to the peer. Each peer owns the reputation information pertaining to all its past transactions with other peers in the network and stores it locally.

A two party cryptographic protocol not only protects the reputation information between the two peers participating in the transaction. Proposing technique not only reduces the percentage of malicious transactions in the network but also significantly reduces the network traffic compared to other reputation systems.

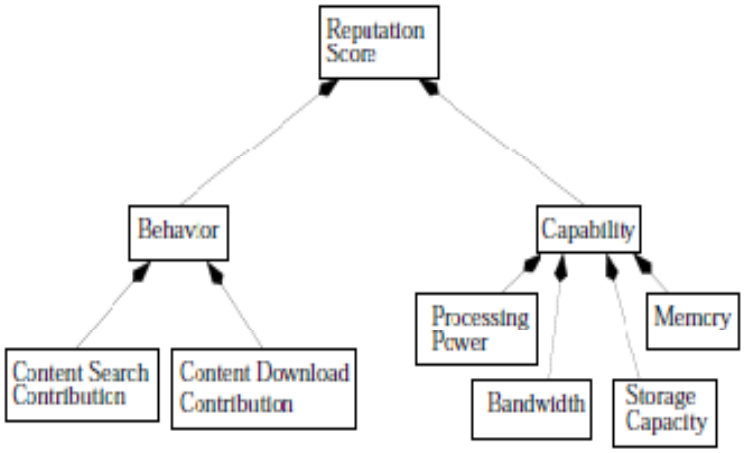
Advantages of this paper are

1. A self-certification based identity system using cryptographically blind identity mechanisms.
2. A light weight and simple reputation model.
3. Attack resistant cryptographic protocol for generation of authentic global reputation information of the peer.
4. The latency associated with a file replication in a P2P system. The peers accounts for the file the size distribution.
5. The search time.
6. Load distribution at peers.

2. RELATED WORK

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. At that point the content download typically uses a HTTP or a TCP connection.

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.

IDENTITY MANAGEMENT

P2P NETWORKS

In Gnutella, a peer is identified by its servant id that is a function of its IP address. IN the DHT based system, like chord, CAN, and Pastry the peers are assigned identifiers on the basis of their location in the network. All the above systems predominantly use the peer identifiers for locating content and not for evaluating reputation or for enforcing security. The identifier allocation strategies in these networks are able to meet their objectives. However these allocation strategies are not apt for a reputation-based system (on a P2P network) because they do not restrict the number of identities a peer can possibly generate.

3. REPUTATION SYSTEM

3.1 Threat Model

In Gnutella-like network Peers follow predefined join and leave protocols. The peers are connected to insecure communication channels. Concept is mainly to reduce leechers. Leechers are peers who derive benefits from the system without contributing to the system. Peers need to judge the quality of the content before making Go/No-Go decision in transactions and thereby develop trust relationships with other peers. A perfect reputation system needs achieve the above goals.

3.2 Self-Certification

In order to participate in the reputation system, a peer needs to have a handle. The reputation of the peer is associated with its handle. This handle is commonly named as the 'identity' of the peer even though it may not "identify" a peer. A peer receives a recommendation for each transaction performed b it, and all of its recommendations are accumulated together to the calculation of the reputation of the peer.

A malicious peer can use self-certification to generate a large number of identities and thereby raise the reputation of one of its identities by performing false transactions with other identities. The malicious peer does not even need to collude with other peers to raise its reputation, but only needs to generate a set of identities for itself. Such a large set of identities managed by one peer is called identity farm. The set of identities that issue false recommendation is called a liar farm.

Each peer runs its own CA that issues the identity certificates to the peer. Peer is denoted by P while the authority is denoted by A. Here P→A: X denotes that the peer (P) sends a message X to the authority (A). Te 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 encryption of the phrase ( ) with key K, while $EB_K(X)$ represents blinding phrase X with key K.

1. P→A: B1= $\{EB_{Ka}(I_{Alice\ r})\}, I_{Alice}$
   The peer Alice generates a BLINDING KEY, Ka 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 Ka. B1 represents the blinded identity. Alice sends B1 to the authority with her real identity that proves her membership to a group.
2. A→P: B2 = $E_{PAuthorityK2}\{B1=EB_{Ka}(I_{Alice\ r})\}$
   The authority signs the blinded identity, B1 and sends it (B2) back to the peer.
3. P: $E_{P\ AuthorityK2}\{ I_{Alice\ r} \} = \{EB_{Ka}\{B2\}\}$
   The peer unblinds the signed identity and extracts the identity authorized by the authority $E_{P\ AuthorityK2}\{I_{Alice\ r}\}$.

REPUTATION MODEL

Once the peer is obtained its identity, it joins in the P2P network using join method of that network. Peer searches for one or more files using the search method provided by the network. If peer have the response corresponding to that the particular peer is responded. The number of peer who offers a particular file is denoted by RANGE. The requester selects the highest reputation peer from the list then initiates the protocol sends the recommendation cryptographic protocol. Depending on the verification MIN_RECOMMENTATION and MAX_RECOMMANDATION is given to the provider. The recommendation have constrains that one recommendation completely nulls or improves the reputation of the any peer in the network. The proposed model is independent of the topology of the P2P.

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},REC_{N-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→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→R: B1 ‖ E_{PK2}(H(B1),nonce),nonce

b) R→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 unblind 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 a MIN_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$\|E_{N\text{-}RK2}${H(TID$\|$RTS))}}).

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.

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 Recommandation 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.
[7] R.Zhou, K.Hwang, and M. Cai, " Gossiptrust for Fast reputation Aggregation in Peer-to-Peer" *IEEE Trans. Knowledge and Data Eng.*,vol.20,no. 9, pp. 1282-1295, Aug. 2008.
[8] L.Xiong and L. Liu, "PeerTrust: Supporting Reputation-Based Trust in Peer-to-PeerCommunities", *IEEE Trans. Knowledge and Data Eng.*, vol.16,no. 7, pp. 843-857, July 2004.
[9] B.C.Ooi, C.Y.Kiau, and K.Tan, "Managing Trust in Peer-to-Peer Systems Using Reputation-Based Techniques," Proc .Fourth Int'l Conf.Web Age Information Management, Aug. 2003.