False Proof Reputation Management for P2p Networks

V.Thejaswini Reddy
M.Tech
Department of CSE
Jntuace, Anantapur
India

B.Lalitha
Assistant Professor
Department Of Cse
Jntuace, Anantapur
India

ABSTRACT

The wide qualities of peer-to-peer (P2P) network have given us many advantages and threats for enhancement of distributed computing. The best way to reduce threats is adding a reputation-based globally trusted model. Many present trust models are failing to restrain effectively some behaviors like collusive attacks, but pay no heed towards the security of this mechanism.

General Terms

Security, Reputation system, Reputation Exchange Protocol.

Keywords

Peer-to-Peer Networks, Distributed System, Reputation-Based, Trust model, Security.

1. INTRODUCTION

Off late, pee-to-peer (P2P) computing has become popular and well recognized in a wide range of applications, like file-sharing, digital content delivery, and P2P computing [1-2]. But the fact remains that, peer anonymity and autonomy make P2P networks easy towards attacks by any peer who is not rust worthy. The recent works [3-4] are a benchmark to the fact that the trust theories in social networks construct well recognized trust models, to find a solution for these kinds of behaviors.

The present reputation-based trust model designs trusted rank of a peer based on its past transactions, and its similar to the peer with full trust value is offered the role of service provider. This method has some advantages on some malicious behaviors to a certain extent, but has a meager effect when it comes to complex attacks and when disturbances are created on these reputation systems, like collusions. The researches now a day's focus on the design and working of the trust system in all practical arenas, and hardly care about the security problem it faces which can damage the tag "reputation management". The security of reputation management is the most important element which assures safe working of the trust management system (TMS). Thus, it is vital to develop and discuss about the security mechanism of the TMS.

Dealing with these research problems, we present a reputation based distributed P2P trust model with the security mechanism for the reputation information management (DSRM), for P2P networks.

2. RELATED WORK

This sector gives a wide review of some of the present P2P reputation systems, concentrating on problems like storage and integrity. We would like to initially give an overview of the reputation systems. Kevin A. Burton designed an open privacy distributed reputation system [5] on p2p, which hails from the distributed trust model which bought to us the idea of reputation network, which is made up of identities and certificates. Hence, the belief of the identities is valued from a visible sub-graph of the reputation network. P2PREP et.al [6] which is a reputation sharing protocol designed for Gnutella, where every peer keeps track and shares the reputation of their peers. Reputation sharing is made by distributed polling protocol. Service requesters use this trust by polling peers.

Karl Aberer et.al. [7] Made a trust managing system on the P2P system which combines the trust and data management to construct a complete P2P architecture for information systems. The reputations here are expressed as complaints; higher the complaints, less trustworthy it is. After every transaction, if there is dissatisfaction, a peer files a complaint stating the problem. To examine the reputation of a peer involves searches for complaints about the peer. Kamvar et.al [8] proposed a reputation management system, for P2P file sharing systems such as Gnutella fighting against the spread of inauthentic file. Here, every peer has a global reputation that shows experiences of every peer with it. Stocia and Morris et.al [9] gave an idea for security of p2p networks.

Their model permits nodes to make packets with arbitrary material, but lets the nodes not to intercept arbitrary traffic. They gave taxonomy of all varied attacks and at the routing layer, they find a node lookup, routing table maintenance, and network partitioning / virtualization as threat to security. They deal also with multilevel protocols, like file storage, where nodes need not have the necessary invariants, like storage replication. They work also on denial-of-service attacks, and rapidly joining and leaving the network, or arranging for various nodes which sends bulk volumes of data to overload a victim's network connection (i.e., distributed denial of service attacks).Douceur [10] work on address spoofing attacks also. Having many potentially dangerous nodes in the system and with no trusted central head which certifies node identities and become complex to know whether you can trust the claimed identity of somebody to an unknown. Bellovin [11] finds many problems with Napster and Gnutella.

He discusses how complex it is to extent the use of Napster and Gnutella use via firewalls, and the ways they pass information that users feel is personal, like the search queries given. Bellovin researches also on Gnutella's "push" feature, which functions on firewalls, useful for denial of service attacks. He feels Napster's central architecture more safe against these kind attacks, even if it needs users to trust the central server. It is to be noted that an alternative answer for secure routing table maintenance and forwarding that we denied. This answer exchanges every node by a bunch of replicas as told by Lynch.

The replicas are run using a state machine replication algorithm like BFT that can sustain faults like Byzantine. BFT can replicate arbitrary state machines and, therefore, it can look like Pastry's routing table maintenance and forwarding protocols. Here, we look into Reputation Systems [12] for P2P networks—highly useful design which protects the P2P network without a central component, and amplifies all the advantages of the P2P network.

3. REPUTATION SYSTEM

A vital corollary of a good reputation management is the online auction system eBay [13]. Here, buyers and sellers rate each other post transaction, and the final reputation of a contestant is the ratings he has over the last 6 months. This system depends on a central system to store and manage these ratings.

In varied areas peers rate each other post transaction, like in eBay system. Like, every time peer I gets a file from peer j, it rates the transaction as positive (tr (i, j) = 1) or negative (tr (i, j) = −1). Peer i can rate a download as negative, if he finds the file inauthentic or tampered with, or if interrupted. Like in the eBay model, we may define a local trust value s_{ij} as the sum of the ratings of the individual transactions that peer i has downloaded from peer j: $s_{ij} = \Sigma \text{ tr }_{ij}$.

Similarly, every peer i can store many transactions it has had with peer j, sat (i, j) and the number of unsatisfactory transactions it has had with peer j, unsat (i, j) Then, sij is defined:

S_{ij} = sat (i, j) – unsat (i, j)

Previous work in P2P reputation systems are based on same notions of local trust values. The obstacle is in an environment is how to deal with the local trust values s_{ij} without a central storage and management. Every previous system named above finds this problem; every system proposed has a couple of negatives. It mostly averages the ratings of some peers and has no wide view about a peer's reputation, or it averages the ratings of the peers and congests the network with system messages questioning for every peer's local trust values for each query.

3.1 Threat Model

A Gnutella-like network has a power-law topology and helps Insert and Search techniques. The peers have a predefined Join & Leave protocols. The peers are connected with a communication channel which is not secure. As the peers have opposing interests, a motivation is required to decrease leechers. Leechers are the ones who gain benefit from the system without giving anything to the system. The rogue peers send malware in the network. Finally, peers judge the quality before making Go/No-Go in every transaction and develop trust relationships mutually.

A good reputation system gives the way to achieve the target. Any reputation system is open to ballot stuffing and bad mouthing as told in [14].A poor reputation system naturally gives problems that exploit the attackers. Peers should have unique way to handle to which their reputations are tagged. If they are absent in trusted central agency, an attacker gathers infinite identities and gives recommendations to itself. A peer can alter the reputation data in the network to uplift its reputation and there are problems that are in the picture based on how a given reputation system is made. We discuss those problems and their mitigation in the sections where the design decision is made.

3.2 Self-Certification

To participate in the reputation system, a peer should have a handle. The reputation of a peer is represented with handle. This handle is the "identity" of the peer even if does not "identify" a peer, i.e., it may not lead to the real-life identity of the peer. A peer gets advices for every transaction, and all advices are stored together for calculation of the reputation of a peer.

In a central system, the head gives these identities. In a decentral reputation system, self-certification [15] divides the trusted entity among the peers and gives their own identities. Every peer has its own CA that gives the identity certificate(s) to the peer. All the certificates used here are same to SDSI certificates [16]. The name of a peer is with its identity and the reputation of a CA is the reputation.

3.3 Reputation Model

The standard Join methodology is made use of by peer to connect itself to a specific P2P network. The search appeal entails the peer supplicant to produce a list of peers who have the demanded file(s) with them. RANGE indicates the count of peers who tender a mentioned meticulous file. The peer supplicant chooses the provider with the peak status by instigating the cryptographic procedure which involves the peer supplicant making use of the Download methodology of the network for downloading the relevant file mentioned by the supplicant, which again assists in validating the reliability, authenticity and the value of the file. A suggestion is then sent to the peer supplicant between MIN_RECOMMENDATION and MAX_RECOMMENDATION, which are limited to the margins ensuring that a single suggestion doesn't entirely annul or radically improve the meticulousness of a supplicant. On receiving the suggestions from the supplicant, it averages the earlier received suggestions and incorporates the recently received ones to estimate its repute.

The factors mentioned above can be assigned values by the means of Decision Theory, Game Theory, and Probability and function F( ) is identified on the basis of intensity levels of menace faced by peers in the P2P network. The function F ( ) in this paper is described as the arithmetic average of the suggestions that are collected by the peer supplicant. The recommended reputation copy is self governing as compared to topology of the P2P network, nodal addressing formats, bootstrap procedures, joining and leaving protocols of the peers present and the name service.

3.4 Contract Signing Between Peers: A Signcryption Approach

The entire process starts here with the employment of RSA signature algorithm [17] otherwise known as Signcryption. Here, the 1st user splits his private key d into d1 and d2 such that d=d1+d2 mod Φ (n), by following park et al.did in [18].The signature of this user has to be exchanged with the other and this signature is $\sigma_A = h(m)^{d1} \bmod n$. The partial signature generated by the 1st user is to assure that he has zero-knowledge base and this is done by Gennaro protocol[19].The connections we have are unreliable due to network failure or router's attacks. But, TTP is reliable since the messages inserted reach the destination for sure but with some delay.

3.4.1 Registration Protocol

The receiver of the information has only to register i.e. only the registration of the initiator with TTP is enough. He then gets a long-term voucher along with CA. After this, the following processes are done: (for our convenience, let the sender be BOB and receiver as ALICE.)

i. Alice first sets an RSA modulus $n = pq$, where *p* and *q* are two -bit safe primes, i.e., there exist two primes p' and q' such that $p = 2p'+1$, $q = 2q'+1$. After, Alice selects her random public key $e \in_R \square^{*}_{\phi(n)}$, and calculates her private key $d = e^{-1} \bmod \phi(n)$, where $\phi(n) = (p-1)*(q-1)$. At last, Alice registers her public key with a CA to get her certificate C_A, which binds her identity and the corresponding pubic key (n,e) together.

ii. Alice randomly splits d into $d1$and $d2$ such that $d = d1 + d2 \bmod \phi(n)$ by choosing $d1 \in_R \square^{*}_{\phi(n)}$, and computes $e_1 = d_1^{-1} \bmod \phi(n)$. She also generates a sample message-signature pair (ω, σ_ω), where $\omega \in \square^{*}_{n} \setminus \{1,-1\}, ord(\omega) \geq p'q'$ and $\sigma_\omega = \omega^{d1} mod\, n$. Then, Alice sends $(C_A, \omega, \sigma_\omega, d2)$ to the TTP but keeps (d, d_1, d_2, e_1) secret.

iii. The TTP first checks for the validation of Alice's certificate C_A . After that, the TTP checks that the triple $(\omega, \sigma_\omega, d2)$ s prepared correctly. If everything is in correct order as per its rules, TTP saves d2 and generates a voucher V_A by computing $V_A = Sign_{TTP}(C_A, \omega, \sigma_\omega)$. This proves the TTP's signature on message $(C_A, \omega, \sigma_\omega)$, which guarantees that the TTP can issue a valid partial signature on behalf of Alice by using the secret $d2$.

3.4.2 Signature Exchange Protocol

Before all this, a contract has to be agreed between bob and Alice and they should sign it. It should also has a deadline, and identify the Alice, Bob, and TTP.

a) Initially, the initiator Alice has to compute her partial signature $\sigma_1 = h(m)^{d1} \bmod n$, and then sends the triple $(C_A, \omega, \sigma_\omega)$ to the responder Bob. Here, $h(.)$ is a cryptographically secure hash function.

b) After receiving (C_A, V_A, σ_1), Bob first verifies that $C_A$is whether issued by CA, and V_A is Alice's voucher created by the TTP. Then, Bob checks if the identities of Alice, Bob, and the TTP are correctly mentioned as part of the contract '*m'*. If all these checking are ok, Bob initiates the below interactive zero-knowledge protocol with Alice to check whether σ_1 is Alice's valid partial signature on contact.

   i) Then Bob selects two numbers $i, j \in_R [1, n]$ at random, and a challenge c to Alice is sent by computing $c = \sigma_1^{2i} \sigma_w{}^{j} \bmod n$.

   ii) Receiving the challenge c , Alice calculates the response $r = c^e \bmod n$ She then returns her commitment $\bar{r} = TCcom(r,t)$ to Bob using a random number t, where $TCcom$ is the commitment algorithm.

   iii) After receiving the commitment $\bar{r}$, Bob sends Alice the pair (i, j) to acknowledge that he is done with the challenge c properly.

   iv) Alice verifies for correct preparation of c, that is $c \equiv \sigma_1^{2i} \sigma_\omega^{j} \bmod n$. If ok, Alice withdraws his commitment $\bar{r}$ by knowing the responses (r,t) to Bob. With this (r,t) , Bob knows σ_1 as valid if and only if $r \equiv h(m)^{2i} \omega^{j} \bmod n$ and $\bar{r} \equiv TCcom(r,t)$.

c). Bob checks the σ_1 Alice's valid partial signature and the deadline t mentioned in contract m is whether enough for resolving the dispute resolution from the TTP. Then only he sends his signature σ_B to Alice.

d). After receiving σ_B , Alice has to check whether it is Bob's valid signature. If it is, she sends Bob the partial signature σ_2 by computing $\sigma_2 = h(m)^{d2} \bmod n$. As Bob receives σ_2, he sets $\overline{\sigma_A} = \sigma_1 \sigma_2 \bmod n$, and accepts

σ_2 as valid if and only if $h(m)^2 = \overline{\sigma}_A^{2e} \bmod n$. Here, Bob can receive Alice's standard RSA signature σ_A on message m from $\overline{\sigma}_A$. If all this do not happen, Bob seeks the help of TTP for connection before the expiry of the date.

3.5 Reputation Exchange Protocol

The status swapping procedure is commenced with the peer supplicant when the peer applicant chooses the supplicant with the highest status. This procedure requires the applicant to be represented as R and the peer supplicant is represented as P. As in R→P: X represents that the peer sends a message X to the supplicant (P). P_{k2} denotes private key of peer P while P_{K1} denotes public key of the peer $P.E_k$ (τ) denotes encryption of the phase (τ) with key K and E B_K (X) symbolizes blinding phrase with a key K. H(λ) denotes a one way hash of the value λ. This procedure supposes that obtainable functions are inserting and search, but are not flexible enough for peers which may not be proposed tag along the join and leave procedures of the network. The status swapping procedure contains the following phases:

Step 1: R→P: RTS & IDR a REQUEST FOR TRANSACTION (RTS) is sent by the peer applicant along with its own IDENTITY CERTIFICATE (IDR) to the peer supplicant as it is required for authentication purposes in Step 7.

Step 2: P→ R: IDP & TID & $E_{p_{k2}}$ (H(TID)||RTS . The peculiar IDENTITY CERTIFICATE (IDP), the CURRENT TRANSACTION ID (TID) and the signed TID, $E_{p_{k2}}$ (H(TID)||RTS is sent by the peer supplicant wherein signed TID is essential for the supplicant to avoid duplication of the usage of the same transaction id again. The applicant also applies for this signed TID and piles it up in the network at the end of the procedure for admission to other peers.

Step 3: R : LTID (Max (Search(PK1|| TID)). The value of the LAST TRANSACTION ID (LTID) that was used by the supplicant is gathered by the peer applicant who then combines the public key P of the peer supplicant along with the string TID and a search operation is carried out. Any peer present in the network responds only when it has the relevant TID that is specified by the applicant and the peer applicant chooses the highest TID out of all the TIDs received. The highest TID value becomes the LTID. It is certainly possible that the peer supplicant may conspire with the peer who piled up its last LTID and may modify it, but this is impossible as the applicant registers relevant information.

Step 4: R : IF(LTID≥ TID)GO TO Step 13
Foul play is presumed if the value of LTID initiated by the peer applicant is originally from some other random transaction and applicant jumps to Step13

Step 5: R→P: Past Recommendation Request & r. If the step 4 check gives successful results, then applicant requests the supplicant for the earlier received proposals. If the current transaction being performed is, say Nth transaction, the applicant makes a head-on request for N-1th,N-2th,….,N-nth proposals where r<N. The peer applicant is solely responsible for deciding the value of r and is considered to be directly proportional to the applicant's venture in the transaction.

Step 6: P→R: CHAIN, $E_{p_{K2}}$ (CHAIN)

CHAIN=({RE C_{N-1} || EZ_{N-1K2} (H(RE C_{N-1})}||

{RE C_{N-2} || $E_{Z_{N-2K2}}$ (H(RE C_{N-2} ,RE C_{N-1}))} ||

{RE C_{N-3} || $E_{Z_{N-3K2}}$ (H(RE C_{N-3} ,RE C_{N-2}))}||

{RE C_{N-4} || $E_{Z_{N-4K2}}$ (H(RE C_{N-r} ,RE C_{N-r-1}))})

The earlier received proposals RE C_{N-1} , RE C_{N-2} ,……, RE C_{N-3} which were provided by peers (Z_{N-1}, Z_{N-2} ,….., Z_{N-3}).is sent by the supplicant. The CHAIN is singed so as to enable the applicant to hold supplicant responsible for the chain. The supplicant can, in no way, change the proposals that have been assessed by the earlier applicants. Consider an applicant (say Z_1) has signed both the (ι th) and the previous (ι-1th) recommendation using its private key Z_{K2}, as $E_{Zn_{K2}}$ (H(RE C_{N-3} || RE $C_{N-(\iota-1)}$)), in no way can a supplicant alter the CHAIN.

Step 7:R : Result=Verify(RE C_{N-1} ;RE C_{N-2} . . .RE C_{N-r})

If Result! = Verified GO TO STEP 13

A simple public key cryptography protocol is employed by an applicant to authenticate the CHAIN. The authentication process is easier when a supplicant possesses certificates of all the peers with whom it had connections earlier. In case it doesn't have one, it accumulates it from the supplicant itself. The provider had obtained its requester's certificate are checked for by the applicant. The applicant jumps to Step 13 in case the authentication process fails.

Step 8: Contract signing between peer selected under reputation check and peer that requesting the service

Signature exchange protocol will get into action between Peer "SRP" that requesting the service and Peer "SPP" that selected as service provider by reputation check.

Initially, the initiator SRP has to compute her partial signature $\sigma_1 = h(m)^{d1} \bmod n$, and then sends the triple $(C_A, \omega, \sigma_\omega)$ to the responder SPP. Here, $h(.)$ is a cryptographically secure hash function. After receiving (C_A, V_A, σ_1), SPP first verifies that C_A is whether issued by CA, and V_A is SRP's voucher created by the TTP. Then, SPP checks if the identities of SRP, SPP, and the TTP are correctly mentioned as part of the contract '*m*'. If all these checking are ok, SPP initiates the below interactive zero-knowledge protocol with SRP to check whether σ_1 is SRP's valid partial signature on contact. Then SPP selects two numbers $i, j \in_R [1,n]$ at random, and a challenge c to SRP is sent by computing $c = \sigma_1^{2i}\sigma_w^{\ j} \bmod n$. Receiving the challenge c , SRP calculates the response $r = c^e \bmod n$ She then returns her commitment $\bar{r} = TCcom(r,t)$ to SPP using a random number t , where $TCcom$ is the commitment algorithm. After receiving the commitment $\bar{r}$, SPP sends SRP the pair (i, j) to acknowledge that he is done with the challenge c properly. SRP verifies for correct preparation of c, that is $c \equiv \sigma_1^{2i}\sigma_\omega^j \bmod n$. If ok, SRP withdraws his commitment $\bar{r}$ by knowing the responses (r,t) to SPP. With this (r,t) , SPP knows σ_1 as valid if and only if $r \equiv h(m)^{2i}\omega^j \bmod n$ and $\bar{r} \equiv TCcom(r,t)$. c). SPP checks the σ_1 SRP's valid partial signature and the deadline t mentioned in contract m is whether enough for resolving the dispute resolution from the TTP. Then only he sends his signature σ_B to SRP. After receiving σ_B , SRP has to check whether it is SPP's valid signature. If it is, she sends SPP the partial signature σ_2 by computing $\sigma_2 = h(m)^{d2} \bmod n$. As SPP receives σ_2 , he sets $\overline{\sigma_A} = \sigma_1\sigma_2 \bmod n$, and accepts σ_2 as valid if and only if $h(m)^2 = \overline{\sigma_A}^{2e} \bmod n$. Here, SPP can receive SRP's standard RSA signature σ_A on message m from $\overline{\sigma_A}$. If all this do not happen, SPP seeks the help of TTP for connection before the expiry of the date.

Step 9: P→R : File or Service

The file or service is afforded as per the obligation specified concerning search operation performed for the supplicants.

Step 10: R →P : B1 =E B_{Ka} (REC|| TID|| $E_{R_{K2}}$ {H(REC, || TID)})

A BLINDING KEY (Ka) is produced by an applicant on receiving the service, who then combines the RECOMMENDATION (REC) and the TRANSACTION ID (TID) it had received in Step 2 and signs it. Consequently, the signed proposal is blinded along with the blinding key, Ka. This is done in order to entrust the supplicant to the proposal received before it actually knows the value, lest it disowns it on recognizing that it is low. It is also involves the fact that the supplicant made use of TID in a blinded suggestion from the peer applicant, which is also authenticated by the applicant itself. The blinded proposal includes the Chain that is consequently used by the supplicant to certify its status to some other applicant.

Step 11:

a. P →R: B1|| $E_{P_{K2}}$ (H(B1),nonce),nonce

b. R→P: Ka

A NONCE is sent by the supplicant after signing the proposal even though it is unable to see the proposal and acknowledges it back to the applicant, who then authorizes the signature and sends blinding key Ka to the supplicant to unblind the received string in Step11a and confirms the received proposal.

Step 12: Insert

(IDR ;{ REC ||TID|| $E_{R_{K2}}$ {H(REC) || H(TID)}})

The proposal assigned to the supplicant (REC), the transaction id (TID), and its own identity certificate is verified by the applicant and is then accumulated in the network using Insert methodology of the P2P network which marks the end of the transaction.

Step 13: Step 13 is concerning the methodology executed by an applicant when foul play is anticipated.

ABORT PROTOCOL

R: Insert (IDR; {CHAIN ||TID|| $E_{R_{K2}}$ {H(CHAIN) || H(TID)}})

If the authentication process in Step7 fails, the applicant takes the CHAIN that was verified b the supplicant and also the TID is taken into consideration after which, it is signed and the Insert methodology is preferred to be made use of to insert the chain and also its own identity certificate into the network. Subsequently, any suitable applicant will be able to confirm with the statistics of the failed authentication efforts and a MIN RECOMMENDATION for that TID is presumed for the supplicant. Fake proposals cannot be encouraged to be inserted into the network as TID is to initiated that is verified by the supplicant. If an applicant reaches Step 13 from Step 4 without any possible hindrances, it will then apply for the Chain form the supplicant and will then afterward execute R: Insert(IDR,{CHAIN ||TID || $E_{N-R_{K2}}$ {H(TID || RTS))}}).

3.6 Analysis of the Protocol

Only a single search request is supposed to be commenced in the network so as to gather the already received proposals that were previously received by the supplicant. Also able to prevent the tampering reputation provided by SRP to SPP by peers that in path. This procedure is entailed the responsibility of tackling the issue of unbalanced nature of availability of peers in the network, which is considered to be a major issue concerning P2P networks.

1. The supplicant unintentionally forwards the wrong TID in Step 2. Consider that id which the supplicant forwards as TID and the LTID be the last Transaction ID for the supplicant. The value of TID is always supposed to be equivalent to LTID + 1. If in case of TID' > LTID+1, there arises a situation wherein there will be inexplicable misplaced proposals. If again in case of TID' < LTID+1, then the supplicant will be caught up with in the Step 4 of the procedure, as the last id issued and used by the supplicant was made public and accessible to all the peers. The value of TID is considered as 0 if a peer is for the first time donning the role of a supplicant.

2. The transaction in Step 9 will not be terminated by the supplicant. A supplicant is allowed to abandon the transaction after providing the applicant with the requested requisite information in Step 9 and also can abandon the transaction after Step 10. In both the cases, there is an absence of a proposal by the supplicant for the transaction id TID. The proposal in Step 12 can be liberated by the applicant provided the supplicant fails to verify and sign the blinded proposal, without acquiring the supplicant's signature. In the next transaction, precisely TID+1, the supplicant again fails to illustrate the proposal for that relevant transaction, TID to the transaction's applicant, TID+1 and hence the new applicant entrusts itself with the job of scanning the network making use of Search methodology for TID. In case TID is found, the suggested proposals are also found out pertaining to the suppliant in the transaction. The applicant will then be responsible as the TID would by then have been signed b the supplicant, who will have to acknowledge the proposal as it comprises the signature of the supplicant, TID & $E_{P_{K2}}$ (H (TID)). A minimal suggestion TID is presented to the supplicant by the peer applicant in the absence of the availability of the required proposal. If in Step 11, the supplicant acknowledges the signed blinded proposal B1 & $E_{P_{K2}}$ (H (B1)), the applicant refuses to send the key, Ka and directs itself to Step 11, missing all the requisite steps, and then the supplicant scans the entire network and acquires the verified proposal of the applicant. If an applicant skips or fails to execute Step 10, then in the upcoming transaction TID+1, LTID is looked for by the new applicant and fails in his endeavor. Hence, TID can be considered as terminated and the next transaction can be continued with the transaction id provided, TID.

3. Collusion by rogues or liar farms. All status systems are prone to complicity on account of its nature. It is possible for two or more liar farms to combine and conspire in order to augment each other's status. The influence of the conspiracy can be alleviated by classifying proposals on the basis of individual identities, authenticating agencies etc. The list of conspirators can be circulated, thereby, guarding the remaining peers from an possible attack. Peers when recognized as conspirators will not be permitted to get back into the stream of network and hence they have an impetus against conspiracy. The chain of proposals of the conspirators will aid in offering support that few peers are conspiring, thereby, protecting good peers and from the intrusion of bad peers into the network.

4. Multiple requesters and concurrency. A supplicant in the presently used procedure will not be provided with the facility of making use of the same identity in the synchronized transactions. The first choice for procedure augmentation is that the supplier identifies and familiarizes all its applicants with each other. Consequently, the authentication process performed in Step 4 is performed amidst a group of applicants and results are arranged in accordance with the fact that TID dissimilarity needs to be initiated due to more number of applicants. After integrating the augments, there would be a bi party procedure that would still be prevalent where the cluster of applicants is considered to the second party while the supplicant is supposed to be the first party. The figure 1 explores the ability of the proposed model to prevent the false reputation submitted by unauthorized peers that acts as service request peer SRP.

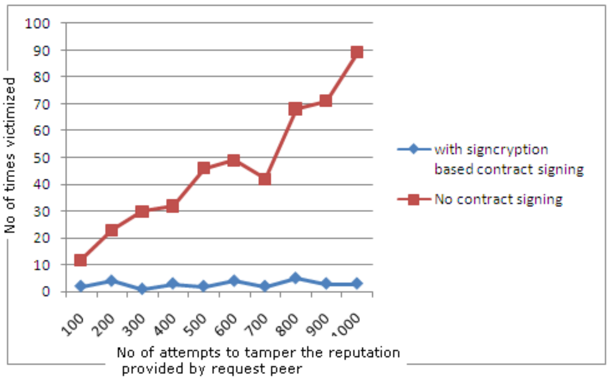

Fig 1: Line chart comparison between reputation check with contract signing and without contract signing.

We can observe that contract signing by signcryption approach is most effective to prevent the reputation tampering attack. Even peer communication with contract signing also victimized few times but victimization occurred due to contract sign breakage. Hence if contract sign is alive then attack to tamper the reputation is almost null. The figure 2 confirms the stable growth in execution time when consider this contract signing process, which was compared with peer communication process without contract signing.

Hike in peer communication execution time that is negligible when consider the improvement in prevention of reputation tampering attack attempts.

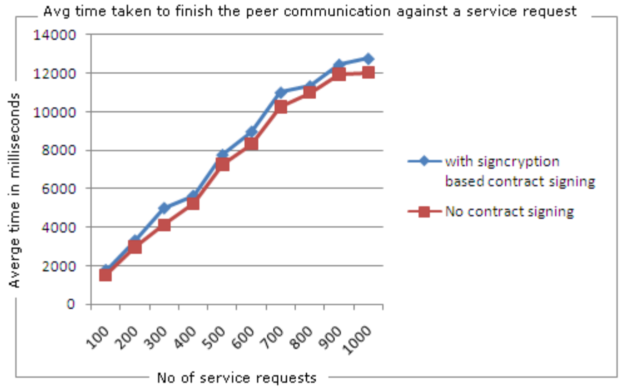

Fig 2: Comparison chart for average time taken for process service request with contract signing and without contract signing.

4. CONCLUSION

Here in this paper we proposed a signcryption based contract signing for peer communication based on reputation check. The results are evident that proposed two way reputation check model is effective to avoid the reputation tampering attack attempts. The proposed model is showing some hike in average process time for peer communication, which can be negligible in the context of reputation tampering attack avoidance. In future we plan to find solution to avoid the contract sign breaching.

5. REFERENCES

[1] Prashant Dewan and Partha Dasgupta,"P2P Reputation Management Using Distributed Identities and Decentralized Recommendation Chains", IEEE Transaction On Knowledge and Data Engineering, VOL.22, NO.7, JULY 2010, PP.1000-1013.

[2] M. Ripeanu, I. Foster, and A. Iamnitchi, "Mapping the gnutella network: Properties of large-scale peer-to-peer systems and implications for system design," IEEE Internet Computing Journal, vol. 6, no. 1, 2002.

[3] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content addressable network," in ACM SIGCOMM, Aug. 2001.

[4] E.Adar and B.A.Huberman,"Free Riding on Guntella" Tech.Rep.Xerox PARC 2000.

[5] N.Li and J.C.Mitchell,"RT:A Role Based Trust management frame work",proc.Third DARPA Information survivability conference and Exposition(DISCEX III)Apr,2003.

[6] S. Saroiu, P. K. Gummadi, and S. D. Gribble, \A measurement study of peer-to-peer sharing systems," in SPIE Conference on Multimedia Computing and Networking (MMCN), Jan. 2002.

[7] Karl. Aberer and Z. Despotovic, "Managing trust in a peer-2-peer information system," in Ninth International Conference on Information and Knowledge Management (CIKM), Nov. 2001.

[8] S. D. Kamvar, M. Schlosser, and H. Garcia-Molina, "Eigenrep: Reputation management in p2p networks," Unpublished work, 2003.

[9] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, "Chord: A scalable Peer-To-Peer lookup service for internet applications," in ACM SIGCOMM, Aug. 2001, PP. 149-160.

[10] J.Docuccur,"The Sybil attack", proc IPTPS 02 workshop 2002.

[11] Gnucleus home page," http://www.gnucleus.com/.

[12] P.Resnick, R.Zeckhauser, E.Friedman and K.Kuwabara

"Reputation systems" communication ACM, Vol 43 PP 44-45 Dec 2000.

[13] eBay website www.ebay.com.

AUTHORS PROFILE

V.Thejaswini Reddy received the M.Tech degree in computer science in 2012 and the B.Tech degree in 2007 from Jawaharlal Nehru Technological University. Her research interests are peer-to-peer networks, security and reputation systems.

[14] C.Dellarocas,"immunizing online reputation reporting systems against unfair ratings and discriminating behavior", Proc.ACM Conf. Electronic Commerce.PP.150-157, oct 2000.

[15]R.Dewan,"Injecting Trust in peer-to-peer systems", technical report, Arizona state univ 2002.

[16] L.Rivest and B.Lampson,"SDSI: A simple distributed security infrastructure" proc.crypto 96 pp 104-109 aug 1996.

[17] R.L.Rivest, A.shamir and L.adleman."A method for obtaining digital signatures and public-key cryptosystems communications" of the ACM, Feb 1978 21(2):120-126

[18] J.M.Park, E.Chong, H.J.Siegel and I. Ray "constructing fair exchange protocols for e-commerce via distributed computations of RSA signatures" In: Proc of 22th annual ACM symp on principles of distributed computing (PODC'03) PP 172-181.

[19] R.Gennaro, T.Rabin and H.Krawczyk. "RSA based undeniable signature journal of cryptology" 13(4):397-416, 2000.A preliminary version of this paper appeared in the proceedings of CRYPTO'97.

B.Lalitha is an assistant professor in the Department of Computer Science at Jawaharlal Nehru Technological University. She leads the PG lab in JNTU University. She is doing research in distributed computing specialized in p2p networks.