Managing P2P Reputation System Using Decentralized Approach

B. V. S. N. Lakshmi, C. Prakasa Rao

Abstract— In Peer-to-Peer networks, many security mechanisms are based on specific assumptions of identity and are vulnerable to attacks when these assumptions are violated. The traditional security techniques developed for the centralized distributed systems like client-server networks are insufficient for P2P networks by the virtue of their centralized nature. The absence of a central authority in a P2P network poses unique challenges for reputation management in the network. These challenges include identity management of the peers, secure reputation data management, Sybil attacks, and above all, availability of reputation data.

In this paper, we present a cryptographic protocol for ensuring secure and timely availability of the reputation data of a peer to other peers. The past behavior of the peer is encapsulated in its digital reputation, and is subsequently used to predict its future actions. As a result, a peer's reputation motivates it to cooperate and desist from malicious activities. The cryptographic protocol is coupled with self-certification and cryptographic mechanisms for identity management and countering Sybil attack. We illustrate the security and the efficiency of the system analytically and by means of simulations in a completely decentralized P2P network.

Keywords– Peer-to-Peer networks, security, identity management, reputation system.

I. INTRODUCTION

PEER-TO-PEER (P2P) networks are self-configuring networks with minimal or no central control. The traditional mechanisms for generating trust and protecting client-server networks cannot be used for pure P2P networks. This is because the trusted central authority used in the traditional client-server networks is absent in P2P networks. Introduction of a central trusted authority like a Certificate Authority (CA) can reduce the difficulty of securing P2P networks. The major disadvantage of the centralized approach is, if the central authority turns malicious, the network will become vulnerable. In the absence of any central authority, repository, or global information, there is no silver bullet for securing P2P networks.

In this paper, we investigate Reputation Systems for P2P networks [1]—a more ambitious approach to protect the P2P network without using any central component, and thereby harnessing the full benefits of the P2P network. 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 ostracized from the network as the good peers do not perform any transactions with the malicious peers. Expulsion of malicious peers from the network significantly reduces the volume of malicious activities.

Revised Manuscript Received on October 2012.

B.V.S.N.Lakshmi, CSE Department, Prakasam Engineering. College, Kandukur, India.

C.Prakasa R, Associate Professor, CSE Department, Prakasam Engineering .College, Kandukur, India.

All peers in the P2P network are identified by identity certificates [2]. 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 certificate authority which issues the identity certificate(s) 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 from its owner, but also facilitates secure exchange of reputation information between the two peers participating in a transaction.

The experiments show that the proposed reputation infrastructure not only reduces the percentage of malicious transactions in the network, but also generates significantly less network traffic as compared to other reputation-based security solutions for P2P networks. The main contributions of this paper are:

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.

II. REPUTATION SYSEM

2.1 Threat Model:

The peers follow predefined Join & Leave protocols. The peers are connected with insecure communication channels. As the peers are likely to have conflicting interests, a source of motivation is needed to reduce the number of leechers. Leechers are peers who derive benefit from the system without contributing to the system. The rogue peers can also spread malware in the network (when other peers download content from them). Finally, peers need a mechanism 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 can provide the means to achieve the above goals. Any reputation system is vulnerable to ballot stuffing and bad mouthing as described in [3]. An imperfect reputation system by itself generates vulnerabilities that can be exploited by attackers. Peers need to have a unique handle to which their reputations can be attached. In the absence of any trusted central agency, an attacker can gather infinite identities and start issuing recommendations to itself. A peer might modify the reputation data stored in the network to maliciously raise its own reputation. Finally, there are other vulnerabilities that come in the picture depending on how a given reputation system is designed. We explain those vulnerabilities and their corresponding mitigation in the sections where we enumerate the design decisions.

2.2 Self- Certification:

In order to participate in the reputation system, a peer needs to have a handle. The reputation of a peer is associated with its handle. This handle is commonly termed as the"identity"of the peer even though it may not "identify" a peer, i.e., it may not lead to the real-life identity of the peer. A peer receives a recommendation for each transaction performed by it, and all of its recommendations are accumulated together for calculation of the reputation of a given peer.

A malicious peer can use self-certification [5] 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 distinct 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 an identity farm. The set of identities that issue false recommendations is called a liar farm [10][11]. This attack belongs to the class of attacks termed Sybil attacks [4]. In simple words, a peer having an identity farm is equally capable of subverting a reputation system as a peer that has colluded with a large number of other peers.

The 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). The symbol P_{K2} represents the private key of the peer P and P_{K1} represents the public key of the peer P. $E_K(\Gamma)$ represents encryption of the phrase(Γ) with key K, while $EB_K(X)$ represents blinding phrase X with key K.

1. P→A: B1= $\{EB_{Ka} (I_{Alicer})\}$, I_{Alice}
   The peer Alice generates a BLINDING KEY, Ka and another identity for herself (I_{Alicer}). Alice cannot be identified from her identity (I_{Alicer}). Subsequently, she blinds her identity (I_{Alicer}) with the blinding key Ka. B1 represents the blinded identity [6]. Alice sends B1 to the authority with her real identity that proves her membership to a group.
2. A→ P: B2 = $_{EPAuthorityK2}$ $\{B1= EB_{Ka} (I_{Alicer})\}$
   The authority signs the blinded identity, B1 and sends it (B2) back to the peer.
3. P: $E_{PAuthorityK2}$ $\{I_{Alicer}\}=\{EB_{Ka}\{B2\}\}$
   The peer unblinds the signed identity and extracts the identity authorized by the authority
   $E_{PAuthorityK2}\{I_{Alicer}\}$.

The fundamental assumption in the group-based approach is that in a P2P network, peers would be more interested in the ranks of the prospective providers than in the absolute value of the reputations. The simulations show that this approach changes the absolute reputations of peers considerably but it has only a minimal impact on the relative ranks of the peers.

2.3 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. $E_K(\Gamma)$ represents encryption of the phrase (Γ) with key K, while $EB_K(X)$ represents blinding phrase X with key K. H(λ)denotes a one way hash of the value λ. 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 steps in the reputation exchange protocol are as follows:

Step 1: R→P: RTS & IDR

The requester sends a REQUEST FOR TRANSACTION (RTS) and its own IDENTITY CERTIFICATE (IDR) to the provider. The provider needs the identity certificate of the requester as the provider has to show it to the future requesters in Step 7.

Step 2: P→R: IDP & TID & E_{PK2} (H (TID || RTS))

The provider sends its own IDENTITY CERTIFICATE (IDP), the CURRENT TRANSACTION ID (TID) and the signed TID, E_{PK2} (H (TID||RTS). The signed TID is needed to ensure that the provider does not use the same transaction id again. In the end of the protocol, this signed TID is signed by the requester also and stored into the network where it will be accessible to other peers.

Step 3: R: LTID =Max (Search (PK1 || 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 public key of the provider with the string TID and performs the search. Any peer having the TID for the provider replies back with the TID and the requester selects the highest TID out of all the TIDs received. The highest TID becomes the LTID. It is possible that the provider might collude with the peer who stores its last LTID and change the LTID. As the LTID and related information would be signed by the requester, the provider cannot play foul.

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

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. Hence, it is trying to get another recommendation for the same transaction number (TID). The requester suspects foul play and jumps to Step 12.

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

If the check in Step 4 succeeds, i.e., the requester is sure that the provider is not using the same transaction number; it requests the provider for its previous recommendations. In other words, if the current transaction is the Nth transaction for the provider, the requester makes a request for N-1th,N-2th and so on recommendations till N-r th recommendation where r is less than N. The value of r is decided by the requester and it is directly proportional to the requester's stake in the transaction.

Step 6: P→R: CHAIN, E_{PK2} (CHAIN)

CHAIN = $(\{REC_{N-1} \| E_{ZN-1K2} (H (REC_{N-1})\} \| \{REC_{N-2} \| E_{ZN-2K2} (H (REC_{N-2, RECN-1}))\} \|$

$\{REC_{N-3} \| E_{ZN-3K2} (H (REC_{N-3}, REC_{N-2}))\}) \| \{REC_{N-4} \| E_{ZN-4K2} (H (REC_{N-r}; REC_{N-r-1}))\})$

The provider sends its past recommendations (REC_{N-1}, REC_{N-2} ….. $RECN_{-3}$) which were provided by 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. As the recommendations have been signed by the previous requesters, the provider could not have maliciously changed them. If the requester (say Z_l) has signed both the (*l*th) and the previous (*l*-1th) recommendation

using its private key Z_{K2}, as E_{ZnK2} (H (REC_{N-3}||$REC_{N-(l-1)}$)), there is no way a provider can modify the CHAIN. In other words, the provider cannot simply take away a bad recommendation and put in a good recommendation in order to increase its reputation.

Step 7:

R: Result = Verify (REC_{N-1}, REC_{N-2} . . . REC_{N-r})
If Result! = 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. In the case it does not have the required certificates; it obtains the certificates from the provider itself. The provider had obtained its requester's certificate in Step 1. In addition, the requester checks for liar farms. If the verification fails the requester jumps to Step 12.

Step 8: P→R: File or Service

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

Step 9:

R→P : B1 = EB_{Ka}(REC|| TID ||E_{RK2}{H(REC,||TID)})

Once the requester has received a service, it generates a BLINDING KEY, Ka. The requester concatenates the RECOMMENDATION (REC) and the TRANSACTION ID (TID) it had received in Step2 and signs it. Subsequently, it blinds the signed recommendation with the blinding key, Ka. The recommendation is blinded in order to make the provider commit to the recommendation received before it sees the value of the recommendation such that it does not disown the recommendation if it is low. The provider receives the blinded recommendation from the requester. The blinded recommendation is 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 then sends the blinding key Ka to the provider which can unbind the string received in Step 10a and checks its recommendation.

Step 11: Insert

(IDR,{REC || TID || 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: Step 12 explains the steps a requester executes when it expects foul play:

ABORT PROTOCOL

R: Insert (IDR, {CHAIN || TID || E_{RK2} {H(CHAIN)||H(TID)}})

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 to insert the chain and its own identity certificate into the network. As a result, any subsequent requester will be able to see failed verification attempt and will assume a MIN_RECOMMENDATION recommendation for that TID for 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 and subsequently will perform R: Insert (IDR, {CHAIN||TID|| E_{N-RK2} {H(TID || RTS))}}).

III. EXPERIMENTS AND RESULTS

3.1 Evaluation of Self- Certification

We simulated a peer-to-peer network to understand the effect of the Network Size (N), the Number of Transactions (T), and the Group Size (d) on the Mean Rank Difference (M) over all the peers in the network. The Mean Rank Difference (M) is calculated by averaging the rank difference of all the peers in the network for one instance of the simulation. The rank difference for a peer is the difference in the rank of the peer when the proposed identity mechanisms are used, and when it is not used. Specifically, we tried to answer the following two questions:

1. Is the mean rank difference a good predictor of the rank difference of individual nodes? What is the variation in the rank difference of individual peers in the simulation for a given value of d? Mathematically, what percentage of nodes has their rank difference equal to the mean rank difference?
2. Does the network size (N), group size (d), or the number of transactions (T) in the network impact the mean rank difference in the network? In other words, what is the expected mean rank difference for other network configurations which are different (in terms of size, group size d, or number of transactions) than the networks simulated by us? The simulated network consisted of 1,000 peers in which the peers performed 20,000 transactions per instance simulation, at a group size d = 3.

The peers were assigned unique IP addresses. Additionally, each peer was assigned a goodness factor to account for the fact that a good peer is likely to participate in higher number of transactions (as a provider) than a bad peer. Hence, the reputation of a good peer is likely to escalate faster than the reputation of a bad peer. MAX RECOMMENDATION was set to 1 while the value of MIN RECOMMENDATION was set to -2. This was done to ensure that a peer lost more reputation on performing a malicious transaction as compared to the reputation gained by doing a good transaction. In addition, the results did not vary much when the values of MAX RECOMMENDATION and MIN RECOMMENDATION were selected to be outside the set [-2, 1]. The percentage of malicious peers was varied from 10 percent to 90 percent. The probability that a peer would cheat was set to 1/2 in order to account for the fact that in the real world honesty is not constant and varies with time and stakes.

For each iteration of the simulation, a randomly selected peer became the provider and another randomly selected peer assumed the role of a requester.

After the transaction, the requester gave a recommendation to the provider. For each recommendation received by the provider, its reputation was incremented by its goodness factor. After 20,000 transactions, the ranks of the peers were calculated without using the proposed identity management mechanism (d = 3). The differences of the ranks were averaged for all peers in the network and the results were statistically analyzed.

The above steps were repeated 20 times and results averaged, for each of the following three scenarios:

1. The group size was varied from 5 to 50 (Step 5) and the other parameters were constant. This case enabled us to investigate the impact of averaging on mean ranks of peers,
2. The group size and the network size were kept constant and varied the number of transactions from 2,000 to 20,000 (step 2,000), and
3. The number of transactions and the group size were kept constant to 20,000 and 10, respectively, and the network size was varied from 200 to 1,000 peers (step 200).

Finally, we calculated the mean rank difference for each set of 20 simulations and statistically analyzed using regression analysis, T-Test, and Analysis of Variance test (ANOVA) [7] to deduce the answers for the above two questions.

3.2 Self-Certification Results and Analysis

In the first experiment (N = 1,000, T = 20 K and d = 3), the rank difference averaged across the peers was 13.246±0.81 with a 95 percent confidence level. Although the result of the first experiment (Fig. 1) does not resemble a normal distribution, we decided to treat it as normally distributed data as per the recommendations of the central limit theorem [8] for a sample size larger than 30. We inferred that 68 percent of the nodes in a P2P network will have a rank difference in the range of 13.246±13.07. In other words, 680 nodes for every 1,000 nodes will have a rank difference of less than 27. Besides the standard error for this analysis was very low (≈0.4), hence the Mean Rank Difference gives a reasonably good picture of the rank differences of the individual nodes.

In order to answer the second question, we did an incremental regression analysis and an ANOVA test to ascertain how do the variation in network size, number of transactions, and group size change the mean rank difference. Heuristically, we expected that the mean rank difference should change with a change in the group size but it should remain invariant when either of the other two parameters is varied. Our null hypothesis (H0) was that none of the three factors had any impact on the mean rank difference. The following regression equation quantifies the impact of the variability of the three factors on mean rank difference:

Mean Rank Difference

$$= \Delta + \beta 1 * d + \beta 2 * N + \beta 3 * T$$

The Significance F value for the ANOVA test was less than zero. Hence, the null hypothesis (H0) was proved to be untrue and had to be rejected. In other words, the group size, network size, and transactions had an effect on the mean rank difference and the coefficients, $\beta 1$, β2, and$\beta 3$ could not be zero. This inference was substantiated by the fact that the p values for the coefficients, $\beta 1$, β2, and $\beta 3$ were extremely low ($«10^{-80}$). Hence, all the three parameters had an impact on the value of the mean rank difference. This result was contrary to our heuristic.

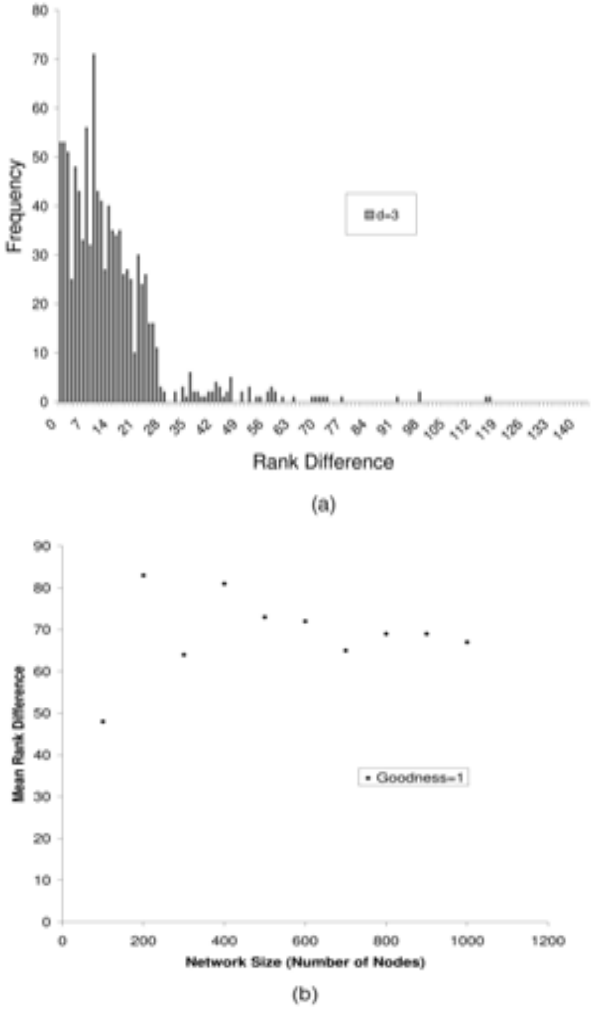
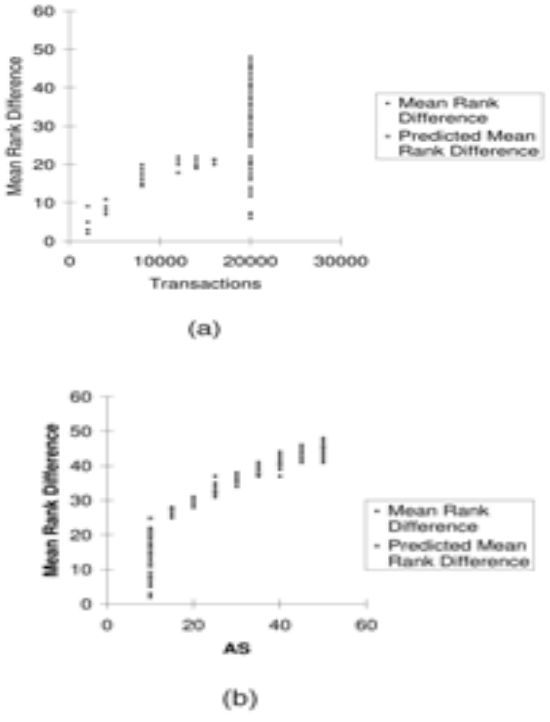


Fig. 1. Variation in mean rank difference. (a) d=3. (b) Goodness=1.



Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

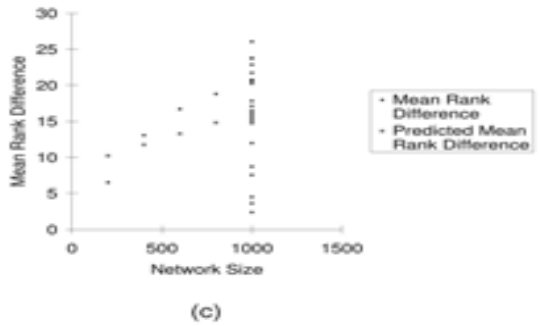


Fig. 2. Variation in mean rank difference (line fit plots). (a) Transactions d= 3. (b) Group size. (c) Network size.

Interestingly, the values of the coefficients, $\beta 1$=0.52±0.02, $\beta 2$=0.02, $\beta 3$=0.0008, and Δ=-19.5±0.98 at the 95 percent confidence level, were in partial conformance to our heuristic. The value of _1 (Fig. 2) was the highest among the three coefficients;

hence, as expected, a small variation in the group size had the highest impact among the three factors on the mean rank difference. The increase in the number of transactions in the network had a minimal impact ($\beta 2$=0.02) on the mean rank difference (Fig. 2). This is ascribed to the fact that with every additional transaction, with a peer in a new6 security zone, the probability of the fact that the next transaction of the provider will again be with a peer in a newer security zone becomes lesser and lesser. In addition, with every transaction of the provider with a peer in the old security zone, the mean rank difference increased. Hence, the reputation calculated with the proposed identity mechanism did not increase while the information calculated without it increased by a very small factor. Therefore, the mean rank difference increased.

The value of $\beta 2$=0.02 was against our initial hypothesis that the network size should not change the mean rank difference (Fig. 2) In order to find out why the network size was changing the mean rank difference, we had to perform an additional experiment. In this experiment, we made an assumption that irrespective of the reputation, the probability of a node being involved in a transaction is the same. In other words, the goodness factor of all the nodes was set to 1 and a bad node and a good node had an equal likelihood of participating in a given transaction. This was different from the assumption made in the earlier experiments where a good node had a higher goodness factor, thereby, accounting for involvement in a larger number of transactions. The result of this experiment showed that there was little or no variation in the mean rank difference w.r.t. the network size, when the likelihood of participation of all the nodes was equal. The correlation coefficient (r) was 0.09247 as is visible in Fig. 1. Hence, we inferred that in the first set of experiments, the increase in the network size raised the number of highly reputed nodes or the nodes with a high value of goodness factor. The reputation difference between the instances, when the proposed identity mechanism was used and when it was not used, was higher for the higher ranked (highly reputed) nodes as compared to the lower ranked nodes; hence, the change in the rank for higher rank nodes was larger. As a result, the mean rank difference for the network was higher.

3.3 Evaluation of the Cryptographic Protocol and the Reputation Model

The protocol was evaluated with a simulating 5,000 peers participating in 20,000-140,000 transactions. The RANGE variable was set to 10. In other words, it was assumed that each file was available with 10 possible providers. The requesters did not have any a priori knowledge of rogue nodes. The rogue nodes performed maliciously with a probability of 1/2, i.e., rogues cheated in one out of every two transactions. Each simulation was executed five times and the results were averaged. The number of repetitions was set to 5 because after the fifth iteration of the simulation, the average values were more or less constant.

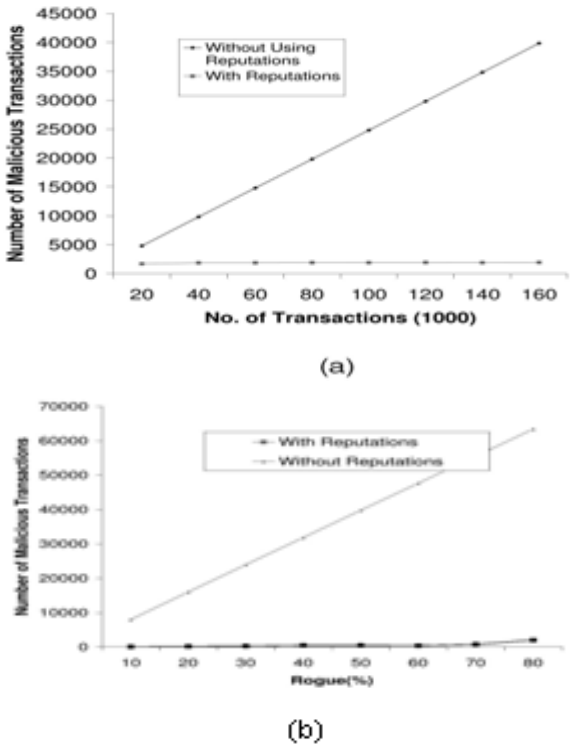


Fig. 3. Variation in total number of malicious transactions. (a) When reputations are used d = 3. (b) Variations in number of rogues

3.3.1 Cumulative and Individual Benefits of Using Reputation

We wanted to quantify the holistic and individualistic benefits of using the proposed reputation model for a P2P network. We measured the change in the number of malicious transactions with an increase in the total number of transactions in the network. The number of rogues was set to a constant at 50 percent and the number of transactions was incrementally raised from 20,000 to 140,000. As is visible in Fig. 3, the total number of malicious transactions increased considerably with an increase in the number of transactions when the proposed model was not used but are more or less constant when the proposed model was used. In the presence of an increasing number of rogues (10-90 percent), when the total number of transactions is constant (= 140,000), the rate of increase in the number of malicious transactions was much less when reputations were used (Fig. 3).

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Subsequently, we analyzed the experience of each peer when the reputations are not used as compared to when they are used. As visible in Fig. 4, when the reputations were not used, the mean of the number of malicious transactions experienced by each good node was 7.966±5.52 with a 95 percent confidence. This mean drastically reduced, when the reputation model is used, to 0.4 ±1.2 with a 95 percent confidence. From the first three simulations, we concluded that the proposed model reduces the number of malicious transaction from the perspective of the network and from the perspective of each peer.

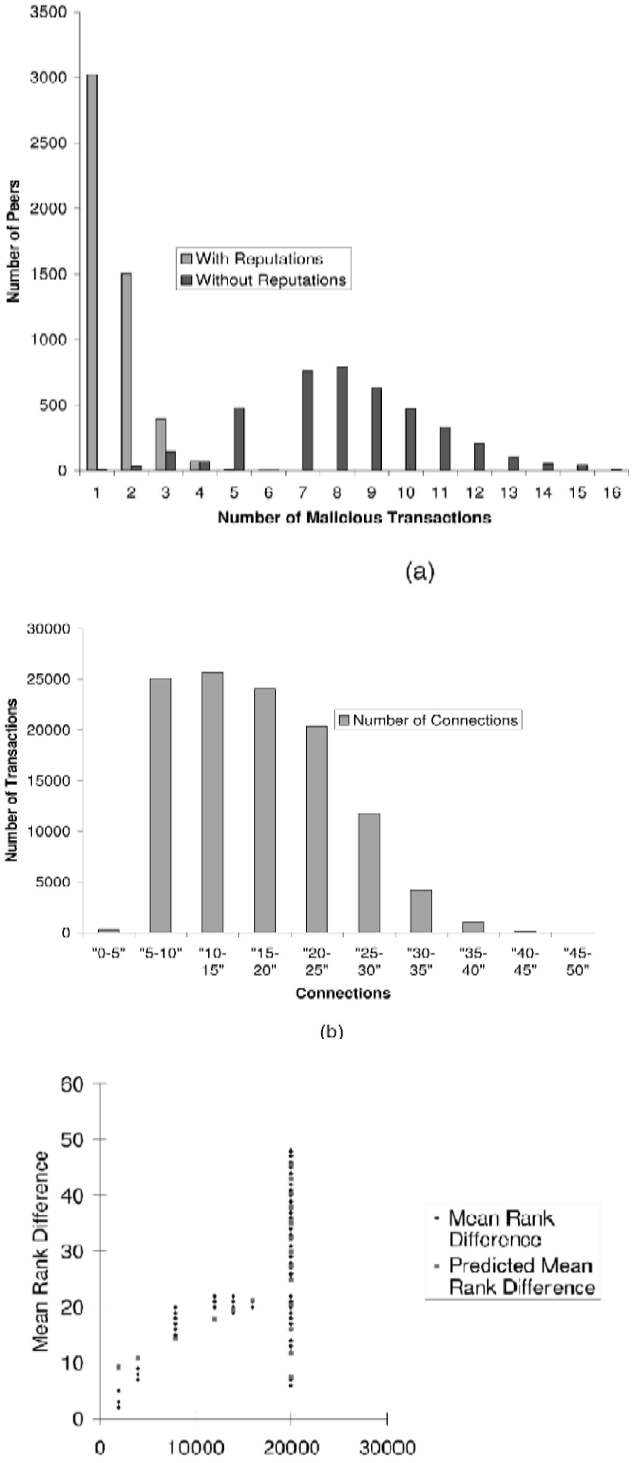


Fig. 4. Variation in total number of malicious transactions. (a) Number of peers versus malicious transactions d =3. (b) Mean rank difference versus number of connections. (c) Mean rank difference versus number of transactions

3.4 Overall Evaluation of the System

In order to evaluate the combined benefit of self-certification, the cryptographic protocol, we modified the experiments done for the evaluation of the cryptographic protocol, and added an availability factor, AF, to each node. The availability factor accounts for the erratic availability of the past recommenders of a given peer. AF values from 50 percent to 90 percent were randomly allocated to peers. The number of malicious transactions were counted and compared with the results obtained in Section 3.3. As is visible in Fig. 5, the number of malicious transactions in the system is the same when the protocol is used separately and when used along with self-certification

IV. DISCUSSIONS

4.1 Network Traffic

We compared the cryptographic protocol with P2PRep [9]. Bandwidth consumption is the Achilles heel of any peer-to-peer network. The proposed protocol generates less network traffic than the P2PRep voting protocol. In both, P2Prep and the proposed protocol, the requester maintains a list of possible information providers when the discovery protocol finishes. P2PRep is a stateless system while the proposed system is stateful. P2PRep is highly communication intensive because of its statelessness. In phase 2 of P2PRep, 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. On the other hand, the proposed system uses minimal communication as the security of system lies in the reputation values stored by the provider itself.

4.2 Low Stake Fast Transaction

In the proposed protocol, it is recommended for the requester to search the network for LTID but it is not mandatory. The requester does not have to ever disclose the LTID in the elicitation-storage protocol.

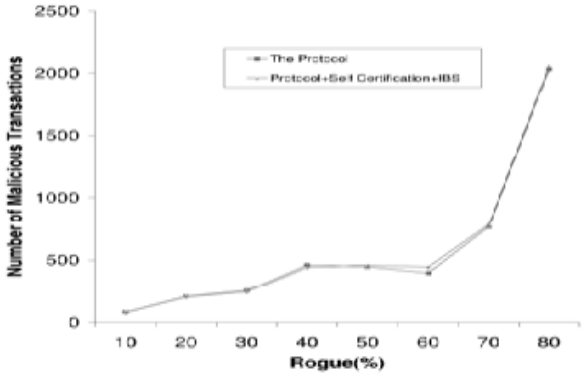


Fig. 5. Number of malicious transactions when self-certification, identity management, and the cryptographic protocol are combined.

In other words, even if the peer who wants to perform a transaction does not know the LTID, it can still perform the transaction. For example, if a peer only wants a "standards file,"

which is commonly available, it does not have to search for LTID and still the probability of the provider cheating is low. In P2PRep, the vote polling cannot be bypassed 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.

4.3 Stale LTID

It is possible (though unlikely) that the requester might never be able to retrieve the correct LTID from the network or it retrieves a stale LTID. If the LTID retrieved by the requester is less than the one offered by the provider then the requester can simply go ahead with the transaction because then it will be the responsibility of the provider to explain the transactions of the missing LTID. If the LTID retrieved by the requester is greater than the one offered by the provider, then the provider is surely trying to cheat. The requester can abort the transaction.

V. CONCLUSIONS AND FUTURE WORK

This paper presents self-certification, an identity management mechanism, reputation model, and a cryptographic protocol that facilitates generation of global reputation data in a P2P network, in order to expedite detection of rogues. A reputation system for peer-to-peer networks can be thwarted by a consortium of malicious nodes. Such a group can maliciously raise the reputation of one or more members of the group. There is no known method to protect a reputation system against liar farms and the absence of a third trusted party makes the problem of liar farms even more difficult. 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 Identity mechanism is based on the fundamental that the ranks of the peers are more relevant than the absolute value of their reputation. The cost of this security is the difference in the ranks of the providers because of the use of the proposed mechanism. 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 than the other reputation systems proposed in its category. It also handles the problem of highly erratic availability pattern of the peers in P2P networks. Currently, the reputation of the provider is considered and the reputation of the requester is ignored. This system can be extended to encapsulate the reputations of both the provider and the requester. In addition, instead of generic number values, the reputation values can be modified in accordance with the context of the reputation.

REFERENCES

[1] 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.
[2] L. Zhou, F. Schneider, and R. Renesse, “COCA: A Secure Distributed Online Certification Authority,” ACM Trans. Computer Systems, vol. 20, no. 4, pp. 329-368, Nov. 2002.
[3] C. Dellarocas, “Immunizing Online Reputation Reporting Systems against Unfair Ratings and Discriminatory Behavior,” Proc. ACM Conf. Electronic Commerce, pp. 150-157, Oct. 2000.
[4] J. Douceur, “The Sybil Attack,” Proc. IPTPS ’02 Workshop, 2002.
[5] P. Dewan, “Injecting Trust in Peer-to-Peer Systems,” technical report, Arizona State Univ., 2002.
[6] D. Chaum, “Blind Signatures for Untraceable Payments,” Proc. Advances in Cryptology (Crypto ’82), 1983.
[7] D.C. Montgomery, Design and Analysis of Experiments. J. Wiley and Sons, 2000.
[8] D. Rumsey, Statistics for Dummies. J. Wiley and Sons, 2003.
[9] 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.
[10] T. Rabin and M. Ben-Or, “Verifiable Secret Sharing and Multiparty Protocols with Honest Majority,” Proc. 21st Ann. ACM Symp. Theory of Computing, pp. 73-85, 1989.
[11] Cachin, K. Kursawe, A. Lysyanskaya, and R. Strobl, “Asynchronous Verifiable Secret Sharing and Proactive Cryptosystems,” citeseer.nj.nec.com/ cachin02asynchronous.html, 2002.

AUTHOR PROFILE

B.V.S.N.Lakshmi, pursuing Master of Technology in CSE branch from Prakasam Engineering College, Kandukur, Prakasam(Dt.),Andhra Pradesh Affiliated to JNTUK, Kakinada, A.P., India.

C.Prakasa Rao, M.Tech, (Ph.D) working as Associate Professor, in CSE Department at Prakasam Engineering College, Kandukur, Prakasam(Dt.).Andhra Pradesh, India. Affiliated to JNTUK, Kakinada, A.P., India.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication