

SECURING REPUTATION DATA IN PEER-TO-PEER NETWORKS

Prashant Dewan
Ira A. Fulton School of Engineering
Arizona State University
Tempe, AZ, USA
email: dewan@asu.edu

Partha Dasgupta
Ira. A. Fulton School of Engineering
Arizona State University
Tempe, AZ, USA
email: partha@asu.edu

ABSTRACT

Alice, a peer in a peer-to-peer (P2P) network can evaluate the reputation of another peer, Bob, either based on her own experiences (with Bob) or based on the evaluation (of Bob) by others (say Carol and David). If Alice uses her own experiences only, she will get cheated atleast once for every malicious peer she identifies. If Alice uses third party experiences (Carol and David), she can reduce the number of times she gets cheated. Besides, Alice will have to verify the third party information receives from other peers. The verification (if at all possible) is likely to be both network and computation intensive.

In the propose solution, the reputation holder (Bob) stores its reputation data and protects it from the other peers. The reputation data is stored in a cryptographic chain in order to protect the reputation data from the reputation holder itself. Bob cannot modify the chain because the head and tail of the chain are public information and all recommendations are digitally signed. We show that the proposed solution reduces the number of malicious transaction experienced by a peer (Alice), along with a reduction in network traffic.

KEY WORDS

Peer-to-Peer, Security, Trust, Reputation

1 Introduction

Pure P2P networks are completely decentralized and unregulated. These networks cannot be effectively policed, thereby making them more vulnerable to malicious activities, as compared to traditional client server networks. As a result the peers are highly vulnerable to dissemination of malicious or spurious content, besides being susceptible to malicious peers.

A large body of literature [1, 2, 3, 4] corroborates the fact that *reputation based systems* can make the P2P networks more secure, besides motivating the peers against cheating. This body of the reputation literature can be broadly categorized into three main groups: 1) systems in which peers use only their own experience (*local information*) for evaluating other peers [5], 2) systems in which peers use the experiences of other peers (*global information*) [2] and 3) systems in which peers use *both* local and global information [3]. In the absence of any central

authority the global information is generated from the local information (managed by each peer), using various decentralized schemes [1]. The global information is highly susceptible to peers that falsify their local information. Although the local information is more trustworthy for a peer, the global information considerably speeds up the process of identification of malicious peers, as peers learn from each other's transactions. This paper proposes a system for the secure ¹ generation, storage and disbursal of the global reputation data in an unmanaged P2P network.

In the proposed system a peer *owns* the reputation information pertaining to all its past *transactions* ² with other peers in the network and stores it locally. As a result every peer owns and stores its own reputation data and thereby protects the information from other malicious peers. The challenge now is to protect the reputation information from malicious modification by its owner. The proposed system consists of a reputation model coupled with a two-party cryptographic protocol. The proposed system not only protects the reputation information from its owner, but facilitates secure exchange of reputation information between the two peers participating in a transaction.

Each peer is identified by one and only one identifier and its reputation is associated with its identifier. This requirement has been relaxed in Section 5.2 and the possible repercussions have been enumerated. A peer (*requester*) looking for a specific file, uses the *search* function of an unstructured or a DHT based network and obtains a list of peers (and their corresponding reputations), who have the file. It downloads the file from the peer (*provider*) having the highest reputation. After the completion of the download, the requester provides a *recommendation* to the provider. The recommendation is positive if the downloaded file is satisfactory, and negative otherwise. The provider stores its recommendations in its local storage and manages them by itself. It accumulates its recommendation(s) to evaluate its own reputation. In the future transactions, the provider presents its reputation information to other peers, who request the information in order to ascertain its trustworthiness. Digital signatures of the requester(s) protect the recommendations against malicious modification by the provider (reputation owner). The

¹Non tamperable, authentic

²A *transaction* can be as simple as a transfer of a file from the source (peer) to the destination (peer) or as complex as a securities transaction

owner stores the recommendations in the form of a *recommendation chain* in which the signature of the $N - 1^{th}$ transaction is included in the N^{th} transaction. A signed record of the last transaction in the chain is stored in the network thereby making the complete chain verifiable. The transaction chain prevents any malicious addition or deletion of a recommendation by the owner. The main contributions of this paper are:

1. A light weight and simple reputation model.
2. An attack resistant cryptographic protocol for generation of authentic global reputation information w.r.t a peer.

2 Related Work

P2P networks are classified into two categories. The first category consists of the unstructured P2P networks of the class of Gnutella and the second category comprises the DHT based systems like Chord, CAN and Pastry. These networks are built on the all-peers-are-good premise and are intrinsically insecure.

Many techniques have been proposed to secure P2P networks. PGP is based on PKI and enables users to construct their own *web of trust* without using any central trusted server³. It is based on a peer-to-peer model, where each certificate represents a peer. A commercial application, Groove, builds self-administering, context-sensitive and synchronized share spaces for exchanging files in fine granularity. It ensures the secrecy of shared spaces and facilitates authentication of the members of the group. Lots of reputation models have been developed by researchers in order to emphasize the importance and usefulness of reputation systems [6, 7, 8].

Levien has proposed Advogato [9], which is based on the maxflow-mincut theorem. Schafer *et al.* [10] provide a good overview of the recommender systems being used in E-Commerce websites. Dellarocas [4] has enumerated the design challenges in the online reporting systems. In addition, he has reported a list of attacks on reputation systems and techniques for foiling those attacks. Aberer *et al.* [11] have proposed a completely distributed solution to the trust injection problem. They store reputation data in the form of a binary search tree, over the network, as explained in [11]. Any agent looking for the recommendation data of another agent searches the peer-to-peer network and computes the reputation from the recommendations received.

Another category of systems consists of the reputation based infrastructure implemented on Gnutella like P2PRep and RCert. In P2PRep [12], the reputation of a peer is a function of positive and negative votes polled from the current neighbors of the peer. Therefore in P2PRep the transactions of a peer with other peers that were neighbors of the peer in the past, do not impact its reputation. P2PRep is a stateless system and does not consider departure and arrival

³although PGP servers are available now but they more for convenience than necessity

of the same peer with different identity or a peer with multiple identities (one user controlling multiple peers). Peers in RCert [13] store their own reputation but the peer identity management is not a part of the RCert model.

3 Reputation Model & Reputation Exchange Protocol

We divide the system into two parts for easier exposition. The first part is an extremely light weight *reputation model* and the second part is a *cryptographic reputation exchange protocol*. The reputation model specifies the value of different reputation parameters, definition of functions used for calculation of reputation value from a set of recommendations, and the constraints on peer identities. The cryptographic protocol enumerates the steps for the exchange of reputation information such that the information cannot be altered maliciously.

3.1 Reputation Model

A peer (requester) joins the network using the standard *Join* method of the particular P2P network. The requester searches for one or more files using the *Search* method provided by the network. On the basis of the responses received, as a result of its search request, the requester generates a list of peers who have the requested file(s) and are ready to share the file(s). Besides the location of the peers, the responses to the search request contain the reputation information of the responding peers. Let RANGE be the number of the peers who offer a particular file. The requester selects the peer (provider) with the highest reputation from the list and initiates the cryptographic protocol. The cryptographic protocol is presented in detail in the next section. The requester uses the *Download* method of the P2P network, to download the file. It verifies the integrity, authenticity and the quality of the file. Depending on its verification results, it sends a recommendation between MIN_RECOMMENDATION and MAX_RECOMMENDATION to the provider. The recommendations are constrained to boundaries in order to make sure that one recommendation does not completely nullify or drastically raise the reputation of a provider. Once the provider receives the recommendation, it uses the function $F()$ over all the previous recommendations received by it and the recent recommendation to calculate its reputation. The above mentioned steps are repeated for every transaction.

There is a big body of work in *Decision Theory, Game Theory* and *Probability* [14, 15, 16] which can be used for selecting appropriate values of the above-mentioned parameters and the function $F()$ depending on the threat faced by the peers in the network. In this paper we define the function $F()$ as the arithmetic average of the recommendations received by the provider. *The reputation model is independent of the topology of the P2P network, address-*

ing schemes for its nodes, bootstrap mechanisms, joining and leaving protocols of peers. In other words the choice of any of these components has no impact on the reputation model and vice versa.

A unique identifier is associated with each peer. The identifier is a SPKI certificate [17] that is signed by a globally trusted *Certificate Authority* (CA). The user identities are different from the node identifiers used to route packets in P2P networks. It can be argued that a presence of a central CA will form a single point of failure or at least a bottleneck for smooth functioning of the network. As the CA is only used for one time issuance of identities and the periodic renewal of the identity certificates: 1) it will not experience a high request traffic 2) a decentralized P2P network is not dependent on 24*7 availability of the CA. Hence a CA downtime will have no or minimal impact on the network. Alternatively, the possibility of a hierarchical set of CA's has been explored in [18] and self-signed certificates have been reported in [19].

3.2 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 $R \rightarrow P : X$ 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. In the protocol, the words, 'Identity Certificate' and Identity have been used interchangeably. The steps in the reputation exchange protocol are as follows:

STEP 1) $R \rightarrow P : RTS \& IDR$

The requester sends a REQUEST FOR TRANSACTION (RTS) and its 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) $P \rightarrow R : IDP \& TID \& E_{P_{K2}}(H(TID||RTS))$

The provider sends its own IDENTITY CERTIFICATE (IDP), the CURRENT TRANSACTION ID (TID) and the signed TID, $E_{P_{K2}}(H(TID||RTS))$. 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. This attack is foiled by storing the LTID with multiple unrelated peers.

STEP 4) $R : IF (LTID \geq TID) GO TO STEP 12$

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 jumps to STEP 12.

STEP 5) $R \rightarrow P : Past Recommendation Request \& r$

If the check in the 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 N^{th} transaction for the provider, the requester makes a request for $N - 1^{th}, N - 2^{th}$ 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) $P \rightarrow R : CHAIN, E_{P_{K2}}(CHAIN)$
 $CHAIN = (\{REC1 || E_{Z1_{K2}}(H(REC1))\} ||$
 $\{REC2 || E_{Z2_{K2}}(H(REC2, REC1))\} ||$
 $\{REC3 || E_{Z3_{K2}}(H(REC3, REC2))\} || \dots$
 $\{RECr || E_{Zr_{K2}}(H(RECr, RECr-1))\})$

The provider sends its past recommendations (REC1, REC2... RECr) which were provided by peers (Z1, Z2,... Zr). 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 Zr) has hashed both the (r^{th}) and the previous ($r - 1^{th}$) recommendation using its private key Z_{K2} , as $E_{Zr_{K2}}(H(RECr, RECr-1))$, 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) $P : Result = Verify (REC1, REC2... RECr)$
If Result != Verified GO TO STEP 12

The provider 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. If the verification fails the requester jumps to STEP 12.

STEP 8) $P \rightarrow R$: 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) $R \rightarrow P$: $B1 = EB_{K_a}(REC \parallel TID \parallel E_{P_{K_2}}\{H(REC \parallel TID)\})$

Once the requester has received a service, it generates a BLINDING KEY, K_a . The requester concatenates the RECOMMENDATION (REC) and the TRANSACTION ID (TID) it had received in Step 1 and signs it. Subsequently, it blinds the signed recommendation with the blinding key, K_a . 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 negative. The provider receives the blinded recommendation from the requester. The blinded recommendation is also signed by the requester.

STEP 10) a) $P \rightarrow R$: $B1 \parallel E_{P_{K_2}}(H(B1), nonce), nonce$
b) $R \rightarrow P$: K_a

The provider cannot see the recommendation but it signs the recommendation and send the NONCE and the signed recommendation back to the requester. The requester verifies the signature and then sends the blinding key K_a to the provider which can unblind the string received in Step 10 a) and check its recommendation.

STEP 11) Insert (IDR, { REC || TID || $E_{R_{K_2}}\{H(REC \parallel H(TID))\}$ }

The requester signs its the recommendation 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 || $E_{R_{K_2}}\{H(CHAIN \parallel H(TID))\}$ }

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

they have to include TID signed by the provider. If the requester reaches STEP 12 from STEP 4. It will perform

R : Insert (IDR, { CHAIN || TID || $E_{R_{K_2}}\{H(CHAIN \parallel H(TID))\}$ }

4 Analysis of the Protocol

The requester needs to initiate only one search request in the network in order to collect the recommendations received by the provider in the past. This protocol handles the problem of irregular availability of the peers in the network, which is one of the major problems in P2P networks. This protocol does not prevent *bad mouthing* and *ballot stuffing* [4] but only mitigates their effect as it is resilient to collusion.

4.1 The provider will not (intentionally) send the wrong TID in Step 2.

Let the id that the provider sends be TID' and the last Transaction Id for the provider be LTID. The TID' should always be equal to LTID+1. If TID' > LTID+1, then there will be unexplained missing recommendations. If TID' < LTID+1, then the provider will be caught in Step 4 of the protocol, as the last id used by the provider was made a public information (by the previous requester) and is available to all the peers now. If a peer is taking the role of a provider for the first time, then the TID will be 0.

4.2 The provider will not abort the transaction in step 8.

It is possible for the provider to abort the transaction after giving the requester the requested information or file in Step 8. It is also possible for requester to abort the transaction after step 9. In both scenarios, the provider will not have a recommendation for the transaction id TID. If the provider does not sign the blinded recommendation that the requester sent her, the requester can release the recommendation in step 11 without obtaining provider's signature.

In its next transaction TID+1, the provider will not be able to show the recommendation for transaction, TID to the requester of transaction, TID+1. Therefore, the new requester will search the network using the Search method for TID. If it finds TID, it will also find the recommendation provided to the provider in the transaction. The requester will be accountable because the TID was signed by the provider. The provider will have to accept the recommendation because it will also include the signature of the provider, TID & $E_{P_{K_2}}(H(TID))$. If the new requester does not find the recommendation, then it can believe the provider and provide him minimal recommendation for the transaction, TID.

If the provider returns the signed blinded recommendation in step 10, $B1 \parallel E_{P_{K_2}}(H(B1))$, but the requester

does not send the key, Ka and jumps to Step 10 without performing the intermediate steps, then the provider can search the network and get the signed recommendation of the requester. If the requester never performs Step 10, then in the next transaction, TID+1 the new requester will search for LTID and he will not find it. Hence the transaction TID can be considered as aborted and the next transaction can be done with transaction id, TID.

4.3 Collusion by Rogues

Two or more rogues might collude in order to increase each others reputation. The requester can sort the recommendations by public key in Step 7 of the protocol, and if the provider has "too many" recommendations from a given public key, the requester suspects collusion and aborts the protocol.

5 Evaluation

The proposed system was simulated on a 2.4 GHz machine using SUSE Linux 9. The simulation consisted of 5000 peers participating in 20,000 - 160,000 transactions. The requesters did not have any apriori knowledge of rogue nodes and the rogue nodes performed maliciously with a probability of 1/2 i.e. rogues cheated in 1 out of every 2 transactions. Each simulation was performed 5 times.

5.1 Cumulative & Individual benefits of using reputation

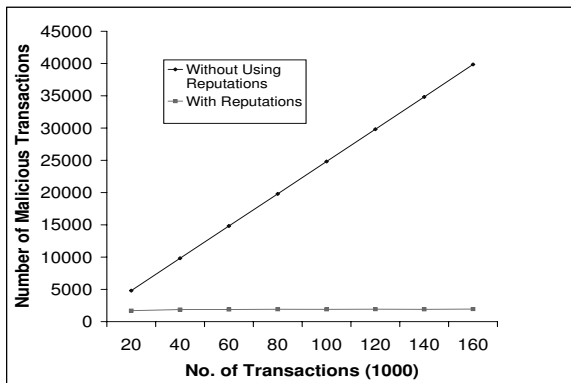


Figure 1. Reduction in Number of Total number of Malicious Transactions when Reputations are used

In the first part of the simulation we wanted to quantify the benefits of using the proposed reputation model for a peer-to-peer network. The number of rogues was set to a constant at 50% and the number of transactions was incrementally raised from 20,000 to 140,000. As is visible in Figure 1 the total number of malicious transactions increases considerably with an increase in the number of transactions when the proposed model is not used but are more or less constant when the proposed model is used.

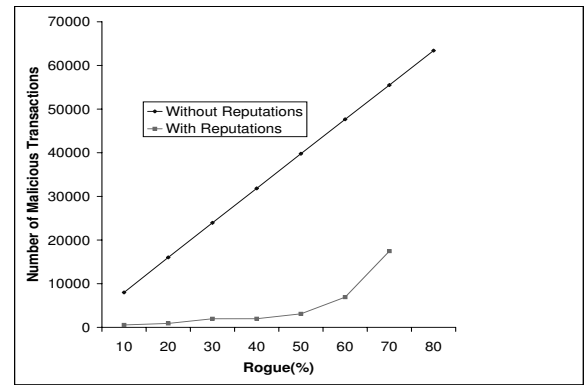


Figure 2. A comparison of number of malicious transactions with a variation in the number of rogues

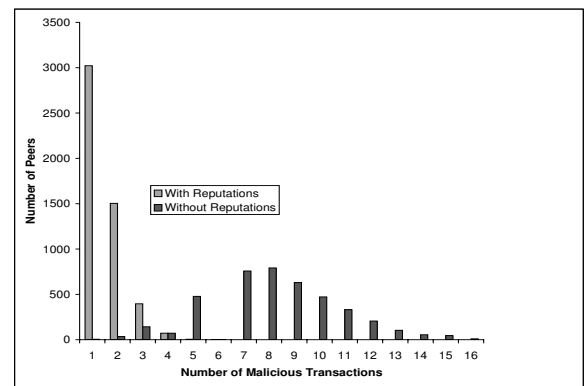


Figure 3. Distribution of Malicious Transactions across 5000 Peers

When the reputations were not used the mean of the number of malicious transactions experienced by each good node is 7.966 ± 5.52 with a 95% confidence (Refer Figure 3). This mean drastically reduces when the reputation model is used, to 0.4 ± 1.2 with a 95% confidence.

In the second part of the simulation we intended to estimate the impact of variation in the number of malicious nodes on the network. The number of transactions was kept constant to 140,000 transactions for 5000 nodes. We varied the number of rogue nodes from 10% to 90%. As visible in Figure 2, there was a considerable reduction in the number of malicious transactions when reputations were used as compared to the event when they were not used. In the presence of an increasing number of rogues, the rate of increase in the number of malicious transactions is much less when reputations are used.

We compared the cryptographic protocol with the reputation based system presented in [12], called P2PRep. 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 highly communication intensive because of its

statelessness. On the other hand the proposed system uses minimal communication as the security of the system lies in the cryptographic techniques used.

5.2 Identities of Peers

In this paper we have assumed that each peer possesses only one identity. Identity management for reputation based systems in P2P networks is an important problem but is not addressed in this paper. More information about identity management can be found in [20, 21]. If the peers are able to obtain multiple identities then the requester has to ensure that the identities that recommend a peer are not owned by the peer itself. Dewan [22] uses IP-Based Safe-guard to protect the system from peers who use their own identities to give recommendations to themselves.

In the scenario where the peers have multiple identities, the verification step of the cryptographic protocol would have to be changed to not only ensure the 'correctness' of the recommendations of the provider but also ensure that the identities that provide the recommendations belong to distinct peers.

6 Conclusions & Future Work

This paper presents a reputation model and a cryptographic protocol that facilitates generation of global reputation data in a peer-to-peer network, in order to expedite detection of rogues. The global reputation data is protected against any malicious modification by the third party peer and is immune to modifications by its 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] S. D. Kamvar, M. T. Schlosser and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. *Proceedings of Twelfth International World Wide Web Conference*, 2003.
- [2] E. Damiani, S. D. C. di Vimercati, S. Paraboschi, and P. Samarati. Managing and sharing servants' reputations in p2p systems. *IEEE Transactions on Knowledge and Data Engineering*, 15(4), 2003, 840-854.
- [3] Y. Wang. Bayesian network-based trust model in peer-to-peer networks. *Proceedings of Workshop on "Deception, Fraud and Trust in Agent Societies" at the Autonomous Agents and Multi Agent Systems 2003 Conference (AAMAS-03)*, Melbourne, Australia. 2003.
- [4] C. Dellarocas. Building trust on-line : the design of reliable reputation, *Sloan school of Management and College for eBusiness at MIT.*, Technical Report. 2001.
- [5] P. Dewan, P. Dasgupta & A. Bhattacharya. On using reputations in ad hoc networks to counter malicious nodes. *Proceedings of QoS and Dynamic Systems Workshop, International Conference for Parallel and Distributed Systems (ICPADS04)*. 2004.
- [6] L. Xiong. Peertrust: Supporting reputation-based trust in peer-to-peer communities. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*. IEEE, 2004.
- [7] A. Abdul-Rahman and S. Hailes, Supporting trust in virtual communities. *Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 6*, Washington DC, USA. 2000.
- [8] M. Chen and J. P. Singh. Computing and using reputations for internet ratings. *Proceedings of the 3rd ACM conference on Electronic Commerce*, Tampa, Florida, USA. 2001, 154-162.
- [9] R. Levien. Advogato. Webpage, 2003.
- [10] J. B. Schafer, J. A. Konstan, and J. Riedi. Recommender systems in e-commerce. *Proceedings of ACM Conference on Electronic Commerce*. 1999, 158-166.
- [11] K. Aberer and Z. Despotovic. Managing trust in a peer-to-peer information system. *Proceedings of 10th International Conference on Information and Knowledge Management (CIKM01)*, New York, USA. 2001, 310-317.
- [12] E. Damiani, D. C. Di Vimercati DEA and others. A reputation-based approach for choosing reliable resources in peer-to-peer networks. *Proceedings of Conference on Computer and Communications Security (CCS 02)*, Washington DC, USA. 2002, 207-216.
- [13] K.-L. T. Beng Chin Ooi, Chu Yee Kiau. Managing trust in peer-to-peer systems using reputation-based techniques. *Proceedings of 4th International Conference on Web Age Information Management*. Chengdu, China. 2003.
- [14] G. Shafer and J. Pearl. *Readings in uncertain reasoning* (Morgan Kaufmann series in representation and reasoning, 1990)
- [15] F. K. Robert A. Wilson. *The MIT Encyclopedia of the Cognitive Sciences (MITECS)*. Bradford Books, 1999.
- [16] D. P. Foster and H. P. Young. On the impossibility of predicting the behavior of rational agents. *John Hopkins University*. Technical Report. 1999.
- [17] R. L. Rivest and B. Lampson. SDSI: a simple, distributed, security infrastructure. *Proceedings of Crypto*, Santa Barbara, USA. 1996.
- [18] K. Jiang and P. Dasgupta, "A secure information management system for large-scale dynamic coalitions, *Proceedings of The 2nd. Darpa Information and Security Conference and Exposition*. July 2001.
- [19] P. Dewan. Injecting trust in peer-to-peer systems. *Ph.D. Dissertation Proposal*, Arizona State University, 2003.
- [20] J. Douceur. The Sybil attack. *Proceedings of IPTPS02 Workshop*, California, 2002.
- [21] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. S. Secure routing for structured peer-to-peer overlay networks. *Proceedings of Fifth Symposium on Operating Systems Design and Implementation*, Boston. 2002.
- [22] P. Dewan, P. Dasgupta. PRIDE: Peer-to-peer Reputation Infrastructure for Decentralized Environments *Proceedings of Alternate track papers & posters of the 13th international conference on World Wide Web (WWW2004)*. 2004.