Peer-to-Peer Reputations

Prashant Dewan
Department of Computer Science and Engineering
Ira A. Fulton School of Engineering
Arizona State University
Tempe, AZ 85287 - 8809
dewan@asu.edu

Abstract

Peer-to-peer (P2P) networks are designed with an assumption that the nodes in a P2P network will cooperate each other. In the absence of any common goals shared by the nodes of a P2P network, external motivation to cooperate and be trustworthy is required. Digital Reputations can be used to inject trust among the autonomous nodes of a network and motivate the nodes to contribute resources. This paper summarizes a self-certification scheme for the identification of peers using digital certificates similar to SDSI certificates, techniques to mitigate the problem of 'a consortium of liars' and an elicitation-storage protocol for procuring and storing recommendations.

1. Introduction

In P2P networks a peer can cheat other peers by not performing what it promised to do (or ought to do). For example a malicious peer might send a corrupt copy of the file requested by the other peer. Hence motivation to abstain from cheating has to be injected explicitly. Digital reputations (a representation of the history of on line transactions performed by a peer in the system) can be used to inject the necessary motivation [3, 9, 14]. The proposed self-certification scheme enables a peer in a P2P network of the class of Gnutella [4], to run a Certificate Authority (CA) to generate identities for itself. A peer is restricted from generating a large number of identities, called an Identity Farm, by using 'IP Based Safeguard' and 'Reference Based Prevention'. Lastly the peer that owns(i.e. to whom the recommendation was granted) the reputation stores its own reputation information locally. Therefore neither the other peers have to search the network for the reputation information nor are they dependent on any third node.

2. Digital Reputations

One of the goals of a digital reputation system is to enable a peer to determine the likelihood of achieving the desired level of satisfaction from its next transaction. This is achieved by ranking the service providers on the basis of their reputation accrued from their past interactions. The reputation information in the reputation-based systems used by websites like Amazon, Epinions, and EBay is stored at a central location, which forms a single point of failure and requires peers to trust the central authority. Most of the reputation models, like P2PRep [5] and RCert [13], are based on Gnutella and use the servent id (hash of its IP address) to identify peers. Unlike identities generated using self-certification, servent id is not always 'owned' by the same peer and hence is inappropriate for a reputation-based system. Other reputation models [2, 11, 12] assume that each peer has only one identity. While peer reputation in P2Prep, which is based on voting and the reputation information, is not stored anywhere; EigenTrust [10] is based on Distributed Hash Table (DHT), and the peers use DHT properties to store reputation information in the network. P2Prep assumes that the neighbors of a node will know whether the node is a good node or a rogue. This paper does not make any such assumptions.

3. Peer Identity Management

The relation between the peer identity and the peer(a human being or a computer)for any P2P network should be one-to-one or many-to-one. Douceur has shown in [8] (also acknowledged in [15, 1]) that in the absence of a centrally trusted party or an external verification, one-identity-for-one-entity cannot be enforced in a decentralized distributed system (e.g. a pure P2P network) except in certain impractical circumstances. Self-certification enables the peers to generate their own identities. Each peer's CA issues identity

Proceedings of the 18th International Parallel and Distributed Processing Symposium (IPDPS'04)

0-7695-2132-0/04/$17.00 (C) 2004 IEEE

certificates (SDSI certificates [15]) to the peer. Using self-certification a peer can generate a large number identities-unless a peer is restricted explicitly. Identity farms can become the Achilles heel of a reputation-based system. In other words, a peer can use the identities in an identity farms to give false recommendations to a subset of the identities in the farm and raise the reputation of the subset. The set of identities that issue false recommendations is called a Liar Farm. 'IP Based Safeguard' (IBS) and 'Reference Based Prevention' (RBP) can be used to mitigate the impact of liar farms.

'IP Based Safeguard' uses a security zone that is a subsection of the linear IP space, and its size is set by a parameter called security distance(SD) which is defined as the number of IP's in one security zone and denoted by d. Only the average of the recommendations from a given security zone is included in the calculation of the peer's reputation. Each peer can choose its own value of d. A higher value of d provides a higher protection against an identity farm, but less accurate reputation, and vice versa. As mentioned before, one of the main goals of a reputation system is to rank other peers. Therefore the relative ranking of the peers is more informative than their absolute reputation values. Experiments show that use of IBS with self-certification in Gnutella, only causes minimal changes to the relative ranks of the peers. An important thing to note here is that, IBS does not prevent identity farms but only safeguards a peer from being a potential victim to an identity farm. Currently we are trying to find out ways to make IBS work with NAT and anonymizers.

'Reference Based Prevention' makes the generation of an identity farm difficult for a peer. For every identity generated by a peer, the peer has to show a Total Reference Inflow (TRI) greater than the threshold TRI. Only a peer that has a verifiable identity (issued outside the system), like an X.509 certificate is eligible to be a referrer. This implies that a new peer has to find other peers that have non-anonymous identities with good reputations. In addition, the reference provider becomes accountable for the references that it provides; i.e., the recommendations received by the referred peer, percolate to the referrer (identity of the referrer used for granting the reference) . Currently we are trying to design bootstrap mechanisms such that the threshold TRI can be achieved by peers at system startup. A more detailed discussion of both these techniques can be found in [6].

4. Self-storage of Reputation Information

When the reputation owner stores its reputation information locally, it necessitates prevention of malicious modification of the information by the owner. The elicitation-storage protocol cryptographically prevents malicious modification of reputation information and facilitates self-storage of the reputation information.

4.1. Elicitation-Storage(ES) Protocol

The requesting peer (requester) obtains a list of information providers (providers) who have the information or the content, satisfying the peer's query. The requester also receives the reputation of each of the providers in the list. The requester selects the 'best' peer based on the reputation of the possible providers and initiates the ES protocol. We assume that only the information provider receives the recommendation, and the information requester provides the recommendation.

On the requester's initiation, the provider that is selected by the requestor on the basis of its reputation, generates a new transaction id (TID) by using the last transaction id as a seed for a one-way function. The requester verifies if the same TID has been used for any other transaction by the provider. Once the TID is verified, the requester checks (at least some of) the past recommendations[1] received by the provider and, once satisfied, performs the transaction. In order to verify recommender identity, the requester performs a challenge response with the past recommenders of the provider by contacting the recommenders at the IP address in their identity certificate. If the verification fails, due to the unavailability of the peer at the IP address, or if the peer disowns the public key present in the identity certificate of the peer, the requestor recalculates the reputation of the provider by omitting unverified recommendations. In other words, the identity is considered to be anonymous and the recommendations provided by anonymous identities are not included in the reputation of the provider (both for IBS and RBP). Following IBS the requestor recalculates the reputation of the peer by averaging the recommendations received in each security zone. If the recalculated value of peer's reputation is above the requestor's threshold it performs the transaction and if not it contacts the second peer in the list and so on. Once the transaction (file download) is complete, the requester gives a signed recommendation to the provider, which is stored by the provider. In addition, the requestor signs the TID and stores it in the P2P network. The details of the protocol can be found in [7]

5. Conclusion

In the absence of a centrally trusted party or an external verification, one-identity-for-one-entity cannot be enforced in a decentralized distributed system. Self-certification by peers offers an interesting alternative to centralized issuance of identities. While IBS protects peers against liar farms;

[1] Recommendation is the reputation information pertaining to one interaction between two distinct peers

0-7695-2132-0/04/$17.00 (C) 2004 IEEE

RBP makes generation of identity farms (and hence liar farms) more difficult. Self-storage of reputation information by the owner necessitates cryptographic protocol to protect the reputation data from malicious modification. The peers in a reputation-based system can weed out the rogues by giving them bad recommendations and refusing to interact with peers with low reputation. The fear of bad reputation gives the providers an incentive to provide accurate and timely information in order to obtain good recommendations from requesters. [7]

References

[1] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. S. Wallach. Secure routing for structured peer-to-peer overlay networks. *SIGOPS Oper. Syst. Rev.*, 36(SI):299–314, 2002.

[2] M. Chen and J. P. Singh. Computing and using reputations for internet ratings. In *Proceedings of the 3rd ACM conference on Electronic Commerce*, pages 154–162. ACM Press, 2001.

[3] U. o. M. Chrysanthos Dellarocas, MIT Paul Resnick. Online reputation mechanisms: A roadmap for future research, 6 2003.

[4] Clip2. The gnutella protocol specification v0.41. Technical report, Clip2.com, 2002.

[5] E. Damiani, D. C. di Vimercati, S. Paraboschi, P. Samarati, and F. Violante. A reputation-based approach for choosing reliable resources in peer-to-peer networks. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 207–216. ACM Press, 2002.

[6] P. Dewan and P. Dasgupta. Countering identity farms in p2p networks. Technical report, Arizona State University, Tempe, AZ 85281, USA, 10 2003.

[7] P. Dewan and P. Dasgupta. Pride: Peer-to-peer reputation infrastructure for decentralized environments. Technical report, Arizona State University, Tempe, AZ 85281, USA, 07 2003.

[8] J. Douceur. The sybil attack. In *Proceedings of the IPTPS02 Workshop*, 2002.

[9] M. N. Doukidis G. and P. N. *Information Society or Information Economy? A combined perspective on the digital era*, chapter 1. Idea Book Publishing, 2004.

[10] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the Twelfth International World Wide Web Conference (WWW)*, 2003.

[11] H. Kung and C.-H. Wu. Differentiated admission for peer-to-peer systems: Incentivizing peers to contribute their resources, 6 2003.

[12] H. Lee and K. Kim. An adaptive authentication protocol based on reputation for peer-to-peer system, 2003.

[13] B. C. Ooi, C. Y. Liau, and K.-L. Tau. Managing trust in peer-to-peer systems using reputation-based techniques, 8 2003.

[14] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman. Reputation systems. *Commun. ACM*, 43(12):45–48, 2000.

[15] R. L. Rivest and B. Lampson. SDSI – A simple distributed security infrastructure. Presented at CRYPTO96 Rumpsession, 1996.