International Journal of Research in Engineering, Science and Management
Volume-1, Issue-11, November-2018
www.ijresm.com | ISSN (Online): 2581-5792

SORT-That Aim to Decrease Malicious Activity

Shilpa A. Irlapalle[1], B. M. Patil[2]
[1]*ME Student, Department of CNE, MBES's College of Engineering, Ambajogai, India*
[2]*Professor, Department of CNE, MBES's College of Engineering, Ambajogai, India*

***Abstract*: Anonymous nature of peer-to-peer (P2P) systems exposes them to malicious activity. Establishing trust among peers can mitigate attacks from malicious peers. This paper presents distributed algorithms used by a peer to reason about trustworthiness of others based on the available local information which includes past interactions and recommendations received from others. Peers collaborate to establish trust among each other without using a priori information. A peer's trustworthiness in providing services and giving recommendations is evaluated in service and recommendation in trust. Defining trust metric in separate contexts makes possible to measure trustworthiness of peers more precisely. Peer may be a good service provider and a bad recommender at the same time. Interactions among peers have varying importance. An interaction loses its importance with time. These effects are considered along with the satisfaction of peers while evaluating an interaction. A recommendation contains the recommender's confidence in the information provided. This factor is considered with trustworthiness of the recommender when evaluating recommendations. A file sharing application simulated to understand advantages of the proposed algorithms in mitigating attacks related with services and recommendations. The results of several empirical studies are used to simulate peer, resource, and network parameters. This enables us to study the effects of external parameters on the algorithms and the evolution of trust relationships among peers. Individual, collaborative and pseudonym changing attack scenarios simulate nine different malicious behaviors. In most experiments, we find that malicious peers are isolated from other peers and their attacks are mitigated. There are cases where they obtain a high reputation but their attacks are still contained.**

***Keywords*: Recommendation, reputation, trustworthiness, security**

1. Introduction

Peer-to-peer (p2p) systems rely on collaboration of peers to accomplish tasks. Ease of performing malicious activity is a threat for security of P2P systems. Creating long-term trust relationships among peers can provide a more secure environment by reducing risk and uncertainty in future P2P interactions. However, establishing trust in an unknown entity is difficult in such a malicious environment. Furthermore, trust is a social concept and hard to measure with numerical values. Metrics are needed to represent trust in computational models. Classifying peers as either trustworthy or untrustworthy is not sufficient in most cases. Metrics should have precision so peers can be ranked according to trustworthiness. Interactions and feedbacks of peers provide information to measure trust among peers. Interactions with a peer provide certain information about the peer but feedbacks might contain deceptive information. This makes assessment of trustworthiness a challenge. A peer sends trust queries to learn trust information of other peers. We propose a Self-Organizing Trust model (SORT) that aims to decrease malicious activity in a P2P system by establishing trust relations among peers in their proximity. No a priori information or a trusted peer is used to leverage trust establishment. Peers do not try to collect trust information from all peers. Each peer develops its own local view of trust about the peers interacted in the past. In this way, good peers form dynamic trust groups in their proximity and can isolate malicious peers. Since peers generally tend to interact with a small set of peers, forming trust relations in proximity of peers helps to mitigate attacks in a P2P system. In SORT, peers are assumed to be strangers to each other at the beginning. A peer becomes an acquaintance of another peer after providing a service, e.g., uploading a file. If a peer has no acquaintance, it chooses to trust strangers. An acquaintance is always preferred over a stranger if they are equally trustworthy. Using a service of a peer is an interaction, which is evaluated based on weight (importance) and recentness of the interaction, and satisfaction of the requester. An acquaintance's feedback about a peer, recommendation, is evaluated based on recommender's trustworthiness. It contains the recommender's own experience about the peer, information collected from the recommender's acquaintances, and the recommender's level of confidence in the recommendation. If the level of confidence is low, the recommendation has a low value in evaluation and affects less the trustworthiness of the recommender. A peer may be a good service provider but a bad recommender or vice versa. Thus, SORT considers providing services and giving recommendations as different tasks and defines two contexts of trust: service and recommendation contexts. Information about past interactions and recommendations are stored in separate histories to assess competence and integrity of acquaintances in these contexts. SORT defines three trust metrics. Reputation metric is calculated based on recommendations. It is important when deciding about strangers and new acquaintances. Reputation loses its importance as experience with an acquaintance increases. Service trust and recommendation trust are primary metrics to measure trustworthiness in the service and recommendation contexts, respectively. The service trust metric is used when selecting service providers. The recommendation trust metric is important when requesting

recommendations. When calculating the reputation metric, recommendations are evaluated based on the recommendation trust metric. Since there is no central server in most P2P systems, peers organize themselves to store and manage trust information about each other.

2. Literature survey

In the presence of an authority, a central server is a preferred way to store and manage trust information, e.g., eBay. The central server securely stores trust information and defines trust metrics. Since there is no central server in most P2P systems, peers organize themselves to store and manage trust information about each other [1], [2]. Management of trust information is dependent to the structure of P2P network. In distributed hash table (DHT) - based approaches, each peer becomes a trust holder by storing feedbacks about other peers [1], [3], [4]. Global trust information stored by trust holders can be accessed through DHT efficiently. In unstructured networks, each peer stores trust information about peers in its neighborhood or peers interacted in the past [2], [5], [6]. Since peers generally tend to interact with a small set of peers [7], forming trust relations in proximity of peers helps to mitigate attacks in a P2P system. In SORT, peers are assumed to be strangers to each other at the beginning. A peer becomes an acquaintance of another peer after providing a service, e.g., uploading a file. If a peer has no acquaintance, it chooses to trust strangers. An acquaintance is always preferred over a stranger if they are equally trustworthy. We implemented a P2P file sharing simulation tool and conducted experiments to understand impact of SORT in mitigating attacks. Parameters related to peer capabilities (bandwidth, number of shared files), peer behavior (online/ offline periods, waiting time for sessions), and resource distribution (file sizes, popularity of files) are approximated to several empirical results [8], [9], [10]. This enabled us to make more realistic observations on evolution of trust relationships. We studied 16 types of malicious peer behaviors, which perform both service and recommendation- based attacks. SORT mitigated service-based attacks in all cases. Recommendation-based attacks were contained except when malicious peers are in large numbers, e.g., 50 percent of all peers. Experiments on SORT show that good peers can defend themselves against malicious peers without having global trust information. SORT's trust metrics let a peer assess trustworthiness of other peers based on local information. Service and recommendation contexts enable better measurement of trustworthiness in providing services and giving recommendations.

3. Proposed system architecture

We define secure routing and outline our solution. Throughout this paper, most of the analyses and techniques are presented in terms of this model and should apply to other structured overlays except when otherwise noted. We define an abstract model of a structured Distributed routing overlay, designed to capture the key concepts common to overlays such as CAN, Chord, Tapestry and Pastry. The protocol routes messages with a given key to its associated root. To route messages efficiently, all nodes maintain a routing table with the node IDs of several other nodes and their associated IP addresses. Moreover, each node maintains a neighbor set, consisting of some number of nodes with node IDs nearest itself in the id space. Pastry node IDs are assigned randomly with uniform distribution from a circular 128-bit id space. Given a 128-bit key, Pastry routes an associated message toward the live node whose node ID is numerically closest to the key. Each Pastry node keeps track of its neighbor set and notifies applications of changes in the set. Secure routing ensures that (1) the message is eventually delivered, despite nodes that may corrupt, drop or misroute the message; and (2) the message is delivered to all legitimate replica roots for the key, despite nodes that may attempt to impersonate a replica root. Secure routing can be combined with existing security techniques to safely maintain state in a structured Distributed overlay. For instance, self-certifying data can be stored on the replica roots, or a Byzantine-fault-tolerant replication algorithm [10] can be used to maintain the replicated state. Secure routing guarantees that the replicas are initially placed on legitimate replica roots, and that a lookup message reaches a replica if one exists. Similarly, secure routing can be used to build other secure services, such as maintaining file metadata and user quotas in a distributed storage utility. The details of such services are beyond the scope of this paper. Fig 1. Shows network architecture. In this dig downloading a file is an interaction. A peer sharing files is called an up loader. A peer downloading a file is called a downloader. The set of peers who downloaded a file from a peer are called downloaders of the peer. An ongoing download/ upload operation is called a session. A good peer uploads authentic files and gives fair recommendations. A malicious peer (attacker) performs both service and recommendation-based attacks. Four different attack behaviors are studied for malicious peers: naive, discriminatory, hypocritical, and oscillatory behaviors. A non-malicious network consists of only good peers. A malicious network contains both good and malicious peers.

4. Forward security model

PEER to PEER algorithms enable a peer to reason about trustworthiness of other peers based on past interactions and recommendations. By using local information available, Peers create their own trust network in their proximity and do not try to learn global trust information. In proposed system peers do not collect information of all pairs in the network they only keep information of neighbors. This system has following main roles: A) Service trust matric B) Reputation Trust Metric. C) Recommendation Trust Metric.

Peers establishment: Self-Organizing Trust model (SORT) that aims to decrease malicious activity in a P2P system by establishing trust relations among peers in their proximity. Peers do not try to collect trust information from all peers. Each peer develops its own local view of trust about the peers

interacted in the past. In this way, good peers form dynamic trust groups in their proximity and can isolate malicious peers.

Files uploading, downloading: peers are assumed to be strangers to each other at the beginning. A peer becomes an acquaintance of another peer after providing a service, e.g., uploading a file. If a peer has no acquaintance, it chooses to trust strangers. An acquaintance is always preferred over a stranger if they are equally trustworthy. Using a service of a peer is an interaction, which is evaluated based on weight (importance) and recentness of the interaction, and satisfaction of the requester.

Service Trust Matric: Using the information in its service history a peer first calculates competence and integrity belief values when evaluating an acquaintance's trustworthiness in the service context. How well an acquaintance satisfied the needs of past interactions represented by Competence belief? Let in the service context the friend request denote the competence belief of P_i about. Average behavior in the past interactions is a measure of the competence belief. Consistency is as important as competence. Integrity belief is the level of confidence in predictability of future interactions. Let in the service context, I denote the integrity belief of P_i about. A measure of the integrity belief.

Reputation Trust Matric: The reputation metric measures a stranger's trust worthiness Based on recommendations. We assume that is a stranger to P_i and is an acquaintance of P_i in the following two sections [4]. If P_i starts a reputation query to collect recommendations from its acquaintances, if it wants to calculate r_{ij} value [2]. Trustworthy acquaintances and requests their recommendations. Let the maximum number of recommendations denoted by max that can be collected in a reputation query and the size o S_{ij} if a set S denoted by S_{ij} . P_i Sets a high threshold for recommendation trust values and requests recommendations from highly trusted acquaintances first, in the getting recommendation algorithm.

Recommendation Trust Metric: Assume that a particular service want to get to P_i.P_j a probable service provider and is a stranger toP_i. P_i Requests recommendations to learn P_j's reputation, from its acquaintances. Assume that recommendation send back to P_i from P_k[8]. After collecting all recommendations P_i calculates r_{ij}. Then, P_k's recommendation evaluates P_i, and stores results in rh_{ik}, and also updates rt_{ik}. Assuming P_j is trustworthy enough, P_i sets the service from P_j.Then, $P_i$and stores the results in sh_{ij}, and updates st_{ij} by evaluating this interaction.

5. Algorithm

A. *Get-recommendation algorithm*

Here we use get recommendation algorithm for correct feedback. The Recommendation algorithm is used here which is having following steps.

- First initialize the peers in the network.
- Then initialize threshold value

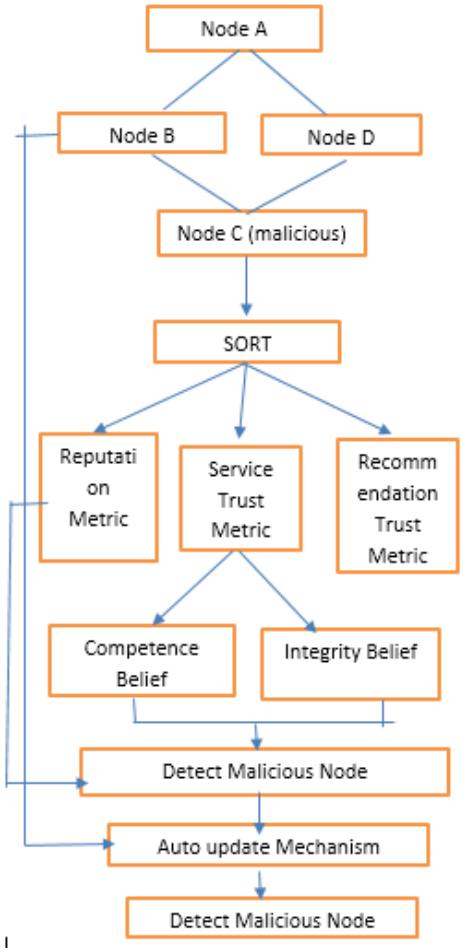

Fig. 1. Proposed system architecture

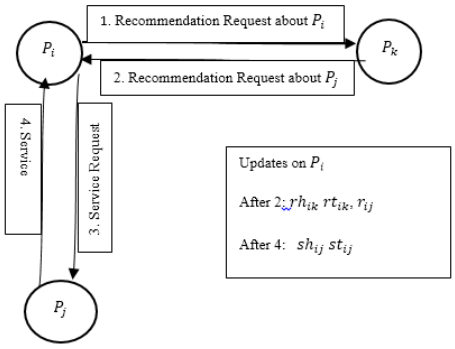

Fig. 2. Network architecture

- Trust values Calculate threshold for recommendation.
- Calculate the threshold from highly trusted acquaintances for requests recommendations.
- Evaluate Recommendation according to trust value of the recommender.
- Decreases the threshold and repeats the same process.
- When maximum recommendations are collected, if excessive network traffic then the algorithm stops.

6. Results and discussion

For the proposed system performance evaluation, we deploy the system on java 3-tier MVC architecture framework with INTEL 3.0 GHz i7 processor and 8 GB RAM. Here each graph shows the system performance with different experiments that has been classified in graphs. Below graph defines three trust metrics. Reputation metric is calculated based on recommendations. It is important when deciding about strangers and new acquaintances. Reputation loses its importance as experience with an acquaintance increases. Service trust and recommendation trust are primary metrics to measure trustworthiness in the service and recommendation contexts, respectively. The service trust metric is used when selecting service providers. The recommendation trust metric is important when requesting recommendations. Second graph shown below is the graph for recommendation is evaluated based on recommender's trustworthiness. It contains the recommender's own experience about the peer, information collected from the recommender's acquaintances, and the recommender's level of confidence in the recommendation. If the level of confidence is low, the recommendation has a low value in evaluation and affects less the trustworthiness of the recommender.

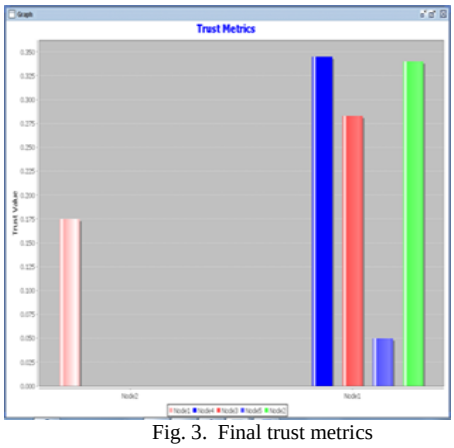

Fig. 3. Final trust metrics

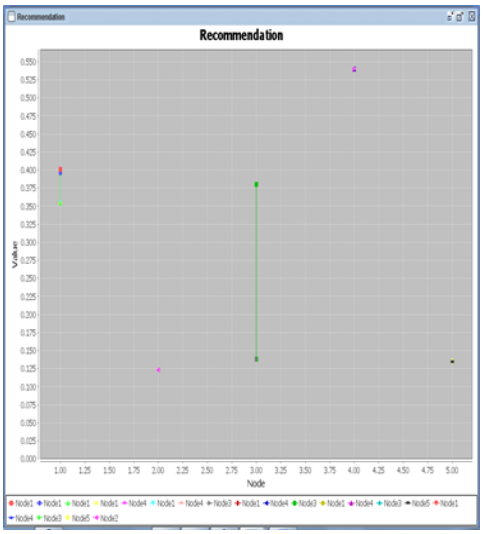

Fig. 4. Final recommendation metrics

Peername	Giver	rating	Feedback
system-2	system-1	7.5	good peer
system-1	system-2	8.5	Trustable peer
system-3	system-5	8.0	very nice peer
system-1	system-2	9.0	good peer
system-1	system-5	6.5	Average peer
system-4	system-5	9.0	exelent peer
system-1	system-2	1.5	great response

Fig. 5. All peer feed back

Above figure shows the module describing the feedbacks of all the peers.

Sysna...	s1	s2	s3	s4	s5
system...	yes	no	no	no	no

Fig. 6. Neighborhood peer status

7. Conclusion

We discussed a trust model for P2P networks is presented, in which a peer can develop a trust network in its proximity. A peer can isolate malicious peers around itself as it develops trust relationships with good peers. Two context of trust, service and recommendation contexts, are defined to measure capabilities of peers in providing services and giving recommendations. Interactions and recommendations are considered with satisfaction, weight, and fading effect parameters. A recommendation contains the recommender's own experience, information from its acquaintances, and level of confidence in the recommendation. These parameters provided us a better assessment of trustworthiness. Another issue about SORT is maintaining trust all over the network. If a peer changes its point of attachment to the network, it might lose a part of its trust network. These issues might be studied as a future work to extend the trust model. Using trust information does not solve all security problems in P2P systems but can enhance security and effectiveness of systems. If interactions are modeled correctly, SORT can be adapted to various P2P applications, Peer-to-peer (P2P) systems, peers often must interact with unknown or unfamiliar peers without the benefit of trusted third parties or authorities to mediate the interactions. A peer will need reputation mechanisms to incorporate the knowledge of others to decide whether to trust another party in P2P systems.

References

[1] K. Aberer and Z. Despotovic, "Managing Trust in a Peer-2-Peer Information System," Proc. 10th Int'l Conf. Information and Knowledge Management (CIKM), 2001.

[2] F. Cornelli, E. Damiani, S.D.C. di Vimercati, S. Paraboschi, and P. Samarati, "Choosing Reputable Servents in a P2P Network," Proc. 11th World Wide Web Conf. (WWW), 2002.

[3] S. Kamvar, M. Schlosser, and H. Garcia-Molina, "The (Eigentrust) Algorithm for Reputation Management in P2P Networks," Proc. 12th World Wide Web Conf. (WWW), 2003.

[4] L. Xiong and L. Liu, "Peertrust: Supporting Reputation-Based Trust for Peer-to-Peer Ecommerce Communities," IEEE Trans. Knowledge and Data Eng., vol. 16, no. 7, pp. 843-857, July 2004.A.A. Selcuk, E. Uzun, and M.R. Pariente, "A Reputation-Based Trust Management System for P2P Networks," Proc. IEEE/ACM Fourth Int'l Symp. Cluster Computing and the Grid (CCGRID), 2004.

[5] R. Zhou, K. Hwang, and M. Cai, "Gossiptrust for Fast Reputation Aggregation in Peer-to-Peer Networks," IEEE Trans. Knowledge and Data Eng., vol. 20, no. 9, pp. 1282-1295, Sept. 2008.

[6] J. Kleinberg, "The Small-World Phenomenon: An Algorithmic Perspective," Proc. 32nd ACM Symp. Theory of Computing, 2000.

[7] S. Saroiu, P. Gummadi, and S. Gribble, "A Measurement Study of Peer-to-Peer File Sharing Systems," Proc. Multimedia Computing and Networking, 2002.

[8] 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, vol. 6, no. 1, pp. 50-57, Jan. 2002.

[9] S. Saroiu, K. Gummadi, R. Dunn, S.D. Gribble, and H.M. Levy, "An Analysis of Internet Content Delivery Systems," Proc. Fifth USENIX Symp. Operating Systems Design and Implementation (OSDI), 2002.

[10] S. Marsh, "Formalising Trust as a Computational Concept," PhD thesis, Dept. of Math. And Computer Science, Univ. of Stirling, 1994.

[11] A. Abdul-Rahman and S. Hailes, "Supporting Trust in Virtual Communities," Proc. 33rd Hawaii Int'l Conf. System Sciences (HICSS), 2000.

[12] B. Yu and M. Singh, "A Social Mechanism of Reputation Management in Electronic Communities," Proc. Cooperative Information Agents (CIA), 2000.