Security Enhancement using SORT: Self Organizing Trust model

Ravikiran Kendre [1] ,Nikhil Jain [2], Gaurav Kedari[3], Sudhir Dhekane[4]
Department of Computer Engineering
[1,2,3,4]Studenta,RSCOE, Savitribai Phule Pune University, India

Abstract-*Peer-to-peer (P2P) content distribution network (CDN) technologies have such innovative technological improvement which claims low cost, efficient high demand data distribution and it gradually involves to the next-generation CDNs. This paper presents distributed algorithms Heuristic Algorithm 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. Interactions with a peer provide certain information about the feedbacks might contain deceptive information. The interactions and recommendations are evaluated based on importance, recentness, and satisfaction parameters. Metrics are needed to represent trust in computational models. Classifying peers as either trustworthy or untrustworthy is not sufficient in most cases. A file sharing application is stimulated to understand advantages of the proposed algorithms in mitigating attacks related with services and recommendations. For simplicity of discussion, following type of interaction is considered in the service context. i.e. File upload & download. Good peers were able to form trust relationships in their proximity.*

Keywords-Peer-to-peer network, Content delivery network, trust metrics, Heuristic Algorithm.

I. INTRODUCTION

P2P systems rely on collaboration of peers to accomplish tasks. Peers need to trust each other for successful operation of the system. A malicious peer can use the trust of others to gain advantage or harm. Feedbacks from peers are needed to detect malicious behavior.

Since the feedbacks might be deceptive, identifying a malicious peer with high confidence is a challenge. Determining trustworthy peers requires a study of how peers can establish trust among each other. Long-term trust information about other peers can reduce the risk and uncertainty in future interactions. Interactions and feedbacks provide a means to establish trust among peers.Trustworthiness of a peer is measured based on complaints. A peer is assumed as trustworthy unless there are no complaints about it. Peer Trust defines community and transaction context parameters in order to address application specific features of interactions. SORT assumes that all peers are strangers to each other at the beginning.

Peers must contribute others in order to build trust relationships. Malicious behavior quickly destroys such relationships . Measuring trust using numerical metrics is hard. Classifying peers as either trustworthy or untrustworthy is not sufficient. Metrics should have precision so peers can be ranked according to their trustworthiness. As in this, SORT's trust metrics are normalized to take real values between 0 and 5. In SORT, trust values are considered with the level of past experience.

A peer with more past interactions is preferred among peers assigned to the same trust value. A recommendation contains suspicious information. Combining these two types of information in one metric and using it to measure trustworthiness for different tasks may cause incorrect decisions. SORT defines three important trust metrics: reputation, service trust and recommendation trust.

Reputation is the primary metric when deciding about strangers. Recommendations are used to calculate the reputation of a stranger. Providing services and giving recommendations are different tasks. A peer may be a good service provider and a bad recommender at the same time. SORT defines two contexts of trust: service and recommendation contexts. Metrics on both contexts are called service trust and recommendation trust respectively. When a peer gives misleading recommendations, it loses recommendation trust of others but its service trust remains same. Similarly, a failed service interaction decreases the value of service trust metric. Quality of an interaction is measured with three parameters: satisfaction, weight, and fading effect. A binary value for satisfaction does not reflect how good a service provider was during the interaction. Interactions with larger weight values are more important on trust calculation. The effect of an interaction on trust calculation fades as new interactions happen. This makes a peer to behave consistently.

A peer with large number of good interactions can not disguise failures in future interactions for a long period of time. A recommendation from a highly trusted peer is more important. If the level of confidence has a small value, the recommendation is considered weak and has less effect on the reputation calculation. After giving a recommendation, the recommender's trust value changes in proportion to its level of confidence. If a weak recommendation is inaccurate, the trust value does not diminish quickly. When a peer wants to get a service and it has no acquaintance, it simply chooses to trust any stranger that provides the service. If the peer has some acquaintances, it may either select an acquaintance or a stranger. An acquaintance is always preferred over a stranger if they are equally trustworthy. Selection of a stranger is based on its reputation. If many interactions happened with anacquaintance, service trust metric is more important than reputation metric. Otherwise, reputation of the acquaintance is more important during the selection process. A P2P file sharing application has been simulated. 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. This enables us to make realistic observations about the success of the algorithms and how trust relationships evolve among peers. A malicious peer who performs collaborative attacks rarely but behaves honest in other times is harder to deal with The main contributions of this research are outlined as follows:

- Trust metrics are defined in service and recommendation contexts. Two contexts of trust distinguish capabilities of peers based on services provided and recommendations given.
- SORT's distributed algorithms enable peers make autonomous decisions without requiring any trusted peers. A peer adaptively adjusts the necessary level of trust according to its trust relationships.
- Arecommendationevaluationschemeisdefined.Arecommender'strustworthinessandlevel of confidence about the recommendation is considered for a more accurate calculation of reputations and fair evaluation of recommendations.
- Simulation experiments on a file sharing application verify SORT's ability in mitigating attacks. An evaluation scheme is defined for service interactions based on the application parameters.

An attacker model is presented including service and recommendation based attacks and nine types of malicious peers. Outline of the paper is as follows. Section II discusses the related research. The algorithms and formal definitions of SORT are explained in Section III. The simulation experiments of SORT is presented in Section IV. We outline future research opportunities to extend SORT in Section V and present conclusions in Section VI.

II. LITERATURE SURVEY

EXISTING SYSTEM:

In the existing system 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. Management of trust information is dependent to the structure of P2P network. In distributed hash table based approaches, each peer becomes a trust holder by storing feedbacks about other peers. 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. A peer sends trust queries to learn trust information of other peers. A trust query is either flooded to the network or sent to neighborhood of the query initiator.

DISADVANTAGES OF EXISTING SYSTEM:

- Calculated trust information is not global and does not reflect opinions of all peers.
- 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.
- Trust models on P2P systems have extra challenges comparing to e-commerce platforms.
- Malicious peers have more attack opportunities in P2P trust models due to lack of a central authority.

III. PROPOSED SYSTEM

PRODUCT PERSPECTIVE

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.

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. Management of trust information is dependent to the structure of P2P network.

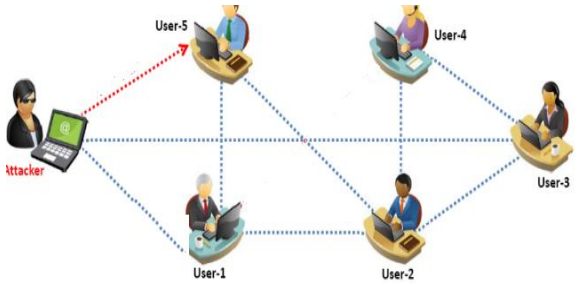


FIGURE Peer to Peer network

Product Functionality:-

1. User registration.
2. User login.
3. Upload document.
4. Download document.
5. Save service history.
6. Calculate importance.
7. Calculate satisfaction.
8. Calculate fading effect.
9. Calculate recommendation trust.
10. Calculate reputation.
11. Classify Peers.
12. Calculate competence belief.
13. Calculate integrity belief.
14. Calculate service trust.
15. Store peer data.
16. Storage of document.

A. USERS AND CHARACTERISTICS

The users are peer to peer systems and Database Server. The application gives the output in the visual formats.

- Peer to peer system:

The peer can be data owner and users

Peer can see the data as per the permission granted to them.

- Database

The database server is responsible to serve the peer request.It stores the uploaded file and maintains the log file. It is responsible for processing the query and storing the user information, log file. It is also responsible for holding access privilege related data.

OPERATING ENVIRONMENT

The environment is a platform independent.The product can operate in a minimum of Windows 7 operating system and the later versions of it. Other software's required are Netbeans. Product will run with a minimum configuration of 1GB RAM.

DESIGN AND IMPLEMENTATION CONSTRAINTS

The application is developed on Java and J2EE technologies, Netbeans, MySQL and has basic GUI. However it can also be implemented on platform.

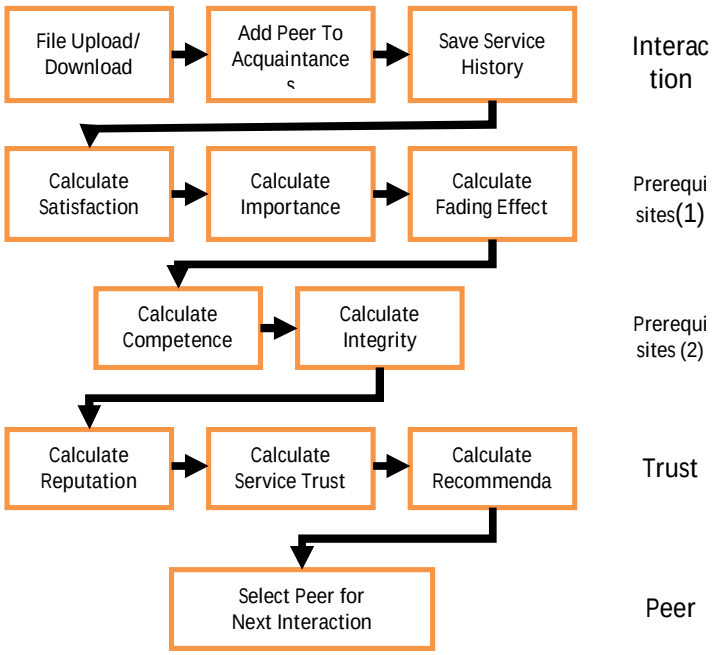
USER DOCUMENTATION

For this particular project there is no need of any hardware. This project is completely based on the network security. The project is in developmental stage and after implementation of project user manuals can be generated for user support.

ASSUMPTIONS AND DEPENDENCIES

The systems security constraint plays an important role. If the attacker bypasses the front end application as well as back end security constraints then attacker will be able to access and temper the database. This system is designed for different types of attacks but still if attacker uses another attacking technique then the application will be unable to detect thosetypes of attack.

IV. SYSTEM ARCHITECTURE



Interaction Phase

- A file is downloaded/uploaded among stranger peers.
- The peers are added to each other's set of acquaintances.
- Maintenance of service history starts.

If the service history becomes full, the first/oldest history is removed, thus giving preference to recent interactions

Prerequisites Phase (1)

- Satisfaction of a peer with another peer's service is calculated based on:
    - Average download speed
    - Average delay
    - Retransmission rate of packets
    - Online/offline periods of the service provider
- Importance of an interaction is calculated based on:
    - Size of file
    - Popularity of file (number of times it has been downloaded/uploaded)

Fading Effect of an interaction ensures that an old interaction loses importance and thus prevents it from misbehaving by relying on its good history

Prerequisites Phase (2)

- Competence Belief represents how well an acquaintance satisfied the needs of past interactions.
- Integrity Belief represents the level of confidence in predictability of future interactions.
- Both metrics are calculated based on following factors:
    - Satisfaction of interactions
    - Importance of each interaction
    - Fading effect (old interactions get less preference)

Trust Metrics Phase

- **Service Trust:**Check service history for competence belief & integrity belief for all interactions, before interaction among peers**.**
- **Reputation:**
    - Check with acquaintance peers for reputation of stranger peers based on service trust developed among them, before interaction with strangers.
- **Recommendation Trust:**
    - Compare the recommendation given and the service provided to check validity of the recommendation.

Takes into account the case of good service provider but bad recommender and vice versa

Peer Selection Phase

- A peer is selected for further interaction based on the 3 trust metrics in the previous module.
- When strangers, the peer which needs the service will consult an acquaintance peer which has already interacted with the stranger peer.
- The consultant peer will give a recommendation about the stranger peer and the consulting peer will get the stranger peer's reputation from all such consultant peers. Recommendation trust would define the validity of a peer.
- For peers already in acquaintance, peer selection follows the hierarchy as below:
    - Service trust
    - Service history size
    - Competence belief
    - Integrity belief
- Upload bandwidths

V. ALGORITHM

Assumptions:

We make the following assumptions. Peers are indistinguishable in computational power and responsibility. There are no privileged, centralized, or trusted peers to manage trust relationships. The majority of peers are expected to be honest but some might behave maliciously. Peers occasionally leave and join the network. A peer provide services and use services of others. For simplicity of discussion, one operation is considered in the service context, e.g., file download

Notations:

pi denotes the ith peer. When pi uses a service of pj,aservice interaction for pi occurs. Interactions are unidirectional. For example, if pi downloads a file from pj, no information is stored on pj about this download. If pi have not had a service interaction with pj, pj is a stranger to pi. An acquaintance of pi is the one whom pi had at least one service interaction with. pi's set of acquaintances is denoted by Ai. A

peer stores a separate history of service interactions for each acquaintance. SHij denotes pi's service history with pj. Since new interactions are appended to the history, SHij is a time ordered list.

Parameters for Interactions:

After finishing a service interaction, pi evaluates quality of the service. 0 ∑ ek ij ∑ 1. denotes pi's satisfaction about kth service interaction with pj. If theinteraction is cancelled, ek ij gets 0 value. k is the sequence number of the interaction in SHij. A service interaction is associated with a weight to quantify importance of the interaction. 0∑wk ij ∑1 denotes the weight of kth service interaction of pi with pj. The importance of an interaction fades as new interactions happen. 0 ∑ fk ij ∑ 1 denotes thefading effect of kth service interaction of pi with pj. It is calculated as follows:

fk ij =k /shij, 1∑k ∑shij…….. (1)

Service Trust Metric(stij):

This section describes the calculation of service trust metric. A peer first calculates competence and integrity belief values using the information about service interactions. Competence belief is based on how well an acquaintance satisfied the needs of interactions. cbij denotes the competence belief of pi about pj in the service context. Average behavior in the past interactions can be a measure of competence belief. pi calculates cbij as follows:

$$cb_{ij} = \frac{1}{\beta_{cb}} \sum_{k=1}^{sh_{ij}} (e_{ij}^{k} \cdot w_{ij}^{k} \cdot f_{ij}^{k})$$

A peer can be competent but may present erratic behavior. Consistency is as important as competence. The level of confidence about the predictability of future interactions is called integrity belief. ibij denotes the integrity belief of pi about pj in the service context. Deviation from the average behavior is a measure of integrity belief. Therefore, ibij is calculated as an approximation to the standard deviation of interaction parameters:

$$ib_{ij} = \sqrt{\frac{1}{sh_{ij}} \sum_{k=1}^{sh_{ij}} (e_{ij}^{k} \cdot w_{ij}^{\mu} \cdot f_{ij}^{\mu} - cb_{ij})^{2}}$$

Based on the past interactions with pj, pi has an expectation about future interactions. pi wants to maintain a level of satisfaction according to this expectation. Assume that the satisfaction parameter follows a normal distribution and cbij and ibij are the approximations of mean (μ) and standard deviation(æ) of this parameter respectively. According to the cumulative distribution function of normal distribution, an interaction's satisfaction is less than cbij with a ©(0) = 0.5 probability. If pi sets stij = cbij, half of the future interactions will likely to have a satisfaction value less than cbij.Thus, stij = cbij is an over-estimate for pj's trustworthiness. A lower estimate makes pi more confident about future decisions with pj. pi may calculate stij as follows:

stij = cbij -ibij/2.

V. ACNOWLEDGMENT

I would like to take this opportunity to express my profound gratitude and deep regard to my Project Guide,Prof. V.D.Shinde (Faculty of Computer Engineering Department), for her exemplary guidance, valuable feedback and constant encouragement throughout the duration of the project. I also want to thanks Prof. Gopal D.Upadhye(Project Coordinator) for his valuable suggestions were of immense help throughout my project work. His perceptive criticism kept me working to make this project in a much better way. Working under both of them was an extremely knowledgeable experience for me.

VI. CONCLUSION

In this paper, we proposed an algorithm called Heuristic Algorithm for calculating reputation values of nodes in a network. A number of experiments are conducted. And the simulation results show that Heuristic Algorithm outperforms SORT in four kinds of attack models. In particular, the false positive rates and the false negative rates are decreased significantly. However, the loads of nodes in the network are slightly increased when Heuristic Algorithm is used. In the future, we will further investigate relative techniques to decrease it.

REFERENCES

[1] Can A B, Bhargava B. SORT: A Self-ORganizing Trust Model for Peer-to-Peer Systems[J]. Dependable & Secure Computing IEEE Transactions on, 2015, 10(1):14-27

[2] Yu H, Kaminsky M, Gibbons P B, et al. SybilGuard: Defending Against Sybil Attacks via Social Networks[J]. Networking IEEE/ACM Transactions on, 2015, 16(3):576 - 589.

[3] K. Aberer and Z. Despotovic, "SORT using HP2PSORT" Proc. IEEE transaction on dependable and secure computing 2015.

[4] A.B. Can, "Decentralised Reputation Based Trust Model for Peer-to-Peer Content Distribution Networks"PhD thesis, Dept. of Computer Science, Purdue University, 2014.

[5] F. Cornelli, E. Damiani, S.D.C. di Vimercati, S. Paraboschi, and P.Samarati, "SORT to avoid malicious

peers from P2P Network," Proc.11th World Wide Web Conf. (WWW), 2015.

[6] 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 2014.

[7] 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), 2014.

[8] 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. 20011.

[9] J. Kleinberg, "The Small-World Phenomenon: An Algorithmic Perspective," Proc. 32nd ACM Symp. Theory of Computing, 2013.

[10] S. Saroiu, P. Gummadi, and S. Gribble, "A Measurement Study of Peer-to-Peer File Sharing Systems," Proc. Multimedia Computing and Networking, 2012.

[11] 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. 2012.