Secure Communication in Peer-to-Peer Network Based on Trust-Based Model

Vijay Paul Singh, Tulika Vijay, Tushita, Sonal, Rigzin Angmo, and Naveen Aggarwal

1 Introduction

Peer-to-peer network approach has evolved over the years. It has been accepted as an efficient communication model because of its distinctive characteristics and applications such as self-scalability, dynamic nature, content distribution, and effective search [1]. In terms of computer networks, it refers to a network that uses a distributed network architecture. Whereas the devices or computers or nodes that are part of this network are called peers [2] and are completely independent of each other without any central authority. In this paper, we use the acronym P2P for peer-to-peer network.

Earlier, for communication, a simple client–server architecture was followed over the Internet and flourished throughout the 1990s. During the late 1990s, new technologies of data compression for mp3, MPEG, etc. became popular and the traditional systems were unable to deal with the increase in overall demand of data consequently, it was difficult to manage the increase in bandwidth costs for the clients. P2P network provides solutions for this problem by introducing applications like "Napster" that can be used to download and share free mp3 files. In a P2P network, delivery costs get reduced further with the increase in demand of a file which leads to more seeding of that file in the network by the peers [3]. The availability, cost, and capability of personal computers and broadband Internet services to the general population also led to an inevitable increase of interest in the P2P architecture. Since the 90s there has been a sudden boom in P2P network applications to share multimedia files. Some of the most famous file-sharing protocols used are Freenet, Napster, Direct Connect, Gnutella, eDonkey2000, and Bit Torrent [4].

In P2P communication, file sharing with the help of peers, akin to Napster, Bit Torrent both uses centralized directory server, Gnutella that uses Query Flooding to provide an easy way and avoids the central point of failure [5]. Usually, in the

V. P. Singh (✉) · T. Vijay · Tushita · Sonal · R. Angmo · N. Aggarwal
CSE, UIET, Panjab University, Chandigarh, India

R. Agrawal et al. (eds.), *Cyber Security in Intelligent Computing and Communications*,
Studies in Computational Intelligence 1007,
https://doi.org/10.1007/978-981-16-8012-0_13

file system, security breaches due to infected files sharing with this whole peer get affected or even other files get affected. To resolve the risk of file sharing and to provide secured documents from malicious, a trust-based P2P system is needed [6]. Trust is a widely used term whose definition differs among researcher and application areas [7]. Trust can be measured numerically for a specific level of the subjective with which other nodes or agents will perform a particular action and take further action [8]. In addition, the Trust often refers to the mechanisms to ensure that the source of information is actually what claims to be a source [9]. The trust in network communication is further categorized [10].

(i) Policy-based trust: Establish trust using the third party or trusted third party to serve as an authority for issuing verified credentials [11].
(ii) Reputation-based trust: It is based on past interaction performance or history of an entity action/behavior [12].
(iii) General model of trust: It is also based on transaction and reputation.
(iv) Trust in information resources: It is related to web resources and websites in which rating by the user about the quality of information and services [13].

To provide trust in communication, various trust-based model techniques are there, some of them are described in this paper. It is the most cited trust and name primarily based model employed in peer to see system [14]. It fights against malicious peers and permits peers to see file sharing. Every peer maintains a neighborhood trust with another peer and every peer calculates international trust worth by all alternative peers:

(i) Eigentrust: It is the most usable and cited trust. It identifies trustable versus non-trustable peers during file sharing using the reputation-based model. Each peer maintains local trust with another peer and global trust by all alternative peers [15, 16].
(ii) Peer trust: It measures trust factor based on two important factors, i.e., transaction feedback collected from other peers and general metric, using these two parameter peer credibility measures [17].
(iii) Cuboid trust: It is based on global trust and pre-trusted peer involvement; these two techniques support identifying trustable peers in the network [12, 18].
(iv) Ant reputation: It maintains a trustable table similar to the distance vector routing protocol table, in which peer information is maintained related to the trust factor [9].

Eigenvalue uses almost every reputation-based model, based on that local trust, global, the number of transactions, and also the past reputation of the peer are important factors to identify trustable peers in the network. There are three possibilities during file sharing in the P2P network [9, 12, 15], i.e., Only a few responses, No one Response, and Everyone Response. This peer requires its acquaintances about their opinions about the other peer, which affects it to decrease the number of unauthenticated files. In this, the system should be self-policing and should maintain anonymity, not assign any profit to newcomers, minimal overhead. Each peer i can store the information of successful transactions with peer j [19].

Each peer i can store the information of successful transactions with peer j [19].

$$S_{ij} = Sat(i, j) - Unsat(i, j). \tag{1}$$

In Eq. (1), *Sat* is a satisfaction transaction, *Unsat* is an un-satisfaction transaction.

$$C_{ij} = \max\left(S_{ij}, 0\right) / \sum j \max\left(S_{ij}, 0\right). \tag{2}$$

In Eq. (2), C_{ij}, compute local trust value using max function which finds out higher values between S_{ij} and zero based on that local trust value calculated. To mitigate the risk during file sharing in such systems, we have proposed the model to improve the system scenario, proposed model is using trust and reputation mechanism to find the malicious nodes. The proposed model categorizes the peers into two main categories: (i) trusted nodes and (ii) normal nodes, based on their defined access level. In this paper, eigenvalue is used to measure the trust value, and the reputation analysis is done based on time. In the proposed network, nodes are rated by each other and based on this rating they communicate with each other in the network. To validate the approach, simulation using simply has been implemented. The implementation is aimed to figure out the presence of malicious peers in the network based on the trust mechanism. With the help of graphs, it can be visualized that outcome has been achieved. In a decentralized network, this mechanism to find the malicious node is effective and sensitive. This paper is organized as: the next section, i.e., Section 2, pointed to some related work; Sect. 3, problem statement; Sect. 4, proposed model; Sect. 5, discuss implementation and experimental results; and Sects. 6 and 7, compare, conclude, and discuss future work.

2 Related Work

P2P network is an example of Bit Torrent [20] in which each node depends on each other. In client–server model, information is not revealed to the client but in P2P some internal information may be exposed and attackers can easily attack or communicate with these nodes through various tactics [21, 22]. There are various levels of P2P networking [23] such as Hybrid P2P in which the central server keeps the information about the network, Pure P2P, where there is no absolute server, Mixed P2P similar to Gnutella no central server but a cluster of nodes. There are various attacks which can be possible in the P2P network such as DoS, DDoS, Masquerade, Man-in-the-middle Attack, Worm Propagation, Rational Attacks, File Poisoning, Eclipse Attack, Sybil Attack, and many more [24–26]. These attacks are generally possible due to several vulnerabilities. DoS is an attack in which a network or node loses the service. Detecting a DoS attack is a challenging task [27]. El Defrawy et al. [19] make use of the BitTorrent system and perform real-life experiments that demonstrate the practicability and severity of such attacks. Yusof et al. [28] surveyed

DDoS attack, where they had provided answers to the 6 proposed research questions and 48 techniques that are used to DDoS attack on various kinds of the network system in which one of them is P2P. Man-in-the-middle attack is common in file sharing where an attacker easily inserts in the network and spreads polluted files on behalf of the authorized node [29]. To mitigate this attack in P2P, trustable authority is required, which generally does not exist in P2P. Worm propagation is the biggest threat, popular worms are Code Red or Nimda that can infect thousands of nodes within an hour [30]. There are various factors that make P2P worms infected such as using the same software by all nodes of P2P [21], during transfer of large file set limit in order to hold one TCP packet [30], and easily accessing normal user computers and retrieving sensitive information. Once a worm propagates inside the network then its next goal is to launch the DDoS attack [21]. The human is also one of the factors of attacks, sometimes novice users download files that are infected, and due to their inaction with regard to security created difficulties [31]. Whereas rational attack, file poisoning, and Sybil attack are enormously possible in the P2P network. In the rational attacks in P2P, a large number of nodes consume system resources and less involvement in the network [31]. File poisoning, the actual file has been spoofed by the attacker. This attack is controlled by deleting corrupted downloaded files on the user's end or by trying not to download unauthorized files and is detected by various smart algorithms. Finally, there is the Sybil attack in which a single malicious identity can present multiple identities that effectively take control of the network [32]. To handle Sybil attacks one approach is the trustable central authority which is not possible in P2P. Another approach is a reputation-based system which might be able to control this attack. In this paper, we have used a trust-based model which is related to the reputation of nodes in the network.

3 Problem Statement

In peer-to-peer communication, the security threats mainly seen in file systems are attacks by malicious nodes or malicious collective, while using the eigentrust concept in P2P might be less control of malicious nodes due to an increase in satisfactory transaction Eq. (1). These security threats make the system unreliable and non-robust. The lack of any scrutiny and the open nature of decentralized systems have made it prone to security hazards that adversely affect the performance of the network. To mitigate this situation, we have proposed an approach that can easily disallow the malicious node to communicate in the network.

4 Proposed Model

The proposed model states that the peers are categorized into two nodes: (i) trusted node and (ii) normal nodes. The access permissions are provided to these nodes according to their categories. The trusted nodes are those that make the network and are provided with all kinds of access permissions. Whereas all other nodes that subsequently join the network are normal nodes with restricted permissions. In the proposed model, the trust and reputation mechanism is used to track the behavior of the nodes. Within the network, a normal node records local trust values about its experience with some nodes and the trusted nodes record the aggregate trust values that summarize the experiences of all nodes in the network with some nodes. The main contributions of the paper are as follows:

- It presents a simple and easy-to-implement model for P2P network security by providing the categorization of nodes, differentiation in access permissions, and monitoring of the behavior of peers based on trust and reputation mechanism.
- The model also provides a fair and unbiased opportunity to peers that enhance their access permissions.
- The malicious nodes are efficiently detected in the network with the help of this model and used for the prevention of the underlying security risk in an advanced file-sharing system.
- The experimental results under various cases indicate that our approach is more effective and sensitive in detecting malicious peers as compared to other similar trust models.

Table 1 contains symbols related to the proposed approach.

A. ***Algorithm for Joining of Nodes***

1. Each node joins the network as a normal node.

B. ***Algorithm for Trust Calculation***

1. Each node i in the network maintains local *eigentrust value* $S_{i,j}$ and normalized local trust value $C_{i,j}$ based on the number of successful and unsuccessful transactions with peer *j.*
   for each node i

Table 1 Symbol table

T_i	Trusted nodes
U_i	Normal nodes
$Sat_{(i,j)}$	Successful transaction between nodes i and j
$Unsat_{(i,j)}$	Unsuccessful transaction between nodes i and j
$S_{i,j}$	Local trust values of node i with node j
$C_{i,j}$	Normalized local trust between node i and node j
G_i	Aggregate trust

$$S_{ij} = Sat(i, j) - Unsat(i, j) \tag{3}$$

$$C_{ij} = \max\left(S_{ij}, 0\right) / \sum j \max\left(S_{ij}, 0\right). \tag{4}$$

2. T_i stores aggregate trust values of each node in the network
   for each T_i

$$Gi = \left(\sum Sat(i, j) - \sum jUnsat(i, j)\right) / \left(\sum Sat\left(i, j + \sum jUnsat(i, j)\right)\right). \tag{5}$$

3. The trusted nodes maintain a minimum threshold value. If the aggregate trust value of any node becomes less than this value, the node can be identified as harmful to the network and its access permissions are further decreased.
4. The trusted nodes maintain a maximum threshold value. If the aggregate trust value of any node becomes greater than this value, the node can be identified as reliable and its access permissions are increased.
5. The measure of aggregate trust G_i is a probabilistic and normalized measure and thus is successful in differentiating between two peers whose difference between successful and unsuccessful transactions is the same but one peer has a higher probability of unsuccessful transactions than the other.
6. The threshold calculation by the trusted node is assumed to be dependent on various reputation parameters like the amount of time spent in the network, the number and type of transactions done, and the study of how malicious nodes have behaved in the past in the same network or in peer-to-peer networks in general.

III. *Algorithm for Upgradation of Nodes*

Some nodes may be upgraded from the normal category to the trusted one, according to the following algorithm:

1. The trusted node keeps track of the total number of nodes in the network.

2. If total number of(U_i)>> total number of(T_i), then

3. G_{max} is set as U_1 and j=0
4. For each node i,
   if (U_i>G_{max}), then
   G_{max} is set as U_i
5. End for
6. For each node i,
   (If U_i== G_{max})
   j=j+1
7. End for
8. If j==1, then
9. Ui with G_{max} is set as Ti
10. Else
11. Reputation parameter based on time of the node that is the amount of time node has spent in the network is used (Ri).
12. The node with G_{max} and R_{max} is upgraded.
13. End if
14. End if

The trusted node keeps track of the total number of nodes in the network. If there is an imbalance between the number of trusted and normal nodes, they take the decision of upgrading certain nodes from the normal category to the trusted one.

When there is a need for upgradation, trusted nodes find a node with maximum aggregate trust.

(a) If there is no other node with the same aggregate trust the node is upgraded.
(b) If there are more than one node in the network with this same aggregate trust, we use a reputation parameter based on time of the node, which is the amount of time it has spent in the network. The node which has this maximum aggregate trust and has been in the network for a longer time is upgraded.

IV. ***Algorithm for Leaving of Nodes***

```
1. If the node which wants to leave is a normal
   node(U_i), then
2. Delete (U_i).
3. End if
4. If leaving node is a trusted node(T_i), then
5. If total number of(U_i)>> total number of(T_i),
   then
6. Call upgradation function.
7. Else
8. Delete(T_i)
9. End if
10.End if
```

From the above algorithm, if the leaving node is a normal node, then it is a simple leave operation. If the leaving node is a trusted node, the remaining trusted nodes use the *UPGRADE* algorithm.

5 Implementation and Results

To simulate the peer-to-peer network, we use SimPy [33] which is a discrete-event, process-based simulation platform based on standard Python. The implementation is aimed to figure out the presence of malicious peers in the network based on the trust mechanism. The simulation implements the presented algorithm, calculates local trust, normalized local trust and aggregate trust of each peer, and enlists nodes that are harmful to the network.

We assume that the network consists of a total of 50 nodes consisting of five peers that are labeled trusted and 45 other nodes. The trusted nodes are provided with read and write access and the others are provided only read access. The connections formed between the nodes are random and they are assumed to communicate using three types of messages—read, write, and inauthentic. All peers in the network can send a read message as they all have the read permission. The write message by any normal node notifies them that the write access is denied. The inauthentic messages which represent malicious behavior are sent only by the malicious nodes that decrease their reputation in the network.

The nodes maintain a record of their communication. Each node *i* maintains a log of the number of successful and unsuccessful transactions it has with node *j* which is used to calculate the local trust.

At the end of simulation, aggregate trust is calculated for each peer which is a measure of its overall behavior and helps to determine which nodes are malicious in the network. In this paper, three cases are presented to understand the effectiveness of the model to provide secure communication based on the trust model in P2P communication network.

Case 1: A trusted peer communicates with any other peer: The trusted peers have all access permissions and they can communicate successfully with any peer in the network.

Case 2: A normal node communicates with any other peer: All other nodes in the network except the trusted nodes have only read permissions. When a node sends a read message, the transaction can be successful or unsuccessful based on the delay which is calculated using randomly allotted bandwidth. When a node sends the write message, it is alerted that this access is not provided to it.

Case 3: A malicious node communicates with any other peer: The model deals with read and write messages sent by the malicious node in a similar manner as that of a normal node discussed above. The malicious behavior of such nodes is simulated by using a third type of message, inauthentic, which decreases their reputation in the network considerably.

As per the above cases, we have implemented and visualized in graph form. In Fig. 1, the graph depicts local trust values for each connection (over a range of positive and negative values) as the number of connections increases. Here, peers are chosen randomly and sequential numbers are assigned to the connections established.

In Fig. 2, the graph depicts local trust values for each connection as the number of connections increases but the values are normalized (in the range (0, 1)).

Note: peers are chosen randomly and sequential numbers are assigned to the connections established.

In Fig. 3, It affirms that the peers numbered 40, 41, 42, 43, 44 are malicious nodes as they have the least aggregate trust values.

In Figs. 1 and 2, the graph depicts local trust values for each connection (made up of two nodes) as the number of connections increases. Figure 2 is a modification of Fig. 1 as it shows normalized local trust values. In Fig. 3, the graph depicts the aggregate trust (calculated with the help of local trust) for each peer, and as a result

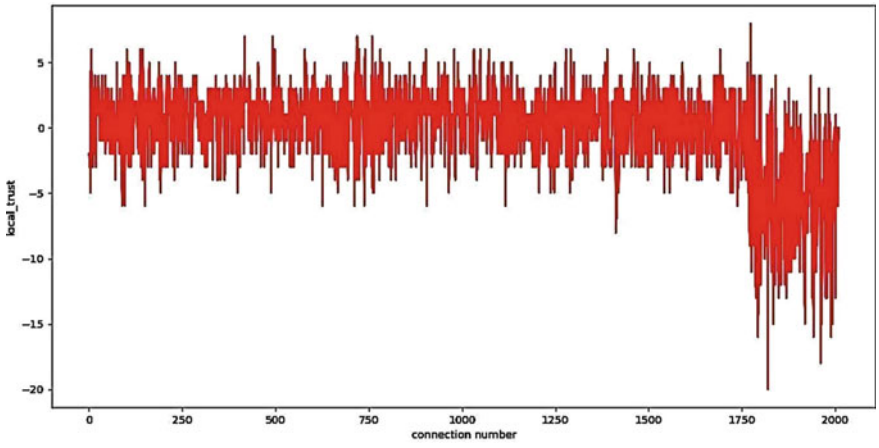

Fig. 1 Local trust values versus the number of connections

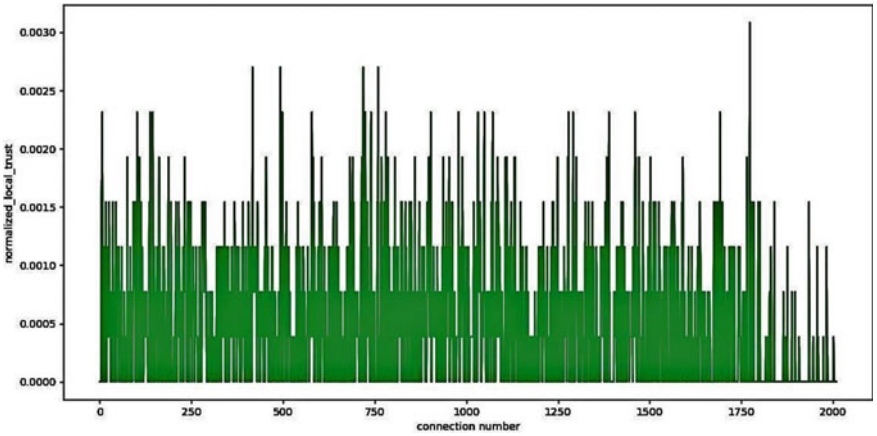

Fig. 2 Normalized local trust values versus the number of connections

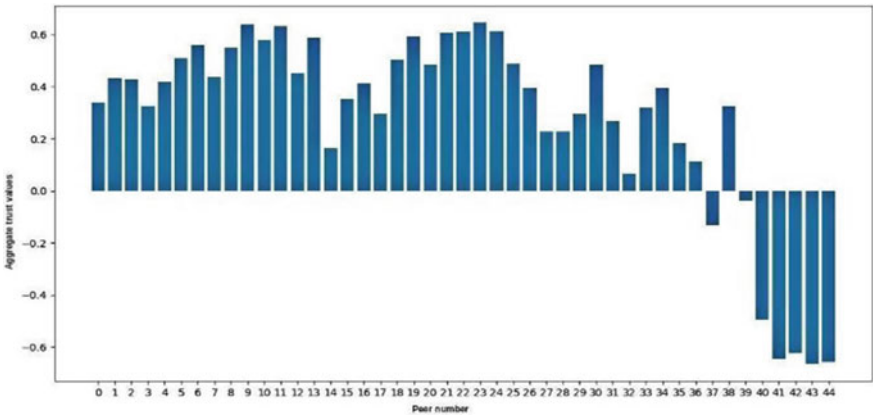

Fig. 3 Aggregate trust values versus the peer number

we are able to distinguish between non-malicious and malicious peers, with Fig. 4, we are able to see the activity of the malicious peer (in this case the most malicious peer) an amount of time the number of connections it forms increases. In Fig. 5, we are able to see the activity of a non-malicious node (in this case, the most trustworthy node) after a short interval the number of connections it forms increases.

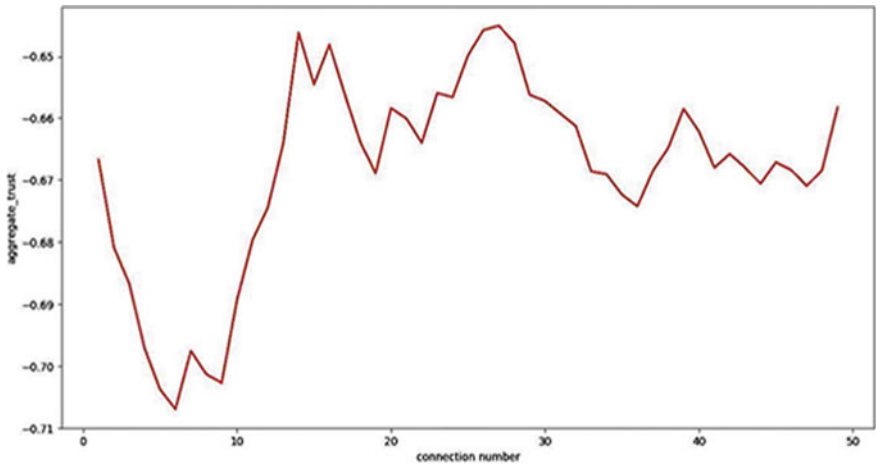

Fig. 4 Aggregate trust values of the most malicious peer (P43) versus the connections made by the said peer over a period of time

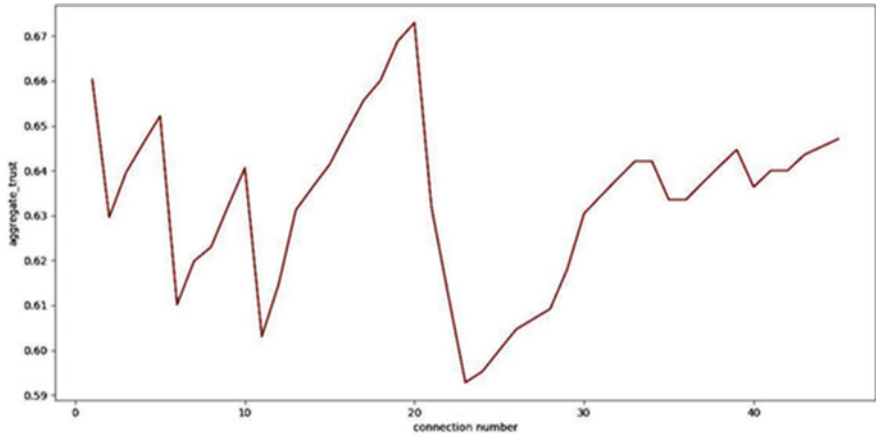

Fig. 5 Aggregate trust values of the most trustworthy peer (P23) versus the connections made by the said peer over a period of time

6 Comparison of Proposed Trust-Based Models

Peer-to-peer network has many implementations—one of the implementations is in electronic markets where SLA (service-level agreement) is used to state agreements based on transactions between client and service provider [31, 34] deals with the same problem as the one discussed in this paper, that is, in a network of clients and service providers as peers, malicious behavior in a network can affect the overall trust values of the peers. The ground of comparison is that our proposed model categorizes the

nodes into (i) trusted and (ii) normal with differentiation in access permissions and the rearrangement of permissions based on the trust values (upgradation of normal peers to trusted peers).

In a peer-to-peer network, peers can review each other negatively or positively. In the team-based learning model, [35] peers can assign biased scores to other peers to inflate or deflate grades. Similarly, this paper deals with malicious nodes which can increase their trust values in the network with other malicious nodes to increase their aggregated trust value. The cited paper proposes Michaelsen, Fink, Kole methods to build a trust model while we propose a model based on eigentrust algorithm with aggregated and local trust values. In paper, [32] proposes a model for secure transaction in mobile P2P networks. It defines an adaptive reputation factor similar to the trust values in our proposed model. With historical interactions of the peers, Bayesian game theory is used to design the trust model in MP2P network whereas our model uses the eigentrust algorithm with a record of transactions to maintain local trust and aggregate trust to build a secure network.

7 Conclusion

In the evaluation of the proposed model, the experimental results under various cases evince that our approach is more effective and sensitive in detecting malicious peers as compared to other similar trust models. This paper presents a simple and easy-to-implement model for P2P network security. The model is based on the categorization of nodes, differentiation in access permissions, and monitoring of the behavior of peers based on trust and reputation mechanism. The model also provides the opportunity to peers to enhance their access permissions with being fair and unbiased. It is found to be efficient in detecting the presence of malicious nodes in the network and thus preventing the underlying security risk in an advanced file-sharing system.

Acknowledgements Authors would like to thank the team of DIC, Chandigarh, India (dic.puchd.ac.in) and are grateful to the Ministry of Human Resource Development (MHRD) of Government of India for supporting this research under Design Innovation Center (MHRD-DIC).

References

1. S. Androutsellis-Theotokis, D. Spinellis, A survey of peer-to-peer content distribution technologies. ACM Comput. Surv. (CSUR) **36**(4), 335–371 (2004)
2. J. Risson, T. Moors, Survey of research towards robust peer-to-peer networks: Search methods. Comput. Netw. **50**(17), 3485–3521 (2006)
3. S. Nakamoto, A. Bitcoin, A peer-to-peer electronic cash system. Bitcoin. https://bitcoin.org/bitcoin.pdf. 4 (2008)
4. J. Li, A survey of peer-to-peer network security issues. Retrieved November 29 (2007), 2010

5. S. Saroiu, K.P. Gummadi, S.D. Gribble, Measuring and analyzing the characteristics of Napster and Gnutella hosts. Multimedia Syst. **9**(2), 170–184 (2003)
6. H. Tran, M. Hitchens, V. Varadharajan, P. Watters, A trust based access control framework for P2P file-sharing systems, in *Proceedings of the 38th Annual Hawaii International Conference on System Sciences* (Big Island, HI, USA, 2005), pp. 302c–302c
7. M. Momani, S. Challa, Survey of trust models in different network domains. arXiv:1010.0168 (2010)
8. D.J. Kim, D.L. Ferrin, H.R. Rao, A trust-based consumer decision-making model in electronic commerce: The role of trust, perceived risk, and their antecedents. Decision Support Syst. **44**(2), 544–564 (2008)
9. A.A. Selcuk, E. Uzun, M.R. Pariente, A reputation-based trust management system for P2P networks, in *IEEE International Symposium on Cluster Computing and the Grid, 2004*. CCGrid 2004 (IEEE, 2004)
10. D. Artz, Y. Gil, A survey of trust in computer science and the semantic web. J. Web Semant. **5**(2), 58–71 (2007)
11. P. Bonatti et al., An integration of reputation-based and policy-based trust management. Networks **2**(14), 10 (2007)
12. L. Xiong, L. Liu, Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. IEEE Trans. Know. Data Eng. **16**(7), 843–857 (2004)
13. S.A. Morris, T.E. Marshall, R.K. Rainer Jr., Impact of user satisfaction and trust on virtual team members. Inf. Res. Manage. J. (IRMJ) **15**(2), 22–30 (2002)
14. F.G. Mármol, G.M. Pérez, Towards pre-standardization of trust and reputation models for distributed and heterogeneous systems. Comput. Stand. Interfaces **32**(4),185–196 (2010)
15. S.D. Kamvar, M.T. Schlosser, H. Garcia-Molina, The eigentrust algorithm for reputation management in p2p networks, in *Proceedings of the 12th International Conference on World Wide Web* (2003)
16. A. Abdul-Rahman, S. Hailes, A distributed trust model, in *Proceedings of the 1997 Workshop on New Security Paradigms* (1998)
17. G. Suryanarayana, R.N. Taylor, A survey of trust management and resource discovery technologies in peer-to-peer applications (2004)
18. R. Chen et al., CuboidTrust: a global reputation-based trust model in peer-to-peer networks, in *International Conference on Autonomic and Trusted Computing* (Springer, Berlin, Heidelberg, 2007)
19. K. El Defrawy, M. Gjoka, A. Markopoulou, BotTorrent: misusing BitTorrent to launch DDoS attacks. SRUTI **7**, 1–6 (2007)
20. B. Cohen, The BitTorrent protocol specification, 11 October 2013. Retrieved April 24, 2017, from http://www.bittorrent.org/beps/bep0003.html
21. N. Naoumov, K. Ross, Exploiting p2p systems for ddos attacks, in *Proceedings of the 1st International Conference on Scalable Information Systems* (2006).
22. J. Liang, N. Naoumov, K.W. Ross, The index poisoning attack in P2P file sharing systems. INFOCOM (2006)
23. J. Buford, Y. Heather, E.K. Lua, *P2P Networking and Applications* (Morgan Kaufmann, 2009)
24. J. Mirkovic, P. Reiher, A taxonomy of DDoS attack and DDoS defense mechanisms. ACM SIGCOMM Comput. Commun. Rev. **34**(2), 39–53 (2004)
25. J. Seedorf, Security challenges for peer-to-peer SIP. IEEE Netw. **20**(5), 38–45 (2006)
26. B. Pretre, Attacks on peer-to-peer networks. Dept. of Computer Science Swiss Federal Institute of Technology (ETH) Zurich Autumn (2005)
27. G. Carl et al., Denial-of-service attack-detection techniques. IEEE Int. Comput. **10**(1), 82–89 (2006)
28. A.R. Yusof, N.I. Udzir, A. Selamat, Systematic literature review and taxonomy for DDoS attack detection and prediction. Int. J. Digi. Enterprise Technol. **1**(3), 292–315 (2019)
29. D.S. Wallach, A survey of peer-to-peer security issues, in *International Symposium on Software Security* (Springer, Berlin, Heidelberg, 2002)

30. V. Vlachos, S. Androutsellis-Theotokis, D. Spinellis, Security applications of peer-to-peer networks. Comput. Netw. **45**(2), 195–205 (2004)
31. J. Dinger, H. Hartenstein, Defending the Sybil attack in p2p networks: taxonomy, challenges, and a proposal for self-registration, in *First International Conference on Availability, Reliability and Security (ARES '06)* (IEEE, 2006)
32. Z. Li, J. Bi, An adaptive trusted request and authorization model for mobile peer-to-peer networks, in *2013 IEEE 10th International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing* (Zhangjiajie, 2013), pp. 1274–1280. https://doi.org/10.1109/HPCC.and.EUC.2013.181
33. SimPy. https://simpy.readthedocs.io/en/latest/. Accessed 1 May 2019
34. I. Petri, O. Rana, Y. Rezgui, G.C. Silaghi, Evaluating trust in peer-to-peer service provider communities, in *7th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)* (Orlando, FL, 2011), pp. 407–414. https://doi.org/10.4108/icst.collaboratecom.247125 (2011)
35. M.S. Patil et al., Trusted relative peer review: A novel approach to assess an individual in team based learning, in *2016 IEEE 4th International Conference on MOOCs, Innovation and Technology in Education (MITE)* (Madurai, 2016), pp. 54–59. https://doi.org/10.1109/MITE.2016.021
36. B.N. Levine, C. Shields, N.B. Margolin, A survey of solutions to the Sybil attack. University of Massachusetts Amherst, Amherst, MA 7, 224 (2006)