Malicious Trust Managers Identification (MTMI) in Peer to Peer Networks

Adwan Alownie Alanazi

Department of Computer Science and Engineering
University of Hail
a.alanazi@uoh.edu.sa

Abstract

Peer to Peer Networks play an increasing role in today's networks, also it's expected that this type of communication networks evolves more in the future. Since the number of users that is involved in Peer to Peer Networks is huge and will be increased more in the future, security issues will appear and increase as well. Thus, providing a sustainable solution is needed to ensure the security of Peer to Peer Networks. This paper is presenting a new protocol called Malicious Trust Managers Identification (MTMI). This protocol is used to ensure anonymity of trust manager, that computes and stores the trust value for another peer. The proposed protocol builds a secure connection between trust managers by using public key infrastructure. As well as experimental testing has been conducted to validate the proposed protocol.

Keywords:

Peer to Peer, Malicious Trust Managers Identification (MTMI), Trust Mangers

1. Introduction

Recently, P2P networks are widely used and expected to spread more in future. In peer to peer networks any node can be server and client at any time. A peer could be a server when it provides service to other peers, such as uploading document. On the other hand, if a peer is receiving a file then it is called client. Another important feature of P2P networks is the varying nature of nodes in the network; nodes join and disjoin the network frequently. Each peer in P2P is assigned a trust value, representing a reputation the network. Based on the trust value of the peer, other peers decide to contact that peer or not. Client peer try to interact with server peers with high trust value to get data, and in most cases server peers response to those clients with high trust value and refuse the interaction with clients with low trust values. Therefore, trust values of peers are very critical issue.

Moreover, decentralized nature of P2P networks makes it very hard to figure out malicious peers in the network. In order to overcome the threat of malicious peers, a trust value which is assigned to each node in the network becomes a necessity. A peer trust value represents the reputation of the peer in the network. A peer is assigned a trust value based on it is performance in the network and it is managed by other peers (Trust Managers). These trust values are very sensitive data in distributed Peer to Peer networks. Therefore, the big challenge is to manage and distribute those trust values in a very secure way, such that peers - especially malicious peers- are forbidden from hacking the place where those values are stored, or how peers exchange those values.

In this paper, a number of essential security services are achieved through our new scheme that creates a secure connection between trust managers such as, anonymity, confidentiality, authentication and integrity. The rest of this paper is organized in the following way: Section 2 present some of the related work to this paper.
Section 3 makes an overview of the main characteristics of the peer-to-peer architecture we focus
on. Section 4 introduces an overview of trust value calculation and the trust value mechanism we have used and considered in our trust management protocol. Section 5 explains the design and the implementation of the proposed protocol in more detail. Section 6 discusses the conducted experiment and the results of the proposed design. Section 7 presents the discussion. Finally, section 8 draws the conclusions.

2. Related work

In a P2P networks there are two main important issues that should be taken into account. First, how Trust value can be computed for each node effectively? Second, how trust values for peers can be stored and distributed securely among peers in the network? However, the field of securely store and distribute the trust values are still very hot field [1, 2]. Centralized networks are less vulnerable to attacks and malicious threat than P2P networks, because in the later, network resources are shared among too many peers. Therefore, the security issues in P2P networks forms a big challenge in nowadays[3]. However, eBay reputation-based trust metric shows a successful system to manage and distribute trust value efficiently, but unfortunately it is based on a centralized server [7].

Most of P2P research such as [8] focused in identity, or on the Distributed Hash Tables as in [12], but research does

Manuscript received September 5, 2021
Manuscript revised September 20, 2021
https://doi.org/10.22937/IJCSNS.2021.21.9.11

not pay a lot of attention for the malicious peers in order to identify it. [4] Proposed a multitier architecture that is self organized without central point to manage the network. The first tier is used for message routing and its implemented as multi tree where each node in tree has a trust value that is computed according to its availability, response time, bandwidth, etc, by its neighbors, where the second tier links divided to two types, links to all node siblings and one random link to one of parent sibling, moreover in this tier each node will maintain a list of recent active links.

[5] examined the usage of Trusted Computing Group (TCG) protocols for making P2P networks more secure, they have summarized P2P security problems as the conflict between the need for anonymity for users synchronized with need to offer access control, data integrity, and confidentiality services. Another point is P2P networks do not have any centralized authority that may be responsible for identifying identities. The Trusted Computing Group (TCG) is an industry standards body formed to develop, define, and promote open standards for trusted computing and security technologies. TCG has developed an open architecture and standards for Network Access Control called Trusted Network Connect (TNC).

[6] discusses a secure protocol which is called TrustMe. TrustMe provides a secure way to store the trust values of the peers, as well as a secure way to exchange the trust values between the peers which hold values and the peers which request the trust values. In addition, TrustMe provides anonymity for those peers. [7] propose a new way to compute the trust values for peers. The trust use the historical transaction of peers to compute peers trust values and store them in a global table. Moreover ThTrust decreases the trust values of malicious peers by using the time decay factor and the equivalent evaluation value.

[8] proposed an anonymous and secure protocol to access and manages the trust information using cryptography for decentralized peer-to-peer systems. The peers that hold the information about the trust value should be anonymous and they are called trust mangers. A peer ranks or evaluates others and that vote is kept in the system regardless if the one who voted still in the network or not. When a peer joins the network, he will try to find the proper place and will be associated with a parent in the in work who has high rank .Initially, he will be assign to zero as his trust value .Before starting the communication with other nodes, a peer should contact his parent to get the trust value of that peer. Then, the trust value is computed based on the feedback from theses voters who communicate before.

The paper [9] discuses general ideas about the security and trust in P2P network and explain some related concepts. In mobile peer to peer system, there are some situations may threaten the trust including: unknown peers, cheater identification and replication. The author proposed digital signature to detect malicious peer. It suggest each peer should keep it trust based on the feedback that it received from other peers and that feedback should be encrypted using the public key of that peer. Also, the universal trust set (UTS) was introduced to be the parameters on which the trust score is calculated. UTS allows trust to be extended over the communication between peers.

[10] introduced different security issues such as methods for protection, and suggestions for further enhancements on the security mechanism in PZP systems. It suggests some security principles that should be applied with carefully due to lack of the control server. Including intruders will use any means of penetration, items only need to be protected until they lose value and controls must work, efficient, easy to use, and appropriate. it also discussed some trusts issues between peers and some security protocols. It shows the importance of the trust as major factor in the growth of P2P. There are some issues must be taken into account to ensure trust in the system. The communication between nodes must be secure, the infrastructure must also make it possible to identify the other nodes and the recourses that is shared must be checked carefully.

In [11], it proposed a new trust model of ecommerce security based on voting another peers depending on the agreement between the two sides. According to that vote, it can rank and predicate the behavior of that peer and it gives recommendation to other peers about it. It suggests a trust model in e-commerce environment that can evaluate the communication between peers to insure the security in the whole network.

This paper [12] introduced a new model for peer-to-peer networks defense. It shows some possible attacks in the network especially on peer-to-peer users that can threaten the security easily. Also, it proved clear weakness and low resistance of these networks. It proposed categorization of the attacks in P2P network and shows some analysis and evaluation for them.

Related work provides an extensive survey about the state-of-the-art technology, mechanisms, or algorithms related to our project. To the best of our knowledge, [8], [6] have attempted to build a secure protocol to store and distribute the trust values of peers. In contrast in our work we argue that, it is not only important to ensure the anonymity of the requester and the trust value holder peer but also to identify the malicious peers especially the peer who hold trust value of other peers on the network. In our work, we built a secure connection between trust managers in order to identify malicious peers. Moreover, we relied on

the super nodes in order to help in identify these malicious trust managers.

3. Decentralized Peer to Peer Architecture

In this section we will give an overview about the Decentralized P2P Architecture concentrating on Self-Organizing Decentralized Peer-to-Peer System Based on Well Balanced Multi-Way Trees [1] [5]. Architecture is shown in Figure 1 as in [1]. The first tier is a balanced multi-way tree and the purpose of it is for routing the message. The mechanism naming is based on the topology of this tier. In the second tier, there are two types for each node. One link related to a random sibling of its parent and all the nodes links are in the same cluster. Also, every node has a list of all the active links for the nodes that they communicated before. The trust value for every node is calculated by each neighbor depending on it response time, bandwidth, availability, bandwidth etc. The most reliable node is selected as root. This root makes a list for the most reliable nodes that can be access publicly by a website.

However, that root will not involve in the routing process because it may result in some problems such as a bottleneck. Any node joining the network for the first time, it must contact the web server to get the peers list. After that, the new node will start communicating with one of its closest nodes. Based on its geographical position and using a location algorithm, it will get its position and will communicate with the closest nodes. That node can communicate with its parent in order to get all the nodes related to its parent. The ultimate number of nodes in the network is set in advanced. If any super node exceeds this threshold, it must select one of its children as a new parent for the other children.

The multi- way architecture is mapped by node naming. The identifier for each node is a fixed size of slots and at every level there is an identifier. A node will be on the k_{th} level where the identifier ahs the same as his parent for the first K-1 slots. For each node the connection order is in the slot K and the other slots will be having the value 0. The second tier can help to improve the performance of the network in different ways. It increases the fault-tolerance, decreases routing time by checking all the links related to siblings in the network, and it can help to cope with possible parent bottleneck.

We will use similar concepts to build our network, in order to insure the security and to maintain the network as decentralized. We divided the network into areas where every area has a super node and every super node has number of trust managers. The trust managers are responsible to store and modify the trust values for all nodes in each area.

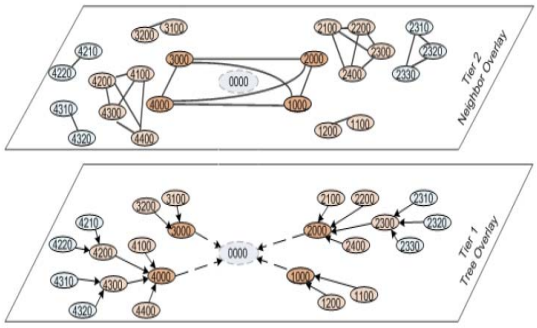

Figure 1. The SOPSys architecture.

4. Trust Value Computation

The trust ranking for nodes in peer to peer network can be derived using two main approaches; transaction-based and user-based rating. In the case of transaction based approach, after finishing each transaction, nodes send the feedback based on their transactions. For each node the trust ranking is calculated after it finish all the transactions with the corresponding nodes. In the user-based approach, a node sends the result of the rating to another peer according the previous communications with the desired node. The trust ranking for a node is the accumulation of all the feedbacks that were sent by other nodes.

In general, more researches have been done based on the user-based approach since it can give better results than transaction based approach. Its reaction is fast like in the case of some groups of malicious nodes behaving together. In order to improve the efficiency, the result of the vote must be stored in the system and be secret and unknown for other nodes even of a node have left the network. That will help to avoid a problem when a malicious node leaves the network only to hide it identity if it has been know as entrusted node.

In [5] the trust ranking is kept in a selected set of nodes and these nodes must acts in fast way to respond to all the requests that have been sent by other nodes. Any nodes can send a request to obtain the trust rating for other nodes even of the trust manger is unknown to that node. A feedback message issued by other nodes is sufficient to judge other nodes. In [3] a trust value can be computed then stored and distributed to other peers using a secure protocol called Trust me based on Public Key cryptography schemes and while exchanging the data, the response sources must be anonymous.

In our work, each peer is assigned a trust value according to the following scenario. Initially a new peer is assigned a trust value of (0), when he joins the network. The peer's trust value is increasing or decreasing based on its behavior in

the network. There are numbers of factors represent the peer's behavior such as, availability of the peer in the network, availability of the claimed service, whether the service causes a harm or not, and bandwidth. Each factor has a weight of a scale of 10 degrees. Therefore, after each interaction the peer which requested a service, will evaluate its partner based on the previous factors, and then he will send a feedback to its partner trust managers.

For simplicity, we give an example to clarify how the trust value is calculated. As we mentioned, each new peer joins the network will be assigned a (0) trust value. Then, if peer A interacts with a peer B for some service, peer A has to generate a feedback based on the following scenario:

- If peer A request a service from peer B, but peer B disjoins out of the network before he offers the service for peer A. Then peer A will decrement the feedback by -2. Otherwise, if peer B grants the service for peer A, then peer A will increment the feedback by 5.
- If the service that was offered to peer A from peer B is the same service which peer A requested from peer B, then peer A increment the feedback by 2. Otherwise peer A gives peer B 0 for the claimed service.
- If peer B causes any harm for peer A, then peer A will set the feedback value to -2.
- If the bandwidth of peer B was relatively fast, then peer A will increment the feedback by 2. Otherwise, if it was slow, then peer A will increment the feedback by 1.

5. MTMI Design and Implementation

This section discusses our proposed algorithm to identify malicious trust manager in P2P networks. Our main goal to ensure that any peer can get the true trust value for another peer. Therefore, our proposed algorithm ensures the anonymity of trust managers which computing trust values for the peers, and identify the malicious trust manager among the group. Each trust manager should be accountable. As well as, a malicious trust manager should be identified, and should not be trust manager any more.

5.1. Trust Manager selection

Figure 2 represents the process of selecting the list of trust managers for new peers. For simplicity, we chose a small part of the P2P network to explain our algorithm. When a new peer joins the network, it will search for its right place. After he finds its place in the network, directly it sends a massage (M1) to the super node, the message request the super node to assign a set of trust managers. Super trust manager associate to the new peer a set of trust managers, as well as super trust manager informs the chosen trust managers about the new peer. Super node sends a message (M2) for each chosen trust manager singed using its private key and the trust manager public key. The message contains a time stamp (T), The identifier of the trust manager, and the identifier of the new peer.

M2: Pr_super(Pu_Tr(newPeerId, T), newPeerId)

Trust manager can know the identifier of the peer whom it has to maintain and update the trust value, and it can ensure that this message is issued by the super node. As well as, this massage is send for each peer in the network requesting the trust value for the peer from the trust manager. Super node selects the trust managers randomly and independently that covers the new peer area, as well as Super Node will send an encrypted list of peer's assigned Trust Managers. Super node just send a message for the new peer to inform it that its trust managers are chosen. In order to protect the message (M3) from replay attack, message (M3) will be time stamped and signed by the super node private key and send to the new peer.

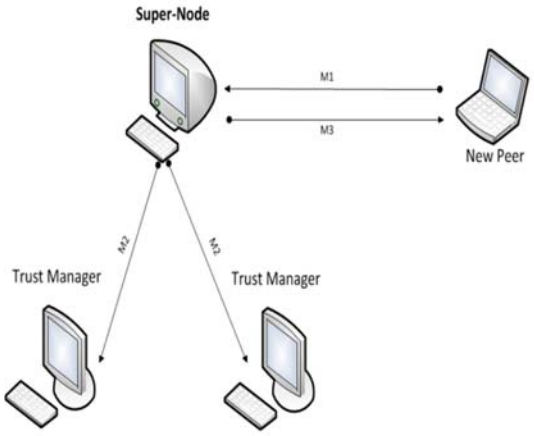

Figure 2: Trust Manager Selection

5.2. Peer request to get a trust value

When a peer wants to interact with another peer, it sends a message (M1) throughout the network, requesting the trust value for its partner. As a result, we ensure the anonymity of the trust managers. Each trust manager who is responsible of the trust value of intended peer, when it receives the message, it sends to the requester peer a replay message (M2). The message is singed and time stamped by the trust manager private key, in order to ensure that no one

can alter the message. The message contains the trust value and the proof that it was chosen to be a trust manager for that peer. The proof is the message which sent by a super node in order to select a peer as a trust manager for that peer.

PrTr(Trust_value||T||Pr_super(Pu_Tr(newPeerId, T)|| newPeerId))

The requested peer decrypt the message using the trust manager public key, then it checks the proof, in order to ensure that the message comes from the true trust manager. Based on the received trust values from the trust managers, the peer will then determine if he should interact with that peer or not.

5.3. Trust Managers Mutual Authentication Protocol

Each trust manager should know the identity of other trust managers which is correlated with them to keep the trust values up to date. In case one of the trust managers was identified as a malicious, the super node will send a message to other trust managers in order to notify them that their identity could be compromised. As a result, they must suppress their current IDs and adapt a new ID in order to maintain the anonymity of their identities. To sum up, each trust manager must maintain at least two unique IDs. The first ID will be in service and the others will be deployed when one trust manager identified as a malicious.

5.4. Message Exchange Between Trust Managers

As it can be seen in Figure 3, Trust managers share secure connections between each other. Over these connections, trust managers exchange trust values of all peers that they share. Each trust manager sends a message (M1) to other trust managers containing all of the peers' IDs and their trust values. This message will be signed by the sender's private key, and it will be encrypted by the receiver's public key. Additionally, this message is time stamped in order to prevent replay attacks.

M1:Pu_Receiver_TM(Pr_Sendr_TM(Peers_IDs||Trust_Values||T))

Once a trust manager receive the trust values from other trust managers, this trust manager compare each peer's trust value that was received with the trust value that he owns for this peer. For each peer's trust value, the trust manager will check this trust value if it matches with the one he has or not. If the peer's trust value matched with the one he has, then nothing can be done. Otherwise, the trust manager will firstly create a list of all peers whose trust value does not match with the one he has. trust manager will add its own trust value for each peer in the created list. As well as, he will add the identity of the trust manger who sends the unmatched trust value with one it has. Secondly, the trust manager will send message (M2) to the super node to inform that there are different trust values that do not match with what he has. This message will be signed by the sender's private key, and it will be encrypted by the super node public key. Moreover, message (M2) will be time stamped to prevent replay attacks.

M2:Pu_Super(Pr_Sendr_TM(Peers_IDs||Trust_Values || Suspended_TM_ID||T))

The above mentioned procedures will be processed periodically to ensure that the information are updated.

Actually most of the trust managers' lists which are sent to the super node will have the same peers. Once the super node receive the messages from all trust managers, he will compute and compare to identify who is the trust manager that gives wrong trust values. If the super node observes that one of the trust managers gives many different trust values. Super node will identify this trust manager as a malicious trust manager. As a result, it will be not a trust manager anymore.

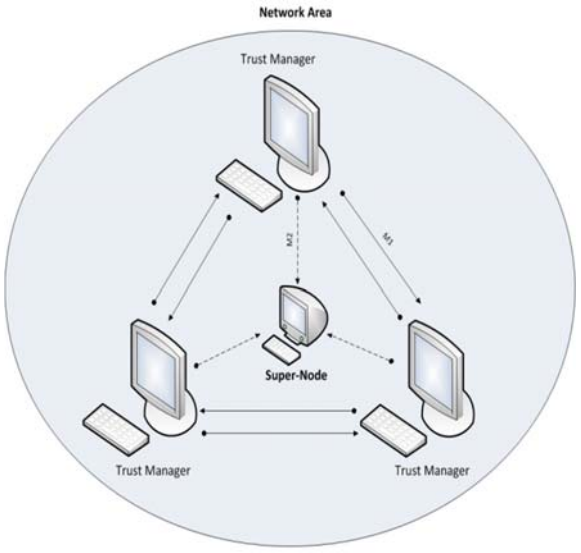

Figure 3: Secure Connection between Trust Managers

6. Experiment and Result

We have implemented a simulation to validate the proposed protocol (MTMI) in peer to peer network, which is designed to detect the malicious trust managers in the network. In P2P environment the number of Peers (nodes) is huge, so we have simulated the peer to peer network environment with different occurrence of malicious TRMG and different network scenarios. We will show only two results as an example. In table 1, we are showing the number

of nodes (N), the number of super nodes (SN), Number of Trust Managers (TrMG), the number of message between TrMG's, and the number of messages between trust managers and super nodes M-TrMG-SN.

In the first scenario, the number of nodes (N) = 10000 nodes, the number of super node (SN) = 4, the number of trust managers (TrMG) = 16. In the second scenario, the number of nodes (N) = 20000 nodes, the number of super nodes (SN) = 8 and the number of trust managers (TrMG) = 32.

Table 1. Scenarios Results

N	SN	TrMG	M-TrMG	M-TrMG -SN
10,000	4	16	240	40
20,000	8	32	992	80

The simulation results was massive, here we only showing part of it that was selected from three different areas to make it clear and easier to understand. The following tables clarify the results more. In table 2 which is based on the second scenario the number of nodes is 20,000, number of trust managers is 32, SN= 8 , and the trust value for each node that was given by each trust manager in each area. Every super node has 4 trust managers, hence 32/8 = 4. The super node will detect the malicious TRMGs in the network according to the result of the trust value for a shared node. All the trust values issued by the trust managers for same node must be the same with some tolerance specified by the super node. If a trust manager gives different result value from other TRMGs, then it will be considered as malicious trust manager. For example, trust managers 21, 22, and 23 respectively, issued trust value 7 for node 12. However, the trust manager 24 gives different trust value as 10 for the same node.

Also, we can see as in table 2, the trust managers 21, 22 and 23 give the same trust values for nodes 13, 14 and 16 while trust manager 24 gives different values. As a result, the super node will identify trust manager 24 as malicious trust manager in the network and must be not be considered as a trust manager anymore in the network. Also, tables 3 and 4 shows the same results.

Table 2

N	TrMG 21	TrMG 22	TrMG 23	TrMG 24	Super Node
P12	7	7	7	10	TrMG 24 is a malicious
P13	5	5	5	8	
P14	7	7	7	9	
P16	7	7	7	5	

Table 3

N	TrMG 35	TrMG 36	TrMG 37	TrMG 38	Super Node
P25	7	7	7	9	TrMG 38 is a malicious
P28	7	7	7	9	
P30	7	7	7	9	
P32	7	7	7	9	

Table 4

N	TrMG 49	TrMG 50	TrMG 51	TrMG 52	Super Node
P39	7	7	7	10	TrMG 52 is a malicious
P42	7	7	7	10	
P44	7	7	7	5	
P45	8	8	8	6	

7. Discussion

Our proposed protocol in this paper ensure authentication, confidentiality, integrity, non-repudiation, reliability and accountability.

Authentication: In our proposed protocol no peer in the network is able to compromise or fake a report message, because all messages are sent encrypted by the intended receiver's public key. Therefore, only the intended receiver peer is able to decrypt an access the message content. In other words, if a peer in a network try to capture a message in order to alter and reply it to the intended peer, the receiver peer will discover and recognize the alter. Moreover, a peer who request a trust value, can ensure about the identity of the trust manager by checking the proof message. A trust manager send to the requested peer the message which is sent by the super node to assign it a trust manager for the peer. As well as, the message is sent to the requested peer is signed by the trust manager private key.

Confidentiality: In peer to peer networks sensitive data such as trust values should be protected. Message holds a trust value should not be sent in clear. The message should be encrypted, therefore no malicious peer can modify or fake it. In our proposed protocol no peer in the network is able to compromise a report message, because all messages are sent encrypted by the intended receiver's public key. Therefore, only the intended receiver peer can compromise and read the message. In other words, if a peer in a network try to capture a message in order to alter and reply it to the intended peer, the receiver peer will discover and recognize the alter.

Integrity: In our proposed protocol we sure about the integrity of all sent messages. As we shown that all the messages are authenticated and encrypted. Therefore no one in the in network is able to alter or change the content of sent message in network. Any attempt to alter any message will be recognized.

Non-repudiation: In our proposed protocol no party can deny a message that he sent it. All messages are signed by the sender's private key then it is sent to the intended receiver. Therefore, no peer is able to deny a message is signed by his private key.

Accountability: Each peer should be accountable and any trust manager are trying to fake the trust values should be identified. Our proposed protocol is able to identify the malicious trust manager, in order to be not a trust manager anymore.

Anonymity: In decentralized system, it is important to hide the identities of the peer that is responsible of computing the trust value, in order to protect them against malicious peers attacking. The proposed algorithm in this paper protects the identities of the peer that responsible to maintain the trust values, and it also ensure their anonymity. The id of the trust manager is encrypted in order to ensure the trust managers anonymity.

Reply messages: It is important to detect the reply message in order to ensure that all messages which are exchanged between the super node, trust managers and peers are not replied messages. In our proposed algorithm Any attempt made by any node to reply any message is avoided by using the timestamp. Any message is out of the reasonable time slot is discarded.
Additionally, there are other important features that being satisfied by our new proposed should be mentioned:

Reliability: Our proposed algorithm ensure the reliability in maintaining the trust value. In the system even when trust manager logged out of the network, we ensure that the trust value is maintain by other trust managers. Each peer trust value is maintain by more than one trust manager to ensure that the requested peer obtain a true trust value and no matter of one of trust manager logged out of the network. As well as, super node can choose a new trust manager quickly and efficiently.

Make decision is very fast: Our proposed algorithm make the decision of the peer to interact with another peer very fast and very efficient. When a peer X want to interact with peer Y, peer X request to get the trust value of peer Y. A single reply message from the trust managers who responsible of peer Y trust value is enough to peer X to decide if he want to interact with peer Y or not. while the case is different in other protocols where a peer asking for a trust value should seek and interact with many parties to get the needed trust value.

No Central Point: P2P networks are decentralized system, as a result, no central entity solution can be practical. Our proposed algorithm take in account this issue. There is no central entity is responsible to maintain trust values. The mission of maintaining and computing the trust values are distributed among multiple entities in the network. In addition, there are several super nodes in the network. The process of selecting Super nodes and TRMGs are dynamic according to varying environment of P2P network.

8. Conclusion

In conclusion, this paper proposed a new protocol to build a secure connection between trust managers based on super nodes in order to identify malicious trust managers in decentralized P2P networks. MTMI makes use of different security services to enforce security based on using public key infrastructure. The security services were proposed are as follows: authentication, confidentiality, integrity, non-repudiation, accountability and anonymity. Moreover, the proposed protocol will help in detecting replay messages and ensure the reliability of the trust values in case of one manager log out from the network. The result presented in this paper prove the efficiency of MTMI in identifying malicious trust managers. Our protocol is resistant to different possible attacks that can happen in P2P networks as we proposed in the discussion section in this paper.

9. References

[1] M. K. Riaz, F. Yangyu, and I. Akhtar, "A multidimensional trust inference model for the mobile Ad-Hoc networks," in 2019 28th Wireless and Optical Communications Conference (WOCC), 2019: IEEE, pp. 1-5.

[2] N. Rahimi, "Security consideration in peer-to-peer networks with a case study application," International Journal of Network Security & Its Applications (IJNSA) Vol, vol. 12, 2020.

[3] S. K. Awasthi and Y. Singh, "Absolutetrust: algorithm for aggregation of trust in peer-to-peer networks," IEEE transactions on dependable and secure computing, 2020.

[4] M. D. Istin, A. Visan, F. Pop, and V. Cristea, "SOPSys: Self-Organizing Decentralized Peer-to-Peer System Based on Well Balanced Multi-Way Trees," in P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2010 International Conference on, 4-6 Nov. 2010 2010, pp. 369-374, doi: 10.1109/3pgcic.2010.61.

[5] S. Balfe, A. D. Lakhani, and K. G. Paterson, "Trusted computing: providing security for peer-to-peer networks," in Peer-to-Peer Computing, 2005. P2P 2005. Fifth IEEE International Conference on, 31 Aug.-2 Sept. 2005 2005, pp. 117-124, doi: 10.1109/p2p.2005.40.

[6] S. Aameek and L. Ling, "TrustMe: anonymous management of trust relationships in decentralized P2P systems," in Peer-to-Peer Computing, 2003. (P2P 2003). Proceedings. Third International Conference on, 1-3 Sept. 2003 2003, pp. 142-149, doi: 10.1109/ptp.2003.1231514.

[7] Q. Shaojie, C. Xingshu, and T. Changjie, "Trust: Transaction History Based Peer-to-Peer Trust Model," in Data, Privacy, and E-Commerce, 2007. ISDPE 2007. The First International

Symposium on, 1-3 Nov. 2007 2007, pp. 242-247, doi: 10.1109/isdpe.2007.34.

[8] A. Visan, F. Pop, and V. Cristea, "Decentralized Trust Management in Peer-to-Peer Systems," in Parallel and Distributed Computing (ISPDC), 2011 10th International Symposium on, 6-8 July 2011 2011, pp. 232-239, doi: 10.1109/ispdc.2011.41.

[9] V. Kumar, "Trust and Security in Peer-to-Peer System," in Database and Expert Systems Applications, 2006. DEXA '06. 17th International Workshop on, 0-0 0 2006, pp. 703-707, doi: 10.1109/dexa.2006.142.

[7] S. Marsh. Formalising trust as a computational concept. In Ph.D. Thesis, University of Stirling, 1994.

[8] J. Douceur. The sybil attack. In IPTPS02 Workshop, Cambridge,MA (USA), March 2002.

[9] E. Sit and R. Morris. Security considerations for peer-to-peer distributed hash tables. In IPTPS02 Workshop, Cambridge,MA (USA), March 2002.

[10] Jung-Tae Kim; Hae-Kyeong Park; Eui-Hyun Paik; , "Security issues in peer-to-peer systems," Advanced Communication Technology, 2005, ICACT 2005. The 7th International Conference on , vol.2, no., pp.1059-1063, 0-0 0.

[11] Yu Wang; Wang Yu; Zhao Yue-long; Hou Fang; , "A New Security Trust Model for Peer-to-Peer E-Commerce," Management of e-Commerce and e-Government, 2008.ICMECG '08. International Conference on , vol., no., pp.399-402, 17-19 Oct. 2008.

[12] Schafer, J.; Malinka, K.; Hanacek, P.; , "Peer-to-Peer Networks Security," Internet Monitoring and Protection, 2008. ICIMP '08.The Third International Conference on , vol., no., pp.74-79, June 29 2008-July 5 2008.