Sybil Node Detection in Peer-to-Peer Networks using Indirect Validation

Chayan Banerjee
Department of Electronics & Communication
Brainware Group of Institutions, SDET
Barasat, India
chayan.in.2014@ieee.org

Shubhra Saxena
Department of Electronics & Communication
SRM University
Chennai, India
shanusaxena_25@yahoo.co.in

***Abstract*—Peer- to- peer networks are used extensively today. Due to this wide use P2P networks is a target of malicious attacks. The most mentionable of them is the Sybil attack. Existing approaches for detection and mitigation of Sybil nodes are either computationally costly or are dependent on belief models found in social networks. It has been found that these belief models are themselves vulnerable to other attacks. In this paper, we propose a new type of indirect validation where we have a two stage validation in place to check that if a suspected node is Sybil or not. We crosscheck belief data from local monitor nodes and community detection data from randomly selected global monitor nodes, and then validate a suspected node. The proposed approach is found to be with less computation overheads and less vulnerable to malicious attacks.**

Keywords— Belief model, Community detection, Indirect validation, Message passing algorithm, Sybil detection

I. INTRODUCTION

Decentralized distributed networks, like peer-to-peer networks are employed in a great deal of applications today. In a typical (distributed) peer-to-peer system, the peers or the participators usually play three important roles, simultaneously. Firstly, they are the source of data and they also share certain data with other peers. This data could be local raw data, such as sensor readings, or could also be other computational results. Secondly, they are the data processors. Each participant locally processes the data according to some rules or algorithms. Thirdly, they are also data transmitters. In a large peer-to-peer system, a direct connection between each pair of nodes is impossible; therefore, the participating nodes usually build up a network. A message is transmitted from one peer to another via the relay operations of multiple intermediary peers (nodes). If attackers control one or more participating nodes then they could modify the local raw data, local computed results, or all of the transmitted data. So by such an attacking mechanism, the attackers can modify the overall computation results of a peer-to-peer system, or even destabilize the entire system [1]. P2P systems are particularly susceptible to Sybil attacks [2]

In Sybil attacks an adversary exploits a system by creating multiple identities and thus taking the disguise of multiple individual nodes. Multiple Sybil nodes then may collude and cooperate with each other and can orchestrate a larger attack by diverting the system resources or disrupting the network operation. The Sybil attack is a particularly harmful attack in sensor networks. The malicious node may behave as if it were a larger number of nodes, for example by impersonating other nodes or simply by claiming false identities. In the worst case, an attacker may generate an arbitrary number of additional node identities, using only one physical device [20].

There has been a prolific research on the formulation of methods to defend networks against Sybil attacks [12]. A popular approach is to use information from social networks [3], [4], [5], [6], [7]. This approach is based on the fact that it is quite costly for Sybil nodes to establish a link with other non-malicious or honest nodes of the network. And so the social network graphs may be used to detect and counter Sybil attacks. But recent studies [13], [14], [15] indicate that the social network based approaches are vulnerable to social engineering attacks. Recommendation system in social networks proposes relationships between users based on background knowledge on users. This knowledge is derived from the interaction between registered users, the friend relationship between them and other things based on their interaction with the social network. Popular social networks make use of this information to make recommendation to users. From the attacker's point of view if he is able to influence this recommendation system and make the network issue targeted recommendations, then he may trick a victim into contacting him [32]. Social networks generally use belief propagation & reputation- based trust model to mitigate the attack and deal with malicious nodes [8], [9]. The reputation based model itself is also susceptible to attacks [10], [11]. Besides these kinds of models, adds a lot of computing, communication and storage overhead.

We propose a two pronged strategy to validate a suspect node (S). The proposed approach which gives a little computational overhead compares Local monitor (local neighborhoods nodes) and Global monitor observations (Global neighborhood nodes) about a certain suspect node before deciding if it is a Sybil node or not.

978-1-4799-5364-6/14/$31.00 ©2014 IEEE

II. Related Work

Certification has been by far the most frequently cited solution as a defense against the Sybil attacks [16]. It involves the presence of a trusted certifying authority (CA) that validates the one is to one correspondence between an entity on the network and its associated identity. This centralized CA thus eliminates the problem of establishing a trust relationship between two communicating nodes [17], [18], [19], [23]. An entity has three potential sources of information about other entities: A trusted authority, itself or another (untrusted) entities. In the absence of a trusted authority either an entity only accepts identities that is has directly validated or if the identity is vouched for by other entities it has already accepted [2].

For Direct validation, the following observations are made:

1. Even when severely resource constrained a faulty entity can counterfeit a constant number of multiple identities.
2. Each correct entity must simultaneously validate all the identities it is presented otherwise a faulty entity can counterfeit an unbounded number of identities.

There are a number of implementation issues specifically about how the CA shall establish the entity-identity mapping. In real-world applications this may incur an appreciable performance cost and a single point of failure.

In Indirect validation, an entity accepts only those identities which were vouched by already accepted entities:

1. A sufficiently large set of faulty identities can counterfeit an unbounded number of identities.
2. All entities must perform their identity validations concurrently; otherwise a faulty entity can counterfeit a constant number of multiple identities.

An obvious danger of accepting indirectly validated identities is that a group of faulty entities can vouch for counterfeit identities [2]. A Sybil node can vote in another Sybil node or it can vote out a healthy node, sometimes called 'stuff the ballot box' attack [20]

III. Proposed Dual Validation Approach

We are proposing an approach where we cross check the results from the two phases of validation (locally as well as globally) in the network., and then decide if the Suspect node is a Sybil or not.

A. Phase 1: Local Validation

Computational puzzles have been a popular method for detection of Sybil nodes [21], [22], [24]. Generally, these types of approaches are based on the fact that if the adversary has finite resource, then it can only support a limited number of malicious nodes in the system. One such defensive scheme Sybil Control [22] enables nodes to distribute computational puzzles to its neighbors and verify that if those neighbors recently used the challenges to solve the computational puzzles, which requires a significant amount of computation to solve but are comparatively easier to verify. The nodes bring fresh puzzles periodically, such that other nodes have to solve newer challenges from time to time to remain in the system.

We have used the same scheme, but not for validating the entire network at once. To reduce the complexity our scheme would limit the distribution of a computational puzzle and its verification to a certain node under suspicion and its adjacent neighbors (or local monitors, LM). The number of local monitors or the adjacent neighbours may be varied to increase the quality of authentication. We have considered a maximum of 4 LM nodes, which will pass challenges on to the suspect node. Now, after the computational phase is over and the suspect is ready with the solution, these 4 individual Local Monitors will verify the response to their requests and generate a belief about the suspect node. The cumulative belief from the 4 local monitors will serve as the *Belief value* of that certain suspect node. So to summarize the steps involved in checking the authenticity of a certain node the neighbour may perform the following, see figure 1:

A. Local Monitors (LM_1, LM_2 . . . LM_n), Challenge the suspect node S. Here 4 adjacent neighbours based neighbourhoods are considered.
B. The suspect solves the computational puzzle.
C. Local Monitor LM_1 requests for a puzzle solution from suspect node S
D. Suspect node S responds with the solution and the challenges from all or select number of local monitors (that it has processed recently).

Belief propagation is a method used to build a reputation system. BP algorithms are used in social network based Sybil defence as discussed in the introduction [8]. The Belief Propagation algorithm is a Message passing algorithm for calculating marginal distributions. Messages are real valued functions that are passed along the edges between variable nodes (of a factor graph). They represent the influence that one variable exerts on its parent or descendent variable [25]. A factor graph is a bipartite graph that contains two kinds of nodes, namely the variable node and the factor node. They are used to model complex real world systems and to derive a practical message passing algorithms like Sum-Product (Belief Propagation), for associated detection and estimation problems [26]. [27] The belief $b_i(x_i)$ of a node i in its value x_i can be understood as how likely the node i think it should be in state x_i. A node bases its decision on messages from its neighbours. A message $m_{a\text{-}i}(x_i)$ from node a to i can be interpreted as how likely node a thinks that node *i* will be in the corresponding state. The belief for a variable node is, thus, proportional to the product of messages from the neighbouring factor nodes (see table 1)

$$b_i(x_i) \propto \prod_{a \in N(i)} m_{a \to i}(x_i) \quad \ldots\ldots\ldots (1)$$

Or in the case of Min-Sum BP algorithm, the belief for a variable node is, thus, proportional to the Sum of messages from the neighbouring factor nodes (or Local Monitor nodes).

$$b_i(x_i) \propto \sum_{x \in N(i)} m_{a \to i}(x_i) \quad \ldots\ldots\ldots (2)$$

In case of the min-sum algorithms the messages are basically using the message and belief that represent the cost of each variable to be in its different possible states. In Max-Product algorithm, the beliefs and messages are represented as probabilities.

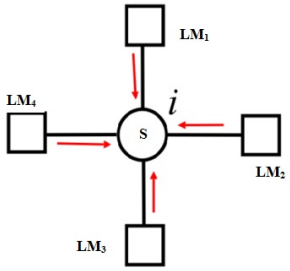

Fig 1: Diagrammatic representation of the belief of a variable node

Table 1: Message passing methods and their corresponding Belief values

Max-Product Method (Pass =1, Fail = 0.2)				
LM_1	LM_2	LM_3	LM_4	S_b
0.2	0.2	0.2	0.2	0.0016
0.2	0.2	0.2	1	0.008
0.2	0.2	1	0.2	0.008
0.2	0.2	1	1	0.04
0.2	1	0.2	0.2	0.008
. . . . .				
1	1	1	1	1

Now the first step of the algorithm is to ask the suspect node to provide its belief value that it will gather from the neighbourhood nodes or its Local Monitors. The suspect node will send messages to the adjacent neighbours, as if to ask about what state they think the suspect node (S) is in. The neighbourhood nodes will initiate the computational puzzle test on the suspect node and will provide a cost or probability value to it depending on their verification of the solution that the suspect node has provided for the challenge that was sent to it. With these feedbacks from the neighbours the suspect node S generates its belief value,

Depending on the belief value we may say that if the node is a Sybil or not. Table 1 shows the verification results of puzzle test by the Local Monitors and the respective belief value calculated by the suspect node. Fig 1, clearly explains the approach, where a suspect node is surrounded by 4 local monitor nodes. Stem plots in fig 3 & 4 shows the different belief values corresponding to the QAVs from the different local monitor nodes.

Limitations of Phase 1: The most mentionable fault that may throw this whole approach in jeopardy is if the belief value is faulty. It may happen due to all or any of the following reasons:

- S is a Sybil / malicious node and is intentionally lying about its belief value.
- S based belief value is low, meaning the suspect nodes have failed in some tests.
- It may happen that the node may be an honest one, but it is surrounded by a community of Sybil or malicious nodes (which are its Local Monitors) and they are deliberately trying to vote the honest node out.

To tackle this kind of situations we crosscheck the belief value by feedback from our Global Monitors (GM). From the GMs we try to have some knowledge about the community or the neighbourhood from which this suspect node belongs to.

B. Phase 2. Global Validation

As already discussed, the basis of indirect validation is validation of new entities by older entities which are already certified by a central authority. We have also seen the problems that may arise in case of a central certifying authority and in case a large number of indirect validating nodes are manipulated.

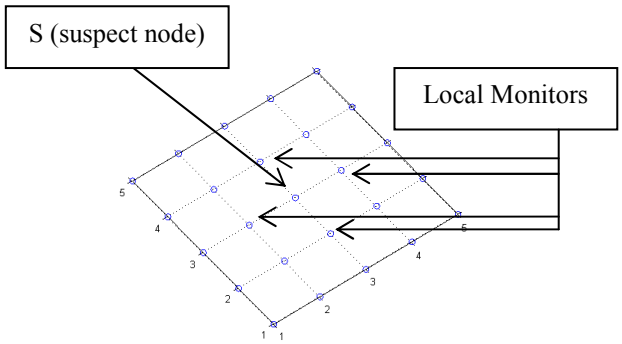

Fig 2: View of a suspect node (S), surrounded by 4 Local Monitors, in a 5x5neighborhoodhbourhood

We propose a new approach where we don't have any direct validation in place, though the nodes may have checks of authenticity among them. But there will be no central authority to provide any security certificates to the nodes. There will be certain randomly selected nodes; the Global monitors (GM), who will validate the suspect node by providing some insight about its community. So the process starts with selecting the GMs from the neighbourhood. We use a random node selected such that it is quite impossible for the adversary to know about the GM beforehand.

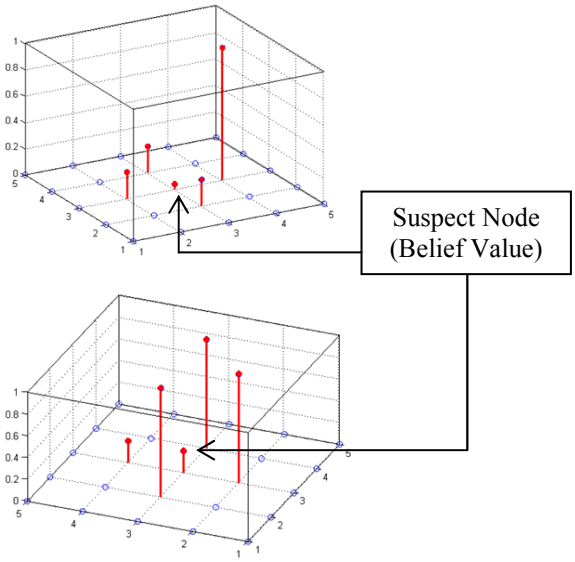

Fig 3: Suspect node- belief value, using Max-Product Method. (a) When LM_2= 1, LM_3=LM_4 =LM_1= 0.2, S_b = 0.08; (b) LM_1=0.2, LM_2=LM_3= LM_4 = 1, S_b = 0.2

Generation of Global Monitor distribution sequence: First we have to know the size of the entire neighbourhood (network size) from which we will select the GM nodes. Let the neighbourhood size be given in a matrix like row-column format as n x n. We have considered a square matrix here for ease of deployment and visualization. The generation of the monitor matrix will require a random sequence of binary bits. Now this random sequence can be generated as shown in the figure 5.

Assumptions: The maximum limit for random Integer numbers to be generated is $(100)_{10}$. It can be represented by a 7 bit sequence $(1100100)_2$. The total neighbourhood dimension (N_{size}) is already known, where $N_{size} = n^2$. Any special characteristic expected of the extracted random sequence is already known, e.g. Min number of 1s and maximum number of 1s.

The complete process of sequence generation is defined in fig 5. The random binary stream is transformed into a matrix, which actually represents the location or the distribution of the Global Monitors in the neighbourhood defined.

Validation by Global Monitor Node: The next step is indirect validation by the individual Global Monitors (GMs). Each randomly selected GMs performs validation checks on the suspect node (S) and submits their observation. The cumulative observation of the total GM nodes is considered.

This paper doesn't deal with the details of the kind of checks or validation algorithm the GMs performs to scrutinize the suspect node. [15] Observes that most Sybil defence schemes work by identifying nodes in the local community around a given trusted node and ranks them as more trustworthy than those outside. As a result Community detection algorithm [30], [31] is used nowadays as a method of detecting Sybil nodes [15], [29]. The logic behind the numerous community detection algorithms is that the Sybil nodes maintain a small community with other Sybil nodes and shares a few numbers of edges with the non-Sybil community (or healthy network). A community can be defined as a group of vertices where the connection is dense. But the connections between those groups or communities are sparse. One of the community detection methods uses random walks from a certain vertex [31] to detect its community.

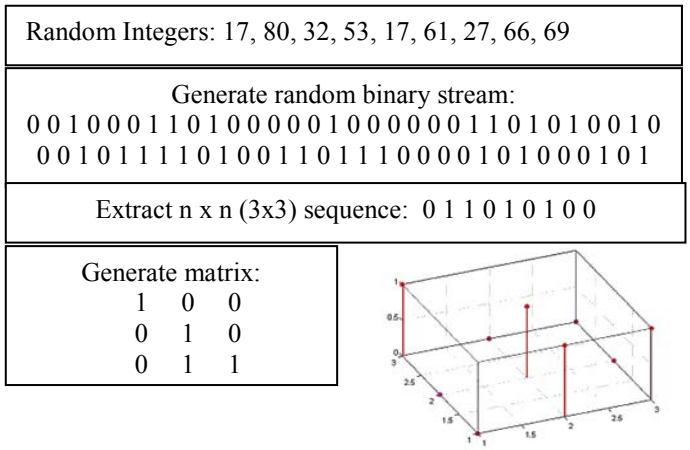

Fig 5: Detailed process of random stream generation

We propose an approach using where each of the elected GMs will individually seek for their respective communities. We have already mentioned the fact that this randomly selected GMs are not validated by any centralized authority and hence may be themselves part of a Sybil or infected community. So to bypass that limitation, after the community identification cycle is, the result from the GMs are aggregated and processed as discussed below.

It is assumed that a Sybil community is very small as compared to a non-Sybil or healthy community. And there are only two possible community clusters in the complete network, namely Sybil and non-Sybil. So the community, to which the majority of GMs belongs, will be tagged as the Non-Sybil community. In case of a tie (node 2, Table2), the community will be tagged non-identifiable (NI). The community chosen by the minimum of the GMs is tagged Sybil community.

Thus, majority of GMs, as their observations will tell, will belong to the same community, i.e. the non-Sybil one.

So as a last step we will check that if the suspect node (S) lies within which of these 2 territories (table 3).

Table 3 shows the decisions about the state of the suspect node, depending on the column 2 & 3 data. The probable states of the suspect nodes are ranked according to their chance of being a malicious one. State 1 corresponds to a healthy (H) or non-Sybil node. State 2 corresponds to a node whose identity is not clear and is quite alarming (A). State 3 corresponds to a Sybil node (I).

Table 2: Detection of Sybil and Non-Sybil communities, similarly coloured GMs belong to same community.

GM Observation				Probable Community
GM1	GM2	GM3	GM4	GM1∈ S
GM2	GM3	GM4	GM1	GM1∈ S
GM1	GM2	GM3	GM4	NI
GM1	GM2	GM3	GM4	NI
GM1	GM2	GM3	GM4	GM4 ∈ NS
GM1	GM2	GM3	GM4	GM2 ∈ S
GM- Global Monitor, S- Sybil community, NS- Non-Sybil or Healthy Community, NI- Non-Identifiable Community				

The location of suspect node is checked and a decision about its state (Sybil or not) is reached, keeping in consideration, both the belief value (from Phase1) and the community information (from Phase 2).

Table 3: Crosschecking the data from Phase 1 & Phase 2 validations

Suspect Node ID	Belief (max-prod)	Probable community	Probable state of S
1	01/0.2	S	A
		NS	H
		NI	H
2	0.008/0.04	S	I
		NS	A
		NI	A
3	0.0016	S	I
		NS	A
		NI	I
I- Sybil Node, A-Alarming, H-Healthy			

IV. Simulation & Discussion

For the proper evaluation of the proposed approach a simulation was run using MATLAB. To preserve the variety of the situations, the simulation is run with random belief values & location with respect to Sybil and non-Sybil communities. To make it simpler the Non-Identifiable community (NI) concept is not depicted in the simulation. The location data is taken as a random set of numbers between -5 to +5, where the negative numbers will correspond to nodes in Sybil Zone and positive numbers will represent nodes in Non-

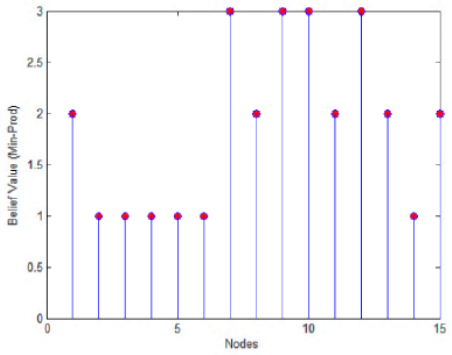

Fig 6: Belief data of node

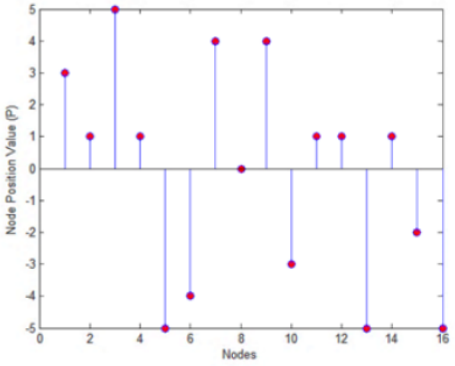

Fig 7. Location data of nodes

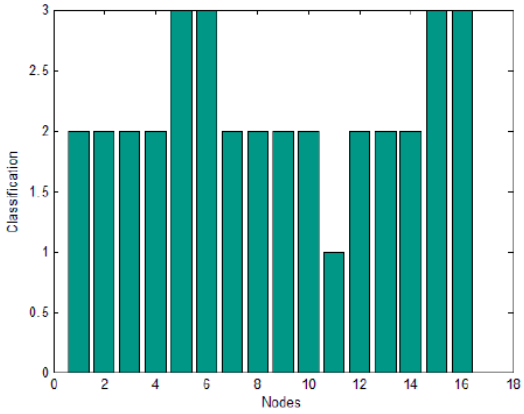

Fig 8. Probable states of the suspect nodes, 1-H, 2-A & 3-I

Sybil zones (fig 7). Finally the bar plot (Fig 8) shows the comparison result following table 3. The 'classification' (along the Y axis) shows the ranks of the suspect node. The most important benefit of this approach is that unlike other prevalent approaches, no single validation method is applied throughout the whole network and through each candidate node. Here the 4 neighbouring nodes of the suspect participates in belief based validation and 4 randomly selected GMs participates in community detection to see if the LMs belong to a malicious community or not.

Now belief based validation of a sybil node fails if the LMs are themselves becomes infected or malicious. From fig. 9 we can see a random belief observations by LMs, about a healthy across a network with 50% malicious nodes. The malicious nodes will deliberately try to vote out the healthy node. But when coupled with the community detection scheme, which comprises of our phase 2 of validation, this problem is mostly eradicated. The reason is that community detection algorithms are not susceptable to attacks which infects the LMs and makes them vote for sybil nodes and vote against the healthy node. The community detection algorithms needs to keep a check on the attack edges of sybil clusters, since the community detection capability decreases with increase in connection between the sybil and healthy clusters in the network.

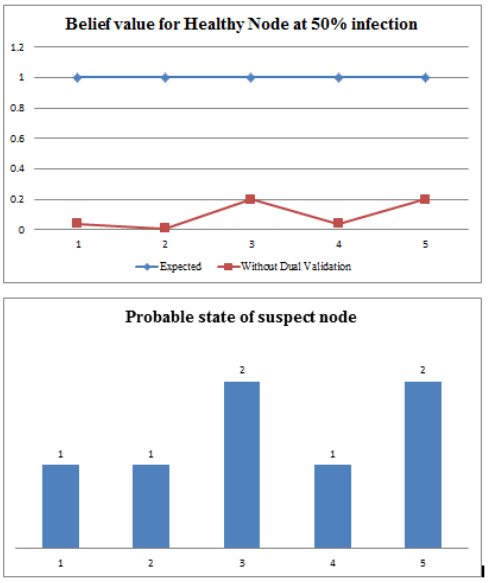

Fig 9: Probable state of suspect node, using dual validation

Above figure depicts that without the dual validation scheme the belief results from a compromised network (50% infection) will lead to a value far less than expected. But dual validation brings forward the participating LMs belonging to the Sybil / malicious cluster and hence the probable state of the suspect node (after proper changes to existing crosschecking table 3) is estimated correctly.

V. Conclusion

We have proposed a new type of indirect validation for detection of Sybil nodes. Our approach though uses a combination of well-established validation methods, but it's less on computing and storage complexity. The primary reason behind it is that we are not using any primary validation method covering the whole network. Since no direct certification is employed hence it is low in communication overhead also. The Monitor or the validating nodes are selected randomly and hence there is a very less chance for the adversary to guess and eventually manipulate the validating nodes.

References

[1] Wei Chang and Jie Wu, 'A Survey of Sybil Attacks in Networks', Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122

[2] John R. Douceur, The Sybil Attack, in Proceedings of 1st International Workshop on Peer-to-Peer Systems (IPTPS),2002

[3] Lu Shi, Shucheng Yu, Wenjing Louy and Y. Thomas Houz, 'SybilShield: An Agent-Aided Social Network-Based Sybil Defense among Multiple Communities',Proceedings IEEE Infocom, 2013

[4] G. Danezis, and P. Mittal, SybilInfer: Detecting Sybil Nodes using Social Networks', NDSS, 2009

[5] C. Lesniewski-Laas and M.F. Kaashoek, 'Whanau: A Sybil Proof Distributed Hash Table', NDSI, 2010

[6] H. Yu, P. Gibbons, M. Kaminsky and F Xiao, Sybillimit: A near Optimal Social Network Defence against Sybil Attack, IEEE Security & privacy.

[7] H. Yu, M. Kaminsky, P. Gibbons, and A Flaxman, SybilGuard: Defending against Sybil Attacks via Social Networks, SIGCOMM, 2006

[8] Gaeta, R.; Grangetto, M., "Identification of Malicious Nodes in Peer-to-Peer Streaming: A Belief Propagation-Based Technique," Parallel and Distributed Systems, IEEE Transactions on , vol.24, no.10, pp.1994,2003, Oct. 2013

[9] Tien Tuan Anh Dinh; Ryan, Mark, "A Sybil-Resilient Reputation Metric for P2P Applications," Applications and the Internet, 2008. SAINT 2008. International Symposium on , vol., no., pp.193,196, July 28 2008-Aug. 1 2008

[10] Avinash Srinivasan , Joshua Teitelbaum , Huigang Liang , Jie Wu , Mihaela Cardei, Reputation and Trust-based Systems for Ad Hoc and Sensor Networks

[11] Kevin Hoffman, David Zage and Cristina Nita-Rotaru, A survey of attack and defence techniques for reputation systems, ACM Computing Surveys (CSUR) Surveys Homepage table of contents archive, Volume 42 Issue 1, December 2009

[12] Aziz Mohaisen, Joongheon Kim, The Sybil Attacks and Defenses: A Survey, arXiv:1312.6349

[13] Leyla Bilge, Thorsten Strufe, Davide Balzarotti, Engin Kirda, All Your Contacts Are Belong to Us: Automated Identity Theft Attacks on Social Networks, Proceeding WWW 2009

[14] Garrett Brown , Travis Howe , Micheal Ihbe , Atul Prakash , Kevin Borders, Social networks and context-aware spam, Proceeding CSCW '08.

[15] B. Viswanath, A. Post, K P. Gummadi and A. Mislove, An analysis of Social Network based Sybil defences

[16] B. N. Levine, C. Shields, and N. B. Margolin, A survey of solutions to the Sybil attack, University of Massachusetts Amherst, Amherst, MA, 2006.

[17] Nitish Balachandran,Sugata Sanyal, A Review of Techniques to Mitigate Sybil Attacks, Int. J. Advanced Networking and Applications.

[18] Karlof, C., Wagner, D., Secure routing in wireless sensor networks: Attacks and countermeasures, Ad hoc Networks Journal (Elsevier) 1(2–3) (2003) 293–315

[19] J. Newsome, E. Shi, D. Song, and A. Perrig. The Sybil attack in sensor networks: analysis & defences, In Proceedings of the third international symposium on Information processing in sensor networks, pages 259–268, 2004

[20] Newsome, J.; Shi, E.; Song, D.; Perrig, A., "The Sybil attack in sensor networks: analysis & defenses," Information Processing in Sensor Networks, 2004. IPSN 2004. Third International Symposium on , vol., no., pp.259,268, 26-27 April 2004

[21] Borisov, N., "Computational Puzzles as Sybil Defenses," Peer-to-Peer Computing, 2006. P2P 2006. Sixth IEEE International Conference on , vol., no., pp.161,166, 6-8 Sept. 2006

[22] Frank Li, Prateek Mittal, Matthew Caesar, Nikita Borisov, SybilControl: Practical Sybil Defense with Computational Puzzles, arXiv:1201.2657

[23] Rowaihy, H.; Enck, W.; McDaniel, P.; La Porta, T., "Limiting Sybil Attacks in Structured P2P Networks," INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE , vol., no., pp.2596,2600, 6-12 May 2007

[24] Tegeler, F.; Xiaoming Fu, "SybilConf: Computational Puzzles for Confining Sybil Attacks," INFOCOM IEEE Conference on Computer Communications Workshops , 2010 , vol., no., pp.1,2, 15-19 March 2010

[25] Ying Hu; Kuh, A.; Yang, Tao; Kavcic, A., "A Belief Propagation Based Power Distribution System State Estimator," Computational Intelligence Magazine, IEEE , vol.6, no.3, pp.36,46, Aug. 2011

[26] H.-A. Loeliger,An introduction to factor graphs,IEEE Signal Proc. Mag, Jan. 2004, pp. 28-41.

[27] Megha Khosla, Prof. Dr. Kurt Melhorn, Dr. Konstantinos Panagiotou, Prof. Dr. Kurt Melhorn, Dr. Konstantinos Panagiotou, Message Passing Algorithms, Master's Thesis in Computer Science, Max-Planck-Institute for Informatics, 2009.

[28] F.R. Kschischang, B.J. Frey, and H.-A. Loeliger, Factor graphs and the sum-product algorithm, IEEE Trans. Inform. Theory, vol. 47, pp. 498-519, Feb. 2001

[29] Zhuhua Ca,Christopher Jermaine,The Latent Community Model for Detecting Sybil Attacks in Social Networks,LDB '11, August 29- September 3, 2011, Seattle, WA

[30] Andrea Lancichinetti and Santo Fortunat1,Community detection algorithms: a comparative analysis, Physical Review E 80, 056117 (2009)

[31] Pascal Pons, Matthieu Latapy: Computing Communities in Large Networks Using Random Walks. J. Graph Algorithms Appl. 10(2): 191-218 (2006)

[32] Danesh Irani, Marco Balduzzi, Davide Balzarotti, Engin Kirda & Calton Pu, Reverse Social Engineering Attacks in Online Social Networks, Lecture Notes in Computer Science Volume 6739, 2011, pp 55-74