SYBIL DEFENSE FOR ONLINE SOCIAL NETWORKS USING PARTIAL GRAPH INFORMATION

by

VRITANT NARESH JAIN

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT ARLINGTON
December 2011

To every penny; ever spent,
To wayward journeys restricted heart bent,
To ever influencing best friends,
To co − operating room mates; benevolent,
To the ones back home who wished me well,
To both those women who gave me birth...

But most of all, I dedicate,
This paper that marks my growth; my tell,
To every one who ever loaned,
To the wise man who′s wisdom′s gold,
and to him, who I my life owe.

−Vritant Naresh Jain.

ACKNOWLEDGEMENTS

I would like to thank my supervising professor Dr. Matthew Wright, whose weekly meetings *always* kept me *on my toes* and *above my head.* I would also like to thank my academic advisors Dr. Manfred Huber and Dr. Vassilis Athitsos for their *natural intelligence* in *Artificial Intelligence*, for their most interesting courses, and especially for their interest in my research and for taking time to serve on my dissertation committee.

Finally, I would like to express my deep gratitude to people who have encouraged and inspired me and sponsored my undergraduate and graduate studies. I am greatful to my best friends, my roommates, iSec lab members, my brother, sisters, and most of all my Parents, who are the best one could have.

November 11, 2011

ABSTRACT

SYBIL DEFENSE FOR ONLINE SOCIAL NETWORKS USING PARTIAL GRAPH INFORMATION

VRITANT NARESH JAIN, M.Sc.

The University of Texas at Arlington, 2011

Supervising Professor: Dr. Matthew Wright

Online social networks (OSNs) today are proprietary, in the sense that communication between users requires the users to be part of the same OSN. This raises privacy issues and reliability concerns among users, and calls for an open, interoperable, and distributed OSN infrastructure that would link different OSNs together. Any decentralized system is vulnerable to *Sybil attacks*, in which an attacker claims multiple identities called Sybils, to overwhelm the OSNs and defeat standard techniques used to protect against attacks such as message spam. The state of the art defense against these attacks is SybilInfer, which utilizes the fast mixing property of social networks to distinguish between Sybil nodes and honest nodes. SybilInfer, however, assumes a centralized system with a complete view of the social network. In this thesis, we investigate the effectiveness of applying SybilInfer on open and decentralized networks, and we propose improvements that would make SybilInfer deployable in such a scenario. These improvements facilitate a user of one OSN to listen to messages from other users of another OSN without the fear of spam due to a Sybil attack. We show that the proposed improvements greatly reduce the number of Sybil nodes misclassified as honest users and make SybilInfer more accurate in classifying members of other OSNs.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure Page

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Well known micropublishing networks today are proprietary in the sense that communication between users requires the users to be part of the same network. This raises privacy issues and reliability concerns among users, and calls for an open, interoperable, and distributed infrastructure that would link different networks together. However, such a decentralized system is said to be under a Sybil attack where an attacker claims multiple identities to compromise a large fraction of nodes in the system. The malicious identities introduced by the attacker are called Sybil identities. Systems that depend on assumptions that at least a fraction of the identities in the system are honest are particularly vulnerable to such attacks, as the attacker is able to out vote the honest users by introducing enough Sybil identities in the system [2]. Some examples of such systems are Byzantine fault tolerant systems [3], reputations systems [4], online voting systems, etc.

A Sybil defense protocol can be classified as either centralized or de-centralized. A centralized system requires a trusted central authority that verifies whether an identity is honest or Sybil on behalf of the users. Some of the popular centralized Sybil defense protocols include IP address binding, where the centralized authority binds each identity to an IP address, binding an identity with government issued identification or an e-mail address, to distinguish between honest identities and Sybil identities as it is too difficult for an attacker to forge such identification [5]. Such a protocol can be discouraging to users due to privacy concerns and concerns about system failure due to high traffic, and can also prove to be a single point of failure for the system when the trusted authority is compromised [2]. In a decentralized Sybil defense protocol, on the other hand, the lack of a centralized authority makes it easier for an attacker to claim an unlimited number of Sybil identities, especially in the absence of an efficient Sybil defense protocol.

In this thesis we study the defense against Sybil attack on an Online Social Network (OSN). An OSN is a social networking application which enables users to communicate with each other. Facebook, Twitter, Google Plus, and Ping are among the popular OSNs today. To defend social networks from Sybil attacks researchers have proposed several Sybil defense protocols like Sybilguard [2], Sybillimit [6], SybilInfer [7], etc. We investigate the efficiency of a Vanilla SybilInfer (an unmodified SybilInfer protocol) on distributed OSNs simulated over the FETHR protocol. We then propose three improvements over a Vanilla SybilInfer that customize SybilInfer for practical use over OSNs and verify the customized SybilInfers improvement in accuracy of results over simulated scale free networks and real world networks.

In the following chapters, we study the practical implementation of a state of the art Sybil defense protocol, on distributed social networks. In Chapter 2, we look into several Sybil defense protocols that are built on an assumptions that Social networks are fast mixing [2]. We discuss two protocols which are frameworks for decentralized OSNs, Mr.Privacy and FETHR and the need for a decentralized OSN. In Chapter 3, we describe an algorithm built over the Barabasi - Albert Model [8] for generating scale free networks. We use each generated networks to simulate a FETHR server as a model OSN and simulate our network model as independent but interconnected FETHR servers to investigate the defense against a Sybil attack on decentralized OSNs. Chapter 4 studies the effect of using an out of the box SybilInfer protocol [7], which is set up on the network generated in Chapter 3. We also propose modifications to the protocol to adapt it for practical implementation on social networks, which are later verified in Chapter 5. Finally, Chapter 6 summarizes the proposed modifications to the SybilInfer protocol for better adaptation on open, decentralized micropublishing networks.

CHAPTER 2

BACKGROUND

In this chapter we discuss social networks and the need for open social networks. We then survey a specific class of Sybil defense protocols and their effectiveness in defense against a Sybil attack on social networks.

2.1 Social Networks

A social network can be simply defined as a network of inter-personal relationships between people. A social network graph (SNG) is a graph representation of a social network in which nodes represent users and the edges represent established trust relations between these users. An Online Social Network (OSN) is a social networking application which enables users to communicate with each other. Facebook, Twitter, Google Plus, and Ping are among the popular OSNs today. A common feature among these OSNs is a centralized authority that is responsible to provide services to users of that particular network. The centralized authority is also responsible to maintain the privacy of the users, as per the privacy policies agreed to by the user and the centralized authority. These privacy policies vary from OSN to OSN and are a key issue that users consider to decide among the various applications [9]. The services offered by the networks also differ to a large extent and might influence a user to choose one network over another.

Table 2.1. Registered users in OSNs (Aug, 2011). Reproduced from [1].

Online Social Network	**Registered Users**
Facebook	750,000,000
Qzone	481,000,000
Twitter	200,000,000
RenRen	170,000,000
Vkontakte	135,000,000
MySpace	125,000,000
Badoo	122,000,000
Orkut	120,000,000
Bebo	117,000,000
Linkedin	100,000,000
Viadeo	35,000,000
Google+	20,000,000
Foursquare	10,000,000
Friendfeed	2,000,000

2.1.1 Need for open micropublishing social applications

Table 2.1, reproduced from Vincenzo Cosenza's Social Media Statistics [1], shows the approximate number of registered users in the most popular OSNs today. As seen in the table, while Facebook clearly has the most number or registered users, it does not distinctively dominate the social networking market, as evidenced by the comparable popularity of so many websites. The lack of a clear leader can be attributed to the fact that different users prefer different features and/or privacy settings. Facebook's privacy concerns [10] and Twitter's fail whale, a symbol of its frequent system failures [11] are perhaps as well known as the OSNs themselves. As a result, there is no clear choice of the best OSN. In China, about a dozen micro-blogging websites like 'Weibo' were until recently [12] the only open platform to discuss sensitive political issues and other public affairs [13]. Thus, while these OSNs are popular, there is no clear leader among them. The issues with these social applications are twofold: they are proprietary and centralized. For instance, to communicate with a Twitter user, and create a social link in the Twitter system (by following another user) and sending Twitter messages (tweets). Also, a centralized approach implies a central point of failure as if the centralized authority is compromised, the entire system which relies on that authority is also compromised. As more number of users join the service, it can also lead to frequent crashes that Twitter is infamous for [11], to add to the woe of users already cencerned about their privacy.

2.1.2 Open micropublishing social applications

A micropublishing social application is an OSN where users publish their posts and subscribe to other user's post. A micropublishing social application is called open if a user registered on that OSN can commumicate with other users registered on other open OSNs. In this section we give two examples of open micropublishing social applications, SocialBar and BIRDFEEDER. SocialBar is powered by Fischer et al.'s proposed system called Mr. Privacy [14], a social application framework based on the Simple Mail Transfer Protocol(SMTP) and Internet Message Access Protocol (IMAP), which are standard emailing protocols. This email-based framework allows for data to be owned by users and improves system scalability. The framework allows OSNs to share rich content, music playlists, GPS locations, etc. through emails. Neither the users nor the OSN designers have to be concerned with bootstrapping the underlying protocols.

Sandler and Wallach have proposed an open decentralized micropublishing service called Featherweight Entangled Timelines via HTTP Requests (FETHR) [15]. FETHR provides for a robust and decentralized framework built over HTTP, with no intent to replace Twitter or other OSNs, but to overcome the limitations of Twitter, by facilitating users to choose their service providers. The clients interact with the usual HTTP GET and POST requests for communication. The protocol however, has several critical issues to be addressed, like group communication, allowing for sharing of rich content and privacy.

Since they are decentralized, open approaches like FETHR and Mr. Privacy are vulnerable to Sybil attacks and require a reliable and robust Sybil defense protocol that can be practically implemented as well.

2.2 Sybil Defense

We now go through a brief history of Sybil defense literature. First we discuss Doucer's paper on Sybil attacks [16] and summarize a survey on Sybil defense by Levine et al. [5], then describe in brief the workings of three Sybil defense protocols, Sybilguard [2], Sybillimit [6], and SybilInfer [7]. Finally we study how these protocols compare against each other through Vishwanath and Post's analysis of social network-based Sybil defenses [17].

2.2.1 Early Sybil defenses

Douceur [16], in their[1] paper termed what was known previously as pseudo-spoofing [18], a Sybil Attack. The author proves that a Sybil attack is always possible unless the identities of the system coordinate with some assumptions of resource par-

[1] The use of plural pronoun for the number of authors here is ironic similar to Douceur's use in The Sybil Attack [16].

ity [16]. The paper also outlines a few limitations of Sybil defense protocols that still hold true, in that even when an attacker is restrained strictly in terms of resources; it will always be able to counterfeit a constant number of forged identities (Sybils). The paper outlines the need for concurrent verification of all identities to prevent the attacker from forging an unbounded number of identities. The author was the first to term such identities as Sybil idenities and to point out the threat of a Sybil attack. The author shows that a sufficient number of attackers may result in an unbounded number of Sybil identities, thereby compromising the entire system.

Levine et al. [5] have classified about 90 papers in Sybil defense literature into 11 categories. According to the authors, a majority of these papers either offer no solution [19, 20, 21, 22, 23, 24] or suggest the use of trusted certification [25, 26, 27, 28, 29] with a central authority that would certifiy a user as honest. Fifteen papers offer the solution of resource testing [30, 26, 31, 32, 33], designed on the assumption that Sybil identities are limited to the attackers' resources and peers can challenge the computing or storage ability of users to ascertain which users are honest. Checks are carried out to determine if a group of identities carry fewer resources than they would if they were honest and independent. A few papers [34, 28, 35, 36, 37] propose regular and periodic resource tests on all identities, thereby making it costlier for the attacker to introduce Sybil identities. Other solution categories include using trusted devices [38, 39], where a special hardware device would ensure unique identities by making it costlier for the attacker to create Sybil identities, and building Reputation systems [40] , in which the user's reputation is based on her reputation graph, which is constructed from on the user's interactions with other users and is dynamic in nature. These approaches are either inefficient or not specific to the use of Online Social Networks for defense against a Sybil attack.

The state of the art Sybil defense protocols are based on the assumption that social networks are *fast mixing*, A graph is said to be Fast mixing if a random walk on it approaches the stationary distribution very quickly [41]. It means that while Sybilguard was the first protocol to use the fast mixing property of social networks for defense against Sybil attacks, Yu et al. first prove that social networks are indeed fast mixing and propose an optimal Sybil defense protocol, Sybillimit [6]. Since a social network graph (SNG) is Fast mixing it means that nodes in a SNG are usually well connected and removal of few trust relations between honest users would not result in disconnections of a large number of honest users, as there is a high probablility that there exist mutliple paths between any two nodes. Under a Sybil attack, however, a large number of Sybil identities are connected to the social network graph through few trust relations between a real identity (honest user) and an attacker. Such relations are called attack edges. Removal of an attack edge would result in disconnection of a large number of Sybil identities. The set of edges which's removal might result in disconnection of a significant number of identities in a social network graph are called as quotient cuts. The problem of detecting Sybil identities is now reduced to finding such quotient cuts. Unfortunately, this problem is NP-hard to solve [2] .

Yu et al. [2] propose SybilGuard, the first in a series of Sybil defense protocols that make a Sybil attack independent of the number of Sybil identities introduced. These Sybil defense mechanisms are among the protocols that assume that social networks are usually *fast mixing*, but a Sybil attack results in quotient cuts as an attacker is limited in its ability to form sufficient trust relations with honest users of a network. Thus all the Sybil identities created by the attacker are connected through these limited number of trust relations (edges in the SNG) and result in a small set of edges, detection of which would free the network from Sybil identities. SybilGuard uses random walks performed on nodes of the social networks based on pre-determined but random routing tables that each node has for its neighbors on the network. Yu et al. first prove that a random walk of length at $O(log\ n)$ is sufficient to reach a node from a stationary distribution with probability of least $(1 - 1/n)$. Then they show that in a network of n nodes and g attack edges, the probability of a random walk of length l having an attack edge is at most $(g * l/n)$. SybilGuard requires each node in the network to have registry tables and witness tables, where nodes along these random walks can register their public keys. These public keys are used to verify if a user's random walks intersect with at least half the honest user's random walks, in which case the user's public key is accepted and the user is considered honest. SybilGuard ensures that in a network of n users, at most $O(\sqrt{(}n) * log(n))$ Sybil identities are accepted with a probability of $(1 - \delta)$, where δ is a small constant, given the number of attack edges is not more than $o(\sqrt{(}n)/log(n))$. However, SybilGuard allows a lot of Sybil nodes to be accepted, and it assumes that social network are *fast mixing*, but does not prove the same for the real world [6].

SybilLimit [6] is a near optimal Sybil defense protocol that improves on SybilGuard by reducing the number of Sybil nodes accepted by a factor of $O(\sqrt{(}n))$. It also provides evidence that social networks are *fast mixing* based on three real world social networks, Friendster, LiveJournal and DBLP. Sybillimit consists of two protocols, a secure random route protocol and a verification protocol. The random routing is similar to SybilGuard except that it performs many walks of shorter length and stores only tails (the last edge of a walk) instead of the whole path as stored by SybilGuard. An honest user will classify another user as honest or Sybil based on the number of tails that intersect between them. SybilLimit accepts $O(log\ n)$ Sybil users, allowing for at most $o(n/log\ n)$ attack edges.

2.2.2 SybilInfer

Danezis and Mittal propose SybilInfer [7], an approach based on Bayesian inference to identify Sybil nodes. In addition to classifying users as honest or Sybil, the protocol labels each node with the degree of certainty of it being honest. This protocol also relies on random walks performed over the social network, stored as a set of traces T. A probabilistic model, built using T, describes the probability that T was generated by a set of honest nodes, X, which gives us the probability P[T/X is Honest]. By applying Bayes theorem, we get P[X is Honest/T], which is the probability that a certain set of nodes is honest. Thus to label a node as honest or Sybil, we need to sample subsets of X, generate traces T with respect to X and find P[X is Honest/T].

To label each of nodes as honest or Sybil, instead of considering every possible subset of nodes for X, any sampling technique can be used iteratively to get accurate estimates of the degree of certainty of each node being a Sybil. The authors suggest Metropolis Hastings [42] for this purpose which is a Markov chain Monte Carlo sampler. At the end of a certain number of iterations, we can ascertain the probability that a node is honest. SybilInfer is very accurate. In a scale free network in which 10% of nodes have trust relations with the adversary, only 5% of the Sybil nodes are wrongly marked as honest.

The accuracy of SybilInfer does not come without a cost. The protocol relies on three assumptions: First, it is required that at least one honest node in the network is known to be honest to break the symmetry between the sets of honest nodes and Sybil nodes. Second, it is assumed that social networks are *fast mixing*; and Finally, it is necessary for a node to know the complete SNG topology. The former assumptions are quite reasonable, but to assume that a node can have the entire networks information is both impractical given that well known OSNs today have millions of registered users and can raise several privacy issues (as discussed in Section 2.1.1). The authors point out that social networks, once mature, are quite stable and non-dynamic to a large extent [7]. While this may be true, it should be noted that a Sybil defense system would be needed more often by a user who is new to the network and does not have stable connections. Also, the space required to store the necessary SNG information is about 187 GB [7], which makes a decentralized implementation of the protocol largely impractical. The authors propose more practical alternative implementation in which every node has partial graph information , probably every node in a two hop boundary of the user [7]. SybilInfer is then applied on this information to assign a degree of trust with every user in this two hop boundary (or neighborhood). For the nodes that do not lie in the honest user's two hop boundary, there is a high likelihood that they would like in the two hop boundary of a node that the honest user has already assigned a degree of trust to. The degree of trust of these nodes can be formulated based on a function of trust of the intermediate nodes, which is more formally proposed in Section 4.

Viswanath and Post [17], in their analysis of social network-based Sybil defenses, provide a comparison of SybilGuard, SybilLimit, SybilInfer and SumUp. The authors treat the algorithms as a black box and compare them to study how they perform on the same kind of network topology and attacker strategy. The authors [17] intuit that these Sybil defense systems work by detecting local communities in the social network graph. Further, they study the application of community detection algorithms [43] for Sybil defense. The authors observe that, while the users of a SNG that are tightly connected to each other are more likely to be ranked higher, different algorithms rank equally well-connected nodes in a different way. The authors also contribute to Sybil defense analysis by showing that when there is a crisp boundary between communities, the nodes in the local community of the trusted node are ranked higher than the ones in other communities, where there is no crisp boundary.

CHAPTER 3

NETWORK MODELS

In this Chapter, we describe two network models to study the effect of a Sybil attack on a social network graph. First, we describe a scale-free model where users would subscribe to their social network services provider, (FETHR or FETHR-like) and could interact with any user who could be subscribing to another social network service provider of choice. Such interoperation could give every user a freedom to choose the kind of services they like, instead of being tied into with social network features that do not meet their needs. For example, Facebook has often been criticized about its privacy issues [44, 10], and Twitter has length limitations per post and other issues [45]. If a Facebook user could communicate with a Twitter user they would each be able to use the system that best met their respective needs. Additionally, this would help distribute users across a range of providers, relaxing server failure concerns due to excessive traffic [11] and privacy concerns [10]. Our second model uses three comparable real world social network graphs with simulated connections between the real SNG's, to validate our results in the real world.

3.1 Scale-free model

Our scale-free model of a SNG is generated using a three step procedure. First, the nodes are assigned to sub-networks based on a power law probability. The nodes are then connected to other nodes in the sub-network in a scale-free manner. Finally, nodes are connected to nodes in other sub-network to simulate connections between sub-networks, also in a scale free manner.

We simulate a hundred sub-networks with a total of 100,000 users, we assign each user to a network n with the probability $(1/n * n)$. As a result, we get a hundred sub-networks with no edges between them yet. An additional user is assigned to each network representing the servers that would run SybilInfer on its subnetwork. These servers are called *FETHR servers*, all of which are connected to each other. Figure 3.1 shows the probability with which a node would become a member of each network and Figure 3.2 shows the number of users in each network for one instance of the assignment.

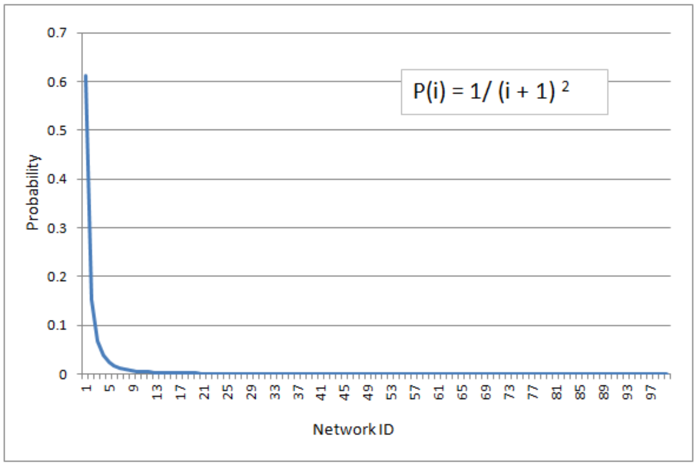

Figure 3.1. Probability with which nodes are added to each network.

These nodes are then connected to other nodes in the sub-network according to the Barabasi-Albert model [8] which generates scale-free networks according to a power law degree distribution. According to the model, a node that has higher number of connections is more likely to get connected to newer nodes as they are

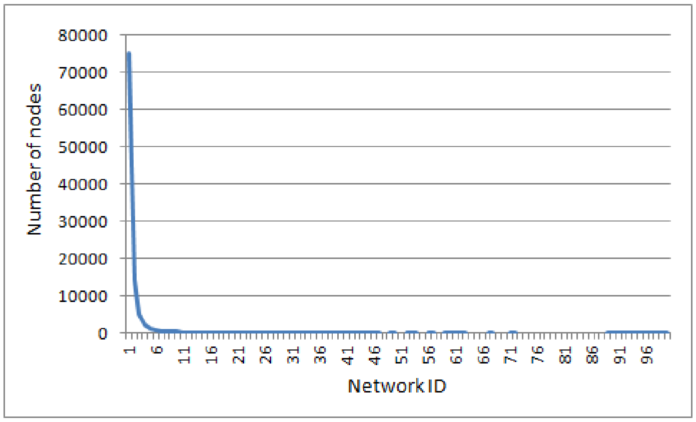

Figure 3.2. Number of members in each network.

added to the network. The network begins with m nodes, where m>2, and each node should should be connected to atleast one other node. Then nodes are added in a scale free manner to the network. The node is connected to every existing node i in the sub-network with the probability $(K_i \ / \ \Sigma_j \ K_j)$, where K_i is the degree of the node i. In real world networks however, the users of several sub-networks would be connected to users of other sub-networks as well. We simulate connections between sub-networks with a probability proportional to the degree of nodes as well.

To simulate connections in a sub-network of our scale-free model, we use a Barabasi-Albert model where the number of initial nodes are 3. To simulate connections between sub-networks, each node is connected in a scale-free manner to every node l of every sub-network with a probability proportional to the degree of that node, as represented by the expression $deg(l) \ / \ \Sigma_m \ deg(m)$. The FETHR servers are provided with connection information of all the nodes within a n hop boundary of all the nodes in its sub-network. For our scale-free model, we use $n = 2$ as proposed in SybilInfer [7].

3.2 Real network model

To study the effect of a Sybil attack on real world networks, In addition to the scale-free model described above, we use a real network model with three social network graphs with comparable sizes: Geom [46], a collaboration network among authors with 7343 nodes; Gnutella [47, 48], a peer to peer network with 6301 nodes; and URV [49], an email network with 1133 nodes. The nodes in these SNGs are connected to nodes in other SNGs with a probablility proportional to the degree of the nodes similar to the interconnections between the sub-networks in the scale-free model.

CHAPTER 4

SYSTEM MODEL

In this Chapter, we study the security aspects of applying SybilInfer to a cluster of interconnected but independent servers which could implement FETHR or FETHR-like services for communication.A user subscribing to a FETHR server may avail services from another user which may be subscribing to the same FETHR server, or might possibly be subscribing to another FETHR server, or even another open micropublishing network provider that inter-operates with the FETHR server of the user that wishes to avail the service. The service could be anything like viewing the user's post, querying for posts with a particular hashtag which is a popular search mechanism in twitter, or even simply looking up a user. In the absence of an efficient Sybil defense protocol, the user might get spammed results from an attacker posing a multiple identities, and if the attacker's responses are ranked higher, the interoperability of the micropublishing networks is compromised. We suggest tweaks to SybilInfer and in Chapter 5 we show that the proposed tweaks reduce the number of Sybil identities classified as honest to a large extent.

4.1 Unpredictability of a Vanilla SybilInfer

SybilInfer [7] was primarily designed with the intention that every node trying to distinguish between honest and Sybil nodes would consider itself as the honest node required to break the symmetry between honest users and Sybil users. Thus, SybilInfer assumes that there is at least one honest node in the entire network and every node runs SybilInfer for itself considering itself as that honest node. However, for the model network defined in Chapter 3, implementing SybilInfer at every node would be highly inefficient for security issues, as it requires the complete social network topology to be known by every node. This raises a lot of privacy issues as some users would like to keep their trust relations confidential.

Every FETHR server(Chapter 3) provides its nodes with services like publishing and subscribing posts, direct messaging, searching other users, etc. These services require every node of the network to be associated with a trust factor. Since each node subscribes to its FETHR server for such services, it is sufficient for the FETHR server to run the SybilInfer protocol on behalf of each node.

In Chapter 5 we demonstrate the risk of implementing a Vanilla SybilInfer on the largest sub-network of the scale-free model as described in Chapter 3. The protocol assigns a trust factor to nodes based on Bayesian inference over a two hop boundary of all the nodes in the sub-network. The protocol selects a node at random in the sub-network to be an honest node, which is required to break the symmetry between honest nodes and Sybil nodes. This honest node is also the initial state of the sampling algorithm used in SybilInfer to sample honest configurations out of the graph. We show that SybilInfer in the current form, for such a network, returns misclassifies users as Sybil in an unpredictable manner, as This is due to its dependance on a single honest node chosen at random. SybilInfer works by partitioning the sub-network into an honest graph and a Sybil graph. In partuclar, when the honest node chosen at random is better connected to the Sybil nodes, the trust factor of the Sybil graph is increased. Thus, there is a need for an improvement in the protocol to make Sybil detection independent of a single node.

4.2 Improvement 1: Change in the initial state of the MCMC Sampler

The FETHR server contains graph information of a two hop boundary of the entire subnetwork that subscribes to it. To choose a single random node as the initial state of the sampling algorithm causes the latter samples to be dependent on that node. To overcome this dependence, we propose to initialize the sampling algorithm with the entire subnetwork. This way, latter samples are no longer dependant on a single node, but a new sample is chosen with a probability which is a function of all the nodes in the sub-network. So even if a new nodes are better connected to the Sybil graph when compared to the honest graph, they do not influence the chosen samples as much as when the honest node chosen at random to solely constitute the initial state is one of those nodes.

4.3 Improvement 2: Using multiple honest nodes

As discussed in the previous section, in addition to sampling subsets of nodes, SybilInfer [7] uses the honest node to break the symmetry between honest graph and the Sybil graph, so as to assign trust to the honest graph. We enhance SybilInfer for use in our setting. Instead of dependence on a single honest node, the FETHR server could choose K distinct nodes it believes to be honest and run SybilInfer K times considering each of the K nodes as the honest node to distinguish between the Sybil sub-graph and honest sub-graph. If any of the K runs classify a node as Sybil, the node is labeled Sybil. The probability of having a 100% false negatives i.e., every Sybil node to be classified as honest and the system being compromised, is reduced to $(fp_i)\ \hat{}\ K$, where fp_i is the ratio of the number of false positives returned by the protocol to the total number of nodes. The FETHR server could choose the K honest users via various methods. It could choose K nodes among the subnetwork that are best connected to the rest of the subnetwork. Alternatively, it could allow each node of the sub-network to choose its own K honest users, or choose the K best connected peers for each user, or just any K friends of a user in the SNG.

4.4 Improvement 3: Assigning partial trust to nodes

An alternative technique is to assign partial trust to both honest graph and the Sybil graph instead of a Boolean trust value to the graph which contains the honest node chosen at random. The trust factor of both the graphs would depend on the number of nodes in the graph. For example, if the sub-network has s nodes in the Sybil graph and h nodes in the honest graph, each node in the Sybil is given a partial weight of $s/(s+h)$ and each node in the honest graph is given a partial weight of $h/(s+h)$. This way, the SybilInfer protocol is no longer dependant on a single honest node chosen at random, as we observe in Chapter 5 that s is much smaller when compared to h. Thus classifying nodes as honest or Sybil after assigning a partially weighted trust to nodes in every iteration reduces the number of false positives (honest nodes missclasified as Sybil) and false negatives (Sybil nodes missclassified as honest).

4.5 SybilInfer on smaller networks

This version of SybilInfer is efficient only when the sub-network has sufficient number of members in this two-hop boundary, as the smaller sub-networks do not have enough information for Bayesian inference. For those sub-networks that do not satisfy that condition, we propose two techniques how SybilInfer can be practically employed for Sybil defense. One solution can be to increase the hop boundary from two hops to three or four hops as necessary. Another alternative is due to the unique feature in SybilInfer which not only classifies a user as honest or Sybil, but also assigns a percentage of trust to the user. If the percentage of trust is more than 50%, the user is considered as honest, other wise it is classified as Sybill. In this alternative, the FETHR server of smaller sub-networks could latch on to a FETHR server of a larger and trustable sub-network, that has at least one member in the two hop boundary of the the smaller FETHR server. The smaller FETHR server could piggy back over the trust factors of a larger Fethserver, where the smaller FETHR server would trust all users by a fraction P of the % of trust returned by the larger FETHR server's SybilInfer after many iterations. The fraction P is the percentage of trust assigned to the most trusted user in the larger FETHR server by the smaller FETHR server. However, some sub-networks might require both these schemes, that is, a "% of % trust" on a three or four hop boundary of the sub-network.

CHAPTER 5

RESULTS

In this Chapter, we verify that the proposed improvements in Chapter 4 improve the accuracy of SybilInfer in classifying nodes as honest or Sybil. First, we show that a Vanilla SybilInfer returns a variant number of false positives (honest nodes misclassified as Sybil) and false negatives (Sybil nodes missclassified as honest). Then we prove that changing the initial state of the sampler decreases the number of false postivies and false negatives considerably. Finally we discuss the effect of assigning a partial trust to nodes as discussed in 4.5, on both the scale-free model (Section 3.1) and the real network model (Section 3.2).

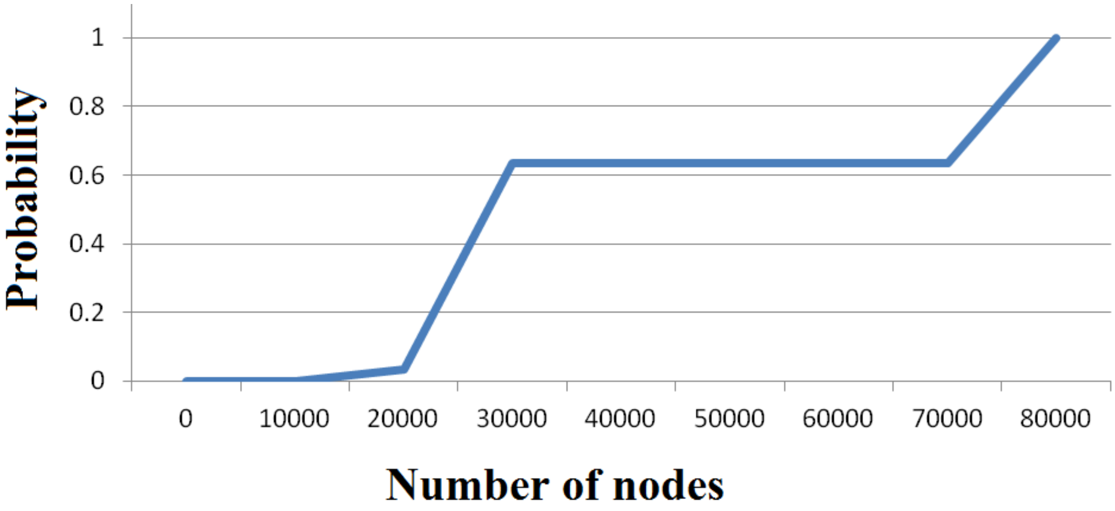

Figure 5.1. False positives (EDF) returned by Vanilla SybilInfer.

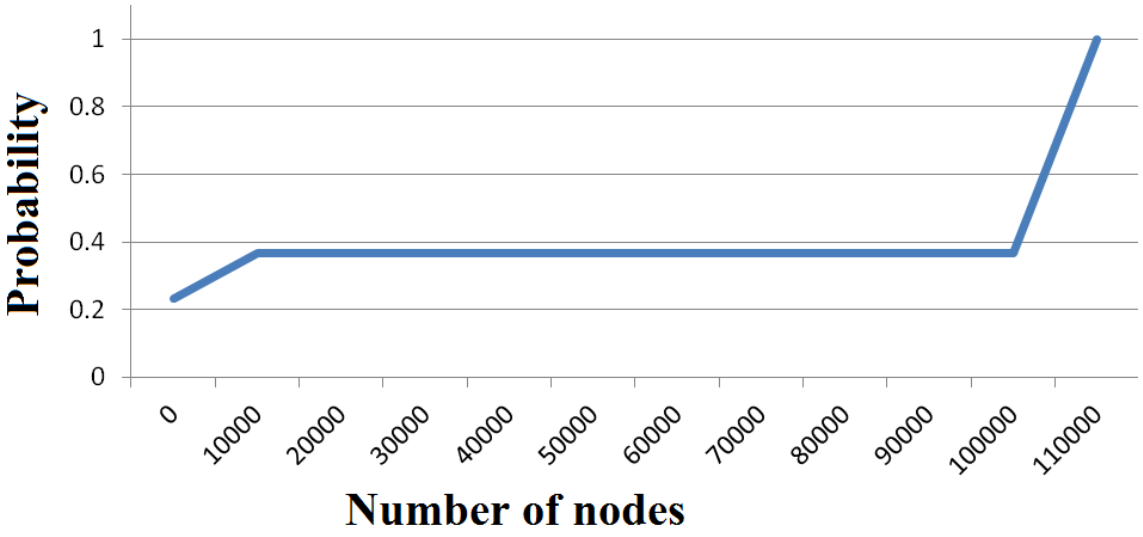

Figure 5.2. False negatives (EDF) returned by Vanilla SybilInfer.

Figures 5.1 and 5.2 illustrate the risk of implementing Vanilla SybilInfer, selecting a node at random in a sub-network from the scale-free model described in Chapter 3 to be an honest node which is also the initial state of the sampling algorithm used in SybilInfer to sample honest configurations out of the graph. The graph in Figure 5.1 is an emperical distribution function of the number of false positives returned by SybilInfer on a sub-network with the most users (74888) in it. The two hop boundary of the sub-network used for the simulations in this experiment contained 98,451 out of the total 100,100 users. The Figure 5.2 is an emperical distribution function plot of the false negatives returned by SybilInfer on the same sub-network. The figures show that Vanilla SybilInfer returns a variant number of false positives for such a network, as illustrated by the step function of the graphs indicating that an unpredictable number of false positives and false negatives are detected. This phenomenon is experienced when the randomly chosen honest node is better connected to the Sybil nodes than honest nodes. This over dependance on a single is improved upon by the proposals as described in Chapter 4.

After changing the initial state of the MCMC sampler, as discussed in Section 4.2, the average number of false positives returned by SybilInfer after thirty runs is approximately 350, where the total number of nodes in the two-hop boundary is 98451. This means that on an average, 350 (with a standard error of 4.41) honest users were wrongly classified as Sybil. The probability that the SybilInfer protocol uses one of the false positive nodes to distinguish between the honest sub-graph and the Sybil sub-graph in this sub-network is 350/98451. Since this probability is very low, this is an unlikely event. However, if this unlikely event occurs, all of the Sybil nodes are accepted as honest, resulting in almost a 100% false negatives. Thus as a result of using the entire sub-network as the initial state, the number of false positives and false negatives decreases considerably and number of false positives returned by the modified SybilInfer protocol is less variant as the samples chosen no longer depend on a single node chosen at random from the network.

Figures 5.3 and 5.4 show the result of assigning a partial trust to all the sub-networks. Figure 5.3 is a plot of the number of false postives and Figure 5.4 is a plot of the number of false negatives returned by SybilInfer [7] on the sub-networks. On the x-axis are the sub-networks arranged in the decreasing order of their size, and the y-axis shows the number of false postives (Figure 5.3) or the number of false negatives (Figure 5.4) returned by SybilInfer, along with the number of members in the sub-networks and the number of members in the two hop boundary of a subnetwork. As observed from the graphs, the number of false positives increases as the size of the sub-networks decreases, until it reaches a peak after which it oscillates unpredictably between maximum false positives and minimum. As the number of members in the

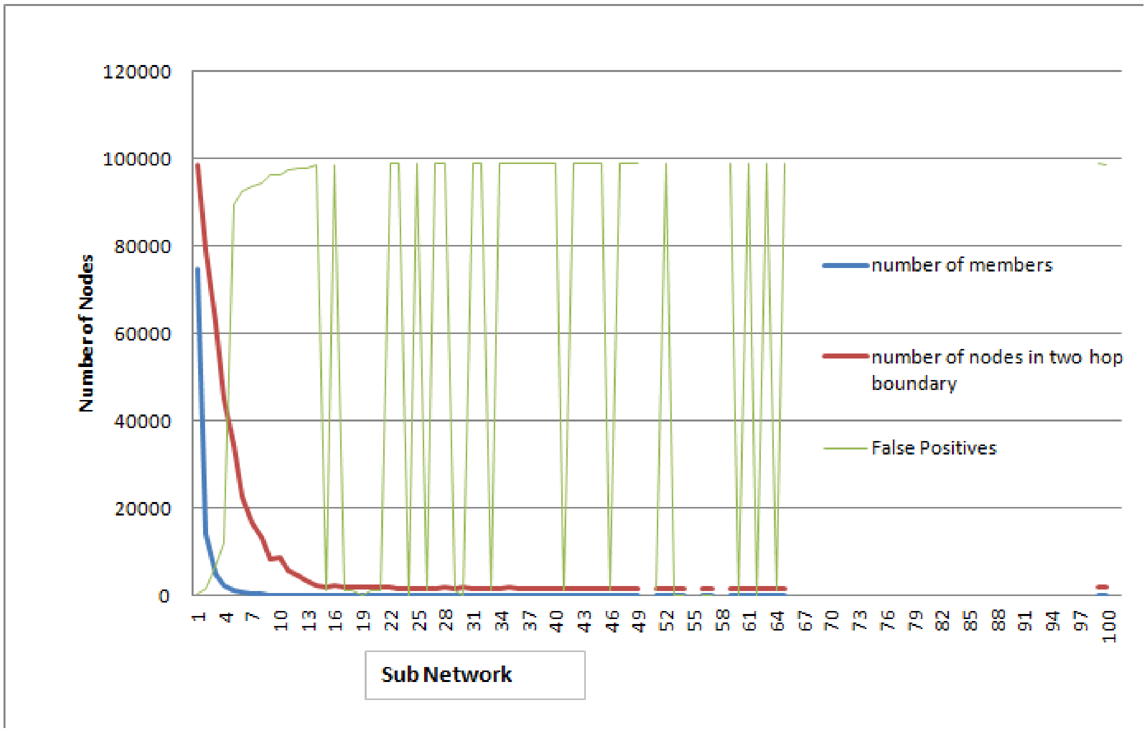

Figure 5.3. False positives after assigning partial weight.

two hop boundary decrease, the number of members in the two hop boundary are so less that there is a small chance of the Sybil identities to be connected with any member in the two-hop boundary. Such sub-networks are not under a Sybil attack and the number of false positives decreases to minimum. Otherwise, we observe that a maximum number of nodes are reported as false postive. Also, there is no possibility of the entire system being compromised, since the trust assigned to each node no longer depends on a single node or a set of nodes, but is an accumulated score of the partial weights a node accumulates over several iterations. The plots of Figure 5.3 and Figure 5.4 were plotted with partial weight gained by each node after five iterations of SybilInfer.

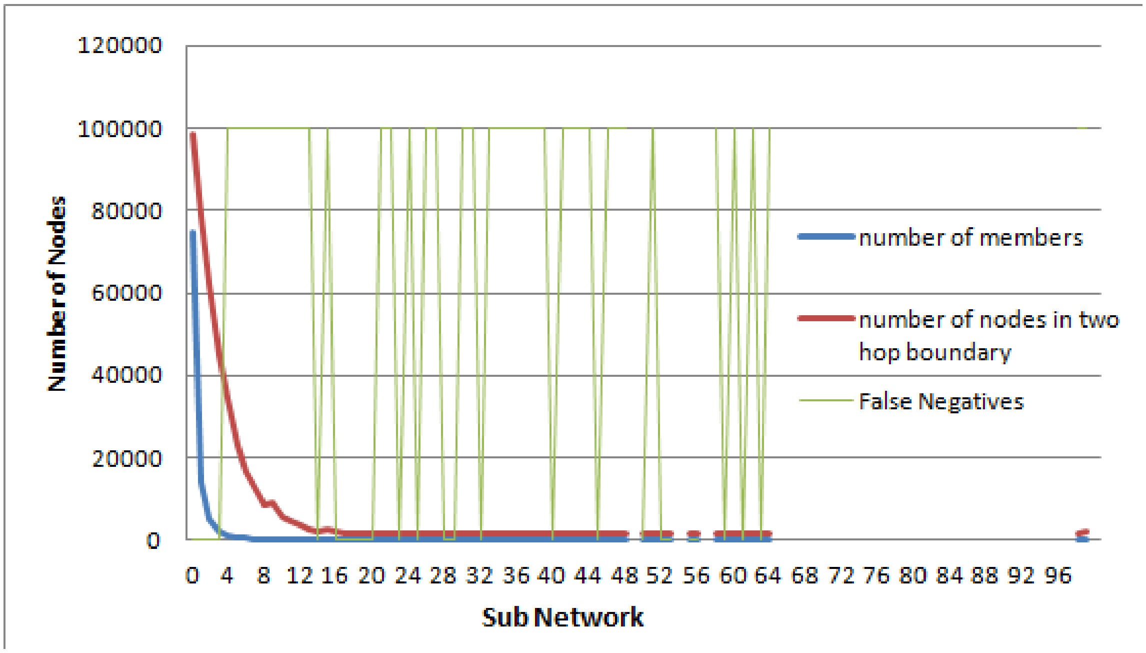

Figure 5.4. False negatives after assigning partial weight.

As seen from Figure 5.3, this verion of SybilInfer is efficient only when the sub-network has sufficient number of members in its two hop boundary. For the smaller networks, The number of false positives is reduced to 576 for the above sub-networks on piggy backing over the largest network as described in Section 4.5, and is reduced to 462 and 344 when piggy backing over two networks and three networks respectively, upon taking a proportional average of the trust factors of the bigger FETHR servers.

Figure 5.5 and Figure 5.6 show the number of false positives and false negatives returned by SybilInfer on the real network model as described in Section **??**. As seen in Figure 5.5, the number of false positives returned by SybilInfer on interoperable real world networks is inversely proportional to the number of members in the network, but comparable networks do not need to piggy back on each other. Figure 5.6 shows that

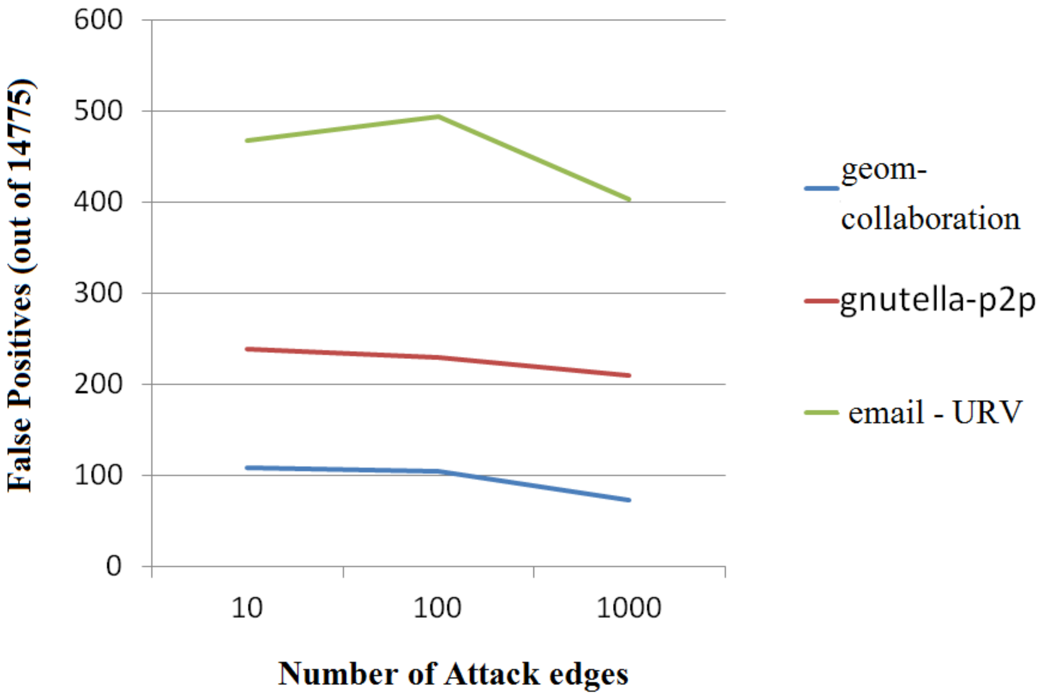

Figure 5.5. False postives on interoperating real world networks.

the number of false negatives returned is directly proportional to the number of attack edges used in the attack. As observed, the attacker gets nearly one false negative per attack edge, which means that for every attack edge, SybilInfer misclassifies only one Sybil node as honest.

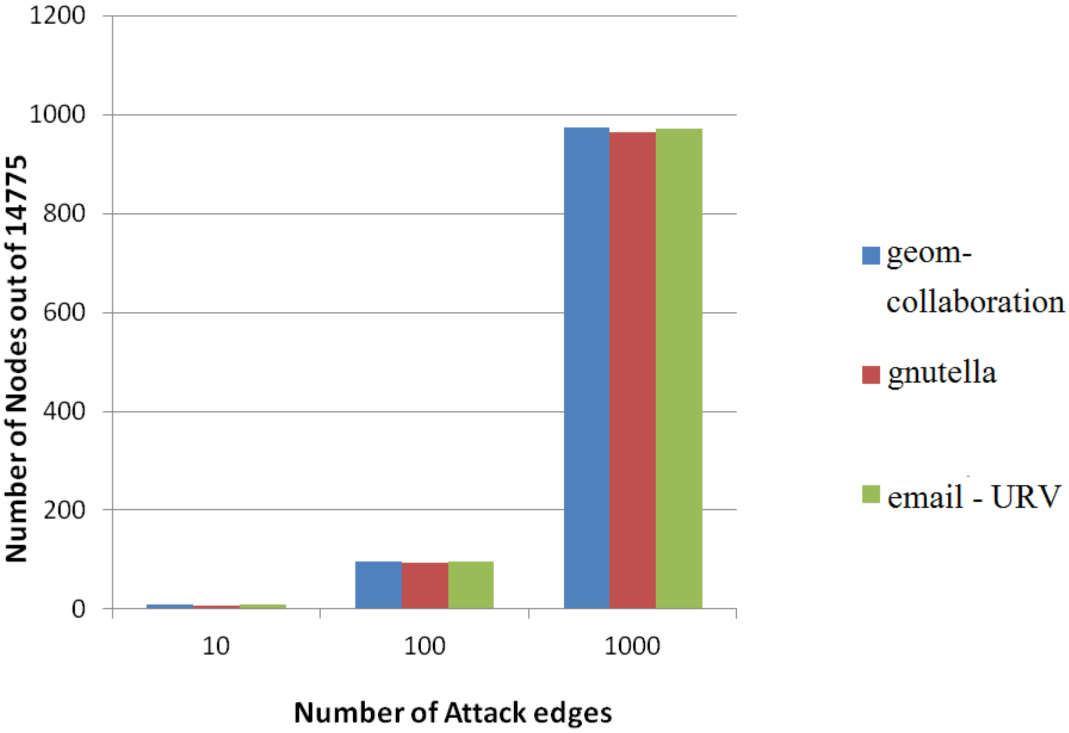

Figure 5.6. False negatives on interoperating real world networks.

CHAPTER 6

CONCLUSION AND FUTURE WORK

Sybil defenses like SybilInfer are critical to open micropublishing networks built over protocols like FETHR and Mr. Privacy to prevent spamming from Sybil identities. In this thesis we explored the application of SybilInfer for defense against a Sybil attack on distributed networks with partial graph information. We discovered that Vanilla SybilInfer's unpredictability in classifying nodes as Sybil or honest. We propose modifications to SybilInfer, including changing the initial state of the MCMC sampler in SybilInfer, using multiple honest nodes instead of jsut one and assigning partial trust to users instead of a boolean trust in every iteration. We showed that in a network model of interoperable micropublishing networks, initializing the MCMC sampler with the entire sub-network instead of a single honest node reduces the number of misclassified nodes. We also showed that assigning a weighted partial trust to nodes instead of a boolean trust in every interation reduces the chances of the entire system being compromised. We further propose piggy backing suggestions for networks which are smaller in size when compared to other networks, and verify the proposed improvements on real world comparable networks.

Having studied the accuracy of SybilInfer over interoperable social networks, it would be interesting to compare community detection algorithms with SybilInfer for defense against Sybil attacks in the distributed scenario. Another interesting research problem is using interaction graphs instead of relationship graphs among users for edges on SNGs. If the amount of interaction between two users is used as the partial graph information used for Bayesian inference in SybilInfer, The results seem intuitively more accurate. Whether or not that is true, remains to be validated. For smaller networks without sufficient graph information, larger hop boundaries need to be considered to observe the change in accuracy of SybilInfer's results.

REFERENCES

[1] V. Cosenza, *Social Media Statistics*, Aug. 2011. [Online]. Available: from http://www.vincos.it/social-media-statistics/

[2] H. Yu, M. Kaminsky, P. Gibbons, and A. Flaxman, *Sybilguard: Defending against sybil attacks via social networks.* ACM Special Interest Group on Data Communcation (SIGCOMM), Sept. 2006.

[3] M. Castro and B. Liskov, *Practical Byzantine Fault Tolerance.* Univ. of Massachussets Amherst: Proceedings of the Third Symposium on Operating Systems Design and Implementation, Feb. 1999.

[4] A. Cheng and E. Friedman, *Sybilproof Reputation Mechanisms.* Univ. of Massachussets Amherst: In ACM Wkshp on the Economics of Peer-to-Peer Systems, Aug. 2005.

[5] B. Levine, C. Shields, and N. Margolin, *A survey of solutions to the Sybil attack.* Univ. of Massachussets Amherst: Technical Report 2006-052, 2006.

[6] H. Yu, P. Gibbons, M. Kaminsky, and F. Xiao, *SybilLimit: A Near-Optimal Social Network Defense against Sybil Attacks.* IEEE Symposium on Security and Privacy (IEEE S & P), Sept. 2008.

[7] G. Danezis and P. Mittal, *Sybillinfer: Detecting Sybil nodes using social networks.* Network & Distributed System Security Symposium (NDSS), Feb. 2009.

[8] R. Albert and A. Barabasi, *Statistical mechanics of complex networks.* Review of Modern Physics, June 2002, vol. 74.

[9] C. Dwyer, S. Hiltz, and K. Passerini, *Trust and privacy concern within social networking sites: A comparison of Facebook and MySpace.* Keystone, CO: Proc. Americas Conference on Information Systems (AMCIS) 2007, Sept. 2007.

[10] B. Debatin, J. Lovejoy, A. Horn, and B. Hughes, *Facebook and online privacy: Attitudes, behaviors, and unintended consequences.* Journal of Computer-Mediated Communication, Oct. 2009, vol. 15.

[11] M. Motoyama, B. Meeder, K. Levchenko, G. Voelker, and S. Savage, *Measuring online service availability using Twitter.* WOSP '10, USENIX 3rd Workshop on Online Social Networks, 2010.

[12] S. Lafraniere, M. Wines, and E. Wong, *To liberalization of popular culture, China says enough*, Dec. 2011. [Online]. Available: from http://www.ndtv.com/article/world/to-liberalisation-of-popular-culture-china-says-enough-144499

[13] N. Ching, *In China, microblogging sites become free-speech platform*, Mar. 2011. [Online]. Available: from http://www.washingtonpost.com/world/in-china-microblogging-sites-become-free-speech-platform/2011/03/22/AFcsxlkB_story.html

[14] M. Fischer, T. Purtell, R. Chu, and M. Lam, *Mr. Privacy: Open and Federated Social Networking Using Email.* Hot Topics in Privacy Enhancing Techniques (PETS), 2011.

[15] D. Sandler and D. Wallach, *Birds of a FETHR:Open Decentralized Micropublishing.* 8^{th} Int. Wkshp. on Peer-to-Peer Systems (IPTPS '09), 2009.

[16] J. Douceur, *The Sybil attack.* Proc. Intl. Wkshp. on Peer-to-Peer Systems (IPTPS), Mar. 2002.

[17] B. Viswanath, A. Post, K. Gummadi, and A. Mislove, *An analysis of social network-based sybil defenses.* Univ. of Massachussets Amherst: Special Interest Group on Data Communcation (SIGCOMM), 2010.

[18] L. Detweiler, *The snakes of Medusa and cyberspace: Internet identity subversion.* Risks Digest: Forum on Risks to the Public in Computers and Related Systems, ACM Committee on Computers and Public Policy, Nov. 1993.

[19] A. Acquisti, R. Dingledine, and P. Syverson, *On the economics of anonymity.* Proc. Financial Cryptography (FC), Jan. 2003.

[20] B. Awerbuch and C. Scheideler, *Robust Distributed Name Service.* Proc. Intl Wkshp on Peer-to-Peer Systems (IPTPS)a, 2004.

[21] A. Bhargava, K. Kothapalli, C. Riley, C. Scheideler, and M. Thober, *Pagoda: a dynamic overlay network for routing, data management, and multicasting.* Proc. ACM Symp on Parallel Algorithms (ACM-SPAA), 2004.

[22] D. Cvrcek and V. Matyas, *On the role of contextual information for privacy attacks and classification.* Proc. Wkshp. on Privacy and Security Aspects of Data Mining, Nov. 2004.

[23] R. Dingledine, V. Shmatikov, , and P. Syverson, *Synchronous batching: From cascades to free routes.* Proc. Privacy Enhancing Technologies workshop (PET), May 2004, vol. 3424.

[24] D. Figueiredo, P. Nain, and D. Towsley, *On the analysis of the predecessor attack on anonymity systems.* University of Massachusetts: Computer Science Technical Report 04-65, July 2004.

[25] A. Adya, W. Bolosky, M. Castro, R. Chaiken, G. Cermak, J. Douceur, J. Howell, J. Lorch, M. Theimer, and R. Wattenhofer, *Farsite: Federated, available, and reliable storage for an incompletely trusted environment.* Proc. USENIX Symposium on Operating Systems Design and Implementation (OSDI), Dec. 2002.

[26] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. Wallach, *Secure routing for structured peer-to-peer overlay networks.* Proc. USENIX Symposium on Operating Systems Design and Implementation (OSDI), Dec. 2002.

[27] M. Castro, P. Druschel, A. Kermarrec, and A. Rowstron, *One ring to rule them all: Service discovery and binding in structured peer-to-peer overlay networks.* Proc. Special Interest Group on Operatign Systems (SIGOPS) European Wkshp, 2002.

[28] B. Dragovic, E. Kotsovinos, S. Hand, and P. Pietzuch, *Xenotrust: Event-based distributed trust management.* Proc. Intl. Wkshp. on Database and Expert Systems Applications, 2003.

[29] B. Dutertre, S. Cheung, and J. Levy, *Lightweight key management in wireless sensor networks by leveraging initial trust.* Technical Report SRI-SDL-04-02, SRI International, 2002.

[30] J. Aspnes, C. Jackson, and A. Krishnamurthy, *Exposing computationally challenged Byzantine impostors.* Yale University Department of Computer Science: Technical Report YALEU/DCS/TR-1332, July 2005.

[31] F. Cornelli, E. Damiani, and S. Samarati, *Implementing a reputation-aware gnutella servent.* Proc. Intl. Wkshp. on Peer-to-Peer Computing, 2002.

[32] M. Freedman and R. Tarzan, *A peer-to-peer anonymizing network layer.* Proc. ACM Conference on Computer and Communications Security (CCS), Nov. 2002.

[33] S. Goel, M. Robson, M. Polte, and E. Sirer, *Herbivore: A scalable and efficient protocol for anonymous communication.* Cornell University: Technical Report 2003-1890, Feb. 2003.

[34] B. Awerbuch and C. Scheideler, *Group Spreading: A protocol for provably secure distributed name service.* Proc. Automata, Languages and Programming (ICALP), 2004.

[35] R. Gatti, S. Lewis, A. Ozment, T. Rayna, and A. Serjantov, *Sufficiently secure peer-to-peer networks.* Wkshp. on the Economics of Information Security, May 2004.

[36] P. Maniatis, D. Rosenthal, M. Roussopoulos, M. Baker, T. Giuli, and Y. Muliadi, *Preserving peer replicas by rate-limited sampled voting.* Proc. ACM Symposium on Operating Systems Principles (SOSP), 2003.

[37] P. Maniatis, M. Roussopoulos, T. Giuli, D. Rosenthal, and M. Baker, *The lockss peer-to-peer digital preservation system.* ACM Transactions on Computation Systems, 2005.

[38] J. Newsome, E. Shi, D. Song, and A. Perrig, *The Sybil attack in sensor networks: analysis & defenses.* Proc. Intl. Symp on Information Processing in Sensor Networks (IPSN), 2004.

[39] R. Rodrigues, B. Liskov, and L. Shrira, *The design of a robust peer-to-peer system.* Proc. ACM Special Interest Groups on Operating Systems (SIGOPS) European Wkshp, Sept. 2002.

[40] A. Cheng and E. Friedman, *Sybilproof Reputation Mechanisms.* Proc. ACM Wkshp on Economics of Peer-to-Peer Systems, 2005.

[41] Mohaisen, A. Yun, and Y. Kim, *Measuring the mixing time of social graphs.* In IMC, 2010.

[42] S. Chib and E. Greenberg, *Understanding the Metropolis Hastings algorithm.* Am. Stat.49:, 1995.

[43] S. Fortunato, *Community detection in graphs.* Physics Reports, 486:75, 2010.

[44] R. Acquisti and R. Gross, *Imagined communities: Awareness, information sharing, and privacy on the Facebook.* In 6^{th} Workshop on Privacy Enhancing Technologies (PETS), 2006.

[45] B. Krishnamurthy, P. Gill, and M. Arlitt, *A few chirps aabout Twitter.* New York, NY: WOSP '08:Proceedings of the first workshop on Online social networks, ACM, 2008.

[46] V. Batagelj and A. Mrvar, *Pajek datasets*, 2006.

[47] J. Leskovec, J. Kleinberg, and C. Faloutsos, *Graph Evolution: Densification and shrinking diameters.* ACM Transactions on Knowledge Discovery from Data (ACM TKDD), 1(1),, 2007.

[48] M. Ripeanu, I. Foster, and A. Iamnitchi, *Mapping the Gnutella Network: Properties of large-scale peer-to-peer systems and implications for system design.* IEEE Internet Computing Journal, 2002.

[49] R. Guimera, L. Danon, A. Diaz-Guilera, F. Giralt, and A. Arenas, *Mapping the Gnutella Network: Properties of large-scale peer-to-peer Systems and implications for system design.* Physical Review E , vol. 68, 065103(R), 2003.

BIOGRAPHICAL STATEMENT

Vritant Naresh Jain was born in Shimoga, Karnataka, India in May 1988. Having spent the initial two years of his life in Bangalore in the sate of Karnataka, brought up in the city of Hyderabad in the state of Andhra Pradesh, where his family migrated from the state of Rajasthan, he belongs to a multi-cultural background. A logic enthusiast, he progressed naturally to a career in Computer Science having completed his Masters Degree from University of Texas at Arlington in the Fall of 2011. His research interests include Security and Artificial Intelligence, and indulges in a wide variety of hobbies including many sports, multilingual poetry, theatre, and a recent addition to this list is strumming old Indian classic tunes on his guitar. He is passionate about studying human behaviour and design patterns, and works best only under pressure.