An Analysis of Social Network-Based Sybil Defenses

Bimal Viswanath
MPI-SWS
bviswana@mpi-sws.org

Ansley Post
MPI-SWS
abpost@mpi-sws.org

Krishna P. Gummadi
MPI-SWS
gummadi@mpi-sws.org

Alan Mislove
Northeastern University
amislove@ccs.neu.edu

ABSTRACT

Recently, there has been much excitement in the research community over using social networks to mitigate multiple identity, or Sybil, attacks. A number of schemes have been proposed, but they differ greatly in the algorithms they use and in the networks upon which they are evaluated. As a result, the research community lacks a clear understanding of how these schemes compare against each other, how well they would work on real-world social networks with different structural properties, or whether there exist other (potentially better) ways of Sybil defense.

In this paper, we show that, despite their considerable differences, existing Sybil defense schemes work by detecting *local communities* (i.e., clusters of nodes more tightly knit than the rest of the graph) around a trusted node. Our finding has important implications for both existing and future designs of Sybil defense schemes. First, we show that there is an opportunity to leverage the substantial amount of prior work on general community detection algorithms in order to defend against Sybils. Second, our analysis reveals the fundamental limits of current social network-based Sybil defenses: We demonstrate that networks with well-defined community structure are inherently more vulnerable to Sybil attacks, and that, in such networks, Sybils can carefully target their links in order make their attacks more effective.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Design studies; C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*

General Terms

Security, Design, Algorithms, Experimentation

Keywords

Sybil attacks, social networks, social network-based Sybil defense, communities

SIGCOMM'10, August 30–September 3, 2010, New Delhi, India.

1. INTRODUCTION

Avoiding multiple identity, or Sybil, attacks is known to be a fundamental problem in the design of distributed systems [8]. Malicious attackers can create multiple identities and influence the working of systems that rely upon open membership. Examples of such systems range from communication systems like email and instant messaging to collaborative content rating, recommendation, and delivery systems such as Digg and BitTorrent. Traditional defenses against Sybil attacks rely on trusted identities provided by a certification authority. But requiring users to present trusted identities runs counter to the open membership that underlies the success of these distributed systems in the first place.

Recently, there has been excitement in the research community about applying social networks to mitigate Sybil attacks. A number of schemes have been proposed that attempt to defend against Sybils in a social network by using properties of the social network's structure [7,29,32,33]. Unlike traditional solutions, these schemes require no central trusted identities, and instead rely on the trust that is embodied in existing social relationships between users.

All social network-based Sybil defense schemes make the assumption that, although an attacker can create arbitrary Sybil identities in social networks, he or she cannot establish an arbitrarily large number of social connections to non-Sybil nodes. As a result, Sybil nodes tend to be poorly connected to the rest of the network, compared to the non-Sybil nodes. Sybil defense schemes leverage this observation to identify Sybils. They use various graph analysis techniques to search for topological features resulting from the limited capacity of Sybils to establish social links.

Our focus in this paper is on the graph analysis algorithms behind the schemes. The literature on Sybil defense schemes is still in its early stages; most papers describe new algorithms, but none provide a common insight that explains how all of these schemes are able to detect Sybils. Each algorithm has been shown to work well under its own assumptions about the structure of the social network and the links connecting non-Sybil and Sybil nodes. However, it is unclear how these algorithms would compare against each other, on more general topologies, or under different attack strategies. As a result, it is not known if there exist other (potentially better) ways to mitigate Sybil attacks or if there are fundamental limits to using only the structure of the social network to defend against Sybils.

In this paper, we take a first, but important, step towards answering these questions. We decompose existing Sybil defense schemes and demonstrate that at their core, the var-

ious algorithms work by implicitly *ranking* nodes based on how well the nodes are connected to a trusted node. Nodes that have better connectivity to the trusted node are ranked higher and are deemed to be more trustworthy. We show that, despite their considerable differences, all Sybil defense schemes rank nodes similarly—nodes within *local communities* (i.e., clusters of nodes more tightly knit than the rest of the network) around the trusted node are ranked higher than nodes in the rest of the network. Thus, Sybil defense schemes work by effectively detecting local communities.

The above insight has important implications for both existing and future designs of social network-based Sybil defense schemes. First, it motivates us to investigate whether a class of algorithms, known as *community detection* algorithms [10], that attempt to find such clusters of nodes directly, could be used for Sybil defense. We find that it is possible to use off-the-shelf community detection algorithms to find Sybils. Unlike Sybil defense, community detection is a well-studied and mature field, implying that our findings open the door for researchers to exploit a variety of techniques from a rich body of community detection literature.

Second, our insight also hints at the limitations of relying on communities for finding Sybils. For Sybil defense schemes to work well, all non-Sybil nodes need to form a single community that is distinguishable from the group of Sybil nodes.[1] In reality, however, users in many social networks form multiple communities that are interconnected rather sparsely. We show that, in these networks, it is hard for a trusted node to distinguish Sybils from non-Sybils outside its local community. Further, we demonstrate how Sybils can launch extremely effective attacks by establishing just a small number of links to carefully targeted nodes within such networks. As systems are beginning to be built on top of Sybil defense schemes [17, 18, 27], our findings question the wisdom of building these systems without a thorough understanding of the limitations of Sybil defense.

2. UNDERSTANDING SYBIL DEFENSE

As noted before, a variety of Sybil defense schemes have been proposed, but each has been evaluated using different social networks and attack strategies by the Sybils. Therefore, it is not well understood how these different schemes compare against each other, or how a potential user of these schemes, such as a real-world social networking site, would select one scheme over another.

2.1 The core of Sybil defense schemes

Given the problem of comparing competing Sybil defense schemes, one approach would be to view the schemes as complete coherent proposals (i.e., treat them as black boxes, and compare them in real-world settings). Such an approach is straight-forward and would provide useful performance comparisons between a *fixed* configuration of schemes over a *given* set of social networks and attack strategies by the Sybils. However, it would not yield conclusive information on how a particular scheme would perform if either the given social network or the behavior of the attacker should change. It also does not allow us to derive any fundamental insights into how these schemes work, which might enable us to build upon and improve them.

An alternative approach is to find a core insight common to all the schemes that would explain their performance in *any* setting. Gaining such a fundamental insight, while difficult, not only provides guidance on improving future designs, but also sheds light on the limits of social network-based Sybil defense. However, we cannot gain such an insight by treating each of these schemes as a black box, with each carrying its own set of algorithms, optimizations, and assumptions. Instead, we need to reduce the schemes to their core task before analyzing them.

At a high level, all existing schemes attempt to isolate Sybils embedded within a social network topology. Every scheme declares nodes in the network as either Sybils or non-Sybils from the perspective of a *trusted node*, effectively partitioning the nodes in the social network into two distinct regions (non-Sybils and Sybils). Hence, each Sybil defense scheme can actually be viewed as a *graph partitioning algorithm*, where the graph is the social network. However, the quality and performance of the algorithm depends on the inputs, namely, the network topology and the trusted node.

Most Sybil defense schemes include a number of useful and practical optimizations that enhance their performance in specific application scenarios. For example, SybilGuard [33] and SybilLimit [32] have a number of design features that facilitate their use in decentralized systems. Similarly, SumUp [29] has optimizations specific to online content voting systems. However, because our goal is to uncover the core graph partitioning algorithm, we study these schemes independent of the assumptions about their application environments as well as the optimizations that are specific to those environments. Later in the paper, we show that this approach not only offers hints for the designers of future Sybil defense schemes, but also helps us understand the characteristics of real-world social networks that make them vulnerable to Sybil attacks.

2.2 Converting partitions to rankings

Even when viewing the schemes as graph partitioning algorithms, comparing the different Sybil defense schemes is not entirely straightforward. The output of each scheme depends on the setting of numerous parameters. At a high level, these parameters can be seen as making the partitioning between Sybils and non-Sybils either more restrictive or permissive, thereby trading false positives for false negatives. While the designers of the schemes offer rough guidelines

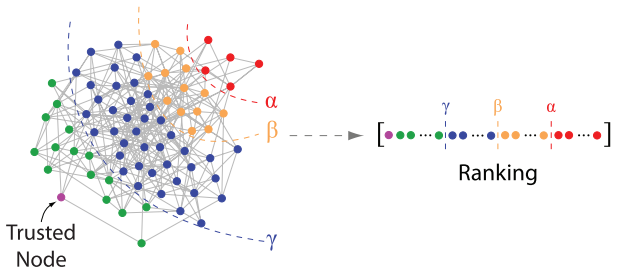


Figure 1: Diagram of converting partitionings into a ranking of nodes. Different parameter settings (α, β, γ) cause increasingly large partitions to be marked as Sybils, thereby inducing a ranking.

[1]Many Sybil defense schemes impose this requirement implicitly by assuming that the non-Sybil region of the network is *fast mixing* [22], meaning a random walk of length $O(\log N)$ reaches a stationary distribution of nodes.

	Assumptions	Algorithm	Ranking	Cutoff	Evaluation
SybilGuard [33]	Non-Sybil region is fast mixing [22]	Random walk performed by each node	Varying random walk length	Whether or not walk intersection occurs	Kleinberg network [12]
SybilLimit [32]	Non-Sybil region is fast mixing	Multiple random walks performed by each node	Varying number of random walks and walk length	Whether or not tails of random walks intersect	Friendster, LiveJournal, DBLP, Kleinberg
SybilInfer [7]	Non-Sybil region is fast mixing, modified walks are fast mixing	Bayesian inference on the results of the random walks	Probability of node being non-Sybil from Bayesian inference	Threshold on the probability that a given node is non-Sybil	Power-law network [24], LiveJournal
SumUp [29]	Non-Sybil region is fast mixing, no small cut between collector and non-Sybil region	Creation of voting envelope with appropriate link capacities around collector	Varying the size of the voting envelope	Whether or not nodes are within the voting envelope	YouTube, Flickr, Digg

Table 1: Overview of the properties and evaluation of social network-based Sybil defense schemes.

for choosing the parameter values (e.g., set a parameter to $O(\log N)$ where N is the number of network nodes), there can be considerable variation in the output from different parameter settings that follow the guidelines. Given the difficulty in selecting the right parameter settings, we would like to compare the schemes independent of the choice of their respective parameters.

We studied the impact of changing parameters on the output of the Sybil and non-Sybil partitions. We observed that as the Sybil partition grows or shrinks in response to parameter changes, an ordering can be imposed on the nodes added or removed.[2] That is, when the Sybil partition grows larger, new nodes are added to the partition without removing nodes previously classified as Sybils. Similarly, when the Sybil partition grows smaller, some nodes are removed from the partition without adding any nodes previously classified as non-Sybils. Figure 1 illustrates how different partitionings obtained by changing parameters can be converted into an ordering or ranking of nodes.

Our observation suggests that one can view the Sybil defense schemes as implicitly ordering or *ranking* nodes in the network, while the parameter settings determine where the boundary between the partitions, called the *cutoff point*, lies. Changing the parameters slides the cutoff point along the ranking, but the resulting partitions uphold the observed ranking of nodes. Thus, we can compare the different schemes independently of their parameters by simply comparing their relative rankings of the nodes.

2.3 Reduction of existing schemes

We reduce each Sybil defense scheme into its component processes using the model presented in Figure 2. At its core, each scheme contains an algorithm, which, given a trusted node and a network, produces a ranking of the nodes in the network relative to the trusted node. Then, depending on the setting of various parameter values, the scheme creates a cutoff, which is applied to the ranking and produces a Sybil/non-Sybil partitioning.

The schemes that we examine in this paper are SybilGuard [33], SybilLimit [32], SybilInfer [7], and SumUp [29]. For each of these Sybil defense schemes, Table 1 identifies the partitioning algorithm, how this partitioning induces a ranking of nodes, and how the algorithm parameters determine a cutoff. We also describe the assumptions the schemes make about their input environment (i.e., the structure of non-Sybil and Sybil topologies), and briefly describe the networks that these schemes were evaluated upon. A more detailed description of how these schemes map into our model is included in the Appendices.

Although we only show how our model applies to four well-known schemes, we believe that it could be applied to other schemes as well. For example, a recent work proposes a Sybil-resilient distributed hash table routing protocol [17, 18], by using social connections between users to build routing tables. The protocol relies on random walks much in the same manner as SybilGuard and SybilLimit, so we believe our analysis would apply to it as well. Similarly, Quercia et al. [27] recently proposed a Sybil defense scheme that relies on a graph-theoretic metric called betweenness centrality to calculate the likelihood of a node being a Sybil. To apply our analysis, the centrality measure can be used directly to induce a ranking of the nodes.

2.4 Rest of the paper

In this section, we have shown that existing Sybil defense schemes all work by inducing an implicit ranking of the nodes. We now take a closer look at these rankings, us-

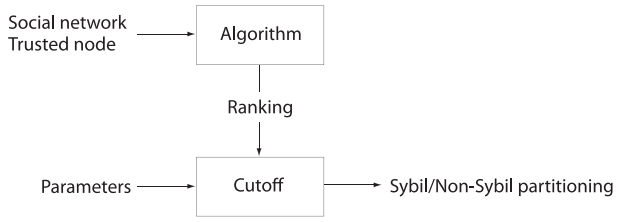


Figure 2: Diagram showing the processes involved in a Sybil defense scheme. In brief, the scheme itself can be split into an algorithm, which when given a social network and a trusted node, produces a ranking. The parameters to the scheme are used to create a cutoff, which defines a Sybil/non-Sybil partitioning from the ranking.

[2]While we do not formally prove that all parameters of any Sybil defense scheme must induce an ordering, it is the case for all schemes, environments, and parameters we analyzed.

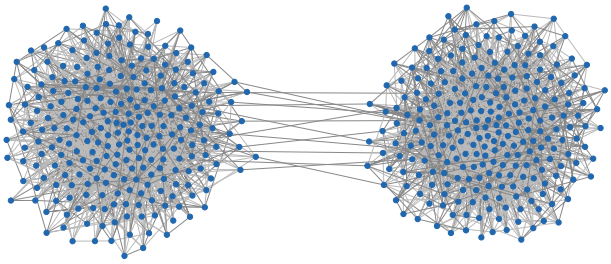


Figure 3: The synthetic network used in Section 3.1 for exploring the rankings. Each of the two communities contains 256 nodes.

ing them to compare the schemes across a wide range of conditions. Our goal in the remaining sections is to better understand the ranking algorithms underlying existing Sybil defense schemes, and through this understanding, to provide a basis for answering the following questions:

- Are the different Sybil defense schemes performing the core task of ranking nodes in the same way, or is each ranking unique? (Section 3)
- Are there other (potentially better) ways to obtain these node rankings? (Section 4)
- What structural properties of the social network determine how well the schemes work? (Section 5.1)
- Are the schemes robust against the different possible Sybil attack strategies? (Section 5.2)

3. RANKINGS AND SYBIL DEFENSE

In this section, we develop a better understanding of the process by which Sybil defense schemes compute node rankings by comparing the rankings of the different schemes.

3.1 Rankings in synthetic networks

We start by examining the node rankings generated by the schemes when run over a synthetic network topology, taken from [3] and shown in Figure 3. In brief, this network is constructed using the Bárabási-Albert preferential attachment model [4], and then rewired[3] to have two densely connected communities of 256 nodes each, connected by a small number of edges.

3.1.1 Comparing node rankings

We randomly selected a node in one of the communities as the trusted node and calculated the node rankings on this synthetic network for the four Sybil defense schemes previously discussed. We then examined how closely the various rankings matched. To compare the rankings, we use mutual information [28], which measures the similarity of two partitionings of a set. In brief, mutual information ranges between 0 and 1, where 0 represents no correlation between the partitionings, and 1 represents a perfect match.

[3]In brief, the rewiring works as follows: Nodes are first randomly assigned to two communities. Then, rewiring works by selecting two links $A \leftrightarrow B$ and $C \leftrightarrow D$ where A and C are in the same community and B and D are in the same community. These two links are replaced with the links $A \leftrightarrow C$ and $B \leftrightarrow D$, thereby increasing the intra-community links without changing the degree distribution or link count.

The results of this experiment are shown in the top graph of Figure 4. For clarity, we only show the mutual information between partitionings of SybilGuard and each of the other three schemes (the other pairs are similar). The x-axis denotes the size of the partition containing non-Sybils. For example, the x-axis value of 10 divides the ranking into two parts, one with the first 10 nodes in the ranking (marked as non-Sybils) and the other with the rest of the nodes (marked as Sybils). Thus, Figure 4 shows the mutual information between pairs of rankings at all possible cutoff points.

Figure 4 shows that the mutual information metric is maximized at a partitioning of size 256. Interestingly, it falls off sharply before and after this cutoff value. To understand this plot better, we investigated the strong correlation between the different node rankings at the partitioning size of 256 and found that the 256 members that each scheme assigned to the non-Sybil partition strongly corresponded to the half of the network in Figure 4 that contained the trusted node. This indicates that all schemes are biased towards ranking nodes in the local community around the trusted node higher than nodes outside of the community. However, there is little correlation between the ordering of nodes within the community, or the nodes outside of it, as the mutual information is low between pairs of rankings before and after this point.

3.1.2 The common factor behind the rankings

One hypothesis that could explain our above observations is that the nodes are being ranked such that nodes well connected to the trusted node are more likely to be higher in the rankings. Since there are several nodes within the local community of the trusted node that are equally well connected, the ranking amongst these nodes is not strictly enforced, i.e., the different schemes rank these nodes differently. Similarly, several nodes outside the local community are equally

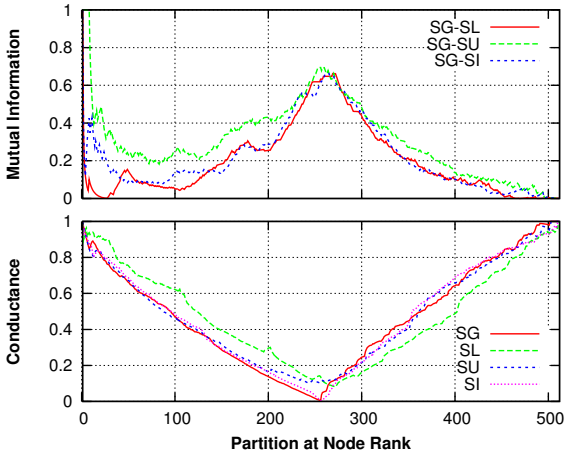


Figure 4: Mutual information between pairs of rankings and conductance of each ranking plotted for various partitions for the synthetic network, using schemes SybilGuard (SG), SybilLimit (SL), SumUp (SU), and SybilInfer (SI). A strong correlation is observed at 256 nodes, indicating a high degree of overlap between the partitionings, and a strong community structure in the non-Sybils, at this point.

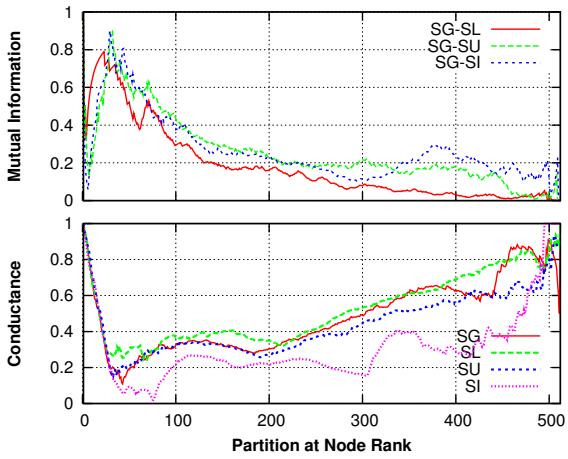


Figure 5: Mutual information between pairs of rankings and conductance of each ranking plotted for various partitions of the four schemes when run on the Facebook network.

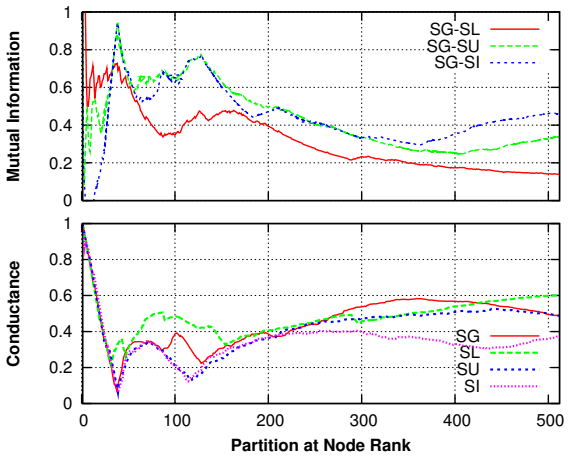


Figure 6: Mutual information between pairs of rankings and conductance of each ranking plotted for various partitions of the four schemes when run on the Astrophysics network.

poorly connected and so their relative ranking is not consistent across the different Sybil schemes. However, there is a sharp distinction between the connectivity of nodes inside and outside the local community, and so the former are ranked before the latter.

To confirm this hypothesis, we used a well known metric called *conductance* [16] for determining how closely a subset of nodes within a network are connected among themselves relative to the rest of the network. Conductance is a widely used metric for evaluating the quality of communities within large networks. In brief, the conductance of a set of nodes ranges between 0 and 1, with lower numbers indicating stronger communities.

We plot the conductance of the non-Sybil subset in the bottom of Figure 4 and notice that there is a sharp inflection point in the conductance at 256 nodes for all schemes. This corresponds to the boundary between the two communities in our synthetic network topology. Adding nodes from another community sharply increases the conductance, so all schemes assign higher rankings to nodes from within the community around the trusted node than to nodes from outside the community. This helps explain why the partitions obtained from the rankings match very well when the cutoff is set at the inflection point.

3.2 Rankings in real-world networks

In this section, we verify that the results we found for our synthetic network also hold in real-world networks. First, we wish to check that nodes are ranked in a biased manner, such that nodes from the trusted node's local community rank higher than any other nodes. Second, we wish to test if the point at which all Sybil defense schemes agree corresponds to a trough in the conductance value, indicating the boundary of the community around the trusted node.

To do show this, we repeat the experiment above for two real world networks: *Facebook*, consisting of the social network between Rice University graduate students taken from Facebook [21], and *Astrophysics*, consisting of the co-authorship network between astrophysicists [25]. Details on these datasets are provided in Table 2.

As we can see in Figures 5 and 6, the mutual information reveals a local cutoff where all rankings have strong correlation, and this cutoff is also characterized by a low conductance value. Taken together, our experiments show that all Sybil defense schemes are identifying a local community that surrounds the trusted node, but that the ranking of nodes they use to reach the local community (and that they use after this point) is not strongly correlated.

3.3 Summary of observations

We now summarize the findings from our comparison of the way in which various algorithms rank nodes:

- The ranking of nodes is biased towards those which decrease conductance. Thus, nodes that are tightly connected around a trusted node (i.e., those that form subsets with lower conductance) are more likely to be ranked higher.
- When there are multiple nodes that are similarly well connected to the trusted node (i.e., they form subsets with similar conductance) they are often ordered differently in different algorithms.
- When the trusted node is located in a densely connected community of nodes, with a clear boundary between this community and the rest of the network, the nodes in the local community around the trusted node are ranked before others.

4. APPLYING COMMUNITY DETECTION

In the previous section, we observed that all Sybil defense schemes work by identifying nodes in the local community around a given trusted node and ranking them as more trustworthy than those outside. In this section, we examine whether algorithms that are explicitly designed to detect communities, called *community detection* algorithms [2, 3, 6, 19], can be used for Sybil defense in the same manner as existing schemes. Our goal is to investigate the

potential for leveraging existing literature in community detection to defend against Sybils. To this end, we first select an off-the-shelf community detection algorithm and generate a node ranking from the algorithm. We then compare its node ranking with those of existing Sybil defense schemes, to determine if it is able to defend against Sybils with similar accuracy.

4.1 Community detection

Community detection in networks is a well studied and mature field. There are numerous approaches that use different mechanisms in order to detect communities and different metrics to evaluate the quality of communities. Below, we give a brief overview of how community detection schemes work.

In this paper, we focus on *local* community detection schemes [3], which do not require a global view of the network.[4] Most of the local approaches work by starting with one (or more [2]) seed nodes and greedily adding neighboring nodes until a sufficiently strong community is found. For example, Mislove's algorithm [21] iteratively adds nodes that improve the the normalized conductance (a metric closely related to conductance) at each step, and stops when the conductance metric reaches an inflection point. For a detailed survey of local community detection algorithms, we refer the reader to the recent survey paper by Fortunato [10], which discusses numerous algorithms for community detection.

As there is a large body of work on community detection, we could theoretically utilize any of these algorithms as the ranking algorithm. For the evaluation presented in this section, we selected Mislove's algorithm [21], but with the conductance metric from Section 3.1.2. We chose this algorithm as it is conceptually easy to understand, since it greedily minimizes conductance. However, our decision is not fundamental, and there may be other algorithms that perform better (especially since different community detection algorithms have been shown to perform better on different networks [15]). Rather, our goal here is simply to investigate how well off-the-shelf community detection algorithms are able to find Sybils.

In order to use community detection to find Sybils, we need to generate a node ranking in the same manner as the other schemes. To do so, we run Mislove's community detection algorithm and record the node that it iteratively adds at each step to minimize conductance. Note that we modify the algorithm to not stop once a local trough is found; instead we allow it to continue running until all of the nodes have been added. This results in a node ranking that we can use to compare against the other schemes.

4.2 Evaluating Sybil detection

We now evaluate the community detection algorithm against our existing Sybil defense schemes. When comparing against each of the Sybil defense schemes, we used experimental settings similar to those described in the paper in which the scheme was proposed. This required us to split our evaluation results in two separate sections; one for SybilGuard, SybilLimit, and SybilInfer and another for SumUp. The split is necessary because SumUp was originally evaluated for its ability to limit the number of votes Sybil identities can place, and not for its ability to accurately detect Sybil nodes. Thus, the experimental settings for evaluating SumUp are quite different from those of the other schemes, necessitating a separate evaluation.

[4]Our decision to focus on local community detection algorithms, as opposed to global ones, is due to the fact that they work in a similar manner as existing Sybil defense schemes by not assuming a global view. However, it has been shown that different global community detection algorithms have many of the same properties as local ones [15], indicating that our results would likely hold for global algorithms as well. We leave this to future work.

Network	Nodes	Links	Avg. degree
YouTube [20]	446,181	1,728,938	7.7
Astrophysicists [25]	14,845	119,652	16
Advogato [1]	5,264	43,027	16
Facebook [21]	514	3,313	13

Table 2: Statistics of datasets used in our evaluation.

A summary of the data sets that we use in the evaluation is shown in Table 2. In addition to the datasets from the previous section, we examine *YouTube*, consisting of the social network of users in YouTube [20], and *Advogato*, consisting of the trust network between free software developers [1].

4.2.1 Measuring Sybil detection accuracy

In order to measure the accuracy of the various schemes at identifying Sybils, we need a way to compute how often a scheme ranks Sybil nodes towards the bottom of the ranking. To do so, we use the metric *Area under the Receiver Operating Characteristic (ROC) curve* or A'. In brief, this metric represents the probability that a Sybil defense scheme ranks a randomly selected Sybil node lower than a randomly selected non-Sybil node [9]. Therefore, the A' metric takes on values between 0 and 1: A value of 0.5 represents a random ranking, with higher values indicating a better ranking and 1 representing a perfect non-Sybil/Sybil ranking. Values below 0.5 indicate an inverse ranking, or one where Sybils tend to be ranked higher than non-Sybils. A very useful property of this metric is that it is defined independent of the number of Sybil and non-Sybil nodes, as well as the cutoff value, so it is comparable across different experimental setups and schemes.

4.2.2 SybilGuard, SybilLimit, and SybilInfer

For comparing SybilGuard, SybilLimit, and SybilInfer to the community detection algorithm, we use the same experimental methodology as the most recent proposal, SybilInfer. Specifically, we use a 1,000 node scale-free topology [4] for the non-Sybil part of the network. Among this set of non-Sybil nodes, a small fraction (10%) of the nodes are compromised by an adversary and become Sybil nodes. These 100 malicious nodes are chosen uniformly at random. These nodes then introduce additional Sybil identities into the network, which form a scale free topology among themselves using the same parameters as non-Sybil region. We vary the number of introduced nodes from 30 to 1,000, and average the results over 100 experimental runs.

We present the results of this experiment in Figure 7. We make two important observations: First, SybilInfer and community detection perform well, with improving accuracy as more Sybils are added. The reason for this increase is that the Sybil region becomes larger and, therefore, easier distinguish from the non-Sybil region. Second, both SybilGuard and SybilLimit perform less well than the other two schemes.

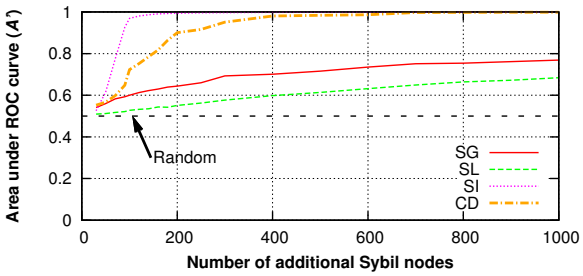


Figure 7: Accuracy for Sybil defense schemes, as well as community detection (CD), on the synthetic topology as we vary the number of additional Sybil identities introduced by colluding entities.

This effect is because the number of Sybil nodes added is lower than the bound enforced by these two schemes, as was observed in the evaluation on SybilInfer [7]. In more detail, the Sybil region is connected to the non-Sybil region by 789 attack edges on the average; SybilGuard and SybilLimit ensure that no more that $O(\log N)$ nodes will be accepted per attack edge, where N is the number of nodes in the network. Since we only add a maximum of 1,000 Sybil nodes, neither of these schemes marks many nodes as Sybils.

We now evaluate these schemes on a real-world social network. Specifically, we repeat this experiment on the Facebook graduate student network from before. This network has similar density as the synthetic network, but is only half the size. The results of this experiment are presented in Figure 8. As we can see, the community detection algorithm performs favorably compared to the explicit Sybil defense schemes, and all become more accurate as more Sybils are added. A careful reader may note that the absolute accuracy of all schemes (community detection included) is significantly lower than that observed above in Figure 7. The underlying reason for this lower performance is a structural characteristic of the Facebook network that makes it inherently harder to distinguish Sybils from non-Sybils. We explore this limitation in greater detail in Section 5.

4.2.3 SumUp

Recall that SumUp provides a Sybil-resilient voting service. To do so, SumUp defines a *voting envelope* wherein the links are assigned a capacity so that all votes from within the envelope can be collected. Outside this envelope, votes are only collected if the voter can find an path with capacity to the vote collector (i.e., the trusted node). In order to apply community detection, we replace the process that determines the voting envelope with a community detection algorithm, pick the community with the lowest conductance value to be the envelope, and unconditionally accept all votes from nodes within this envelope. For nodes outside the envelope, we assign all other links to have capacity one, and we collect their votes if they can find a path with weight to any node within the envelope. This difference is necessary since we don't assign weights to links within the envelope, as SumUp does.

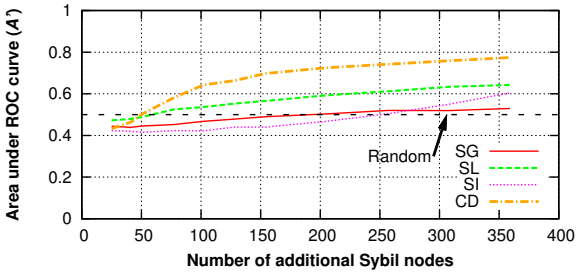


Figure 8: Accuracy in the Facebook network as we vary the number of additional Sybil identities introduced by colluding entities.

We evaluate and compare the community detection scheme against SumUp on three different datasets: Advogato, Astrophysics, and YouTube. We follow the same methodology used in the original SumUp evaluation [29]: for each network, we inject 100 attack edges by inserting 10 Sybil nodes with links to 10 other uniformly randomly chosen non-Sybil nodes. In order to cast bogus votes, each Sybil node is further attached to a large number of Sybil identities by a single link each. As in the original evaluation, we randomly select a vote collector and randomly choose a subset of non-Sybils as voters. We plot the average statistics over five experimental runs for both SumUp and the community detection algorithm.

To evaluate the accuracy of these schemes, we must define a new metric. This is because SumUp does not classify all nodes as Sybil or non-Sybil (needed for A'), but rather, only those nodes which issue votes. Since subsets of both the non-Sybil and Sybil nodes are issuing votes, ideally, the scheme would only count the non-Sybil votes. Thus, our metric should penalize the under counting of non-Sybil votes, as well as the counting of any Sybil votes. The metric we define, *vote accuracy*, is expressed as the number of non-Sybil votes counted divided by the sum of the number of non-Sybil votes issued and the number of Sybil votes counted. Vote accuracy ranges between 0 and 1, where higher values represent better performance.

Figure 9 presents the results of this experiment, as we vary the number of non-Sybil voters (Sybils try to vote as

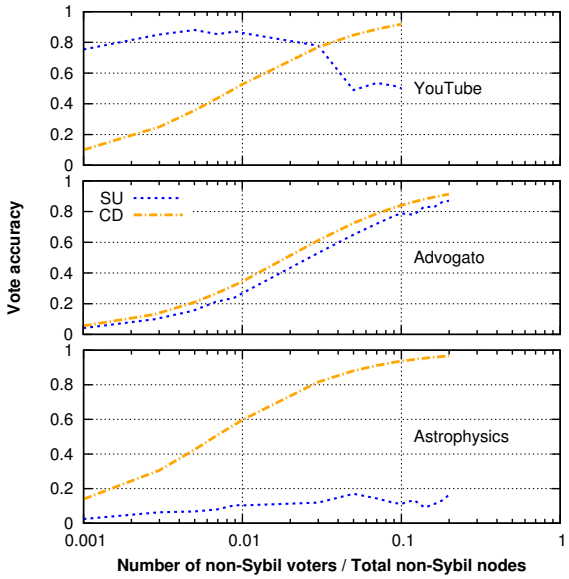


Figure 9: Vote accuracy of SumUp and community detection on three networks.

often as they can). The most salient result is that the accuracy for SumUp varies widely across the three networks; this is a direct result of using the envelope technique. In certain networks, one or more of the Sybil nodes is accepted into the envelope, and a large number of malicious votes are cast. The results for the community detection algorithm are significantly more stable, producing useful results once the number of non-Sybil voters rises above 1%.

4.3 Implications

We began this section by observing that, since all Sybil defense schemes appeared to be identifying local communities, explicit community detection algorithms may be able to defend against Sybils as well. It is interesting to note—even without changing the experimental setup under which existing schemes were evaluated—our simple community detection algorithm gives comparable results to existing schemes. Our results have both positive and negative implications for future designers of Sybil defense schemes.

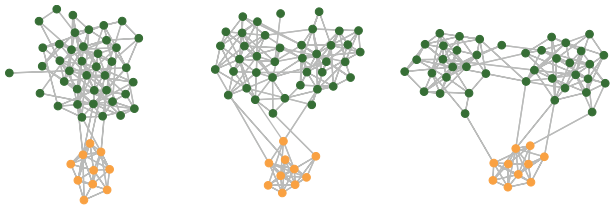
On the positive side, our results demonstrate that there is a opportunity to leverage the large body of existing work on community detection algorithms for Sybil defense [10]. Prior work on community detection provides a readily available source of sophisticated graph analysis algorithms around which researchers could improve existing schemes and design new approaches. On the negative side, relying on community detection for performing Sybil defense fundamentally limits the ability of these schemes to find Sybils in many real-world graphs. We explore these limitations in the next section.

5. LIMITATIONS OF SYBIL DEFENSE

In the previous sections, we showed that Sybil defense schemes work by effectively identifying nodes within tightly-knit communities around a given trusted node as more trustworthy than those farther away. In this section, we investigate the limitations of relying on community structure of the social network to find Sybils. More specifically, we explore how the structure of the social network impacts the performance of Sybil defense schemes and how attackers with knowledge of the structure of the social network can leverage it to launch more efficient Sybil attacks.

Since social network-based Sybil defense schemes use the structure of social networks to distinguish the Sybil nodes from the non-Sybil nodes, we begin by asking the following question: *Are there networks where it is hard to tell these two types of nodes apart?* In other words, could there be networks where the non-Sybil nodes look like Sybils or where it would be easy for Sybil nodes to masquerade as non-Sybils?

Intuitively, one would expect networks where the non-Sybil region is comprised of multiple, small, tightly-knit communities that are interconnected sparsely to be more vulnerable to Sybil attacks. In such networks, nodes within one community might mistake non-Sybil nodes in another community for Sybils, due to limited connectivity between the communities. Furthermore, an attacker can easily disguise Sybil nodes as just another community in the network by establishing a small number of carefully targeted links to the community containing the trusted node. Next, we verify this intuition using experiments over synthetic and real-world social networks where the non-Sybil nodes have different community structures and the Sybil nodes use different attack strategies.



(a) Initial (b) Intermediate (c) Final

Figure 10: Illustrations of the synthetic networks used in Section 5.1 (the actual networks are much larger). Non-Sybils are dark green and Sybils light orange. While the non-Sybil regions of (a), (b), and (c) show increasing amounts of community structure, all non-Sybil regions have the same number of nodes and links, and degree distribution.

5.1 Impact of social network structure

We first examine the sensitivity of Sybil defense schemes to the structure of the non-Sybil region. As in Sections 3 and 4, we analyze synthetic networks and then show that the results from these simple cases apply to real-world networks as well.

We first generate a Bárabási-Albert random synthetic network [4] with 512 nodes and initial degree $m = 8$. This results in a random power-law network with approximately 3,900 links, and without any community structure. We then iteratively generate a series of networks by rewiring [3] five links in same manner as in Section 3 (resulting in a network), then rewiring five more links (resulting in another network), and so on, until only five links remain between the two communities of 256 nodes each (resulting in a final network). The output is a series of networks that all have the same number of nodes, number of links, and degree distribution, but are increasing in the level of community structure that they exhibit. Figure 10 gives a illustration of the initial, intermediate, and final networks.

We use this series of networks to evaluate how well Sybil defense schemes perform on networks with increasing amounts of community structure. To do so, we treat each of these networks as the non-Sybil region, and we randomly attach a Sybil region of 256 nodes using 40 links. We then evaluate how well the existing schemes are able to detect Sybils by using the A' metric. The result of this experi-

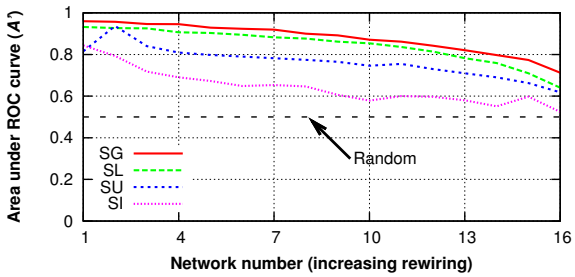


Figure 11: Accuracy of Sybil defense schemes on synthetic networks with increasing community structure induced by rewiring. With high levels of community structure, the accuracy of all schemes eventually falls to close to random.