2013 IEEE Symposium on Security and Privacy

SoK: The Evolution of Sybil Defense via Social Networks

Lorenzo Alvisi
UT Austin

Allen Clement
MPI-SWS

Alessandro Epasto
Sapienza, U of Rome

Silvio Lattanzi
Google Inc.

Alessandro Panconesi
Sapienza, U of Rome

***Abstract*—Sybil attacks in which an adversary forges a potentially unbounded number of identities are a danger to distributed systems and online social networks. The goal of sybil defense is to accurately identify sybil identities.**

This paper surveys the evolution of sybil defense protocols that leverage the structural properties of the social graph underlying a distributed system to identify sybil identities. We make two main contributions. First, we clarify the deep connection between sybil defense and the theory of random walks. This leads us to identify a community detection algorithm that, for the first time, offers provable guarantees in the context of sybil defense. Second, we advocate a new goal for sybil defense that addresses the more limited, but practically useful, goal of securely white-listing a local region of the graph.

I. Introduction

The possibility that malicious users may forge an unbounded number of *sybil* identities, indistinguishable from honest ones, is a fundamental threat to distributed systems that rely on voting [11]. This threat is particularly acute in decentralized systems, where it may be impractical or impossible to rely on a single authority to certify which users are legitimate [20]. The goal of sybil defense is to accurately identify sybil identities[1]—"ideally, the system should accept all legitimate identities but no counterfeit entities" [11]—but simple techniques can be either too brittle (beating a CAPTCHA [40] costs a fraction of a cent) or too blunt (IP filtering penalizes all users behind a NAT).

Against this background, Yu et al. have put forward a radically different approach [45], [46]: protecting a distributed system by leveraging the social network that connects its users. Intuitively, as long as sybil identities are unable to create too many *attack edges* connecting them to honest identities, it may be possible to separate the wheat from the chaff by analyzing the topological structure of the users' social graph. This style of sybil defense[2] promises not only to be more surgical, but offers a mathematically precise and elegant way to characterize the robustness of a sybil defense technique in terms of the number of attack edges it can handle. The vision is to offer *universal* sybil defense to all honest nodes in the system: as long as the social graph conforms to certain assumptions, an honest node will correctly classify almost all honest nodes in the graph while rejecting all but a bounded number of sybil nodes [45].

Several protocols that embrace this style of sybil defense have since been proposed [7], [10], [35], [42], [45], [46] and higher-level distributed applications that rely on them are beginning to emerge [18], [19], [26], [36].

$\sim * \sim$

The first goal of this paper is to examine the promise and the fundamental limits of universal sybil defense. We will see that at the core of this approach are a set of assumptions about the structure of a social graph under sybil attacks that, in essence, amount to modeling the social graph as consisting of two sparsely connected regions: one comprised of sybil nodes; and the other of honest nodes, homogeneously connected with one another. We will report on several studies, confirmed by our own experiments, that suggest that this model over-simplifies the social structure of the honest region of the graph: rather than homogeneous, this region appears as a collection of tightly-knit local communities relatively loosely coupled with one another.

Our second goal for this paper is then to advocate a realignment of the focus of sybil defense to leverage effectively the robustness of communities to sybil infiltration. The intuition that motivates us is not new. Prior work has suggested casting sybil defense as a community detection problem [39] and asked whether it is possible to use off-the-shelf community detection algorithms to find sybils. On this front, we make two contributions. First, we show that this approach requires extreme caution, as the choice of the community detection protocol can dramatically affect whether sybil nodes are accepted as honest. Second, we identify the mathematical foundations on which the connection between sybil defense and community detection rests: we identify a well-founded theory and point to an established literature to guide the development of future sybil defense protocols.

Our conclusion is that instead of aiming for universal coverage, sybil defense should settle for a more limited goal: offering honest nodes the ability to white-list a set of nodes of any given size, ranked accordingly to their trustworthiness. We believe that this is a good bargain, and not just because it results in a goal that, unlike its alternative, is attainable, but because (1) the guarantees it provides are in practice what nodes that engage in crowd-sourcing [47] or cooperative P2P applications [9], [25] need, and (2) the computational cost of providing these guarantees depends

[1] Although this goal may be more accurately characterized as sybil *detection* [38], we use here the term sybil *defense* originally proposed by Yu [45] and widely adopted in the literature.

[2] Henceforth, mentions of sybil defense, unless specified otherwise, refer to techniques that leverage the structure of social networks.

1081-6011/13 $26.00 © 2013 IEEE
DOI 10.1109/SP.2013.33

only on the size of the desired white-listed set rather than, as in techniques that aim for universal sybil defense, on the total number of identities in the network.

The final goal of this paper is to serve as a warning against the danger of falling into a *Maginot syndrome*: the building of an ever more sophisticated line of defense against attacks that the enemy can easily circumvent. Indeed, evidence from the RenRen social network [43] shows sybil attacks that differ from what current sybil defenses anticipate and that, despite their simplicity, can be devastating. We argue that the key to address this challenge is defense in depth, where early defense layers (of which we sketch a few) are designed to catch the simple sybil subgraphs that current defenses are ill-equipped to detect.

Finally, a friendly warning. Achieving the goals we have outlined requires a good mathematical understanding of the problem and of the techniques developed to address it. At times the discussion will be technical; we hope that the persevering reader will be rewarded. Bear with us.

$\sim * \sim$

The paper proceeds as follows. Section 2 examines four fundamental structural properties of social graphs (popularity, small world property, clustering coefficient, and conductance) and asks: which can better serve as a foundation for sybil defense? The answer, we find, is conductance, a property intimately related to the concept of mixing time of a random walk. We then proceed in Section 3 to discuss protocols that exploit variations in conductance as a basis for decentralized universal sybil defense [10], [35], [42], [45], [46]. These protocols provide elegant worst-case guarantees when it comes to their vulnerability to sybil attacks; however, these guarantees are critically sensitive to a set of assumptions that do not appear to hold in actual social networks [6], [17], [23]. This motivates us to explore, beginning with Section 4, an alternative goal for sybil defense that leverages two observations: (1) social graphs have an internal structure organized around tightly-knit communities and (2) the graph properties crucial for sybil defense are significantly more likely to hold within a community rather than in the entire social graph. Section 5 reviews recent work on the theory of random walks that provides a solid theoretical foundation to sybil defense based on community detection; we deepen our investigation of random walks in Section 6, where we show how the well-known concept of Personalized PageRank (not to be confused with PageRank itself) offers honest nodes a path towards a realistic target for sybil defense that is more limited than universal coverage but nonetheless useful: a way to white-list trustworthy nodes that proves efficient and robust in both theory and practice. After all this effort, Section 7 greets us with a sobering result: in spite of their sophistication, state of the art sybil defense protocols are helpless against very crude real-life sybil attacks. However, we show that sybil defense protocols based on random walks continue to be effective when used in combination with very simple checks that leverage structural properties of the social graph other than conductance. Section 8 offers our conclusions and points to directions for possible future research.

II. Sybil defense via social networks

Sybil defense via social networks is predicated on the assumption that it is possible to leverage the structural properties of the social graph G underlying a distributed system to differentiate the honest subgraph H from the sybil subgraph S. In this section, we ask a basic question: which structural property, if any, is most promising towards defending against sybil attacks?

A. Structural properties of social graphs

We consider (and briefly review below) four well-known structural properties that are commonly viewed as characterizing social graphs: the popularity distribution among its nodes [5], the small world property [41], the value of its clustering coefficient [41] and its conductance [17].

Popularity: The node degree distribution of social graphs is heavy-tailed, as in a power-law or lognormal distribution.

Small world property: The diameter of a social graph—i.e., the longest distance between any two nodes in the graph—is small.

Clustering coefficient: A measure of how closely-knit is a social network. When we associate a network vertex v with the user u that it represents, the vertex' clustering coefficient c_v is the ratio between the actual number of friendships between the friends of u and the maximum possible number of friendships between them. Formally, let f_v denote the actual number of edges between neighbors of v, i.e. $f_v := |\{xy : x \in N_v, y \in N_v, xy \in E\}$; and let k be the maximum number of edges between neighbors of v: $k = \binom{\deg(v)}{2}$, where $\deg(v)$ denotes v's degree. Then, $c_v := \frac{f_v}{k}$. The clustering coefficient of a graph is the average clustering coefficient of all its vertices, i.e. $c(G) := \sum_{v \in V(G)} \frac{c_v}{|V|}$.

Conductance: Social graphs are conjectured to be *fast-mixing*, meaning that if we take a random walk in a social graph we will quickly arrive at a random point. This property is at the core of many solutions developed for sybil defense. A graph's *mixing time* [30], which informally conveys the minimum length of a random walk that ends on a uniformly random edge, is intimately related to the concept of conductance: when conductance is high, mixing time is low. Intuitively, the *conductance* of a set S of vertices, denoted by $\varphi(S)$, in a given network is the ratio between the number of edges going out from S and the number of edges inside S. More precisely, given a set of vertices S, the conductance of the set is defined as

$$\varphi(S) := \frac{|cut(S)|}{vol(S)}$$

Graph	Nodes	Edges	Attack Edges	Diameter	90% Diameter	Clustering Coeff	Est. Conductance
DBLP [1]	718115	2786906	0	20	7.43	0.73	0.016
... $p = 0.01$	1436230	5601767	27955	19	7.94	0.71	0.006
... $p = 0.10$	1436230	5851341	277529	17	7.02	0.67	0.031
Epinions [28]	26588	100120	0	16	5.98	0.23	0.020
... $p = 0.01$	53176	201197	957	16	6.72	0.23	0.005
... $p = 0.1$	53176	210291	10051	14	5.97	0.21	0.027
Facebook [37]	63392	816886	0	12	5.15	0.25	0.020
... $p = 0.01$	126784	1641891	8119	14	5.79	0.25	0.005
... $p = 0.10$	126784	1715206	81434	13	5.25	0.23	0.020
WikiTalk [15]	92117	360767	0	9	4.63	0.14	0.048
... $p = 0.01$	184234	725152	3618	10	5.02	0.13	0.005
... $p = 0.10$	184234	757729	36195	10	4.75	0.12	0.053

Table I

STATISTICAL PROPERTIES OF THE LARGEST STRONGLY CONNECTED COMPONENT IN A COLLECTION OF REAL WORLD DATA SETS. THE VALUES REPORTED REFLECT THE PROPERTIES OF THE DATA SET BEFORE AND AFTER THE ATTACK SPECIFIED IN SECTION II–B. THE DBLP GRAPH IS A SNAPSHOT OF THE DBLP CO–AUTHOR GRAPH FROM 2011; THE EPINIONS GRAPH IS A DATASET FROM THE EPINIONS PRODUCT REVIEW SITE OBTAINED IN 2003; THE FACEBOOK GRAPH IS A CRAWL OF THE FACEBOOK–NEW ORLEANS COMMUNITY IN 2007; THE WIKITALK GRAPH IS DERIVED FROM THE WIKIPEDIA PAGE EDIT HISTORY AS OF JANUARY 2008.

where the *volume* of S is defined as $vol(S) := \sum_{v \in S} \deg(v)$ (the sum of the degrees of vertices in S), and the *cut* induced by S is the set $cut(S)$ of edges with one endpoint in S and the other endpoint outside of S. Finally, the *conductance* of a graph G is defined as

$$\varphi(G) := \min_{vol(S) \leq |E|} \varphi(S).$$

B. Which property is most resilient?

Consider a social network G in which every node is honest, and assume a sybil defense that uses a structural property of the social graph to correctly classify every node. An attack that somehow turns some of the nodes in G into sybils, without otherwise affecting the social network, will be undetectable, since it will change nothing tangible. We could term this a *perfect attack*. Similarly, if an adversary can add sybil identities to G without altering G's structural properties, then any sybil defense that tries to leverage those properties will be circumvented.

We can however compare the four structural properties above in terms of the *effort* they require of an adversary bent on evading detection: in particular, we measure the number of attack edges that the adversary needs to create to be undetectable.

To this end, we assume that a graph H with n honest nodes is given and that the attack induces a graph S of sybil nodes. While H is fixed, the adversary has full control over S and can build it so that its structural properties are indistinguishable from those of H—for simplicity, we assume that S is an exact copy of H.

The adversary tries to set up $m := |E(H)|$ potential *attack edges* that connect H with S. We assume that the endpoints of these edges in both H and S are chosen by preferential attachment, i.e. a vertex v is chosen with probability

$$\frac{\deg(v)}{2m} \tag{1}$$

As we will see, preferential attachment is crucial to not alter properties of the social network and in particular its degree distribution.

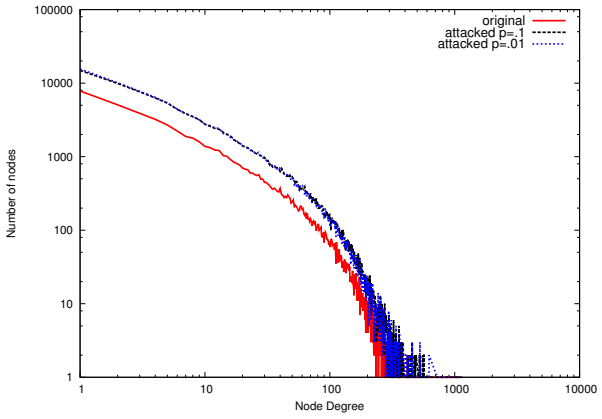
If the attacker is able to create arbitrarily many attack edges, no sybil defense can hope to distinguish between the two regions of the graph. Therefore, as customary in the sybil defense literature [45], [46], we assume that the attacker's ability to create attack edges is limited; in particular, we postulate that tentative attack edges are accepted with probability p and rejected with probability $1 - p$, resulting in a set A of attack edges. To account for the outcome of recent social engineering experiments [6], we allow p to be constant: the expected cardinality of A is then pm. We denote with G the graph that results from joining S to H via A.

Under this simple attack model, how resilient are then the four defining structural properties of social graphs?

1) Popularity: We find that it is trivial for the adversary to make sure that G's popularity distribution is statistically indistinguishable from that of H. We prove [2] that a) the expected degree of an honest node in G is barely higher than in H and b) moving to G will, in essence, at most double the degree of a popular honest node.

Proposition 1. *(a) For each* $v \in H$, $E[\deg_G(v)] = \deg_H(v)\left(1 + \frac{p}{2}\right)$. *(b) If* $\deg_H(v) > 6 \log n$, *then* $\deg_H(v) \leq \deg_G(v) \leq \deg_H(v)\,(2 + p)$ *with probability* $1 - o(1)$.

Figure 1 plots of the degree distribution of the Facebook network before and after two attacks in which attack edges are inserted respectively with probability $p = 0.01$ and $p = 0.1$: the curves before and after the attacks have the same shape. Indeed, an attack that introduced *no* attack edges would produce the same curve! We conclude that popularity is ill–suited as a foundation for sybil defense.



(a) Facebook

Figure 1. Degree distribution of the Facebook graph before and after attack. The attack shifts the distribution up (because it doubles the size of the graph) and to the right (proportionally to the number of attack edges), but does not change the shape of the curves.

2) Small world property: The small world property does not fare much better than popularity, since the adversary can easily keep the diameter of G from growing suspiciously. First, it is easy for the adversary to bound the relative growth of the diameter of G with respect to that of H: if $S = H$ and the adversary succeeds in inserting just one attack edge the diameter can at most double. The following proposition immediately follows [2]:

Proposition 2. *A sybil attack can at most double the diameter of H.*

Second, it is easy for the adversary, who has full control over S, to effect any change to the diameter slowly, so that it appears completely physiological. Our experimental evaluation of several real life social networks shows (90% diameter column of Table I) that the 90%-effective diameter [16], which measures the maximum distance between 90% of the pair of nodes, is indeed barely affected under attack.

3) Clustering coefficient: Leveraging the clustering coefficient appears promising because attack edges reduce its value. Unfortunately, while the clustering coefficient of social networks is typically high, its value varies significantly from network to network [17], from 0.79 in the actor collaboration network of IMDB, down to 0.35 for Live Journal and to a mere 0.09 for the social network of Yahoo! Messenger chat exchanges. Thus, if an attack modifies the clustering coefficient by a small multiplicative factor, the change is hard to detect, especially if made very gradually.

We capture that intuition in the following result [2].

Theorem 1. *Let H be the graph of honest nodes and let G be the network under the sybil attack described in II-B. Also, let $\alpha := 8(1 + \frac{1}{2}p)^2$, where p is the probability that an attack edge is accepted. Then, $c(G) \geq \alpha^{-1}c(H)$ with high probability*

The implications of this theorem are disappointingly clear: the clustering coefficient is not a good basis for sybil defense, since even after the attack its value cannot drop by too much. In fact, if the number of attack edges is smaller than pm, with high probability there will be only a constant change in the clustering coefficient. The *Clustering Coeff* column of Table I confirms the theorem's predictions.

4) Conductance: Yu et al. [45] prove that for graphs whose conductance is asymptotically constant, an adversary that can introduce $O(n)$ attack edges can build a graph G whose conductance is indistinguishable from that of H. We generalize that result to graphs of arbitrary conductance as follows [2].

Theorem 2. *Let H denote a network of n honest nodes and m edges such that $\varphi(H)m = \Omega(\log n)$, and let S denote a network of n' sybil nodes with m' edges such that $\varphi(S) \geq \varphi(H)$ and $\varphi(H)m \leq m' \leq m$. Suppose further that the adversary is able to establish between $\varphi(H)m \log \varphi(H)^{-1}$ and m attack edges. Then, with high probability, $\varphi(G) = \Omega(\varphi(H))$.*

The fundamental implication of the theorem is that if the adversary is able to introduce at least $\varphi(H)m \log \frac{1}{\varphi(H)}$ attack edges (i.e., $O(m)$ attack edges when the mixing time is $O(\log n)$), then the conductance of the graph will with high probability remain very nearly the same.

Table I confirms the theorem's message that an adversary that succeeds in generating sufficiently many attack edges can circumvent any technique that attempts to detect sybil nodes by looking for significant changes in global conductance. As expected, the conductance drops significantly under a weak attack ($p = 0.01$), providing leverage for sybil detection. However, under a strong attack ($p = 0.1$) the conductance may actually *increase* because, by adding random attack edges, the adversary enlarges every cut with some probability, including the cut with minimum conductance which defines the conductance of the entire network.

Note that computing a graph's conductance is NP-hard. The conductance values that we report are approximate and were obtained using the the approximation method proposed by Leskovec et al. [17].

C. Discussion

None of the structural properties of social graphs that we have considered provides full-proof defense against sybil attacks in general, or even against the specific attack we have assumed. However, as Table II shows, when a graph under attack is observed through the lens of conductance, the adversary has to work much harder to look inconspicuous. These results both motivate and justify the insight of Yu and his collaborators to rely on conductance in the work that jump-started sybil defense via social networks [46]. We

Property	Number of edges to circumvent it
Degree distribution	$\lvert A\rvert \geq 0$
Diameter	$\lvert A\rvert \geq 1$
Clustering coefficient	$0 \leq \lvert A\rvert \leq m$
Conductance	$\varphi(G)m\log\varphi(G)^{-1} \leq \lvert A\rvert \leq m$

Table II
THE TABLE SHOW HOW MANY EDGES ARE NEEDED FOR THE ATTACKER TO CIRCUMVENT THE MAIN 4 PROPERTIES OF SOCIAL NETWORKS.

review their approach, its successes, and what we believe to be ultimately its fundamental limitations in the next section.

III. Leveraging conductance towards universal sybil defense

The vision behind the seminal work of Yu and his collab–orators is to develop a decentralized approach to universal sybil defense, with the goal of allowing honest users to correctly assess with high probability the honesty of every other user in the system. False positive and false negatives would still be possible, but they would be few and, further, their number would be bound within a rigorous theoretical framework. This compelling vision, first articulated in the SybilGuard protocol [46], is further refined in their later work on SybilLimit [45] and has inspired several other efforts in sybil defense [7], [10], [35], [42].

We begin this section by discussing the main intuition underlying these techniques and the guarantees that they provide; we then proceed to discuss the crucial role that a set of key assumptions play in ensuring those guarantees and present evidence suggesting that the assumptions do not appear to hold in actual social graphs.

A. Picking whom to trust

In all these protocols, an honest node determines which nodes to trust on the basis of a sample of the social graph collected by using random walks. Different protocols apply sampling in different ways and to different parts of the graph. SybilLimit [46] samples edges; SybilGuard [45] and Gatekeeper [35] sample nodes in the graph; SybilInfer [10] uses the random walks to build a Bayesian model for the likelihood that a trace T was initiated by an honest node. In the following, we provide an overview of how SybilLimit [46] applies the random sampling of edges to identify honest users. While the details of the discussion are specific to SybilLimit, the intuition for how the structural properties of the graph make random sampling effective is common to this entire family of protocols.

Let us consider a particularly simple version of the sybil detection problem. We are given two disjoint graphs H and S—the graph of honest and, respectively, sybil nodes; an honest vertex u—the seed; and a vertex v. Our task is to determine whether v belongs to H or to S. Both nodes select an edge at random: u accepts v if they pick the same edge.

The probability of collision is very low, $\frac{1}{m}$. To boost it we can use the classic birthday paradox. Vertex u picks a set S_u of, say, $\sqrt{m}$ distinct edges, while v picks a set S_v of $\sqrt{m}$ edges independently at random: now u accepts v if there is a collision (i.e. $S_u \cap S_v \neq \emptyset$). This probability is

$$1 - \Pr(\text{no collision}) = 1 - \left(1 - \frac{1}{\sqrt{m}}\right)^{\sqrt{m}} \sim 1 - \frac{1}{e} \qquad (2)$$

a good probability of success. Note now that the set S_u can itself be picked at random. Since $|S_u| = \sqrt{m} \ll m$, almost all edges will be distinct. This simple protocol succeeds with good probability: each vertex picks a set of $\sqrt{m}$ edges independently and uniformly at random. If the two sets intersect, then u accepts v, otherwise it does not. The protocol is symmetric and can be used by both u and v to determine whether to trust one another. This basic idea can be further refined to obtain a test that succeeds with overwhelming probability with small–sized edge sets.

Suppose now we have two disjoint graphs and two ver–tices: we want to determine whether or not they belong to the same graph. If vertices are restricted to pick the edge set from their own graph, the simple protocol above provides the membership test we are looking for: if the two vertices live in different graphs the chance that they trust each other is zero, otherwise it is given by Equation (2).

But how can we implement the test in a distributed fashion? A simple approach is to take a random walk in the graph—which, in the interest of efficiency, should be very short—and pick the last edge of the walk. This is a correct implementation of the test as long as the short random walk picks edges at random (i.e., every edge is equally likely to be selected). It is here that the graph's *mixing time* enters the picture: it is the minimum length of a random walk that selects edges in an unbiased way.[3] Networks for which random walks of length $O(\log n)$ are sufficient (i.e., have mixing time $O(\log n)$) are said to be *fast mixing*.

Therefore, if we assume that the graph of honest nodes is fast mixing, we have a very good protocol for sybil detection, as long as H and S are disjoint. In reality, however, H and S are connected through the attack edges that nodes in S have convinced nodes in H to accept: it is then possible that a random walk starting from $v \in S$ will traverse an attack edge, enter H, and pick one of the edges selected by $u \in H$. The intuition is that, as long as the cut between H and S is sparse, such situations are sufficiently unlikely that the mechanism continues to function with good probability. Indeed, as we already mentioned, Yu et al. prove [46] that as long as the number of attack edges is bound by $o(\frac{n}{\log n})$, then this approach can effectively distinguish between honest and sybil nodes.

Graph	Nodes	Edges	Diameter	90% Diameter	Clustering Coeff	Est. Conductance
DBLP	718115	2786906	20	7.43	0.73	0.016
... preprocessed	191172	1438509	15	5.97	0.60	0.020
Epinions	26588	100120	16	5.98	0.23	0.020
... preprocessed	5624	57341	7	3.89	0.18	0.040
Facebook	63392	816886	12	5.15	0.25	0.020
... preprocessed	40757	632597	7	4.43	0.23	0.023
Wiki–Talk	92117	360767	9	4.63	0.13	0.047
... preprocessed	13069	133343	5	3.78	0.06	0.333

Table III
STATISTICAL PROPERTIES OF THE GRAPHS BEFORE AND AFTER PREPROCESSING. PREPROCESSING DRASTICALLY REDUCES THE GRAPHS' SIZE AND SIGNIFICANTLY ALTERS THEIR STRUCTURAL PROPERTIES.

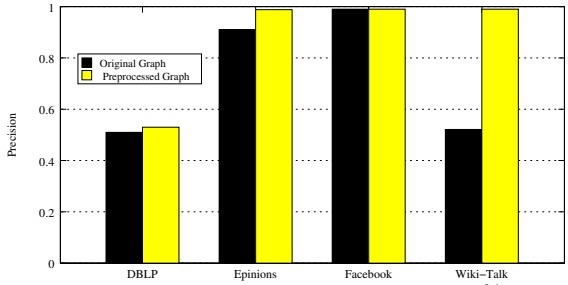


Figure 2. The precision of SybilLimit when recall is 95% on each of the social networks we consider when $p = 0.01$. Other SybilLimit-like protocols show qualitatively similar results.

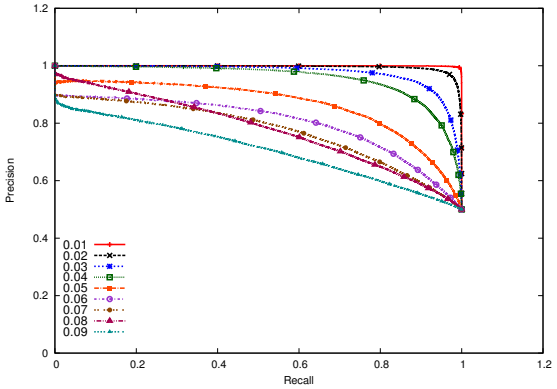


Figure 3. Precision vs Recall of SybilLimit and the Facebook network for p (ranging from 0.01 to 0.10). The number of attack edges is pm.

B. Cracks in the foundations

There are then two fundamental assumptions that underly this elegant approach towards decentralized universal sybil defense. The first is that the cut between the sybil and honest region—the set of attack edges— is suitably sparse. The second is that the mixing time of the honest region is $O(\log(n))$. The combination of these two assumptions ensures that random walks of $\Theta(\log n)$ steps will end in a random edge in the honest region with high probability.

Recent literature has cast doubts on whether these assumptions hold in practice. Social graphs do not seem to be fast mixing after all [17], [23], and the probability with which fake identities are accepted as friends is much higher than anticipated [6], [43], implying that the set of attack edges is not as sparse as assumed. We then ask: to what degree are SybilLimit-like protocols sensitive to their assumptions about sparse cuts and mixing time?

To answer this question, using SybilLimit [46] as representative (we find that the behavior of other SybilLimit-like protocols is similar), we produce, as in [39], a ranking of nodes with respect to a given *verifier node* u, in decreasing order of trust: the first node in the ranking is the node that u trusts the most. We then measure the defensive efficacy of SybilLimit by using two metrics well known in the field of information retrieval: *precision* and *recall*. In particular, we define the precision at position k as the fraction of honest nodes among the k nodes that the protocol ranks the highest. Similarly, we define the recall at position k as the ratio between the number of honest nodes among the top k positions in the ranking and the total number of honest nodes in the network.

SybilLimit-like protocols do not operate on raw social networks: they are to be used only on networks that have been preprocessed by iteratively removing all nodes with degree lower than five [46]. Table III shows the statistical properties of the graphs we use in our experiments.

Sensitivity to sparse cuts. Figure 3 plots SybilLimit's precision versus recall for the preprocessed Facebook data set. SybilLimit proves very effective when the number of attack edges is within the theoretical bound (which corresponds to $p = 0.01$). Once the bound is exceeded, however, the performance of SybilLimit decays rather quickly.

Sensitivity to mixing time. Mohaisen et al. [23] are the first to observe that this step, while boosting the mixing time of social graphs to the level required by SybilLimit to be effective, can also reduce the size of the graph. Table III confirms this observation: in the case of Wiki-Talk, the preprocessing step removes over 85% of the nodes. Removed nodes are effectively considered sybils by the protocol, and while those nodes may still be able in some circumstances to enlist other nodes in the network as proxies [45], it is unclear

[3]The discussion in this section is informal for the sake of clarity.

in general how removed nodes can safely take advantage of honest nodes' resources and vice versa [23].

C. Discussion

The goal of universal decentralized sybil defense with strong theoretical guarantees, which has driven early research on sybil defense via social networks, rests on assumptions (short mixing time and cut sparseness) whose validity is at best dubious. What to do? In a recent survey [44], Yu suggests a couple of ways forward: one could offer sybil defense only to the nodes in the core of the social graph, in effect institutionalizing the removal of nodes that are not as well connected; or one could simply renounce the elegant theoretical worst-case claims of the current framework and rely instead on "weaker but less clean assumptions" [44]. In the next section, we explore a third alternative that offers every honest node a useful degree of sybil protection without compromising on elegance and rigor.

IV. Communities

The theoretical guarantees offered by the protocols discussed so far hold only as long as honest nodes are closely connected to one another everywhere in the social graph and the cut between honest and sybil nodes is sparse. Empirical evidence suggests a different reality: social graphs consist of communities, each a tightly knit sub-network [17], [23]. Indeed, it is quite conceivable that the cut between two tightly-knit communities of honest nodes A and B be as sparse as the cut between A and the sybil region: to an honest node in A using a protocol in the style of SybilLimit, a sybil node would then be indistinguishable from an honest node in B [38], [39].

While these considerations argue against universal sybil defense, they suggest an alternative goal: to provide each honest node u with the ability to white-list a trustworthy set of nodes—namely those in the community to which u belongs. This new goal can be more precisely stated as follows:

Problem 1. *Let u be an honest user and S a subset of the honest region such that: (a) $u \in S$, (b) S has mixing time τ and (c) there are at most $o\left(\frac{|S|}{\tau}\right)$ edges between S and the rest of the social graph. We want an algorithm capable of distinguishing almost perfectly between the nodes in S and the nodes outside of S.*

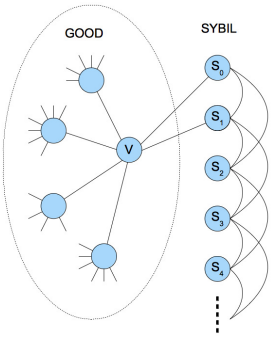


Figure 4. Two edge attack.

We make two observations. First, the problem of universal sybil defense is a special case of Problem 1 in which $\tau = O(\log n)$ and S is the entire honest network. Second, sybil defense appears, informally, to reduce to the task of detecting the "community" S.

The fundamental affinity between community detection and sybil defense has been first observed by Viswanath et al [39]. After pointing out that, from the perspective on an honest node, SybilLimit-like protocols separate the social graph in two communities—honest nodes and sybils—they go on to ask a natural follow-up question: can off-the-shelf community detection algorithms be used to detect sybils? Their answer is mixed: on the one hand, they show that a generic community detection algorithm due to Mislove [21] (also a co-author in [39]) achieves results comparable to those of SybilLimit-like protocols on both a synthetic topology and a real-life Facebook social graph; on the other, they observe that attackers wise to the community substructure of the honest portion of the social graph can manage, as we discussed above, to make the sybil region appear indistinguishable from a sub-network of honest nodes.

We believe that a first step towards a more conclusive answer is to recognize that casting the problem simply in terms of generic community detection leaves it underspecified. While intuitively compelling, the notion of community is ambiguous, as the many community detection algorithms found in the literature, each aiming for a subtly different notion of community, clearly indicate [12]. But what should be the basis for the notion of community to be used in sybil defense?

A. The minimum conductance cut

A somewhat obvious candidate to serve in this role is conductance. Conductance is hard to tamper with (see Section II) and it is intimately related to mixing time, a critical property to leverage against sybil attacks (see Section III).

It is tempting to define the problem of sybil defense in terms of the *minimum conductance cut problem* found in the community detection literature:

Problem 2. *Find a set S whose conductance $\varphi(S)$ is as close as possible to $\varphi(G)$, the minimum conductance of the graph.*

If we believe that the honest region is fast mixing and that it is connected to the sybil region via a sparse cut, the set S should be very close to capturing the entire honest region. This view is of course too simplistic and can lead to community detection algorithms that can be circumvented by an adversary using far fewer attack edges than needed

to dupe SybilLimit–like protocols. Mislove's algorithm [21] serves, in this sense, as a cautionary tale.

Mislove's algorithm is a heuristic algorithm that finds small conductance cuts—which is, in essence, analogous to finding an approximate solution to Problem 2. The set S is built greedily. Starting from a vertex u, the algorithm grows S by incorporating the vertex v connected to S that results in a set $S \cup \{v\}$ with minimal conductance.[4]

Although this simple heuristic appears to capture the intuition behind Problem 2, it fails against the following simple attack. Let v be an honest node, that has no neighbor of degree at most 3. We create the sybil region with nodes $s_0, s_1, \ldots, s_n$ as follows:

- s_0 and s_1 are connected to v.
- For every $i \leq n-2$, s_i is connected with the next two sybil nodes in the sequence s_{i+1}, s_{i+2}, and also with the previous two, s_{i-1}, s_{i-2}.

Figure 4 illustrates the attack, involving only the two attack edges connecting v to s_0 and s_1, that results in Mislove's algorithm deterministically admitting every node of the sybil region[5] (see [2] for a full proof).

B. Discussion

Reframing sybil defense to leverage the community sub–structure that exists in social graphs requires a deep un–derstanding of the relationship between sybil defense and conductance—in essence, understanding when a solution to Problem 2 is also a solution to Problem 1. The key to the approach we explore in subsequent sections relies, at a local scale, on a technique central to the efforts towards universal sybil defense discussed in Section III: random walks.

V. Fast mixing communities

Because of its tight connection with the theory of random walks, the minimum conductance cut problem that we have used to formalize the intuitive relationship between sybil defense and community detection has been studied in depth.

Problem 2, as we have called it, is NP–hard, so the best that can be hoped for is an approximate solution. Several sophisticated algorithms offer non trivial guarantees on the quality of their approximation to the problem [4], [14], [31], but they have two serious drawbacks when it comes to large social graphs: they are not obviously parallelizable and their running time is polynomial in the size of the entire graph. We then consider a different style of techniques that offer less stringent guarantees on the approximations they produce but whose time complexity depends only on the size of the set S we are trying to identify, which we expect to be significantly smaller than the size of the entire social graph.

[4]The original proposal for Mislove's algorithm [21] relies on a normal–ized conductance metric, but in the context of sybil defense the protocol is evaluated using just conductance [39]. For consistency, we follow the approach of the second paper.

[5]Furthermore this attack can be modified to withstand also the prepro–cessing defined in section III–B

The first such "local" algorithm was developed by Spiel–man and Teng [32]. Very roughly, their idea is to associate a weight with each node and to identify as part of the com–munity all nodes whose weight exceeds a certain threshold. To determine the weight of a node they effectively run many truncated random walks of the same length $t \in \tilde{O}(\phi^{-1})$, all originating from the same node (the seed): a node's weight is given by the frequency with which it is visited normalized by degree. The potential of this algorithm for sybil detection becomes evident once one interprets the weight of a node v as a measure of the trust that the seed node puts in v. Indeed, the recent sybil detection protocol SybilRank [7] is essentially an implementation of the algorithm of Spielman and Teng, run using multiple seed nodes.

Since the work of Spielman and Teng, however, the use of truncated random walks for computing low conductance cuts has been further refined. In particular, Andersen, Chung and Lang [3] originate many random walks from the honest seed, as in [32], but the length of their random walks, instead of being fixed, is determined by means of a (geometri–cally distributed) random variable. This algorithm has two properties that are extremely useful in our context. First, it computes a set S whose conductance is smaller than what is computable with the approach used in SybilRank. Second, it is parallelizable and, crucially, its running time depends not on the size of the entire graph, but only on the size of S.

Andersen and Perez [27] and, very recently, Gharan and Trevisan [24] have proposed further improvements. It is not immediately obvious, to us at least, if these algorithms can be used by an honest seed to rank other nodes according to its trust in them. For this reason, we will focus henceforth on the method proposed in [3], which naturally computes such ranking.

A. Discussion

Formalizing community detection in terms of Problem 2 allows us to draw from the rich literature on random–walk–based algorithms. Among them, the algorithm of Andersen, Chung and Lang stands out for the combination of its features: it supports node ranking; the cut it computes has smaller conductance than most of its peers; its running time depends on the size of the community, not that of the graph; and it is easy to parallelize. In the next section we will see that this algorithm solves Problems 1 and 2 simultaneously, i.e., it is able to identify a community of honest nodes containing the honest seed, without being lured into the sybil region. Further, we will prove the first theoretical guarantees on the performance of a community detection algorithm in the context of sybil defense and show experimentally that the algorithm is quite competitive with the state of the art.

VI. A DEEP DIVE: PERSONALIZED PAGERANK AND LOCAL DEFENSE

In this section we analyze in some depth the "variable length" random walk algorithm of Andersen, Chung and Lang [3], which from now we refer to as ACL. Since ACL is based on the *normalized* stationary distribution of the Personalized PageRank [13] (PPR) random walk, we start by reviewing PPR's definition.

Starting from an initial vertex v (which in our application will be an honest seed), at each step in the walk a pebble returns to v with probability α and moves to a uniformly random neighbor of its current location with probability $1-\alpha$. This random walk has a unique stationary distribution [3] that we denote as $p_{\alpha,v} := (p_1, \ldots, p_n)$. Clearly, this distribution depends on the starting node v and the *jumpback* parameter α.

Intuitively, it is as if, starting from the honest seed, we performed many random walks whose length is determined by means of a geometric random variable: a random walk has length k with probability $\alpha(1-\alpha)^{k-1}$ and, as it is well-known, expected length α^{-1}. Note that long walks are likely to be rare—their probability decays exponentially—while short walks in the neighborhood of the honest seed are common. In this fashion, the nodes in the "community" to which the honest seed belongs should be visited most frequently. The weight $p_{\alpha,v}(u)$ that a node u receives, intuitively, is proportional to the number of times it is visited when "many" random walks are performed. ACL uses the PPR limit distribution, for a given honest seed v and a given α, to assign a "trust" value to each vertex u in the network as follows:

$$t_{\alpha,v}(u) := \frac{p_{\alpha,v}(u)}{\deg(u)} \tag{3}$$

Sorting according to $t_{\alpha,v}$ in descending order produces a ranking of the nodes from the point of view of the verifying node v, from the most trustworthy to the least trustworthy.

This ranking is significantly more robust than that obtained by methods based on PageRank (see for example EigenTrust [29], TrustRank [48]) or that apply PPR directly [22]. First, since a random walk can reset only to the seed node, this ranking is immune to all attacks to PageRank based on exploiting random walks that jump back to a spam node [8]. Second, it includes a normalization step that is crucial to obtain the formal guarantees and experimental performance we are seeking: in particular, it prevents high-degree sybil nodes from spuriously outranking less popular honest nodes just by virtue of their high degree.

We now prove that this ranking achieves precisely what we are looking for: it defines a low-conductance cut containing the honest seed and almost no sybil nodes, thereby solving Problem 1.

Let us assume that the degree distribution of the honest region H follows a power law and that S is a subset of nodes in H. Let τ be the mixing time of the graph induced by S, and let $\alpha := (10\tau)^{-1}$.

Theorem 3. *Let $0 \leq \epsilon \leq \frac{1}{2}$ be a constant and let $cut(S, \overline{S}) = o(|S|\tau^{-1})$. Then, there exists a subset $S' \subset S$ of size $|S'| \geq (1-\epsilon)|S|$ such that, for every node $v \in S'$, the first $|S|$ positions of the ranking induced by $t_{\alpha,v}$ contain at least a $1 - o(|S|)$ fraction of vertices from S.*

This theorem, proved in [2], shows that almost all vertices of S can be used as seeds to obtain a ranking whose first $|S|$ positions consist almost only of honest nodes from S, thereby essentially solving Problem 1. Probabilistically, if we pick a random seed inside the honest community S then, with probability $1-\epsilon$ the corresponding ranking will correctly white-list almost all vertices in S.

We are now ready to discuss how ACL provides an arbitrarily good approximation of this ranking.

A. Computing the ranking

The difficult step in producing the ACL ranking lies in producing the PPR distribution, which, as a stationary distribution, is inefficient to compute in general. ACL consequently relies on a push-flow algorithm for approximating it quickly [3]. This algorithm, which we dub Approximate Personalized PageRank (APPR), has three input parameters: a starting vertex v, a jump back probability α, and an error parameter ϵ. APPR computes an approximate vector $q^{\epsilon}_{v,\alpha} := (q_1, \ldots, q_n)$ that is an approximation of the PPR vector $p_{v,\alpha}$.

To produce the approximate vector $q^{\epsilon}_{v,\alpha}$, APPR assigns to the starting node v an amount of "trust" equal to 1, which then flows from v to the rest of the network through a series of "trickle" operations. Each push-flow operation simulates one step of the random walk by transferring a small amount of trust from a vertex u to its neighbor w in proportion to the probability that the random walk moves from u to w in one step. For each node v, APPR keeps track of two quantities: a $ppr(v)$ value and a residual value $r(v)$. The former is the current approximation of the PPR of the node v, while the latter is the amount of total residual trust that the node is allowed to distribute to itself and to its neighbors. The algorithm is described as Algorithm 1 (for a full discussion see [3]).

The final step in ACL is to degree-normalize the approximate vector $q^{\epsilon}_{v,\alpha}$ produced by APPR as follows:

$$\text{ACL}_{v,\alpha} := \frac{q^{\epsilon}_{v,\alpha}(u)}{\deg(u)}. \tag{4}$$

To understand the ACL algorithm it is important to appreciate the effect of changing the α and ϵ parameters. Theorem 3 tells us how we should set the value of α. The dependence on ϵ is also reasonably straightforward. Since ϵ measures how far we are from the limit distribution, smaller values of ϵ imply longer running times. The good news is

Algorithm 1 $APPR(v, \alpha, \epsilon)$

$ppr(u) = 0 \; \forall u \in V$
$r(u) = \chi_v$
$Q = \{v\}$
for $|Q| \neq 0$ **do**
  Extract u from Q.
  while $r(u) \geq \epsilon d(u)$ **do**
    $ppr, r = Push_u(ppr, r)$
    Insert in Q all the nodes w in the neighborhood of u such that $r(w) \geq \epsilon d(w)$
  end while
end for
return ppr

Algorithm 2 $Push_v(ppr, r)$

Ensure: $ppr' = ppr$ and $r = r'$ with the following exceptions
$ppr'(v) = ppr(v) + \alpha r(v)$
$r'(v) = \frac{1-\alpha}{2} r(v)$
for all $u \in V$: $(u, v) \in E$ **do**
  $r'(u) = r(u) + \frac{1-\alpha}{2d(v)} r(v)$
end for
return ppr' e r'

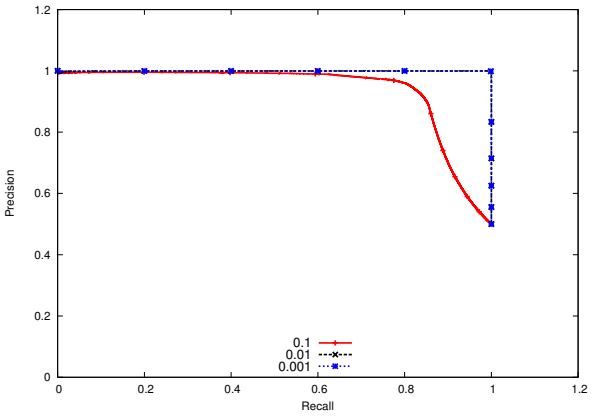
that this dependence on precision is linear: it is possible to show that the running time of the algorithm is $O(\frac{1}{\alpha\epsilon})$ and therefore, for fixed α, the running time is $O(\frac{1}{\epsilon})$. Note that this offers an interesting trade–off between speed and precision.

A second consequence of the choice of ϵ comes from the way the push–flow algorithm works. It can be shown that all vertices w whose frequency p_w in the stationary distribution is smaller than ϵ receive a trust of 0 from APPR. When APPR stops, nodes with a non–zero ppr value define a connected component around the source, while all vertices outside have zero trust.

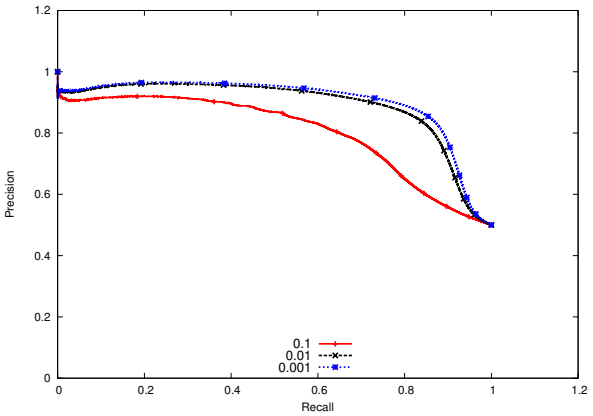
When ACL is computed with respect to the same seed with two values $\epsilon < \delta$, the non–zero portion of the ϵ–ranking is longer than the corresponding prefix of the δ–ranking. The surprising finding is that these two rankings, $u_1^\epsilon, \ldots, u_n^\epsilon$ and $u_1^\delta, \ldots, u_n^\delta$ are almost the same, as can be measured for instance using the Tau–Kendall distance (see Table IV). This is a very useful property: it says that if we want to identify quickly a set of trusted nodes, we can do so just by using a larger value of ϵ. Because the running time of the protocol is dependent on the values of α and ϵ and not the size of the graph, this allows ACL to effectively scale in situations where partial node rankings suffice.

ϵ \ δ	$= 10^{-4}$	$= 10^{-5}$	$= 10^{-6}$	$= 10^{-7}$
$= 10^{-3}$	0.84	0.83	0.82	0.82
$= 10^{-4}$		0.81	0.79	0.79
$= 10^{-5}$			0.73	0.73
$= 10^{-6}$				0.99

Table IV
TAU–KENDALL DISTANCE CORRELATION BETWEEN AN ϵ–RANKING AND A δ–RANKING FOR THE FACEBOOK SNAPSHOT. THE INDEX IS A REAL NUMBER BETWEEN $+1$ (PERFECT CONCORDANCE) AND -1 (REVERSE ORDER). A VALUE OF 0 INDICATES THAT ONE RANKING IS A RANDOM PERMUTATION OF THE OTHER. SIMILAR HIGH CORRELATION WAS OBSERVED FOR DIFFERENT SNAPSHOTS OF SOCIAL NETWORKS.



(a) $p = 0.01$

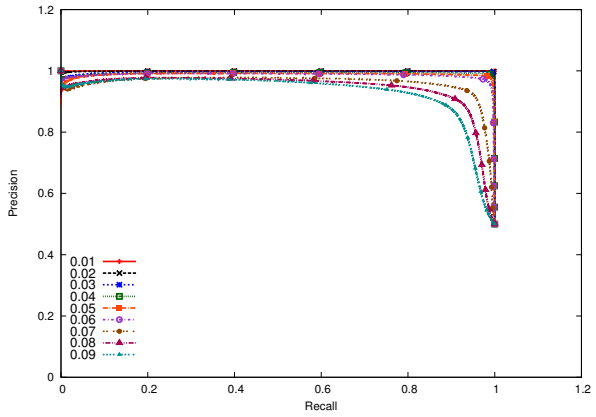


(b) $p = 0.10$

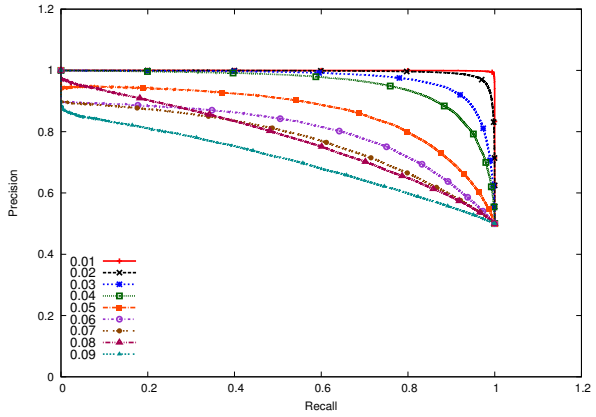
Figure 5. Impact of varying α. Precision vs Recall graph with Facebook–New Orleans data set under (a) a weak attack (edge density $p = 0.01$) and (b) a strong attack (edge density $p = 0.1$).

B. Comparative Evaluation

Our key question in evaluating ACL is to determine whether it expands the guarantees offered by today's social defense systems in two directions: (1) withstanding denser attacks; and (2) providing high quality sybil defense without



(a) ACL



(b) SybilLimit

Figure 6. The impact of varying the attack strength on (a) ACL on the original Facebook graph and (b) SybilLimit on the preprocessed and raw Facebook graph.

relying on the assumption that the entire graph is fast mixing (to avoid the need for preprocessing).

Method and environment: Viswanath et al. [39] observe that, despite their peculiarities, sybil defense schemes are based on the same fundamental principle—community detection—and produce highly correlated results. Hence, for the sake of clarity, the experiments we report compare ACL only against SybilLimit, which we use as the SybilLimit–like champion. Although SybilLimit performed better than its peers, our experiments with SybilGuard, SybilInfer and Gatekeeper returned qualitatively similar results.

The graphs we use to compare their performance are generated by subjecting social networks that we assume to include only honest nodes to the attack described in section II-B. We then run ACL and SybilLimit on the resulting graphs, rank the nodes using the same methodology discussed in Section III, and measure precision (the percentage of nodes in the prefix that are honest) and recall (the percentage of honest nodes that are in the prefix) from the perspective of 10 randomly chosen seeds. We report the average of the values we obtain.

We configure SybilLimit to have $1.5\sqrt{m}$ random walks of length $1.5\log(n)$. ACL is configured with $\alpha = 10^{-3}$ and ϵ sufficiently small to label every node in the attacked graph with non-zero weight. For DBLP $\epsilon = 10^{-7}$; for all other graphs $\epsilon = 10^{-6}$ suffices.

ACL tolerates denser attacks: Figure 6 shows the degree to which ACL and SybilLimit succeed in defending the Facebook graph when the attack strength, measured as the percentage p of attack edges in the graph, varies from $p = 0.01$ to $p = 0.1$. Note that, to respect the "operating range" of each protocol the results we report for ACL are obtained on the *original* Facebook graph while the results from SybilLimit apply to the *preprocessed* Facebook graph.

We observe that the ability of ACL to correctly classify nodes degrades gracefully as the attack increases in strength, remaining relatively high even when $p = 0.1$. Indeed, the selectivity of ACL under an attack of strength $p = 0.05$ is comparable to that of SybilLimit for an attack of $p = 0.01$. SybilLimit on the other hand becomes confused rather rapidly as the attack strength increases.

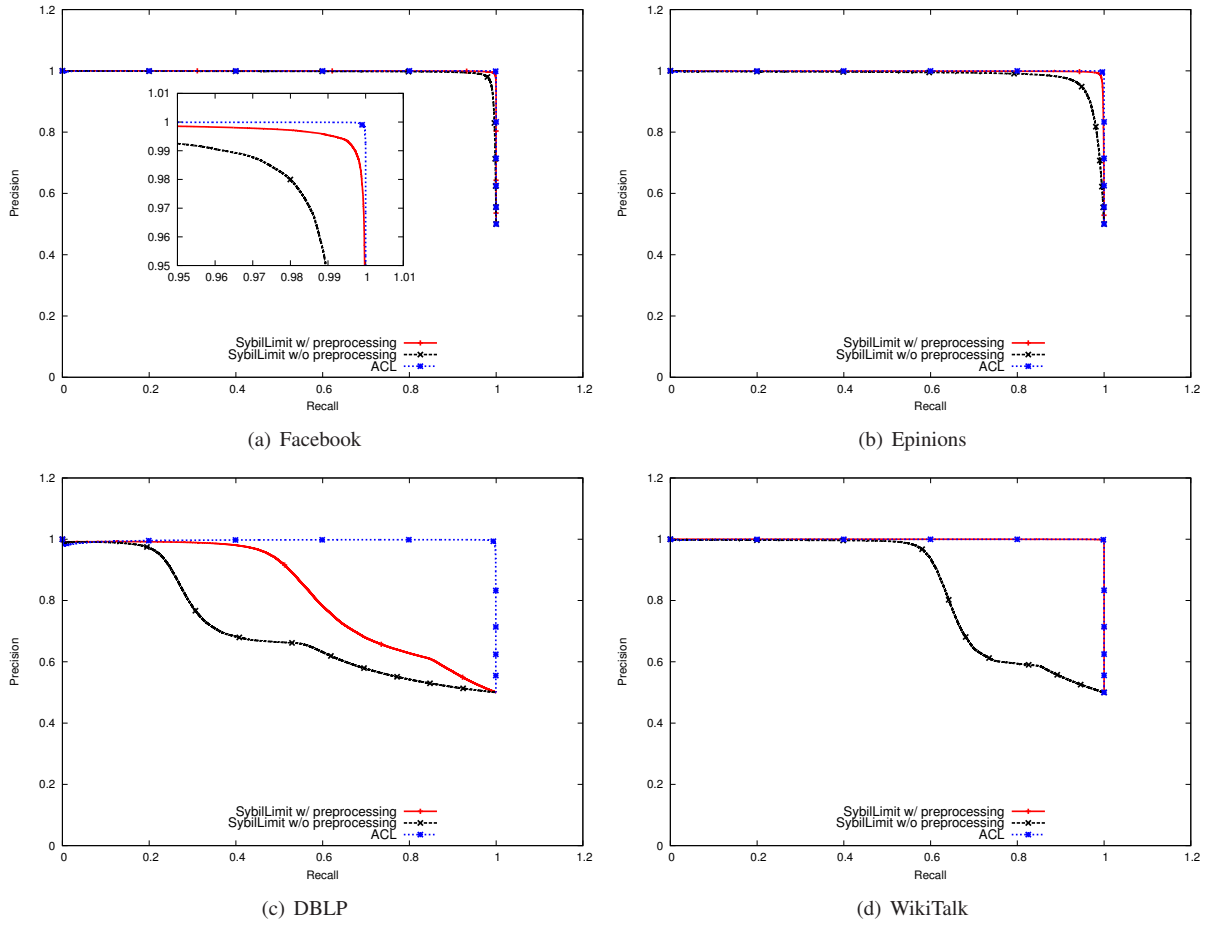
ACL does not need preprocessing: Figure 7 shows the protection offered by ACL and SybilLimit to the Facebook, DBLP, Epinions, and WikiTalk graphs for an attack where $p = 0.01$. For ACL, we report only results from the raw graph. For SybilLimit we report results from both the raw and preprocessed graphs.

Without preprocessing, ACL achieves high precision at high recall. SybilLimit's performance, on the other hand, is mixed. For Facebook, Epinions, and WikiTalk, SybilLimit provides excellent protection as long as the graphs are preprocessed. When the graphs are not preprocessed, the offered coverage degrades to varying extents. The degradation in coverage for Facebook is negligible; for Epinions the degradation is minor but noticeable.

SybilLimit performs poorly on DBLP with or without preprocessing, though preprocessing the graph does provide a significant boost. We speculate that this poor performance is the side effect of the relatively high mixing time observed by Mohaisen et al. [23].

A second attack model: In this section we compare the algorithms using an attack model widely used in the literature [10], [42]. The number of attack edges g is fixed, and random honest nodes are declared to be sybil until g attack edges are obtained. Then more sybil nodes are created from scratch until a total of γ sybils is reached. These γ sybils are then connected among themselves via a scale-free topology. In our attack we use the scale-free topology of Barabasi–Alberts, as in [42].

Figure 8 shows the results for our Facebook graph and $g = 50000$ and $\gamma = 10000$. ACL and Mislove are essentially perfect, outperforming all other algorithm (Gatekeeper, SybiLimit and SybilGuard). We also ran experiments with other graphs obtaining similar results.



(a) Facebook (b) Epinions

(c) DBLP (d) WikiTalk

Figure 7. The precision–recall tradeoffs for ACL and SybilLimit on DBLP, Epinions, Facebook, WikiTalk, with $p = 0.01$. Results for ACL are reported for the raw graphs. Results for SybilLimit are reported for both raw and preprocessed graphs.

C. Local vs Global detection

We have shown that ACL is very effective in practice to address Problem 1. Building a universal sybil defense system for community-structured networks, however, remains an open problem.

In a recently published paper Cao et al. [7] suggest to expand defensive coverage by relying on multiple trusted seed nodes instead of a single one. More precisely, suppose there are several trusted seeds evenly distributed among communities of honest nodes; it is then possible to merge the local ranking of the nodes to get a unified global ranking of the nodes in the network.

While effective in practice, the use of multiple seeds does not immediately lead to strong theoretical guarantees, even assuming that all seeds are honest nodes. For example, suppose we can prove, as it is typical for ACL, that a $1-o(1)$ fraction of the honest seeds will assign a negligible fraction of the overall score to sybil nodes and distribute the rest evenly across the honest region. There is always, however, a fraction of unlucky honest seeds for which such guarantees are impossible—e.g., seeds at the boundary between the honest and sybil regions. Unfortunately, because of the arbitrary nature of the sybil region, walks originating from these nodes might produce an unconstrained (and adversarial) probability distribution among the sybil nodes.

This is not only true for the ACL algorithm, but virtually for any sybil defense algorithm that relies on random walks and mixing time (see for instance [7], [45], [46]).

Unfortunately, it is not clear how an unlucky choice of seeds will affect the overall ranking. While lucky seeds will distribute evenly the score among honest nodes, unlucky ones might concentrate the score to a smaller, but still significant, subregion of the sybil graph, thus letting such nodes overtake the first positions of the ranking.

Despite these words of caution, the results obtained by Cao et al. [7] using multiple seed in real world scenarios are encouraging, and we believe this is a promising research direction.

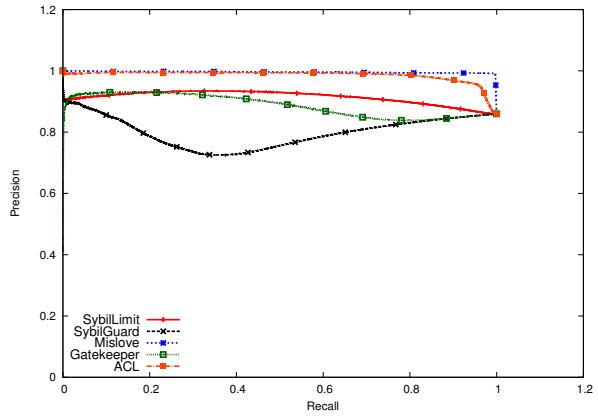


Figure 8. The precision of ACL and the other algorithms on Facebook graph with standard attack model with $g = 50000$ and $\gamma = 10000$.

D. Discussion

We have shown experimentally that ACL is extremely effective at identifying the community of a given honest seed and provided formal guarantees for the rankings it produces. To our knowledge this is the first time that formal guarantees are given for a community detection algorithm in the context of sybil defense. While we have shown that ACL can be used to effectively solve Problem 1, in the next section we will discover a sobering reality: all sophisticated state-of-the-art methods based on random walks, including ACL, are helpless against some of the simple, primitive sybil attacks that are encountered in deployed social networks.

VII. Avoiding the Maginot syndrome

Our appraisal in Section II of the resilience of different structural properties of social graphs indicated that leveraging the complementary notions of mixing time and conductance are the most promising line of defense against sybil attacks; furthermore, techniques based on this approach can provide impressive end-to-end guarantees. Yet one key question remains: how effective are these techniques against actual sybil attacks?

While data on sybil attacks in deployed social networks is not readily available, two recent papers have included experience reports that shed light on the types of attacks that occur in the wild.

Cao et al. report to have successfully used SybilRank to identify sybil users in the Tuenti social network [7]. They observe large clusters of sybil users in regular topologies (star, mesh, tree, etc.) that are connected to the honest communities through a limited number of attack edges. They also report that an unspecified fraction of the remaining accounts are sybil but to preserve confidentiality are unable to report on the number or characteristics of those accounts.

Yang et al.'s experience in analyzing the RenRen social network is significantly different [43]: they do not observe any large clusters of well-connected sybil nodes in turn connected to the honest sub-graph through a small set of attack edges, as would be expected by the sybil defense techniques we have surveyed; instead, they find isolated sybil nodes each connected to the honest sub-graph with a large number of attack edges.

The simple attack observed in the RenRen social network is devastating for conductance-based protocols. We simulated the attack on our Facebook graph and measured the probability that a randomly-chosen honest node be considered more trustworthy than a randomly-chosen sybil one by SybilLimit [45], SybilGuard [46], Mislove [39], Gatekeeper [35], and ACL. A probability of 1 corresponds to the ideal case in which every honest node is ranked higher than any sybil one; a probability of 0 indicates the reverse case; a random ranking correspond to 0.5 probability. In our results, every protocol performs poorly: the probability is 0.45 for SybilLimit, 0.44 for SybilGuard, 0.34 for Mislove, 0.49 for Gatekeeper, and 0.37 for ACL. The vulnerability of conductance-based techniques to an attack where each sybil node can create more than one attack edge is fundamental, as Yu et al. proved [45].

These experiences indicate that while today's socially-based sybil defenses are designed to provide the theoretically-best defense posture, they are also easily circumvented. Much like the real-life Maginot line. [6]

A. Defense in depth

To avoid this fate, we believe that effective sybil-defense mechanisms should embrace a strategy inspired by the notion of defense in depth [34]: rather than relying solely on techniques based on conductance, they should include a portfolio of complementary detection techniques. For example, Yang et al. observe [43] that it is possible to spot sybil nodes by tracking their clustering coefficient (see Section II) and the rate at which their requests of friendship are accepted, both of which in the RenRen graph are significantly higher for honest nodes than for sybils (in the case of the clustering coefficient, this is because a single sybil node that randomly issues friendship requests is unlikely to have many friends who are themselves friends with each other). As a rule of thumb, Yang et al. suggest to report as sybil those users whose friendship-request acceptance rate is less than 50% and whose clustering coefficient is below $1/100$. They report that this is sufficient to correctly identify more than 98% of the sybils, with a false positive rate of less than 0.5%. Note that, while these results sound impressive, they are not cause for unconditional celebration, as it is quite easy for a slightly more sophisticated adversary to circumvent both checks by issuing friendship requests to other sybil nodes under his control. But, at the very least, checks like these make the life of the attacker more difficult and prevent more sophisticated

[6] http://en.wikipedia.org/wiki/Maginot_Line

defenses to be trivially sidestepped. Indeed, they may even nudge the attacker, whether he likes it or not, towards the kind of attacks where conductance–based method can start to be effective. For instance, simply introducing a defense layer that monitors the rate of friendship acceptance introduces a bound (albeit loose) on the conductance of the cut between honest users and sybils.

In particular, assume that honest users accept sybil request with probability p and that the threshold of accepted requests below which a node is flagged as sybil is T. Then the following simple result holds (see [2] for the proof)

Proposition 3. *Sybil nodes, to not be detected, must create fewer than $p\frac{1-T}{T-p}$ of their edges as attack edges.*

So, for example, if honest users accept friendship requests with probability $p = 10\%$ and $T = 50\%$ (as in [43]), then each sybil node must have seven links to sybil nodes for every attack edge to avoid detection.

Proposition 3 bounds the conductance of the cut between honest and sybil nodes in the sense that whenever the sybil region has fewer edges than the honest region, the conductance of the cut is at most $2p\frac{1-T}{T-p}$.

While this bound is loose, it is encouraging that it can be obtained through a defense layer based on a fairly primitive measure such as the rate of friendship acceptance. We speculate that in the near future new defense layers based on advanced machine–learning and profiling techniques [33] will force a sybil attacker who wants to escape detection to generate sybil regions that resemble ever more actual social graphs, connected through a sparse cut of attack edges to the honest portion of the graph: in other words, exactly the scenario suitable for conductance–based sybil defense.

VIII. Conclusions

This work has traced the evolution of social sybil defenses from the seminal work of Yu et al [46] to the developments of the last several years [7], [10], [35], [45] to recent reports [7], [43] that detail their usage in practice.

We have identified two main trends in the literature. The first is based on random walk methods whose goal is to identify fast–mixing (sub)regions that contain the honest seed. The implicit assumption is that social networks under sybil attacks must exhibit a simple structure—a fast–mixing region of honest nodes connected via a sparse cut to the sybil region. We have seen how this initial simplified picture of the world has progressively become more nuanced, leading to methods based on random walks that are able to cope with a more complex world consisting of a constellation of tightly–knit, fast–mixing communities loosely connected among themselves and to the sybil region.

The other trend that we have discussed considers sybil defense as an instance of community detection. While we have revealed the limitation of this approach, we have been able to enucleate its core validity.

As we have shown with our discussion on Personalized PageRank, the two approaches can go hand in hand to yield more robust sybil defense protocols that are competitive with the state of the art. The discussion has highlighted the importance of the body of literature that studies foundational issues on random walks. As we have shown, both algorithms and useful conceptual tools can be distilled from it and successfully deployed in the context of sybil defense.

Despite their growing mathematical sophistication, we have also seen how sybil defense methods can perform poorly when confronted with some real–world attacks that exhibit a very primitive structure. We believe that the defense–in–depth approach that we have advocated as a response to this challenge can be facilitated by moving from the original vision of offering individual honest users decentralized and universal sybil defense [45], [46] towards defense techniques that assume that the defender has com–plete knowledge of the social graph topology [7], [43] and can deploy the kind of parallelizable implementations suitable for handling the large graphs of on–line social networks. In particular, social network operators are in a position to use machine learning techniques, user profiling, and monitoring of user activity to gain additional knowledge that can help them filter sybil attacks not well–suited for de–tection using techniques based on random walks, community detection, and their combination. Still, as attackers increase in sophistication, claims of a silver bullet should be met with healthy skepticism. As the arms race between attackers and defenders continues, it will be increasingly important that new defense mechanisms clearly state the kind of attack they aim to withstand, a landscape that too often is blurred.

Acknowledgements

We thank Bimal Viswanath and Alan Mislove for the code of Mislove's algorithm,Nguyen Tran for the Gatekeeper code, and Krishna Gummadi for his comments on an early draft. Lorenzo Alvisi is supported by the National Science Foundation under Grant No. 0905625. Alessandro Epasto is supported by the Google European Doctoral Fellowship in Algorithms, 2011. Alessandro Panconesi is partially sup–ported by a Google Faculty Research Award.

References

[1] Dblp. http://dblp.uni–trier.de/xml/, 2011.

[2] L. Alvisi, A. Clement, A. Epasto, S. Lattanzi, and A. Panconesi. Communities, random walks and social sybil defense. Technical Report TR–13-04, UTCS, 2013. http://wwwusers.di.uniroma1.it/~epasto/papers/sybil–tr.pdf.

[3] R. Andersen, F. Chung, and K. Lang. Local graph partitioning using pagerank vectors. In *FOCS*, 2006.

[4] S. Arora, S. Rao, and U. Vazirani. Expander flows, geometric embeddings and graph partitioning. *J. ACM*, 2009.

[5] A.–L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 1999.

[6] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda. All your contacts are belong to us: Automated identity theft attacks on social networks. In *WWW*, 2009.

[7] Q. Cao, M. Sirivianos, X. Yang, and T. Pregueiro. Aiding the detection of fake accounts in large scale social online services. In *NSDI*, 2012.

[8] A. Cheng and E. Friedman. Manipulability of pagerank under sybil strategies. In *NetEcon*, 2006.

[9] L. Cox and B. Noble. Samsara: Honor among thieves in peer-to-peer storage. In *SOSP*, 2003.

[10] G. Danezis and P. Mittal. Sybilinfer: Detecting sybil nodes using social networks. In *NDSS*, 2009.

[11] J. Douceur. The sybil attack. In *IPTPS*, 2002.

[12] S. Fortunato. Community detection in graphs. *CoRR*, abs/0906.0612, 2009.

[13] T. H. Haveliwala. Topic-sensitive pagerank: A context–sensitive ranking algorithm for web search. *IEEE Trans. on Knowledge and Data Engineering*, 2003.

[14] T. Leighton and S. Rao. Multicommodity max-flow min–cut theorems and their use in designing approximation algo–rithms. *J. ACM*, 1999.

[15] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Predicting positive and negative links in online social networks. In *WWW*, 2010.

[16] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *KDDWS*, 2005.

[17] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Statistical properties of community structure in large social and information networks. In *WWW*, 2008.

[18] C. Lesniewski-Laas. A sybil-proof one-hop DHT. In *SNS*, 2010.

[19] C. Lesniewski-Laas and M. F. Kaashoek. Whanau: A sybil–proof distributed hash table. In *NSDI*, San Jose, CA, 2010. USENIX Association.

[20] N. Margolin and B. N. Levine. Quantifying and discouraging sybil attacks. Technical report, UMass Amherst, 2005.

[21] A. Mislove, B. Viswanath, K. P. Gummadi, and P. Druschel. You are who you know: Inferring user profiles in online social networks. In *WSDM*, February 2010.

[22] A. Mohaisen, N. Hopper, and Y. Kim. Keep your friends close: Incorporating trust into social network-based sybil defenses. In *INFOCOM*, 2011.

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

[24] S. Oveis Gharan and L. Trevisan. Approximating the Ex–pansion Profile and Almost Optimal Local Graph Clustering. *ArXiv e-prints*, 2012.

[25] J. Pouwelse, P. Garbacki, D. Epema, and H. Sips. The bit–torrent p2p file-sharing system: Measurements and analysis. *Peer-to-Peer Systems*, 2005.

[26] D. Quercia and S. Hailes. Sybil attacks against mobile users: friends and foes to the rescue. In *INFOCOM*, 2010.

[27] Y. P. Reid Andersen. Finding sparse cuts locally using evolving sets. In *STOC*, 2009.

[28] M. Richardson, R. Agrawal, and P. Domingos. Trust man–agement for the semantic web. In *ISWC*, 2003.

[29] H. G.-M. Sepandar D. Kamvar, Mario T. Schlosser. The eigentrust algorithm for reputation management in p2p net–works. In *WWW*, 2003.

[30] A. Sinclair. Improved bounds for mixing rates of markov chains and multicommodity flow. *LATIN*, 1992.

[31] A. Sinclair and M. Jerrum. Approximate counting, uniform generation and rapidly mixing markov chains. *Inf. Comput.*, 1989.

[32] D. A. Spielman and S.-H. Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *STOC*, 2004.

[33] T. Stein, E. Chen, and K. Mangla. Facebook immune system. In *SNS*, 2011.

[34] M. Stytz. Considering defense in depth for software applica–tions. *Security Privacy, IEEE*, 2004.

[35] N. Tran, J. Li, L. Subramanian, and S. Chow. Optimal sybil–resilient node admission control. In *INFOCOM*, 2011.

[36] N. Tran, B. Min, J. Li, and L. Subramanian. Sybil-resilient online content voting. In *NSDI*, 2009.

[37] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi. On the evolution of user interaction in facebook. In *WOSN*, 2009.

[38] B. Viswanath, M. Mondal, A. Clement, P. Druschel, K. Gum–madi, A. Mislove, and A. Post. Exploring the design space of social network-based sybil defenses. In *COMSNETS*, 2012.

[39] B. Viswanath, A. Post, K. P. Gummadi, and A. Mislove. An analysis of social network-based sybil defenses. In *SIGCOMM*, 2010.

[40] L. Von Ahn, M. Blum, N. Hopper, and J. Langford. Captcha: Using hard ai problems for security. *Advances in Cryptol-ogy—EUROCRYPT 2003*, 2003.

[41] D. J. Watts and S. Strogatz. Collective dynamics of 'small–world' networks. *Nature*, 339, 1998.

[42] W. Wei, F. Xu, C. C. Tan, and Q. Li. Sybildefender: Defend against sybil attacks in large social networks. In *INFOCOM*, 2012.

[43] Z. Yang, C. Wilson, X. Wang, T. Gao, B. Y. Zhao, and Y. Dai. Uncovering social network sybils in the wild. In *IMC*, 2011.

[44] H. Yu. Using social networks to overcome sybil attacks. *ACM SIGACT News*, September 2011.

[45] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao. Sybillimit: A near-optimal social network defense against sybil attacks. In *OAKLAND*, 2008.

[46] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. Sybil–guard: Defending against sybil attacks via social networks. *IEEE/ACM Transactions on Networking*, 2008.

[47] M.-C. Yuen, I. King, and K.-S. Leung. A survey of crowd–sourcing systems. In *IEEE Socialcom*, 2011.

[48] J. O. P. Zoltán Gyongyi, Hector Garcia-Molina. Combating web spam with trustrank. In *VLDB*, 2004.