On the State of OSN-based Sybil Defenses

David Koll
University of Göttingen
Göttingen, Germany
koll@cs.uni-goettingen.de

Jun Li
University of Oregon
Eugene, USA
lijun@cs.uoregon.edu

Joshua Stein
University of Oregon
Eugene, USA
jgs@cs.uoregon.edu

Xiaoming Fu
University of Göttingen
Göttingen, Germany
fu@cs.uni-goettingen.de

***Abstract*—A Sybil attack can inject many forged identities (called *Sybils*) to subvert a target system. Because of the severe damage that Sybil attacks can cause to a wide range of networking applications, there has been a proliferation of Sybil defense schemes. Of particular attention are those that explore the online social networks (OSNs) of users in a victim system in different ways. Unfortunately, while effective Sybil defense solutions are urgently needed, it is unclear how effective these OSN-based solutions are under different contexts. For example, all current approaches have focused on a common, *classical* scenario where it is difficult for an attacker to link Sybils with honest users and create attack edges; however, researchers have found recently that a *modern* scenario also becomes typical where an attacker can employ simple strategies to obtain many attack edges.**

In this work we analyze the state of OSN-based Sybil defenses. Our objective is not to design yet another solution, but rather to thoroughly analyze, measure, and compare how well or inadequate the well-known existing OSN-based approaches perform under both the classical scenario and the modern scenario. Although these approaches mostly perform well under the classical scenario, we find that under the modern scenario they are vulnerable to Sybil attacks. As shown in our quantitative analysis, very often a Sybil only needs a handful of attack edges to disguise itself as a benign node, and there is only a limited success in tolerating Sybils. Our study further points to capabilities a new solution must possess; in particular, in defense against Sybils under the modern scenario, we anticipate a new approach that enriches the structure of a social graph with more information about the relations between its users can work more effectively.

***Keywords*—Sybil; Sybil detection; Sybil tolerance; OSN**

I. Introduction

In the past decade a new kind of malicious behavior has been extensively studied. Introduced as the *Sybil attack* [1], the attacker forges multiple identities—hence the name Sybil—in order to subvert a system. For instance, a large number of Sybils can cast manipulated votes to rig the outcome of a voting system. Sybil attacks are not only a threat in theory, but have also been observed in many real-world networks [2].

Among the many schemes that researchers have proposed to defend a target system against Sybil attacks, the most popular approaches in recent years leverage the online social networks (OSNs) of users in the target system, and inspect the structure of their social relations [3]–[8]. The main hypothesis is that OSN identities controlled by the attacker will have difficulties establishing social relations (i.e., edges in the OSN graph) with honest users. These approaches reason that there must be some sort of trust between both ends of a social relation, which should not be the case for relations between honest users and forged identities. Thus, although there may be many relations both among the Sybil identities themselves and among the benign users themselves, there should be very few connections from Sybils to the community of benign users. We call these isolated links, i.e., links between a Sybil node and a benign node, **attack edges**. As a result, the social network graph will have a clear partition between a Sybil region and a non-Sybil region, except for the few attack edges between them. In other words, there is supposed to be a small cut between both regions, as removing a small number of attack edges would result in two disconnected graphs [9]. To summarize, OSN-based Sybil defenses have been predominantly investigating the following scenario:

Although an attacker can create an arbitrary number of Sybil identities, she cannot establish an arbitrary number of attack edges to the non-Sybil identities.

However, recent research has identified a rich set of behaviors of both attackers and honest users that invalidate the above hypothesis. The most worrying observation is that attackers can easily create links to benign users by simply sending out link-establishing requests (e.g., friendship requests on Facebook). The success rate can reach an astonishing 90% for specifically forged profiles or engineered bots [2], [10], [11]. At the same time honest users can also be easily tricked to establish the link and even initiate communication with forged identities [12], [13]. In addition, contrary to what all Sybil defense approaches have suspected, Sybils do not create numerous links mostly between themselves and thereby form a dense Sybil community; instead, almost 75% of links originating at a Sybil are connected to honest users and thus attack edges [2]. We therefore refer to the scenario that matches the assumptions made by current defense approaches as the *classical* scenario, and propose a *modern* scenario:

Rather than predominantly connecting with other Sybils, an attacker is able to establish an increasing amount of social relations to benign users, and becomes more and more integrated within the community of benign users.

Hence, it is uncertain how well existent OSN-based Sybil defense solutions, which were designed toward the classical scenario, would perform under the new, modern scenario. While it is critical to have effective Sybil defense solutions, it is unclear what help and how much help we can obtain from the existent solutions. It is thus also unclear whether a new solution is needed or not; and if so, what capabilities it should have that existent solutions do not.

In this paper, our focus is to systematically analyze, measure, and compare how well or inadequate all existent OSN-based approaches perform, with the goal to qualify and quantify the strengths and weaknesses of these approaches. We investigate *two* classes of Sybil defense approaches: Sybil detection approaches—which try to detect Sybil nodes and exclude them from participation in a target system, and Sybil tolerance approaches—which try to limit the impact of Sybils present in the system. The former includes SybilGuard/SybilLimit [3], [14], SybilInfer [6], GateKeeper [5], and SybilRank [4]. The

ISBN 978-3-901882-58-6 © 2014 IFIP

latter includes Ostra [8] and SumUp [7]. Given that a Sybil node may obtain more attack edges than traditionally assumed, in our analysis we pay particular attention to what a Sybil node has to achieve in order to make itself indistinguishable from honest nodes—and therefore disguise itself from the defense scheme. We investigate different attack strategies where applicable, and for every Sybil defense solution we quantify the cost for the attacker (e.g., the number of attack edges to create) to thwart the solution.

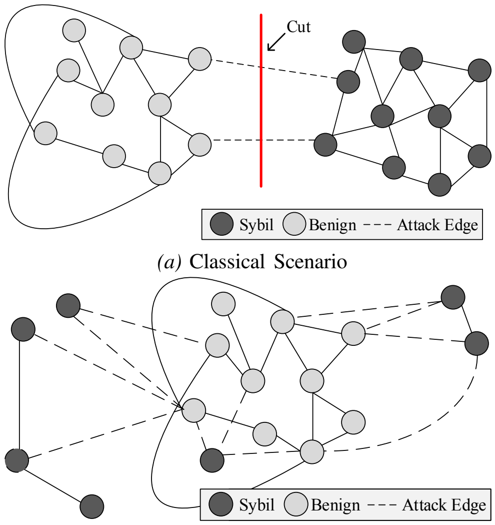
We find that current OSN-based Sybil defense approaches of both classes have difficulty identifying the attack edges and the Sybil nodes in the modern scenario. Although they perform well in the classical scenario, surprisingly little effort is needed to deceive any existent defense scheme. Specifically, we find that in many schemes a Sybil node only needs to create one or two attack edges to random honest nodes in order to successfully masquerade as a benign node. The attacker can further reduce the required effort if she follows more intelligent attack strategies that exploit particular weaknesses in a given defense scheme.

Lastly, our analysis of existent solutions further provides insights on new solutions. In particular, rather than solely exploiting the structural properties of OSN graphs, as done in existent solutions, we anticipate a new approach would be more promising if it enriches the structure of a social graph with more information about the relations between its users, especially if a new approach aims to function under both the classical scenario and the modern scenario.

II. Related Work

Prior to our work, there have been other studies analyzing OSN-based Sybil defenses. In the work closest to ours, Viswanath et al. revealed that Sybil detection schemes could be abstracted to community detection algorithms [9]. In another study, Yu provided a concise summary of existing Sybil defenses and described their working principles under the classical scenario [15]. Viswanath et al. further explored the design space for OSN-based defenses [16], and Boshmaf et al. presented a framework for the evaluation of graph-based Sybil detection [17]. Our work is substantially different from such studies: First, we analyze the performance of Sybil defenses in a completely new scenario, i.e., the modern scenario (Sec. I). For instance, research in [9] requires Sybils to reside outside a distinguishable community, which is not true in the modern scenario. Second, our study includes more recent defense schemes that were not studied before. In particular, we study approaches that can handle **modular** OSN graphs (i.e., those with multiple distinct benign communities) [4], which did not exist at the time of previous studies. Third, in answering under what conditions a Sybil attacker can subvert a defense system, our analysis is not only qualitative, but also quantitative in that for each defense system we provide concrete results under what conditions it may fail. This complements studies such as [17], which do not analyze particular approaches, but rather guidelines on how they should be evaluated or designed.

Besides research focusing on the analysis of Sybil defenses, some works further rethought social graph properties with regards to Sybil defense. Alvisi et al. [18] pointed out that Sybil defense schemes tend to assume a minimal cut between Sybils and honest users, which however may not hold. The authors also analyzed SybilLimit [3] and show the sensitivity of the scheme to the existence of this cut. Our work goes one step further: We provide an in-depth analysis of *all* relevant Sybil defense approaches and their performance. Unlike Alvisi et al., we also consider Sybil tolerance approaches (we will list and categorize our subjects under investigation in Section IV), whose working principles greatly differ from those of Sybil detection approaches (such as SybilLimit [3]).



(a) Classical Scenario

(b) Modern Scenario

Fig. 1: Juxtaposition of Scenarios.

Researchers have also attempted to alter social graphs to help Sybil defense schemes work better. Mittal et al. proposed to introduce noise to a social graph and thus hide its true structure to attackers [19]. However, doing so only prevents the attacker from knowing the exact structure of the graph, but does not constrain her in creating attack edges at all.

III. Revisiting Assumptions for Sybil Defense

In this paper we emphasize that attackers employing a Sybil attack operate differently than previously thought. We now explain *why* we propose a new attacker behavior and *what* this behavior looks like. This section lays the foundation of our in-depth study reported in the later parts of this paper.

A. Troubling Observations

Recently researchers have discovered a multitude of behaviors of both honest and Sybil nodes that have a significant impact on the structure of a social graph. These behaviors can be observed across different OSNs, independent from their working principle and targeted audience [2], [11]–[13]. The most salient observation is that a large fraction of OSN users are credulous and attackers are easily able to create links to honest nodes. Yang et al. found close to 11 million attack edges distributed among roughly 65,000 Sybil nodes—an average of 170 attack edges per Sybil [2]. Moreover, Bilge et al. show that 50% of users with whom the attacker could create an attack edge also clicked on URLs which the attacker sent in a message [11]. This implies that honest users even *trust* the forged profiles to certain extent—even though the content of the message was possibly malicious (e.g., a link to malware).

It is also easy to trick benign users into sending link-establishing requests to Sybils [12], [13]. Using simple attacks (e.g., manipulating the friend recommendation scheme on Facebook), an attacker can obtain hundreds of friend requests *per day* with a single fake profile. Sybils are thus able to obtain thousands of attack edges to benign users without initiating any contact. Furthermore, once a Sybil has established some attack edges to honest users, it will be recommended to more users due to the existence of mutual friends, thus increasing the number of attack edges constantly. Sridharan et al. show that spammers on Twitter are most effective in tricking benign users into following them, i.e., creating a link to attackers [13]. Even with a simple attack, one third of the spamming accounts could accumulate more than 100 honest followers.

B. Modern Scenario vs. Classical Scenario

As a result of these findings, we propose a new structure of the OSN graph than assumed by previous works, with the following modifications: (i) the number of attack edges k increases; and (ii) Sybils create most links to honest nodes, not to other Sybils [2]. Figure 1 shows a juxtaposition of the original classical scenario (Figure 1a) and the modern scenario with such modifications (Figure 1b). The modern scenario has two major distinguishing features:

1) No minimal cut exists. Figure 1a provides a clear distinction between a benign region and a Sybil region based on a small minimal cut in social graph. A minimal cut of a graph is a cut whose cutset (i.e., the set of edges which have to be removed to partition the graph) has the lowest number of edges among all cutsets. In the modern scenario, we do not assume the existence of such a minimal cut between the honest and the Sybil nodes any more. (Note that such cuts may exist between communities sparsely connected to each other [4].)

Some approaches abstract the existence of a minimal cut in a graph to the *mixing time* of the graph [15]. The mixing time indicates how fast a random walk on an OSN graph G approaches stationary distribution. A slow mixing time means that a random walk needs to be long in order to reach the stationary distribution. Defense schemes then compare the mixing times of different subsets of G. If G contains a minimal cut between honest and Sybil nodes, it is supposed to have a slower mixing time than the subset of honest nodes, which is well connected [15]. On the other hand, if there is no minimal cut, the mixing time for G will then be faster. Applied to the modern scenario, the mixing times of different subsets of G are not easily distinguishable anymore.

2) There is not a single, densely connected Sybil community. In the classical scenario, Sybils form a single community, which is densely connected internally. Since Sybils create links to honest nodes in larger counts than to other Sybils, we advocate that Sybils do not form any specific community structure.

IV. Analysis of OSN-based Sybil Defenses

In this section, we analyze the working principle and effectiveness of the well-known Sybil defense approaches under the modern scenario. We investigate both Sybil detection approaches and Sybil tolerance approaches. In particular, we provide a qualitative analysis of the weaknesses of these approaches. We analyze and compare these approaches quantitatively in Section V.

A. Sybil Detection Approaches

To identify Sybils in an OSN graph G, a primary methodology has been leveraging the random walk on the graph, starting at an *a priori* known trusted node. The main idea is that Sybil nodes experience a *low reachability* from the trusted node because it is unlikely for a random walk originating at the trusted node to cross one of the few attack edges. Instead, the random walk is expected to stay within the honest region with very high probability. For instance, consider a random walk starting at node S_1 in Figure 2. Such a walk has a low probability of traversing the attack edge to the Sybil community attached to node A.

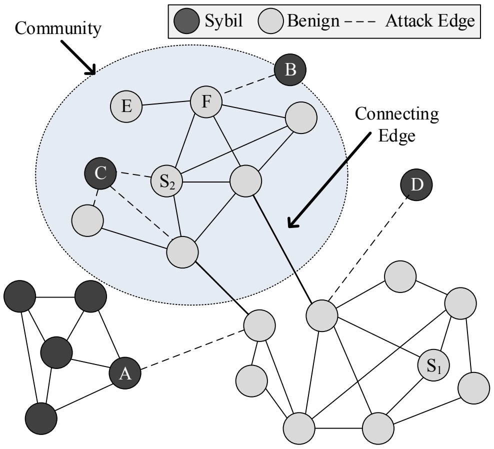


Fig. 2: An exemplary graph with Sybil nodes in different scenarios.

Different methods are employed in leveraging the random walk concept. SybilGuard/SybilLimit use *random routes*, a random walk modification that uses pre-computed random routing tables to determine the next hop of a random walk. SybilInfer conducts random walks—called *traces*—from all nodes, where a particular *a priori* honest node computes the probability that a subset G' of G is composed of entirely honest nodes based on the mixing time of the subset (Section III-B). SybilRank relies on the landing probabilities of short random walks that start from a non-Sybil node.

The only exception is GateKeeper. Rather than using random walk, it employs a breadth-first-search (BFS) ticket distribution strategy to detect Sybils. However, like approaches using random walk, it also exploits the *low reachability* of the assumed Sybil region from the honest region. The idea behind GateKeeper is that since Sybils are not as well connected to the ticket sources, Sybils will obtain substantially less BFS distributed tickets than benign nodes.

All these approaches were designed primarily toward the classical scenario. We now discuss the impact of the modern scenario on each of these detection schemes.

1) SybilGuard/SybilLimit: Because of the similarity of SybilGuard and SybilLimit (the latter is the successor of the former), we study these two approaches together.

Working principle. In SybilGuard, the classification of nodes as honest or Sybil is based on intersecting random routes (i.e., modified random walks). If there are only a few attack edges, it is highly unlikely that random routes from a Sybil will intersect with those from a benign node, causing the Sybil to be unlikely accepted by the benign node. SybilLimit executes a slightly modified SybilGuard protocol multiple times. Route intersections are no longer related to nodes on the routes, but to the tails (i.e., the final traversed edge before a route halts). Each of these routes is also shorter, leading to less possibilities for attackers to fake random routes that cross attack edges. For a Sybil to be accepted by SybilLimit, it needs to come up with one intersecting tail with a *verifier*.

Effectiveness analysis. SybilGuard provides guarantees on the number of Sybils admitted per attack edge. If the number of attack edges k in a system with n benign nodes is at most $O(\sqrt{n}/\log n)$, SybilGuard will admit at most $O(\sqrt{n}\log n)$ Sybil nodes *per attack edge*. If however k increases, SybilGuard is not able to limit the number of Sybil nodes at all [3]. SybilLimit improves on SybilGuard to the point where it accepts at most $O(\log n)$ Sybils *per attack edge*, no matter how many attack edges exist. It does not suffer from SybilGuard's limitation that the number of attack edges must stay below a certain threshold to have an effect. Nonetheless, SybilLimit was designed for a particular scenario: A densely connected Sybil community connected to the honest region by only a *few* attack edges. In this (classical) scenario admitting $O(\log n)$ Sybils per edge is a good effort, since many more Sybils may be connected to the honest region by these edges only (see Figure 1a or node A in Figure 2). However, in an inversion of the argument, this also means that every Sybil that can obtain a single attack edge can be admitted with high probability.

2) SybilInfer: **Working principle:** SybilInfer assumes OSN graphs to be fast-mixing, which is a dubious assumption in itself [3], [20]. The mixing time of a graph $G = (V, E)$ is defined by how fast a modified random walk in G—called a *trace*—reaches the stationary distribution [20]. The basic principle of SybilInfer is that with a small number of attack edges k—i.e., the existence of a small cut in G—the mixing time of G is slower than the mixing time of just its benign region. The reason is that traces originating at a benign node are more likely to remain in the benign region than to traverse one of the few attack edges and end on a Sybil node.

Effectiveness analysis: SybilInfer would work well if k is indeed small *and* the benign region is fast-mixing. However, if the Sybil nodes become more integrated into the graph, the cut will become less distinct (consider nodes B and C in Figure 2). The degree to which the cut is detectable also depends on the structure of the Sybil region: In SybilInfer, an attacker should not introduce many interconnected Sybils (e.g., node A in Figure 2), as otherwise all attacker nodes can be detected due to a slower mixing time of G. The attacker will be most successful in disguising Sybil nodes as benign nodes with a sparse community structure and many attack edges, an attacking behavior which is observed in the modern scenario.

3) SybilRank: **Working principle:** Like SybilLimit and SybilInfer, SybilRank also employs random walk as a means to detect Sybil nodes. SybilRank uses $\log n$ power iterations [21] to compute the probability that an early-terminated random walk would land on a node. The higher the probability is, the more likely the node is benign.

The random walk starts at a seed inside a Louvain-detected [22] honest community, and each major community in the OSN can have a manually chosen seed that is guaranteed to be benign. This method allows SybilRank to deal with the highly modular OSN graph structure. Under the assumption that there are few attack edges connecting Sybil nodes with an honest community, a short random walk is unlikely to finish at a Sybil node since the traversal of an attack edge is needed. Consider node A in Figure 2. As there is only one attack edge towards A, the random walk from seed S_2 is unlikely to land on any node within the Sybil community around node A.

Effectiveness analysis: SybilRank heavily relies on the number of attack edges that Sybil nodes can establish (it admits $O(\log n)$ Sybils per attack edge), but is roughly independent of the Sybil community structure. Consider Sybil nodes A and D in Figure 2, where A is part of a Sybil community but D is attacking as a single node. Since a random walk only has one attack edge to traverse to reach either A or D, both A and D will obtain a low ranking, and are thus likely to be categorized as Sybil nodes.

If the number of attack edges k increases, random walks will be more likely to land on Sybils adjacent to attack edges, thus increasing the ranking of these Sybils and even mislabeling them as benign nodes. The Sybil node C in Figure 2, for example, is completely disguised in the non-Sybil community. In fact, based on random walks starting at S_2, node C will even receive one of the highest rankings due to its position in the graph.

Finally, the introduction of trusted seeds enables more sophisticated attacks. By putting one trusted seed in each major honest community (e.g., S_1, S_2 in Figure 2), honest users will be ranked higher than Sybils since they are well connected to those seeds, whereas Sybils have to rely on few attack edges. However, if Sybils can create links to nodes near the seed, they can increase their rankings. Although the seeds themselves might be extremely careful with accepting link requests, nodes near them may be not.

4) GateKeeper: **Working principle:** In contrast to all previously discussed approaches, GateKeeper does not (directly) leverage random walks for Sybil detection. Similar to SybilRank, a central authority (called **admission controller**) selects a number of seeds (called **ticket sources**). Each ticket source obtains a number of tickets from the admission controller. The ticket source then distributes the tickets equally among its neighbors, and each neighbor again distributes the tickets equally among its neighbors, and so on. To be admitted into the system, a node must obtain tickets from a fraction of the ticket sources (in their experiments, the GateKeeper authors require a node to collect a ticket from 20% of all sources). The reasoning here is that if Sybils are only connected to the honest region by very few attack edges, they will rarely obtain tickets from the requisite f_{admit} fraction of sources, whereas honest users will easily do so.

Effectiveness analysis: GateKeeper is strongly dependent on the number of attack edges k, and accepts $O(\log k)$ Sybils per attack edge, improving from the magnitude of $O(\log n)$ of the previous schemes. If the Sybil community is well-connected among itself and there are few attack edges (see Figure 1a), the system will only admit $k \log k$ Sybils from the Sybil region. The more attack edges (i.e., the further k is increased), the more Sybil nodes will be admitted. In the case of our modern scenario, it becomes easier for Sybils to obtain the required fraction of tickets: each Sybil owns more attack edges and—similar to SybilRank—may be able to place them within a small distance to a ticket source.

B. Sybil Tolerance Approaches

In contrast to Sybil detection approaches, Sybil tolerance schemes aim at limiting the influence of Sybils that may reside in a system without being detected. For instance, these Sybils could—if there are no countermeasures—flood the system with

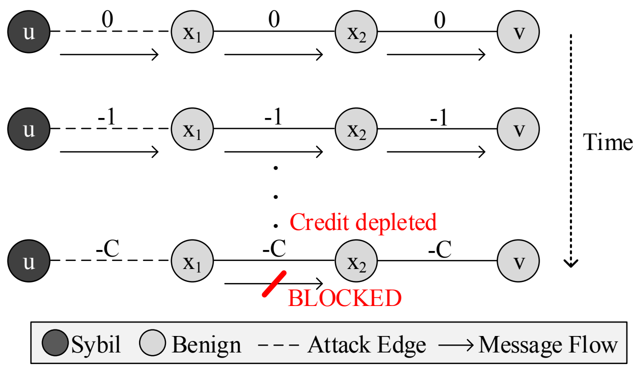


Fig. 3: Credit reduction in Ostra due to spam. As one credit is reduced for every spam, eventually every link between u and v—including benign links—has no available capacity for message delivery.

spam. Another major difference between both classes is that Sybil tolerance approaches are usually designed for a particular purpose, while Sybil detection approaches provide a more universal solution.

We focus on two major Sybil tolerance approaches, Ostra and SumUp. Both schemes are built upon a credit network where each link is granted a certain capacity. Nodes can exchange messages (Ostra) or cast votes (SumUp) only if they find a path with enough credit on every link of the path.

1) Ostra: **Working principle:** Ostra aims to limit spam issued by malicious users in a social network. It assigns credits to links between users, where each link has two credit values, one for each direction. If Ostra finds a path with available credit from a sender to its receiver, the sender's message can be sent; otherwise, the message will be blocked. As a message traverses along a path, starting at the source of the message, credit will be deducted from each traversed link in the direction of message transmission, while the same amount of credit is added in the opposite direction. If the recipient classifies the received message as wanted (i.e., not spam), Ostra will reverse the credit operations previously performed, such that only messages classified as spam will have an effect on the credit available on each link. The main idea limiting the influence of Sybils is that the classification of messages will quickly deplete the capacity on attack edges, leaving Sybils unable to distribute any spam afterwards. As benign users may inadvertently send out unwanted messages from time to time, Ostra also provides a mechanism to forgive a link for retrieving spam over it. That is, Ostra increases the credit on each link periodically, even if it has been depleted before.

Effectiveness analysis: In Ostra, more attack edges will lead to more spam that Sybils can send. Additionally, the credit forgiving scheme effectively makes sure that an attack edge never dies completely. Furthermore, sending spam reduces not only the credit on an attack edge of a spammer, but also the credits on the entire path to the destination. For instance, in Figure 3, with each spam message sent from a Sybil node u to an honest node v that traverses the path $p = u, x_1, x_2, v$, Ostra will penalize benign edges (x_1, x_2) and (x_2, v) as well. If there is no other path, eventually it may not be possible for x_1 to send a regular message to x_2 or v anymore, as the capacity C has been depleted. An intelligent attacker may therefore be able to isolate some honest nodes by depleting credits on many edges in the network. Ostra suggests honest users obtain "a sufficient number of trust links" to reduce the probability of becoming isolated, however this also lowers the bar for the attacker, as users are even more willing to accept link requests.

Even worse, the attacker could specifically target the few links that connect the communities in a highly modular OSN graph G. Consider Figure 2 again. Sybils A and D can block any communication towards the community on top if each can send their spam toward S_2 over one of the two links for reaching the community.

2) SumUp: **Working principle:** SumUp aims at limiting the number of bogus votes that Sybil nodes can cast in a system. It chooses an *a priori* trusted vote collector, which then distributes tickets in a BFS manner downstream along the OSN graph. Every node will keep one ticket, and distribute the remaining tickets equally among its next BFS-level neighbors. The capacity of every link is set to be the number of tickets distributed along the link plus 1. SumUp defines all links with a capacity larger than one to be within the *voting envelope*. The main idea is to keep Sybil nodes outside of the envelope and limit the voting capacity per attack edge to one.

SumUp employs two main mechanisms to modify an OSN graph and reduce the impact of attack edges. Before distributing tickets, SumUp employs a **pruning mechanism** to reduce the number of attack edges available to Sybils (therefore limiting their attack capability) and speed up vote computation with fewer edges. Basically, the number of edges from a node at a BFS-level of $i+1$ can only have at most d edges going to BFS-level i, where the vote collector is at BFS-level 0. It also has a **feedback mechanism**: the vote collector can provide negative feedback to paths through which bogus votes have been cast. Once an edge has too much negative feedback, it is eliminated and no vote can be cast along that edge any more.

Effectiveness analysis: As in Ostra, the damage from Sybil nodes will be directly proportional to the number of attack edges. Moreover, more attack edges would lead to a higher likelihood for a Sybil to be close to a vote collector (e.g., Sybil node C in Figure 2 would be close to S_2). As nodes near the collector can issue more tickets, such a Sybil would gain an increased capacity. Furthermore, as the number of attack edges increases, SumUp's feedback mechanism may penalize links adjacent to benign nodes too, causing collateral damage to benign edges while penalizing attack edges (similar to Ostra). If there are a sufficient number of Sybils with a path through a benign node, it may even cause all edges of the node to be eliminated. To mitigate this negative effect, after SumUp removes certain edges due to penalties caused by Sybil votes, it reintroduces pruned edges into the graph to replace the penalized edges in order to maintain the required number of incoming edges per node. The reintroduced edges, however, could be attack edges, leaving the efficacy of this feature questionable.

V. Are OSN-based Sybil Defenses Still Working?

After providing a qualitative analysis of the weaknesses of Sybil defense approaches under investigation in Section IV, we now conduct a quantitative study of these approaches. We answer the question of how severe these weaknesses are, whether or not OSN-based Sybil defense schemes can still work, and what is the cost for the attacker (e.g., the number of attack edges to create) to thwart a defense solution.

A. Evaluation Methodology

We implemented all aforementioned Sybil defense approaches and simulated their behavior when faced with different attack strategies. Our simulations are based on both a synthetic graph (1000 honest nodes with 2048 edges between them) and a real-world Facebook graph (63,731 nodes with 3,646,662 edges between them). The degree distribution of the synthetic graph follows the OSN-typical power-law distribution [23], which we can also observe in the Facebook graph. Our results are very similar for both datasets; for simplicity we report results for the Facebook set unless stated otherwise.

In order to prevent biased results due to a lucky or unlucky attack edge placement, we simulated 100 different attack edge placements for each *parameter setting*. As parameters, such a setting most importantly includes a varying number of attack edges and a strategy, in which the attack edges are placed. By default, we evaluate each approach with the attacker placing attack edges completely random. However, for some approaches, the attacker can gain an advantage by placing edges close to specific nodes, as explained before. Apart from these parameters, we stick to the parameters of the original evaluation of each approach if possible.

To ensure that we only evaluate the change in the structural properties of OSN graphs, we do *not* allow the attacker to deviate from the protocol of an approach being studied. For instance, in SumUp, a Sybil node obtaining a number of tickets will not try to favor other Sybils but follow the SumUp design and distribute tickets further downstream, even if the recipients are honest nodes. We also do *not* allow Sybils to be selected as seeds where applicable.

B. Sybil Detection Approaches

Our goal for evaluating Sybil detection approaches is to find out what an attacker needs to achieve in order to disguise a Sybil node. We ask two main questions: (1) *How many attack edges does a Sybil node need to establish in order to disguise itself as an honest node?* (2) *Does the location of Sybils on an OSN graph make a difference?*

To determine whether a Sybil detection approach is still able to distinguish between Sybil and honest nodes, we compare the relative performance of both classes of nodes in each detection scheme. In an ideal detection approach, all benign users should perform far better than all the Sybil nodes, thus leading to a clear distinction between both classes without any false positives or negatives. We call the ability of each scheme to differentiate between Sybils and honest nodes its **distinguishing ability**.

1) SybilLimit: The distinguishing ability of SybilLimit depends on the number of intersecting tails that a Sybil has with a verifier. Recall that in the original SybilLimit design, a Sybil node only needs *one* tail to intersect with that of a verifier in order to be verified. Our experiments revealed that a Sybil can become verified with high probability if it can place one attack edge to a random honest node. This is not hard to achieve given that $O(\log n)$ Sybils could be admitted per attack edge (Section IV-A1).

However, with some modification, SybilLimit might still be able to distinguish honest nodes from Sybil nodes by simply looking at the number of intersecting tails. In particular, the worst performing benign node might obtain significantly more intersecting tails with the verifiers than the best performing Sybil. Therefore, we ask how many attack edges a single Sybil node would have to create, which we denote k, in order to be indistinguishable from a benign node in terms of the number of intersecting tails.

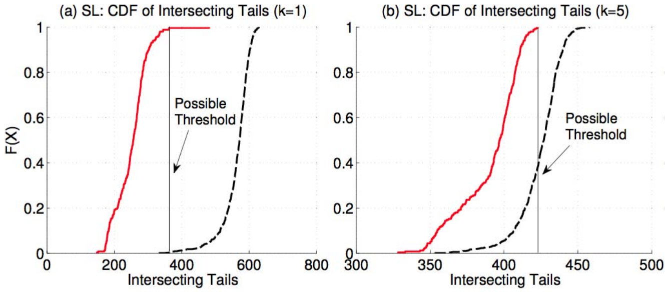


Fig. 4: Performance of SybilLimit (SL). k is number of attack edges per Sybil.

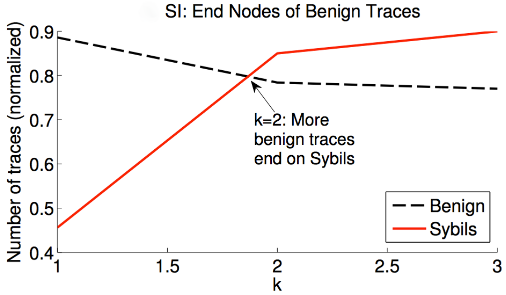


Fig. 5: Performance of SybilInfer (SI). k is number of attack edges per Sybil.

Figures 4a and 4b show the CDF of the number of intersecting tails with a verifier in SybilLimit. When a Sybil node can obtain one randomly placed attack edge (Figure 4a), the distinguishing ability of SybilLimit remains good. As the number of attack edges increases, its distinguishing ability is reduced. We exemplify the results with 5 attack edges per node ($k = 5$) in Figure 4b. We observe that a possible admission threshold, which denies the vast majority of Sybils would also classify 30% of the honest nodes as Sybils.

2) SybilInfer: The distinguishing ability of SybilInfer lies within the landing probability of its modified random walk, i.e., trace. Originating at a benign node, the vast majority of traces should end at another benign node—only then can gaps between the mixing times of different subgraphs be detected. Figure 5 shows the number of traces that end at benign and Sybil nodes, normalized by the number of benign and Sybil nodes in the system, respectively. All traces originate from a benign node, and therefore are called benign traces. As observed in SybilLimit, Sybils cannot obtain a sufficient amount of traces to end at a Sybil node with a single attack edge. However, Sybils succeed as more attack edges are added. As seen in Figure 5, even with two randomly placed attack edges per Sybil node, i.e., $k = 2$, SybilInfer is no longer able to distinguish between benign and Sybil nodes.

We also find that when traces end at a Sybil, they do not concentrate at a few Sybils, but instead are widely distributed. If $k = 2$ and a trace starts from every benign node, altogether the traces can hit 75% of the Sybils. An equivalent amount of Sybils might be admitted.

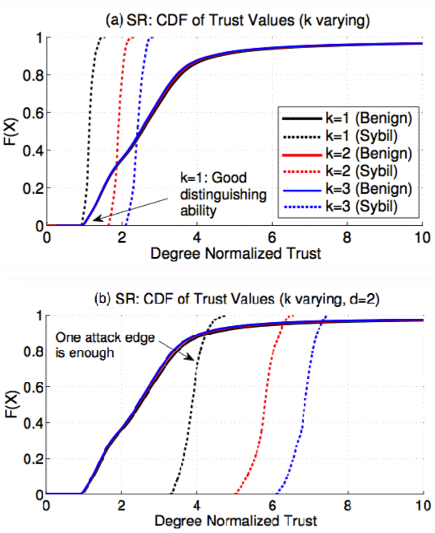


Fig. 6: Performance of SybilRank (SR). k is number of attack edges per Sybil.

3) SybilRank: SybilRank distinguishes nodes according to their normalized trust ranking. The lower the ranking of a node is, the more likely it should be a Sybil. Thus, to determine the distinguishing ability of SybilRank, we need to evaluate how much the rankings of benign nodes and Sybil nodes differ.

In its original design, SybilRank places 50 seeds, with one chosen from ten nodes with the highest degree in the OSN and the other 49 randomly chosen. Our experiments show that this strategy becomes increasingly flawed as the size of the OSN graph grows (not shown in figures). The reason is that due to the modular structure of OSN graphs, in many cases the distance of the honest users to the seeds is larger than that of the Sybil nodes to the seeds, resulting in higher rankings of Sybils than many benign nodes. We therefore place one seed in each honest Louvain-detected community, in order to improve SybilRank's distinguishing ability.

Our results are shown in Figures 6a and 6b. A Sybil node only needs two randomly placed attack edges to obtain a higher ranking than 30% of the honest nodes, leaving SybilRank with either a very high false positive rate (30% of honest nodes ranked as Sybils) or ineffective at detecting Sybils (Fig. 6a).

More worrisome, as shown in Figure 6b, if the attacker can place attack edges only two hops away from a seed (i.e., $d = 2$), one attack edge is sufficient for Sybils to outperform the majority of honest nodes. In further experiments (not shown in graphs), we find that as a rule of thumb, if placing attack edges one more hop away from the seed, a Sybil will only need to add one more attack edge in order to achieve the same effect.

4) GateKeeper: The distinguishing ability of GateKeeper depends on how many tickets Sybil nodes can obtain relative to honest users. Figure 7a shows two CDF curves of acquired tickets for Sybil nodes and benign nodes, respectively, with one randomly placed attack edge per Sybil node ($k = 1$). Clearly, one randomly placed attack edge per Sybil is sufficient to make the two CDFs cross. About 35-40% of the honest nodes obtain fewer tickets than Sybil nodes. If we exclude all Sybil nodes from being admitted, we will also exclude about 90% of the honest nodes. This is caused by the modular structure of the OSN (i.e. multiple distinct benign communities), which GateKeeper does not consider. With only few edges connecting different communities, most ticket sources selected by the admission controller via random walk will be in the same community as the controller, and nodes from other communities will only acquire at most a trickle of tickets.

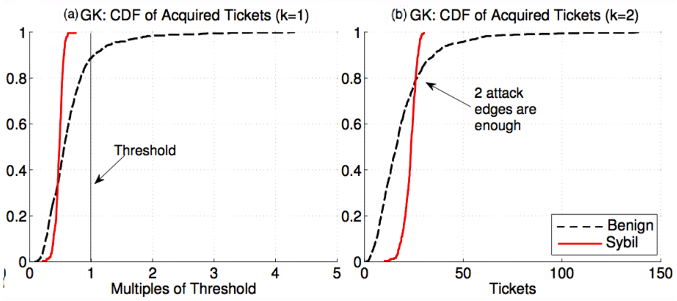


Fig. 7: Performance of GateKeeper (GK). k is number of attack edges per Sybil. Threshold = 35 tickets.

To see whether modifying GateKeeper may help, we alter GateKeeper so that it can reach more benign nodes in modular networks. We place an admission controller in *each* Louvain-detected community [22]. The results are shown in Figure 7b. It shows virtually all honest nodes are admitted, since they only have to be admitted by one controller, and there is one in each community. However, for the same reason virtually all Sybil nodes are admitted as well. If more attack edges are added, the Sybils outperform honest users. In fact, if a Sybil is able to obtain two random attack edges, it can collect more tickets than 80% of the honest nodes. This is because benign nodes have most links within one community, whereas Sybils have a good chance to place attack edges to multiple communities, and therefore in reach of multiple ticket sources.

C. Sybil Tolerance Approaches

Our goal in evaluating Sybil tolerance approaches is to find out to what extent these approaches are able to limit the impact of the Sybil nodes in the modern scenario. In contrast to Sybil detection approaches, it is important to consider the number of attack edges relative to the number of honest edges in a Sybil tolerance system, i.e., the ratio of attack edges to honest edges, also denoted as k. We therefore focus on to which extent the impact of Sybils may grow with a higher ratio of attack edges or intelligent attack strategies.

1) Ostra: Figure 8 provides an overview of Ostra's performance for a varying number of attack edges. On one hand, Ostra does a good job in mitigating spams from Sybils. While the amount of spam messages that can go through does grow proportionally with the number of attack edges in the system, as shown in Figure 8a, Ostra is able to block a large amount of spam messages and keeps the delivery ratio for Sybils quite low. However, the true impact of an increasing number of attack edges lies in Figure 8b, where we evaluate the amount of benign messages that are blocked due to the credit depletion on the path between a source and a destination. Recall when a Sybil node sends spam to a destination, all links on the path—including edges between honest users—can be penalized by Ostra's feedback mechanism. For example, with 1% ($k = 0.01$) edges in the entire system being attack edges, about 5% of the benign nodes will have 5% of their messages blocked. Note since a message that traverses one attack edge may traverse multiple benign edges, every newly added attack edge can

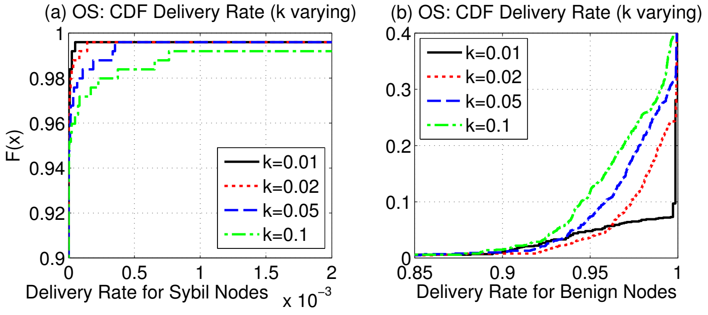


Fig. 8: Performance of Ostra (OS). k is the ratio of attack edges to regular edges.

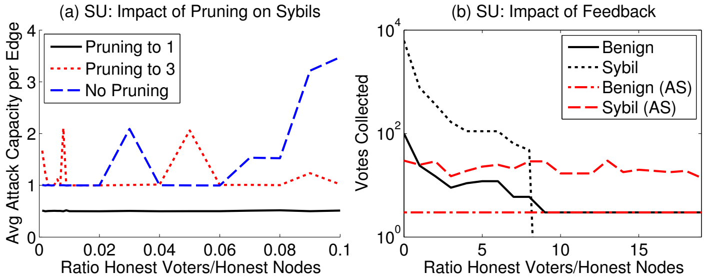


Fig. 9: Performance of SumUp (SU).

multiply the negative impact described here.

2) *SumUp:* Core to SumUp's ability to limit the bogus votes from Sybils are its pruning mechanism and feedback mechanism. We therefore focus on these two mechanisms.

Pruning mechanism: Figure 9a illustrates how pruning may affect Sybils' voting capacity. Specifically, it shows how the capacity of attack edges varies when the pruning has a different level of aggressiveness and when more honest voters vote. Our results in Figure 9a demonstrate that pruning has minimal impact on Sybil users, if any. Only pruning to a single incoming edge drops the average capacity per attack edge significantly. Unfortunately, doing so comes at the price of a reduced ability of collecting honest votes.

Furthermore, when there is no pruning, Sybil voters perform better as more benign users vote. The reason is simple. The more honest votes reach the vote collector, the more tickets to be distributed, thus the higher edge capacities on average and the more Sybil votes.

We also evaluated the impact of pruning on honest users, as a successful pruning should have limited or no impact on them. We discover (not shown in graphs) that this is the case if a node has three or more incoming edges after pruning. Pruning to a single incoming edge, however, has an adverse impact since it depletes vote capacities on edges that could otherwise be routed around.

Feedback mechanism: The best way to evaluate the efficacy of the feedback mechanism is to see how well it works when attackers establish attack edges close to the vote collector (allowing them to flood a large number of bogus votes). We thus designed an experiment with multiple rounds of voting, with the objective to see if the feedback from every round can help the subsequent rounds of voting. We use social graphs where attack edges are directly connected to the vote collector and every node is pruned to have at most three incoming edges. Out of 1,000 honest users at every round a random 10% will vote, thus 100 votes per round. At the same time 10,000 bogus votes try to outvote the honest votes in every round. Figure 9b shows our results. First, coinciding with SumUp's original findings, the feedback mechanism can rapidly reduce the number of Sybil votes after only a few rounds. While Sybils can initially cast 10^4 votes, no bogus votes can go through after round 7. However, the feedback mechanism also reduce the number of honest votes. Although the bogus votes cannot outvote the honest votes after several rounds, in the end out of 100 honest votes cast every round, only 7 can be taken. The cause of this problem is that as the feedback mechanism penalizes more and more links, some honest votes will not be able to reach the vote collector.

Moreover, an intelligent attacker can employ some attack strategy (AS) to counter the feedback mechanism. For example, instead of having all Sybils vote in every voting round, in every round it can cycle through the Sybil nodes and let different Sybils to vote (in our case we gave every Sybil a probability of 0.10 to vote). By doing so, although Sybils cast less votes this way, they can outvote honest votes continuously.

Recall that after SumUp removes penalized links, it can reintroduce pruned links to replace the penalized links in order to maintain the required number of incoming edges for every node (Section IV-B2). Unfortunately, the attacker can take advantage of this feature, and Sybil nodes with a large number of attack edges can cycle through their penalized links and pruned links. If a Sybil has an attack edge that is far from the vote collector and a previously pruned edge that is close to the collector, the Sybil could even cast bogus votes to have the attack edge replaced with the pruned edge, thus moving itself closer to the vote collector.

D. *Lessons Learned*

For Sybil detection schemes, our experiments show that are indeed very vulnerable to an increasing number of attack edges. For some, a single, randomly attached attack edge is sufficient for a Sybil to disguise itself as a benign node. We further show that simple modifications to the approaches are not sufficient to improve their detection capabilities. Out of the approaches under investigation, a modified SybilLimit performed the best, where a Sybil needs to obtain about five attack edges in order to confuse the system. Also, in contrast to other schemes, we were not able to attack SybilLimit so that it rates all Sybils higher than all benign nodes. This is because an increase in attack edges also results in more options for the construction of the benign nodes' and verifiers' traces. As a consequence, the probability to intersect with a verifier decrease for all nodes, including Sybils.

In contrast to Sybil detection schemes, conceptually Sybil tolerance schemes (Ostra and SumUp) are not broken—they still limit Sybil activity to some extent. The main difference between the two classes of Sybil defense schemes is that Sybil tolerance systems do not need to make an ultimate decision on a node's fate, but can rather adaptively react to the behavior of malicious nodes. However, both Ostra and SumUp have problems as well: in Ostra a non-negligible fraction of nodes may be blocked from communicating, and SumUp would allow an intelligent attacker to cast a higher vote count than benign users, leaving both schemes with only limited success in tolerating Sybils.

VI. Prospects of Future Sybil Defense Solutions

Our analysis, measurement and comparison of existent OSN-based Sybil defense solutions further provides insights on new Sybil defense solutions. The main commonality that connects all approaches is that they solely exploit the (same) distribution of edges in the OSN graph. Follow-up suggestions to detect Sybils, such as using the clustering coefficient [2], fall in the same category. As shown before, defenses built according to these properties are very sensitive to changes in the graph structure. Alvisi et al. suggest to force Sybils into the required structure [18] by monitoring the link request acceptance rates of different nodes. If a node has to be accepted by a certain number of other nodes to be classified as benign, Sybils might be forced into creating many links among themselves (which are guaranteed to be accepted). This would eventually lead to a larger density of edges among the Sybils themselves compared to the links with honest nodes—which could ultimately allow detection using existing approaches again. However, Sybils can already achieve acceptance rates of up to 90% and even gather a lot of requests *toward* them with simple attacks [12], [13]. The latter fact removes the need of Sybils to actually reach out to benign nodes and hope for the acceptance of their requests. It also particularly weakens schemes that depend on Sybil nodes to initiate certain actions.

In fact, structural properties only account for a very small fraction of the actual trust that is incorporated within a social relation. On the other hand, there are a lot of meta-data in social networks that add up to the strength of ties between users [24]. For instance, one could measure the intensity of communication between two particular nodes—a major contributing factor to tie strength between users—to detect Sybils. However, such an approach can also result in a large fraction of false positives, as honest users who rarely interact with others might be mistaken for Sybils.

Therefore, in looking forward to future Sybil defense solutions, we anticipate an approach that *enriches* the structure of a social graph with more information about the *relations* between its users in order to defend against Sybils. For instance, attack edges could experience a shorter lifetime than regular edges, since they might be deleted once a benign user realizes he has become connected to a Sybil. One could classify nodes whose links experience suspiciously short lifetimes as Sybils.

VII. Conclusion

As defending against Sybil attacks is of critical importance towards a trusted cyber space, there has been a proliferation of Sybil defense schemes. However, it is unclear how effective these solutions are under different contexts, especially given the modern scenario where Sybils can employ simple strategies to create many links with benign users, i.e., attack edges.

In this work, by focusing on the performance of each Sybil defense solution under the modern scenario, which more truly reflects the evolving behavior of Sybil attackers, we extensively measured and evaluated major Sybil defense schemes, and unveiled that current OSN-based Sybil defenses do contain severe weaknesses. We find that, when Sybils are not herded together in a distinct Sybil community with very few links to the outside world, all of the evaluated schemes suffer in their effectiveness—some more than others. Sybil detection approaches, even with a modified design, have a hard time reliably distinguishing a Sybil node from honest nodes. In fact, whereas it has been shown that Sybils can obtain hundreds of attack edges in real-world OSNs, our study shows all existent approaches can be circumvented by the presence of only a handful of attack edges. Sybil tolerance schemes are application-specific and rather than being fundamentally flawed, their weaknesses are mostly in the details. Nonetheless, they too are vulnerable if Sybils vary their assumed behavior.

Our study provides insights to new Sybil defense solutions. We anticipate a Sybil defense approach to be more effective if leveraging not only structural properties of an OSN, but also more information about the relations between its users.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable feedback, as well as Skyler Berg and Paul Elliot for their proofreading efforts. This work has been partially supported by the NSF (grant no. CNS-0644434 and CNS-1118101) and the Lindemann Foundation. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] J. Douceur, "The Sybil Attack," in *Peer-to-Peer Systems.* Springer Berlin / Heidelberg, 2002, pp. 251–260.

[2] Z. Yang, C. Wilson *et al.*, "Uncovering Social Network Sybils in the Wild," in *IMC'11.*

[3] H. Yu, P. B. Gibbons *et al.*, "SybilLimit: a Near-optimal Social Network Defense against Sybil Attacks," *IEEE/ACM Trans. Netw.*, vol. 18, no. 3, pp. 885–898, 2010.

[4] Q. Cao, M. Sirivianos *et al.*, "Aiding the Detection of Fake Accounts in Large Scale Social Online Services," in *NSDI'12.*

[5] N. Tran, J. Li *et al.*, "Optimal Sybil-resilient Node Admission Control," in *INFOCOM'11.*

[6] G. Danezis and P. Mittal, "SybilInfer: Detecting Sybil Nodes using Social Networks," in *NDSS'09.*

[7] N. Tran, B. Min *et al.*, "Sybil-resilient Online Content Voting," in *NSDI'09.*

[8] A. Mislove, A. Post *et al.*, "Ostra: Leveraging Trust to Thwart Unwanted Communication," in *NSDI*, 2008.

[9] B. Viswanath, A. Post *et al.*, "An Analysis of Social Network-based Sybil Defenses," in *SIGCOMM'10.*

[10] Y. Boshmaf, I. Muslukhov *et al.*, "The Socialbot Network: When Bots Socialize for Fame and Money," in *ACSAC'11.*

[11] L. Bilge, T. Strufe *et al.*, "All Your Contacts are Belong to us: Automated Identity Theft Attacks on Social Networks," in *WWW'09.*

[12] D. Irani, M. Balduzzi *et al.*, "Reverse Social Engineering Attacks in Online Social Networks," in *DIMVA'11.*

[13] V. Sridharan, S. Vaibhav *et al.*, "Twitter Games: How Successful Spammers Pick Targets," in *ACSAC'12.*

[14] H. Yu, M. Kaminsky *et al.*, "SybilGuard: Defending Against Sybil Attacks via Social Networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 3, pp. 576–589, 2008.

[15] H. Yu, "Sybil Defenses via Social Networks: A Tutorial and Survey," *SIGACT News*, vol. 42, no. 3, pp. 80–101, 2011.

[16] B. Viswanath, M. Mondal *et al.*, "Exploring the Design Space of Social Network-based Sybil Defense," in *COMSNETS'12.*

[17] Y. Boshmaf, K. Beznosov *et al.*, "Graph-based Sybil Detection in Social and Information Systems," in *ASONAM'13.*

[18] L. Alvisi, A. Clement *et al.*, "SoK: The Evolution of Sybil Defense via Social Networks," in *IEEE Symposium on Security and Privacy*, 2013.

[19] P. Mittal, C. Papamanthou *et al.*, "Preserving Link Privacy in Social Network Based Systems," in *NDSS'13.*

[20] A. Mohaisen, A. Yun *et al.*, "Measuring the Mixing Time of Social Graphs," in *IMC'10.*

[21] J. Leskovec and C. Faloutsos, "Sampling From Large Graphs," in *KDD'06.*

[22] V. D. Blondel, J.-L. Guillaume *et al.*, "Fast Unfolding of Communities in Large Networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, 2008.

[23] J. Jiang, C. Wilson *et al.*, "Understanding Latent Interactions in Online Social Networks," in *IMC'10.*

[24] E. Gilbert and K. Karahalios, "Predicting Tie Strength with Social Media," in *CHI'09.*