Exploring the design space of social network-based Sybil defenses

Bimal Viswanath*, Mainack Mondal*, Allen Clement*, Peter Druschel*,
Krishna P. Gummadi*, Alan Mislove†, and Ansley Post*

*Max Planck Institute for Software Systems (MPI-SWS)
Kaiserslautern and Saarbruecken, Germany
Email: {bviswana, mainack, aclement, druschel, gummadi, abpost}@mpi-sws.org

†College of Computer and Information Science, Northeastern University
Boston, MA, USA
Email: amislove@ccs.neu.edu

***Abstract*—Recently, there has been significant research interest in leveraging social networks to defend against Sybil attacks. While much of this work may appear similar at first glance, existing social network-based Sybil defense schemes can be divided into two categories: *Sybil detection* and *Sybil tolerance*. These two categories of systems both leverage global properties of the underlying social graph, but they rely on different assumptions and provide different guarantees: Sybil detection schemes are application-independent and rely only on the graph structure to identify Sybil identities, while Sybil tolerance schemes rely on application-specific information and leverage the graph structure and transaction history to bound the leverage an attacker can gain from using multiple identities. In this paper, we take a closer look at the design goals, models, assumptions, guarantees, and limitations of both categories of social network-based Sybil defense systems.**

I. Introduction

Multiple identity, or Sybil [1], attacks pose a fundamental problem in web-based and distributed systems. In a Sybil attack, a malicious user creates multiple (Sybil) identities and takes advantage of the combined privileges associated with these identities to attack the system. For example, in online auction systems like eBay, a fraudulent user can continue to use the system by creating a new user account whenever her existing accounts have acquired a bad reputation. Similarly, in social networking sites like Digg or YouTube, where content is rated based on user feedback, an attacker can create multiple identities to cast bogus votes and manipulate content popularity.

Recently, there has been significant research interest in leveraging social networks to defend against Sybil attacks [2]–[11]. In this paper, we focus on the design of such social network-based Sybil defense schemes.

There are two categories of social network-based Sybil defense schemes. The first category, called *Sybil detection* schemes, operate by detecting identities that are likely to be Sybils [3]–[8]. In contrast, the second category, called *Sybil tolerance* schemes, do not attempt to label identities as Sybil or non-Sybil. Instead, they try to bound the leverage an attacker can gain by using multiple Sybil identities [2], [9]–[11]. Sybil detection and tolerance represent two different approaches towards achieving the higher-level goal of Sybil defense, which is to prevent attackers from gaining an advantage by creating and using multiple identities.

978-1-4673-0298-2/12/$31.00

In this paper, we explore how Sybil detection and tolerance differ in the assumptions they make, the guarantees they offer, their limitations and the challenges they pose in real-world deployment scenarios. While our exploration of the design space of Sybil defenses is not exhaustive and many open questions still remain, our work highlights the need to recognize the fundamental differences between existing Sybil defense designs and the trade-offs they offer. Much of the recent work surveying or analyzing social network-based Sybil defense schemes, including our own, tends toward an overly general characterization of all social network-based Sybil defense schemes based on the study of a few [12], [13].

In the remainder of this paper we discuss both Sybil detection and Sybil tolerance to better understand the design goals, models, assumptions, guarantees, and limitations of each. We close with a discussion of the issues and trade-offs when deploying the schemes in practice.

II. Sybil detection

Sybil detection schemes have been designed for *identity-based* social systems. Each user is intended to have a single identity, and users establish friendship links to the identities of other users they recognize in the system, thereby building a social network. Sybil detection uses this social network as a basis for identifying users with multiple identities. We call a user with multiple identities a *Sybil user* and each identity she uses a *Sybil identity*. The goal of Sybil detection is to label identities in the system as either *Sybil* ('untrustworthy') or *non-Sybil* ('trustworthy') with high accuracy. The system or individual users in the system can then take an appropriate action to

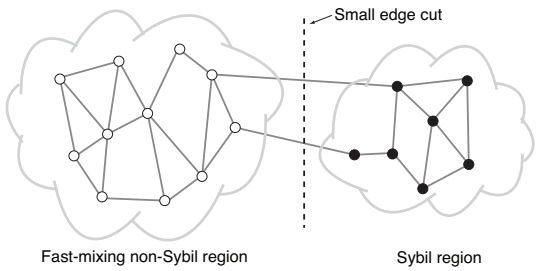

Fig. 1. Sybil detection relies on the small edge cut between the fast mixing non-Sybil region and the Sybil region.

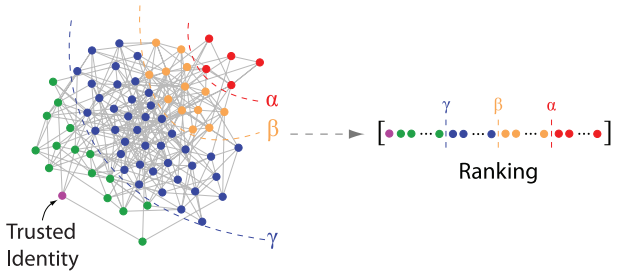

Fig. 2. Diagram of converting partitionings into a ranking of identities. Different parameter settings (α, β, γ) cause increasingly large partitions to be marked as Sybils, thereby inducing a ranking.

handle identities labeled as Sybil. For example, they could block all detected Sybil identities from interacting with other identities in the system.

A. Common assumptions and system model

Social network-based Sybil detection schemes rely on the the assumption that although the attacker can create an arbitrary number of Sybil identities in the social network, he or she cannot establish an arbitrarily number of social connections to non-Sybil identities in the network [12]. Intuitively, this assumption is rooted in the observation that establishing new social links with honest users' identities takes some effort, because honest users are unlikely to accept a friend invitation from an identity they do not recognize.

Effectively, existing social network-based Sybil detection schemes work by analyzing the structure of the social network. To identify Sybils, all schemes make three common assumptions:

1) The non-Sybil region of the network is densely connected (or fast-mixing [14]), meaning random walks in the non-Sybil region quickly reach a stationary distribution.
2) Although an attacker can create an arbitrary number of Sybil identities in social network, she cannot establish an arbitrary number of social connections to non-Sybil identities, i.e., the attacker cannot easily infiltrate the densely connected non-Sybil network.
3) The system is given the identity of at least one trusted non-Sybil.[1]

These three assumptions, together, form the basis of Sybil detection. Since the non-Sybil region of the network is densely connected (assumption 1), and the Sybil region of the network is attached by a limited number of links (assumption 2), existing detection schemes look for resulting topological features to partition the network into Sybil and non-Sybil regions (see Figure 1). They then look for the partition that contains the known non-Sybil identity (assumption 3) to decide which is the non-Sybil region.

[1]This assumption is necessary, as if the systems didn't make this assumption, the Sybil identities could form an identical network to the non-Sybil region, and the system could not distinguish between the two.

B. Example systems

We now give a brief overview of existing social network-based Sybil detection systems. Our goal here is to illustrate that while the precise algorithms used vary greatly between these systems, they all rely on analyzing the network structure to identify Sybil identities.

SybilGuard [3] and SybilLimit [4] are among the first Sybil detection schemes to be proposed. SybilGuard uses the intersections between modified random walks to determine whether identities should be accepted. SybilLimit improves on SybilGuard's bound by using multiple walks, accepting fewer Sybil identities per attack edge. Both of these schemes can be implemented in a centralized or decentralized fashion.

SybilInfer [5] is a centralized protocol that assumes full knowledge of the social graph. It uses a Bayesian inference technique to assign a probability of being Sybil to each identity. Unlike SybilGuard and SybilLimit, SybilInfer does not provide any analytical bounds on the number of Sybil identities accepted per attack edge.

GateKeeper [6] is a decentralized Sybil detection protocol that improves over the guarantees provided by SybilLimit. It uses a variant of the ticket distribution algorithm used in SumUp [2] from multiple random identities in the graph to detect Sybils.

MobID [7] is a Sybil detection system proposed for mobile settings. MobID defends against Sybil attacks on in-range portable devices using two social networks: a network of friends and a network of foes (or suspicious devices). MobID uses network centrality measures to analyze the social network structure and flag identities as Sybil or non-Sybil.

Whānau [8] is a DHT routing protocol built in conjunction with a Sybil identity detection scheme. Whānau only selects nodes for routing if they meet certain random walk intersection criteria over a social network.

C. Understanding Sybil detection

Each of the Sybil detection schemes discussed so far uses different graph analysis algorithms to search for cuts in the social network that mark the boundary between the non-Sybil and Sybil identities. We now turn our focus to the these graph algorithms behind the schemes. Recent work [13] has shown

that, although existing Sybil detection proposals use seemingly different mechanisms, they all work in a similar manner when partitioning the identities in the network graph into Sybils and non-Sybils from the perspective of a single trusted identity.

In more detail, at their core, all Sybil defense schemes can be modeled as, first, inducing a ranking on all the identities from the perspective of the trusted identity and, second, applying a cut-off on the ranking that is determined by scheme-specific parameters. Nodes ranked before the cut-off are marked as non-Sybil; identities ranked after the cut-off are marked as Sybil. This process is illustrated in Figure 2. It has been observed that the different algorithms behind the proposals yield similar rankings of the nodes for a given trusted identity. Thus, the primary challenge when applying Sybil detection schemes lies in configuring their parameters as they crucially determine the position of the cut-off point separating non-Sybils and Sybils in the node rankings.

Moreover, node rankings have been shown to depend on certain properties of the social network—in particular, the community structure of the network [13]. Nodes that are tightly connected to the trusted identity are more likely to be ranked higher. In particular, when the trusted identity is located in a densely connected community of identities, with a clear boundary between this community and the rest of the network, the identities in the local community around the trusted identity are ranked before others. So community boundaries offer a natural cut-off point for separating trusted (non-Sybil) and untrusted (Sybil) nodes.

Thus, the performance of these schemes is heavily dependent on the size and characteristics of the community surrounding the trusted identity: If it includes the whole non-Sybil region, then they will perform quite well; if it is relatively small and localized, the schemes will only be able to reliably classify the small fraction of the non-Sybil identities within the local community as trustworthy. However, in many computer systems, it is necessary for users to interact with others who are outside of this small set of trusted identities. For example, the usefulness of communication systems like email and online marketplaces like eBay would be drastically reduced if users could only interact with local community members.

In summary, for Sybil detection schemes to work efficiently it is crucial that *non-Sybils in real-world social networks form one tightly-knit community devoid of sparse internal cuts*. Sybils would find it hard to infiltrate such a densely connected non-Sybil community as it would require establishing a large number of links.

D. Challenges in building Sybil detection systems

Sybil detection schemes are quite appealing to system designers because they do not rely on any application-specific details and they can be easily integrated into existing systems and applications. To defend against Sybil attacks in distributed systems, designers only need to block all identities declared as Sybils by the detection schemes. Further, some Sybil detection schemes [3], [4] have been designed for deployment in decentralized systems (and decentralized social networks) like Diaspora* [15].

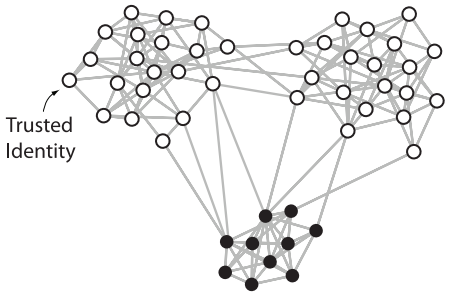

Fig. 3. Non-Sybils are hollow and Sybils filled. The trusted identity cannot distinguish between the non-Sybil community and the Sybil community.

However, by relying solely on analyzing the social network structure, detection schemes are highly vulnerable to misclassifying users as Sybils or non-Sybils, when a real-world network's structure does not conform to the assumptions the schemes make. Specifically, lets examine the requirement that all non-Sybil users form one tightly-knit community, devoid of small internal cuts, holds in practice.

Studies analyzing the structures of large-scale real-world social networks [16], [17] have found that such networks have a significant fraction of nodes on the fringes that are sparsely connected to the rest of the network. These nodes often have low degrees (few friend links) and they constitute the 'heavy-tail' in the power-law node degree distributions observed in these networks. A recent study examining the community structures in these real-world networks [18], found that nodes in the periphery are often organized into small, tightly-knit clusters that connect to the rest of the network via small cuts. As a result, when Sybil detection schemes are run over real-world graphs, the honest identities on either sides of cuts are likely be blocked from interacting with each other [13]. These observations are further corroborated by some recent findings [19] that the mixing time for many real-world networks is substantially lower than was previously thought, implying that the networks are actually *not* fast mixing as previously expected.

Further, users in some small to medium-scale real-world social networks, such as collaboration networks for research [20] or software development [21], have been observed to organize themselves into strong local communities that are sparsely interconnected. In such social networks, all nodes have high degrees (lots of friend links), but the non-Sybil region still possesses sparse internal cuts, causing identities within one community to mistake non-Sybil identities in another community for Sybils. Furthermore, an attacker may be able to disguise Sybil identities as just another community in the network by establishing a small number of carefully targeted links to the community containing the trusted identity. Consider the topology in Figure 3, where the trusted identity will not be able to distinguish between the non-Sybil community (hollow) and the Sybil community (filled) outside its local community. In

this situation, the trusted identity can either conservatively accept only those identities in the local community and mark the rest of the identities as being Sybil (thus wrongly classifying several non-Sybil identities as Sybils) or accept everyone in the network (thus wrongly classifying Sybil identities as being non-Sybil) [13].

In summary, Sybil detection schemes impose strong requirements on the structure of the underlying social network. Many real-world social networks fail to conform to the requirements, limiting the potential deployment scenarios for Sybil defense schemes, despite the ease with which they could be integrated with any application.

III. Sybil tolerance

We now examine Sybil tolerance approaches, which also defend against Sybil attacks, but do so without attempting to explicitly label identities as Sybil or non-Sybil. A number of schemes exist for different applications [2], [9]–[11], [22], [23]; we briefly discuss three of them.

Ostra [10] limits unwanted communication (or spam) sent by users who create Sybil accounts. Ostra uses a social network, with credit values assigned to links. When a message is sent, Ostra finds a path with available credit from the sender to the receiver. If no such path is found, the message is blocked. If a path is found, credit is transferred from each user to the next along the path.

Bazaar [9] protects buyers and sellers in online marketplaces like eBay by limiting the reputation manipulation that is possible through the creation of Sybil accounts. It uses max flow-based techniques to estimate the reputation of users involved in a transaction and flags fraudulent transactions.

SumUp [2] secures online voting against users who create Sybil accounts and vote multiple times. SumUp chooses a vote collector in the network and distributes tokens (or credits) on the links in the network inside a voting envelope. Voters must find a path to the vote collector with available credit in order to cast a vote.

We now demonstrate that this class of schemes, which we refer to as *Sybil tolerance*, shares a common underlying approach.

A. Common assumptions, model, and goals

Similar to the model of Sybil detection described in Section II, each of these schemes is designed to be applied to an existing identity-based system (e.g., a communication system, an online marketplace, or a content-rating system) and assumes the existence of a network connecting the identities. This network may be derived from an external social network (in the case of Ostra and SumUp), or built internally by the system itself (in the case of Bazaar). The schemes make no assumptions about the cost of creating identities, but do assume that an attacker cannot establish an arbitrary number of links to non-Sybil identities (assumption 2 from Section II-A).

Tolerance schemes rely on assumptions about the structure of the network as well as the workload the system experiences.

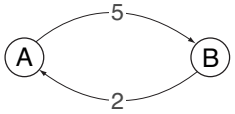

Fig. 4. Simplified credit network between two nodes A and B, with credit available c_{ab} and c_{ba} shown. In this example, A has 5 credits available from B, and B has 2 credits available from A.

In particular, they assume that users perform pairwise transactions (e.g., sending a message, purchasing an item, casting a vote). They achieve a defense against Sybils by assigning *credits* to the network links, and then allowing actions only if *paths with sufficient credit* exist between the source and destination of an action. In Ostra, a message can only sent if a path with at least one credit exists between the source and destination; in Bazaar, a item can only be purchased if a path with the item's price in credits exists between the buyer and seller; in SumUp, a user can only vote if a path with at least one credit exists between the voter and vote collector.

The Sybil tolerance schemes we consider all run alongside an existing system S, reason about a network connecting S's identities (e.g., a social network), and provide a single method

$$\texttt{transaction}(a, b, c \in \{\mathbb{R} > 0\}) \rightarrow \{0, 1\}$$

that decides whether identity a is allowed to initiate a transaction with identity b costing c credits. Thus, to take advantage of Sybil tolerance, S simply queries the Sybil tolerance system when two users are about to interact, and, depending on the result, either denies the transaction or allows it to proceed.

The goal of Sybil tolerance, then, is to ensure that the number of transactions that a (human) user can initiate is independent of the number of identities she possesses. Doing so would remove the creation of multiple accounts as an attack vector, thereby making the application tolerant of Sybils. In comparison to Sybil detection—where the system reasons about guarantees concerning the ability to identify Sybil identities—Sybil tolerance schemes reason about the impact (in terms of transactions) that identities have on one another. As a result, a certain pair of identities may be allowed to participate in certain transactions and not others, and may be allowed to interact at certain times and not others, depending on the state of the system.

B. Understanding credit network-based Sybil tolerance

In this section, we describe how existing Sybil tolerance schemes are implemented using *credit networks*. We first provide some background on credit networks and discuss the Sybil tolerant nature of credit networks.

1) Credit networks: Credit networks [11], [24], [25] were first introduced in electronic commerce to build transitive trust protocols in an environment where there are only pairwise trust accounts and no central trusted entities. In a credit network, identities (nodes) trust each other by offering pairwise credit (links) up to a certain limit. Nodes can use the credit to pay for services they receive from each other. The credit network

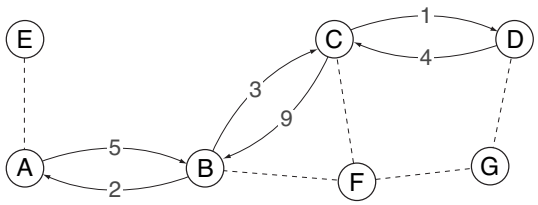

Fig. 5. More complex credit network, with credit available (c_{ij}) shown for each link. In this example, A can transfer 1 credit to D along the path $A \rightarrow B \rightarrow C \rightarrow D$. Note that, for simplicity, the links not on this path are only shown as dashed lines.

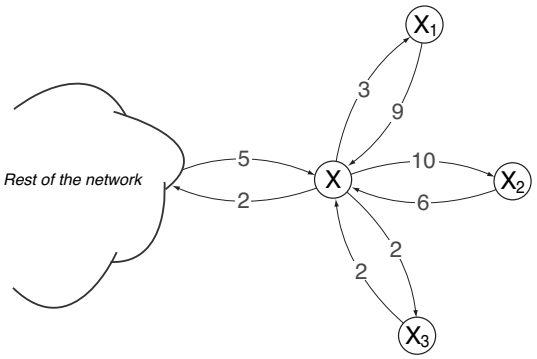

Fig. 6. Credit networks leading to Sybil tolerance. User X can create any number of identities (X_1, X_2, X_3) and arbitrarily assign the credit available between them. However, does not enable any additional available credit with nodes in the rest of the network.

can be used for payments between nodes that do not directly extend credit to each other. For this purpose, nodes can route credit to a node via network paths that traverse over links with available credit. (See Figures 4 and 5.)

Formally, a *credit network* is a directed graph $G = (V, E)$ where V is the set of nodes and E is the set of labeled edges. Each directed edge $(a, b) \in E$ is labeled with a dynamic scalar value c_{ab}, called the *credit available*, and is initialized to C_{ab}. Intuitively, C_{ab} represents the initial credit allocation that b gives to a, and c_{ab} represents the amount of unconsumed credit that b has extended to a. Note that $c_{ab} \geq 0$ at all times.

Transactions between two nodes in a credit network are contingent upon the availability of credit along network paths connecting the nodes. If a node a wishes to obtain a favor or resource from b, then a path

$$a \rightarrow u_1 \rightarrow ... \rightarrow u_n \rightarrow b$$

(which could just be $a \rightarrow b$) must exist where credits are available on each (i, j) link (i.e., $c_{ij} > 0$). If so, the credit available on each directed edge c_{ij} on the path from a to b is decreased and the credit available on each directed edge c_{ji} on the reverse path is increased. As a result of this action, each node "pays" credits to its successor on the path to b, in exchange for the favor or service a obtains from b.

2) Credit networks from social networks: One can build a credit network from a social network as follows: For each identity in the social network, we generate a node in the credit network. For each edge between a pair of identities in the social network, we generate an edge in the credit network between nodes corresponding to the users. Undirected edges in the social network (e.g., Facebook friend links) are replaced by two directed edges, one in each direction, between the nodes adjacent to the edges. Because social networks are known to be richly connected [16], [26], credit networks inherit the rich connectivity they require for liquidity [24].

Further, each directed edge, (a, b), is assigned an initial credit allocation C_{ab} by the destination node b. The system must exercise care when assigning credit allocations. For instance, when a new social link is created, the requesting node should be required to grant the accepting node some initial credit but not vice-versa, to prevent an attacker from obtaining credit by initiating social links.

3) Sybil tolerant nature of credit networks: Next, we show that credit networks built from social networks are naturally tolerant to Sybil attacks. Specifically, we argue that a Sybil attacker cannot increase the credit available to her from the rest of the network.

An attacker can mount a Sybil attack by creating many different identities in the social network, each corresponding to a different node in the credit network. However, per our assumptions about credit assignment to links, having many user accounts does not by itself allow the attacker to obtain additional available credit with other users (though she can create an arbitrary number of links with arbitrary credit between her Sybil identities).

As shown in Figure 6, the total amount of credit available to a single user is the sum of the credit available on her links to other (human) users. An attacker with an arbitrary number of Sybil identities has exactly the same available credit as the attacker with just one identity; in this case, the relevant set of edges is the cut between the subgraph consisting of the attacker's Sybil identities and the rest of the network. Any credit available on edges between the attacker's Sybil identities does not matter, because it does not enable additional "purchases" from legitimate nodes. Thus, available credit in a credit network is resilient to Sybil attacks [27].

C. Challenges building credit network-based Sybil tolerance

We now discuss the key challenges associated with building credit network-based Sybil tolerance systems. This includes

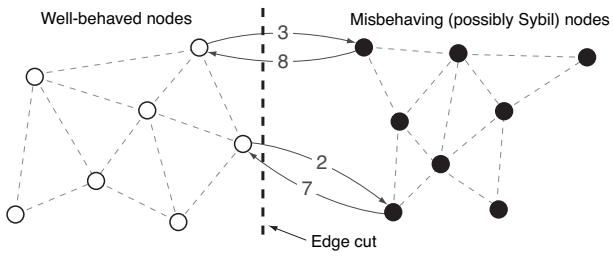

Fig. 7. Edge cut between well-behaved nodes (hollow) and misbehaving nodes (solid). The total credit available to the misbehaving nodes is 5 (3+2), regardless of the number of Sybil identities created. Note that the links that are not along the edge cut are shown as dashed lines, for simplicity.

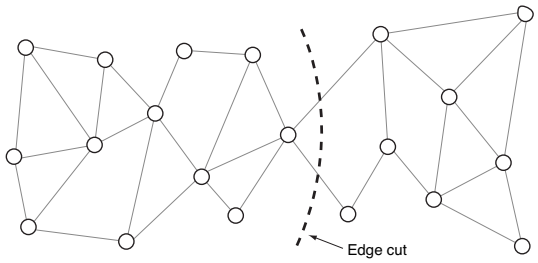

Fig. 8. Edge cut internal to the well-behaved nodes.

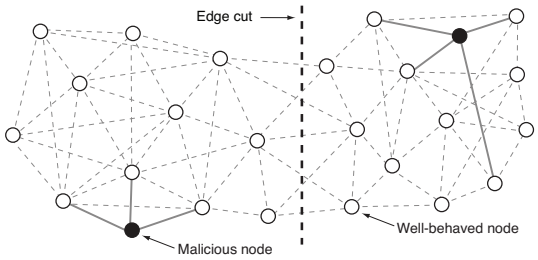

Fig. 9. Diagram of the resilience of credit networks to credit exhaustion attacks by malicious nodes (shown as filled nodes). In real-world social networks, the min cut between nodes occurs at the nodes themselves, rather than in the middle of the network, preventing malicious nodes from exhausting credit between well-behaved nodes

two main challenges: fundamental credit network design challenges and practical implementation/deployment challenges.

1) Credit network challenges: The credit network plays a fundamental role in the operation of Sybil tolerance schemes. Thus, a system designer wishing to apply Sybil tolerance in a given application must make careful choices concerning the initialization of starting credits on links, the adjustments to credits after each transaction, and the replenishment of credits over time. The key goals are to maintain *liquidity in the credit network* such that (1) most attacker transactions are disallowed, (2) most legitimate transactions are allowed, and (3) the credit network does not introduce any denial-of-service attacks on legitimate users. The challenge is to design a credit-network based mechanism that encourages legitimate transactions and discriminates against unwanted transactions. We discuss these goals in more detail using a Ostra as an example.

Bounding undesirable transactions As discussed earlier in Section III-B3, credit networks built from social networks are Sybil tolerant by nature. In the attack topology shown in Figure 7, the imbalance in transactions between the spammers and legitimate users (i.e., the spam in our messaging system) is always bounded by the aggregate credit (the sum of the credit balances on the links) available on the edge cut separating spammers from legitimate users. This is true regardless of the number of Sybil identities the spammers use or the credit balances on the links between the spammers' identities. Thus, the credit network naturally bounds the number of spam transactions, regardless of the number of identities the attacker possesses.

Allowing legitimate transactions The system designer must also ensure that the chosen mechanism does not block legitimate transactions in the common case. So next, we focus on the case when all nodes in our messaging system are legitimate. Consider an edge cut that divides legitimate users into two groups, as shown in Figure 8. Our credit adjustment mechanism would bound the credit imbalance between the two groups to the credit each of the groups make available to the other. If the identities in one group are interested in sending a disproportionately large number of messages to the identities in the other group, the credit along the edge cut could be exhausted, preventing further transactions. This is essentially a *liquidity* problem, where a subset of the legitimate nodes have insufficient liquidity with another subset.

Thus, in the long-term, any subset of legitimate nodes must receive messages from the rest of the legitimate nodes as often as it sends messages to those nodes.[2] The mechanisms should be chosen so that the statistics of a legitimate workload distribution would ensure an approximate long-term trade balance. If not, the use of techniques like credit replenishment (where credits are periodically readjusted by the system) can be used [10]. A number of credit-network mechanism have been designed and evaluated for specific applications [2], [9]–[11], [22], [23]. Designing appropriate mechanisms for many more applications in a principled way remains an open problem.

Vulnerability to attacks on network liquidity Finally, the system designer must ensure that the credit network mechanism does not introduce new vulnerabilities. For example, can a few attacker nodes exhaust the credit along an edge cut separating legitimate identities, thereby preventing the legitimate identities from interacting with each other? For example, consider a small cut A through the network where there are both attacker and legitimate identities on either side. If the attackers have, in aggregate, more credit with the legitimate identities than exists along cut A, it is feasible that the attackers could exhaust the credit along A (e.g., by sending messages to each other, affecting the credit values on A). Fortunately, the topology of social networks (upon which our credit networks are often built) make this scenario unlikely.

First, social networks are sufficiently well connected that the min-cut between any pair of nodes tends to be adjacent to either of the nodes [10]. It follows that a single misbehaving node will run out of credit before the credit on any other cut in the network is exhausted (see Figure 9). Second, assumption 2 indicates that a group of Sybils controlled by an attacker will tend to have a small cut to the rest of the network (because the attacker is unable to create an arbitrary number of links to other real users). Therefore, a group of Sybils is also likely to run out of credit before the before the group can exhaust the credit on any larger cut in the network.

Regardless, a full exploration of the necessary connectivity

[2]Short-term imbalances can be absorbed by setting appropriate initial credit allocations.

of credit networks and the relationship with the transaction workload remains future work.

2) Implementation/deployment challenges: Credit network-based Sybil tolerance schemes face scalability challenges when applied to large social networks. In particular, the scheme must often search for a specified amount of available credit between two identities; this is essentially the maximum flow problem [28], which is known to be a computationally expensive operation. The most efficient algorithms for the maximum flow problem run in $O(V^3)$ [29] or $O(V^2 \log(E))$ [30] time. Also, techniques that pre-calculate the all-pairs maximum flow (e.g., Gomory-Hu trees [31]) can not be applied to Sybil tolerance schemes, as these techniques assume a static network and impose a large, upfront pre-calculation cost (credit networks are constantly changing due to credit manipulations as well as new users and links). Furthermore, techniques for modifying existing pre-calculations as the graph changes [32] often end up being as expensive as simply starting the pre-calculation from scratch.

For example, Bazaar can take over 6 seconds [9] to determine whether sufficient flow exists over a network with 3.3 million links, and Ostra can require over 3.7 seconds [10] over a network with 3.4 million links. Given that both of these are intended to be run in an online fashion, this introduces a significant delay in the processing of transactions.

In order to address this problem, it is worth investigating techniques that can more quickly determine whether sufficient credit exists in very large credit networks. Approximation algorithms [33], [34] represent a promising technique, and hold the potential to be a favorable trade-off between speed and accuracy.

IV. Discussion: Detection vs. Tolerance

Having examined the design trade-offs offered by social network-based Sybil detection and tolerance schemes separately, we now compare them from the perspective of an operator wishing to deploy these schemes to defend her system from Sybil attacks.

Conceptually, Sybil detection schemes offer a simple model that is easy to integrate with any application. For instance, the system can simply deactivate identities that are classified as likely Sybils and allow all activity from identities classified as non-Sybils. However, this simplicity and ease of application comes at a high cost for misclassifying an identity as Sybil or non-Sybil. An innocent user who is misclassified (false positive) is denied all service, while a misclassified attacker identity (false negative) is not limited in its malicious activity. Furthermore, existing Sybil detection schemes rely solely on the network structure to identify non-Sybil and Sybil identities, ignoring other relevant information about the activity of identities.

To achieve accuracy, Sybil detection requires the underlying social network to satisfy certain constraints, such as the absence of small cuts within the non-Sybil region (i.e., non-Sybil region should be fast-mixing). Unfortunately, there is mounting evidence that many real-world social networks fail to meet these requirements, either because a significant fraction of their nodes are sparsely connected or their users organize themselves into small tightly-knit communities that are sparsely interconnected. When applied to such networks, Sybil detection schemes suffer from a high rate of misclassified identities.

Credit network-based Sybil tolerance schemes, on the other hand, allow or deny individual transactions among users based on the prevailing system state. This state reflects the history of transactions among users as well as the social graph structure. Thus, Sybil tolerance schemes are deeply embedded in the operation of the system and have to be tailored for each application; they are limited to applications for which an appropriate mechanism is known that lends Sybil tolerance to the relevant system properties.

Sybil tolerance schemes leverage both social network structure and the transaction history, which enables high classification accuracy. Moreover, they allow or deny individual transactions, which leads to a graceful degradation in the presence of false positives or false negatives. It is highly unlikely that all of a legitimate identity's transactions would be blocked due to false positives, or that all of a Sybil identity's transactions would be allowed due to false negatives.

To illustrate these points, consider applying Sybil detection and tolerance schemes to the problem of email spam. Sybil detection schemes would generate a blacklist and whitelist of Sybil and non-Sybil identities. Any sparsely connected nodes in the fringes of the social network would be blacklisted, while any whitelisted attacker node can send unlimited spam. Sybil tolerance, on the other hand, bounds the rate of spam messages that legitimate users receive from spammers. Sparsely connected legitimate nodes at the fringe of the social network would at worst be limited in the rate at which they can send legitimate messages. Simultaneously, no user has the ability to send an unlimited number of spam messages.

V. Conclusion

In conclusion, this paper considers social-network based Sybil defenses and divides existing proposals into two categories, namely, Sybil detection and Sybil tolerance. Sybil detection is conceptually simple, application-independent, and easy to apply. However, it relies on strong assumptions about the social graph structure. Moreover, misclassifications are potentially costly, because they can ban a legitimate user from the system, or allow an attacker identity free reign. A detailed understanding of the effectiveness of Sybil detection on real social networks remains an open problem.

Sybil tolerance, on the other hand, allows or denies individual transactions between users, which enables its performance to degrade gracefully in the presence of false positives or negatives. Tolerance schemes can potentially achieve higher accuracy because they consider the pattern and history of user transactions, in addition to the social graph structure, as the basis for allowing transactions. However, Sybil tolerance schemes require application-specific mechanisms that distinguish attack activity from legitimate activity, without

making the system vulnerable to denial-of-service attacks. To date, such mechanisms have been designed and evaluated for specific applications. A general understanding of the class of applications that lend themselves to Sybil tolerance, a systematic design methodology for appropriate mechanisms, efficient implementations of credit networks on social networks at scale, and a study of the social dynamics that would shape the combined social graph/credit network in a Sybil tolerant system all remain open problems.

REFERENCES

[1] J. Douceur, "The Sybil Attack," in *IPTPS*, 2002.

[2] N. Tran, B. Min, J. Li, and L. Subramanian, "Sybil-Resilient Online Content Voting," in *NSDI*, 2009.

[3] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman, "SybilGuard: Defending Against Sybil Attacks via Social Networks," in *SIGCOMM*, 2006.

[4] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao, "SybilLimit: A Near-Optimal Social Network Defense against Sybil Attacks," in *IEEE S&P*, 2008.

[5] G. Danezis and P. Mittal, "SybilInfer: Detecting Sybil Nodes using Social Networks," in *NDSS*, 2009.

[6] N. Tran, J. Li, L. Subramanian, and S. S. Chow, "Optimal sybil-resilient node admission control," in *INFOCOM*, 2011.

[7] D. Quercia and S. Hailes, "Sybil attacks against mobile users: Friends and foes to the rescue," in *INFOCOM*, 2010.

[8] C. Lesniewski-Laas and M. F. Kaashoek, "Whānau: A Sybil-proof Distributed Hash Table," in *NSDI*, 2010.

[9] A. Post, V. Shah, and A. Mislove, "Bazaar: Strengthening user reputations in online marketplaces," in *NSDI*, 2011.

[10] A. Mislove, A. Post, K. P. Gummadi, and P. Druschel, "Ostra: Leveraging Trust to Thwart Unwanted Communication," in *NSDI*, 2008.

[11] D. do B. DeFigueiredo and E. T. Barr, "TrustDavis: A non-exploitable online reputation system," *IEEE E-Commerce*, 2005.

[12] H. Yu, "Sybil defenses via social networks: a tutorial and survey," *SIGACT News*, vol. 42, no. 3, 2011.

[13] B. Viswanath, A. Post, K. P. Gummadi, and A. Mislove, "An Analysis of Social Network-based Sybil Defenses," in *SIGCOMM*, 2010.

[14] M. Mitzenmacher and E. Upfal, *Probability and Computing*. Cambridge, UK: Cambridge University Press, 2005.

[15] "Diaspora*," http://joindiaspora.com.

[16] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and Analysis of Online Social Networks," in *IMC*, 2007.

[17] Y.-Y. Ahn, S. Han, H. Kwak, S. Moon, and H. Jeong, "Analysis of Topological Characteristics of Huge Online Social Networking Services," in *WWW*, 2007.

[18] J. Leskovec, K. Lang, and M. Mahoney, "Empirical comparison of algorithms for network community detection," in *WWW*, 2010.

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

[20] M. E. J. Newman, "The structure of scientific collaboration networks," *PNAS*, vol. 98, no. 2, pp. 404–409, 2001.

[21] "Advogato trust network," http://www.trustlet.org/wiki/Advogato.

[22] M. Mondal, B. Viswanath, A. Clement, P. Druschel, K. P. Gummadi, A. Mislove, and A. Post, "Limiting large-scale crawls of social networking sites," MPI-SWS, Tech. Rep. 2011-006, Nov. 2011.

[23] Z. Liu, H. Hu, Y. Liu, K. W. Ross, Y. Wang, and M. Mobius, "P2p trading in social networks: The value of staying connected," in *INFOCOMM*, 2010.

[24] P. Dandekar, A. Goel, R. Govindan, and I. Post, "Liquidity in credit networks: A little trust goes a long way," in *NetEcon*, 2010.

[25] A. Ghosh, M. Mahdian, D. Reeves, D. Pennock, and R. Fugger, "Mechanism design on trust networks," in *NetEcon*, 2007.

[26] Y.-Y. Ahn, S. Han, H. Kwak, S. Moon, and H. Jeong, "Analysis of topological characteristics of huge online social networking services," in *WWW*, 2007.

[27] S. Seuken and D. C. Parkes, "On the Sybil-Proofness of accounting mechanisms," in *NetEcon*, 2011.

[28] L. R. Ford and D. R. Fulkerson, "Maximal flow through a network," *Canadian Journal of Mathematics*, vol. 8, pp. 399–404, 1956.

[29] A. V. Goldberg and R. E. Tarjan, "A new approach to the maximum flow problem," in *STOC'86*, Berkeley, CA, 1986.

[30] E. A. Dinic, "An algorithm for the solution of the max-flow problem with the polynomial estimation," *Doklady Akademii Nauk SSSR*, vol. 194, no. 4, 1970.

[31] R. E. Gomory and T. Hu, "Multi-terminal network flows," *SIAM*, vol. 9, no. 4, pp. 551–570, 1961.

[32] T. Hartmann and D. Wagner, "Fully-dynamic cut tree construction," Karlsruhe Institute of Technology, Tech. Rep. 2011,25, 2011.

[33] A. Gubichev, S. Bedathur, S. Seufert, and G. Weikum, "Fast and accurate estimation of shortest paths in large graphs," in *CIKM*, 2010.

[34] A. Das Sarma, S. Gollapudi, M. Najork, and R. Panigrahy, "A sketch-based distance oracle for web-scale graphs," in *WSDM*, 2010.