Leveraging Trust and Distrust for Sybil-Tolerant Voting in Online Social Media

Nitin Chiluka
Delft University of Technology
the Netherlands
n.j.chiluka@tudelft.nl

Nazareno Andrade
Universidade Federal de
Campina Grande, Brazil

Johan Pouwelse
Delft University of Technology
the Netherlands

Henk Sips
Delft University of Technology
the Netherlands

ABSTRACT

Due to open membership access, voting on content items in online social media (OSM) is susceptible to Sybil attacks. Malicious attackers can create multiple Sybil identities to outvote the real users of the system. This work proposes a mechanism to defend against such an attack by leveraging (i) trust which is inherent in the social network among users in OSM, and (ii) distrust between honest users, who identify some of the spam content items, and the Sybil identities who promoted them. Modeling trust and distrust in the system as a signed network, our method proceeds in two phases. First, we identify nodes and edges that constrain paths along positive edges between the endpoints of each negative edge. Second, we limit the votes from Sybil voters whose paths to honest nodes pass across these bottlenecks. Our simulation results on datasets of popular OSM show both the feasibility of incorporating distrust alongside trust to defend against Sybil attacks, and that our method outperforms the state-of-the-art approach, SumUp.

Categories and Subject Descriptors

H.2.0 [**Database Management**]: General—*Security, integrity, and protection*

General Terms

Algorithms, Experimentation, Security

Keywords

Voting, Social media, Sybil attack, Signed network

1. INTRODUCTION

Voting is a vital component of online social media (OSM). Votes on content items in OSM (e.g., *like*s in YouTube and Facebook, *favorite*s in Flickr, and *digg*s in Digg) are typically incorporated into many of their central features such as recommendations, 'most popular'-like pages and ranking search results. Besides popularity, voting helps in determining the trustworthiness of content, and at a much lower cost than the actual content analysis and human oversight. This last aspect is particularly significant taking into account the scale and rate at which content is published in current OSM.

Due to open membership access in OSM, voting in such systems is vulnerable to Sybil attacks [7]. Malicious users can create multiple cheap Sybil identities that outvote real users of the system. In this manner, Sybils can promote spam content even to the front page of an OSM by voting positively (e.g., Tube Automator for YouTube [2]), as well as demote trustworthy content by voting negatively on it.

SumUp [24] is the state-of-the-art approach to defend against such an attack. SumUp leverages the social network among users to limit the number of fake votes collected from Sybil identities to O(1) per *attack edge*, which is a trust edge between a malicious user and an honest user. Although this is substantially more robust than directly accumulating votes, we contend the resulting solution still leaves room for considerable damage by attackers. For instance, if malicious users constitute a small fraction such as 1% in a large network of 1 million users, SumUp would face more than 10,000 attack edges. Such a scale of the attack even with conservative estimates drives the main research question of this paper: *can the fake votes collected from Sybils in OSM be reduced to less than O(1) per attack edge?*

In this paper, we leverage both trust- and distrust relationships among users to limit the votes collected from Sybil identities. We model the system as a signed network, where positive edges represent trust relationships that are inherent in the social network among users in OSM, and negative edges represent distrust relationships between honest users, who identify some of the spam content items, and the Sybil identities that promoted them. The fundamental rationale of our approach is that the attack edges constrain the paths along positive edges between the endpoints of each negative edge. This is because any path between an honest node and a Sybil node along positive edges passes through at least one attack edge. Based on this rationale, we proceed in two phases. First, we build a *resistance network* to identify such attack edges and their endpoints. Second, we employ a *vote limiter* mechanism to limit the votes from Sybil iden-

PSOSM '12, April 17 2012, Lyon, France

tities whose paths to honest nodes along positive edges pass through the endpoints of attack edges.

Our evaluation based on simulations on large-scale networks of popular OSM shows both the feasibility of incorporating distrust alongside trust to defend against Sybil attacks, and that our method outperforms the current state-of-the-art approach, SumUp. Our results show that negative edges between a small fraction (10%) of honest users and Sybil identities are sufficient to identify endpoints of attack edges with a high probability. Moreover, our method is resilient to negative edges among honest users constituting one-third of all negative edges in the network. In addition, the fake votes collected from Sybil identities are reduced by a significant 80%, when compared to SumUp, by taking into account one negative edge each from just a quarter of all honest users.

The rest of the paper is organized as follows. Section 2 discusses relevant previous studies. Section 3 presents the definitions, notations, and the datasets used in the rest of the paper. Section 4 describes our approach to the problem at hand in detail. Section 5 discusses our evaluation, while the implications of the results and possible future extensions are described in Section 6.

2. BACKGROUND AND RELATED WORK

Here we review two growing research topics, one each in the fields of web-and-distributed systems and social networks, which have not been studied together in detail.

Social Network-based Sybil Defenses (SNSD). *Open* web-based and distributed systems are susceptible to Sybil attacks [7], where attackers can create multiple cheap Sybil identities in order to outvote and outcompete honest users. To defend against such attacks, a large body of work leverages social networks by incorporating their properties such as inherent trust relationships among users and graph structure into the designs of social network-based Sybil defense (SNSD) schemes [5, 15, 22–24, 28, 29].

Each of these schemes makes two fundamental assumptions. First, although an attacker can create arbitrary number of identities, she cannot establish arbitrary number of trust relationships (*attack edges*) with honest users, since forming a trust relationship requires high social engineering cost. This leads to a *sparse cut* between Sybil region containing malicious identities and non-Sybil/honest region containing honest users in the graph, which is then exploited by these schemes. Second, a social network graph is *expander*-like [4] or *fast-mixing* [20] in that a random walk in the graph quickly reaches a stationary distribution. Hence, a short random walk starting from an evaluating node in the non-Sybil region rarely escapes into the Sybil region.

In the context of voting in online content systems, SumUp [24] is the state-of-the-art Sybil-tolerant scheme [25] that leverages the social network among users to bound the number of fake votes cast by Sybils. Given a *vote collector* (VC), SumUp first defines a *vote envelope* by distributing limited number of credits along the edges in a breadth-first fashion starting from VC. Then, VC collects all the votes from voters within this envelope. Finally, VC collects the votes from voters outside this envelope who can find a distinct path (i.e., no two paths share a common edge indicating sufficient credit) to a node within this envelope. As a result, SumUp limits the number of fake votes collected from Sybils – *attack capacity* – to O(1) per attack edge with a high probability. As argued in Section 1, although SumUp is substantially more robust than directly accumulating votes, the resulting solution still leaves room for considerable damage by attackers. We also argue that the number of attack edges used in SumUp's evaluation [24] is conservatively small (e.g., 100 attack edges in a million-node network), particularly for very large networks of the scale of the real-world OSM.

Although all the proposed SNSD schemes have considered positive trust relationships among users to defend against Sybil attacks, only a few have explored the *distrust* factor. MobId [22] is a Sybil detection scheme [25] that operates in a mobile network of devices. A device maintains a network of friends containing honest devices and a network of foes containing suspicious devices, using which it determines whether to accept or reject an unknown device. SumUp [24] incorporates negative feedback from the VC to modify the credit distribution as well as penalize the links from VC to the malicious nodes. Nevertheless, neither study examines the distrust relationships among users other than those from the evaluating node. This implies that an honest node which has few distrust relationships to adversaries, a newcomer for instance, will be vulnerable to attacks irrespective of others' knowledge about the latter.

Signed Network Analysis. Many online social networks comprise not only positive/trust relationships such as friendships among users but also negative/distrust relationships that indicate enmity or antagonistic feelings toward one another. Such networks are referred to as *signed social networks* (in short, *signed networks*). Most of these online social networks keep the distrust relationships such as 'report as spam/abuse', unlike friendships, *private*. However, a few such as Slashdot make these negative links in the form of 'foes' and 'freaks' *publicly* accessible to the outside world.

Recent studies on signed networks have predominantly focused on the *edge sign link prediction problem*: given a social network with signs on all edges, how accurately can we determine the sign of a hidden edge? In one study, Leskovec *et al.* [14] employ social psychology theories of balance and status to predict the signs of edges. In another study [13], they develop a machine learning approach based on various combinations of triads for sign prediction. Kunegis *et al.* propose a number of spectral analysis techniques for sign prediction [11], and later extend them to clustering and visualization [12] using Laplacian matrices of the signed graphs. More recently, DuBois *et al.* [8] propose a promising technique for sign prediction which combines path-probability trust inference algorithm and spring embedding to infer network distance between nodes. Their rationale is that a node-pair with a positive edge attract each other, while a negative edge makes them repel.

Propagation of distrust, unlike trust, is a tricky issue. For instance, an enemy of an enemy is not necessarily a friend, while a friend of a friend can be considered trustworthy. Guha *et al.* [9] propose that trust can propagate multiple steps whereas distrust propagates a single step. Kerchove *et al.* [6] propose PageTrust, along the lines of PageRank, incorporating the knowledge of negative links while performing the random walk. In addition, various other studies explored propagating distrust in the network [3, 18, 30], none in the context of attacks.

Table 1: Datasets of popular OSM. Notations: $|V|$ and $|E|$ represent the number of nodes and edges in the network respectively; $\delta_{0.9}$ is the 90^{th} percentile diameter. The last column gives an example of a vote on an item in the corresponding OSM.

OSM	$\|V\|$	$\|E\|$	$\delta_{0.9}/\delta_{1.0}$	Vote/Item
YouTube [19]	1,134,890	2,987,624	7/15	like/video
Flickr [19]	1,624,992	15,476,835	7/14	favorite/photo
Digg [24]	542,140	4,039,025	6/14	digg/story
Facebook [27]	3,097,165	23,667,394	6/8	like/page
Epinions [14]	75,877	405,739	5/10	review/product
Slashdot [14]	82,168	543,381	5/9	comment/story

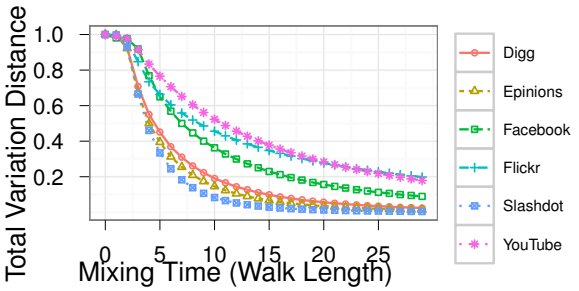


Figure 1: Mixing times of social network graphs in OSM.

3. PRELIMINARIES

Notations and Definitions. Let $G = (V, E)$ denote an undirected and signed graph, where V and $E \subseteq V \times V$ are the set of nodes and edges in the graph respectively. An edge between nodes i and j is represented by (i, j, s), where s represents the sign of the edge. Subgraph $G^+ = (V^+, E^+)$ comprises only positive edges ($E^+ \subseteq E$), while subgraph $G^- = (V^-, E^-)$ comprises only negative edges ($E^- \subseteq E$); also, $V^+, V^- \subseteq V$.

Given a *sample* $S \subset V$ of nodes, the *neighborhood* $N(S)$ is a set of nodes connected to S by positive edges, i.e, $N(S) = \{w \in V - S : \exists v \in S \text{ s.t. } (v, w) \in E^+\}$. *Expansion* of S measures the number of neighbors of S relative to its size: $\frac{|N(S)|}{|S|}$. *Expansion quality* of S measures the extent to which the neighborhood of S covers the network: $\frac{|N(S)|}{|V-S|}$.

Datasets and Mixing Times. Table 1 summarizes the datasets of popular OSM[1] that we use in our paper, to study the problem of voting in OSM under Sybil attack.

We examine the fast-mixing [20] property of these social network graphs, which is an assumption that many SNSD schemes [15,28,29] make, in order to explore the plausibility of incorporating this feature into the design of our method. The mixing times of four out of the six datasets were found to be 'fast' in [21]. For the sake of completeness, we examine the mixing times for all the six datasets using the same methodology as [21]. Specifically, given an initial distribution $\pi^{(i)}$ at node i and transition matrix $P^{(0)}$ which is the adjacency matrix with normalized rows, we compute the total variation distance $|\pi - \pi^{(i)}P^{(t)}|_1$ at each step (random walk length) t where π is the stationary distribution.

Figure 1 plots the total variation distance versus the walk length, averaged over 1000 initial distributions. The mixing times of all the graphs converge to stationary distribution in the order of $O(\log |V|)$, within a total variation distance of 0.2. This shows that the social network graphs of these OSM have 'fast' mixing times, which implies that these graphs exhibit expander-like properties [4]. In the next section, we discuss how the expansion property of the graphs is incorporated into our design.

4. METHOD

In this section, we first formalize the problem along with a brief overview of our approach, and then detail the two main components of our method.

4.1 System Model and Approach

We consider the problem of voting on content items in OSM under Sybil attack, inspired by SumUp [24]. *Votes* on a particular item in OSM is a salient factor considered in determining its popularity.

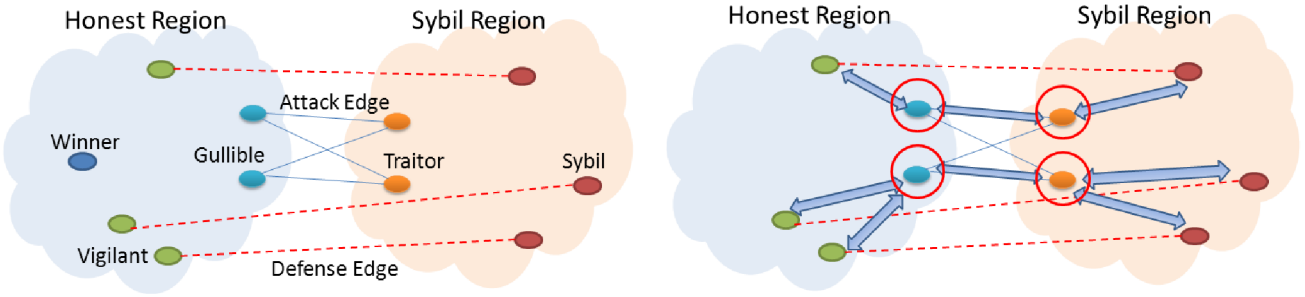
System Model. Our model considers four types of users in the network. First, *traitors* are a fraction of users that turn opportunistic to game the system using Sybil identities to inject and promote spam content through fake votes. Honest (ie. non-traitors) users who have at least one trust relationship with a traitor are considered *gullible*, while honest users who have trust relationships only with other honest users are considered *winners*. Finally, a fraction of honest users who are *vigilant* either form foe-like relationships with adversaries or report some of the spam content items as spam. As a result, a plausible interpretation is that these vigilant users and Sybil identities which have promoted spam content have fundamental disagreement on the quality of this content. Another perspective is that the relationship between these two sets of entities is that of *mutual distrust*.

We model such a system by an undirected signed graph G (Fig. 2(a)), where nodes represent either users or identities and edges represent relationships between nodes. A positive edge represents a trust relationship, while a negative edge represents a distrust relationship. *Attack edges* are the trust relationships between gullible and traitor nodes, while *defense edges* are the distrust relationships between vigilant nodes and Sybil identities that have promoted spam content.

Adversaries can change the graph structure of the Sybil region besides add and delete the Sybil identities to suit their strategies. Since off-the-shelf algorithms in the area of signed networks predominantly depend on graph structure for their effectiveness, these algorithms cannot be directly applied to defend against Sybil attacks.

Goal and Assumptions. Our main goal in this paper is to limit the number of votes collected from Sybils to less than the number of attack edges. We make the following assumptions that influence our design. First, G^+ is connected, i.e., any two nodes in G are reachable along positive edges. Second, the number of traitor nodes is significantly less than the number of honest nodes. As a result, the number of attack edges is limited. Third, creating new attack edges is difficult due to high social engineering cost, similar to the assumption previous SNSD schemes [15, 28, 29] make. Lastly, all

[1]We thank Alan Mislove, Jure Leskovec, Ben Zhao, and Nguyen Tran for making these datasets available.



(a) System Model

(b) Conflict at gullible and traitor nodes

Figure 2: [Best viewed in color.] A pictorial illustration of the system and approach. 2(a) Honest region comprises winner (dark blue), vigilant (green) and gullible (light blue) nodes. Sybil region comprises traitor (orange) and Sybil (dark red) nodes. Defense edges between vigilant and Sybil nodes are represented by dotted red lines. 2(b) Paths along positive edges (blue thick double-headed arrows) corresponding to each defense edge are constrained by gullible and traitor nodes, where red circles indicate high conflict.

honest nodes have a consensus on spam content items, and thus a negative edge exists only between an honest- and a Sybil node. We relax this last assumption in Section 5.1.2.

Approach Overview. We make two observations based on the model and above assumptions. First, attack edges between gullible- and traitor nodes can be viewed as 'bridges' connecting honest- and Sybil regions, with gullible- and traitor nodes being 'bridge nodes' (Fig. 2(a)). Second, these bridge nodes and bridges constrain the paths along positive edges between two endpoints of a negative edge, with one endpoint each in the honest- and Sybil regions (Fig. 2(b)). Notice that any path along positive edges corresponding to a negative edge between an honest- and a Sybil node passes through a gullible- and a traitor node.

Based on this rationale, we (i) use defense edges to build a *resistance network* that identifies bridge nodes and bridges (Sec. 4.2), and (ii) employ a *vote limiter* mechanism to subsequently limit the votes from Sybils whose paths to honest nodes pass across these bridge nodes and bridges (Sec. 4.3).

4.2 Resistance Network

We construct a *resistance network* R by incorporating distrust relationships among the nodes. We model this network by a weighted graph of G^+ where the weight of each edge is initially set to 0. For each negative edge $(i, j, s < 0) \in E^-$ in G, we find a path between i and j along positive edges ($\in E^+$). For each edge on this path, we increase the weight of the corresponding edge on the resistance network by 1. As a final result, the weight of an edge represents the number of paths corresponding to negative edges passing through it. This weight can be viewed as the *resistance* of the edge. The resistance of a node v, $R(v)$, then is the summation of resistance of each of its edges. The resistance of a node or an edge can also be interpreted as the measure of its 'conflict centrality' in the graph. That is, nodes and edges with high resistance indicate the extent of conflict across them.

Such a resistance network has a number of desirable characteristics. First, it presents a *global* perspective of resistance of any given node/edge, which is particularly interesting for administrators/moderators. Second, it is fairly immune to *dynamics* in the network such as additions/deletions of nodes/edges. Third, in personalized voting systems, each vote collector can use the same 'fundamental' resistance network to adapt her vote aggregation mechanism to suit her specific needs. Since the resistance network needs to be computed only once and if it can be done in a scalable manner, leveraging it would not hamper system *scalability*.

We now describe the construction of the resistance network in a scalable fashion. It is important to note that finding a path between any two nodes of a negative edge along positive edges in a large network in an efficient manner is non-trivial. To address this challenge, we proceed in two phases. In the first phase, we build the *backbone* of the network, which is a small connected subgraph comprising nodes that have high connectivity properties to the rest of the graph. In the second phase, we find a path between two endpoints of each negative edge in the graph using this backbone, which is then used to update the resistance network. Following are the details of these phases.

Backbone Construction. We employ the snowball technique to expansion sampling (XSN) [17] (similar to its variant [16]) which constructs a sample S of size-k such that its neighborhood $N(S)$ covers a large portion of the network even for a small sample ($|S| << |V|$). We refer to this sample as the *backbone* of the network. At its core, this technique works by adding a node v from the neighborhood $N(S)$ to S at each step such that v contributes most *new* neighbors to the current sample S. Since the scalability of XSN was not discussed in [16,17] and non-trivial, we develop a dynamic programming method for XSN to allow it to scale for large networks, and discuss its scalability in Sec. 5.3.

Naturally, the efficiency of XSN depends on the network it is applied to. We thus examine how XSN fares on our datasets. We begin with a sample S containing one random node in the graph, and then we iteratively add nodes to S until its size $|S| = 5\%|V|$. We measure the expansion quality of S after each addition of a node. Figure 3 plots the expansion quality versus the sample size relative to all nodes in the graph, averaged over 5 runs. All graphs exhibit high expansion, even for a small sample. For instance, a sample size of just $1\%|V|$ has expansion quality of 0.5 or more, indicating that S is on average within 2 hops from any node ($\in V - S$) in the rest of the network. Also, increasing this sample size reduces the average hop distance between S and $V - S$.

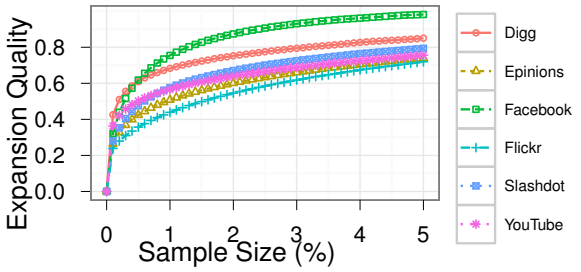


Figure 3: Expansion Quality as the sample grows.

Another benefit of using XSN is that the backbone is a 'representative' sample of the community structure in the network [17]. By adding nodes that best contribute to the expansion of the sample, XSN essentially includes nodes which act as 'bridge nodes' between communities. As a result, the sample consists of nodes from most (if not all) communities in the network. In our context, this implies that the backbone (and surrounding nodes) is likely to contain many of the gullible and traitor nodes.

Pathfinder Algorithm. We exploit the expansion property of this backbone to find a path along positive edges corresponding to a given negative edge with endpoints i and j. We begin two paths P_i and P_j, one at each endpoint, and then add a node each to the paths at every step such that it increases the chances of meeting each other. We now detail this searching process. If the head of P_i (which is the last added element, say v_i) is not in the backbone S, we add to P_i a neighbor of v_i in the large graph which either is in S or contributes to *near* maximal expansion of P_i chosen probabilistically. However, if v_i is on the backbone ($v_i \in S$), we add to P_i a neighbor of v_i from the backbone S which contributes near maximal expansion of P_i. We note here that this reduces the search space, from the large graph to the backbone, by a great extent. Also, if either v_i or its neighbor is in P_j ($v_i \in P_j$ or $N(\{v_i\}) \in P_j$), we merge P_i and P_j which gives us the path between i and j. Similar process is simultaneously carried out on P_j. Section 5.3 shows that this algorithm finds short paths efficiently in large expander-like graphs, where most nodes in the network are one or two hops from the backbone which in turn is itself small.

4.3 Vote Limiter

We now describe the *vote limiter* mechanism, built along the lines of SumUp [24]. Our main aim here is to limit votes from Sybil identities collected by honest nodes. Hence, for simplicity in this paper, we assume a *vote envelope* comprising all honest nodes, and focus on limiting the votes from voters outside this envelope which are essentially traitor- and Sybil nodes.

We define the following. *Attack capability* is equal to the ratio of the collected votes from nodes in the Sybil region and the number of attack edges between the honest- and Sybil regions. *Capacity* of a node v is equal to the maximum number of vote paths that can pass through the node, which is half its degree: $C(v) = \lceil |N\{v\}|/2 \rceil$. *Effective Capacity* of a node v is equal to the 'downgraded capacity' of the node taking into account its resistance $R(v)$: $E(v) = \left\lceil C(v) * \alpha^{R(v)} \right\rceil$, where $\alpha \in [0, 1]$ is a dampening factor which decreases the ability of a node to allow vote paths to pass through it based on its resistance. We use $\alpha = 0.8$ if the node is not on the backbone, and $\alpha = 0.95$ otherwise, since many paths along positive edges corresponding to negative edges pass through the backbone nodes more often than through the other nodes.

We now detail the vote collection process. First, each node is assigned a credit equal to its effective capacity, and each edge outside the vote envelope is allocated a credit of 1 unit. Next, a vote from a voter outside the vote envelope is collected if there exists a distinct path (i.e., no two paths of voters share a common edge) from the voter to a node within the envelope, with each node and edge on the path having credit greater than zero. When a vote is collected in this manner, the credit of each node and edge on this path is decreased by 1 unit. The final number of collected votes divided by the number of attack edges gives the attack capability of the adversaries.

5. EVALUATION

We evaluate our method in this section, aiming to address the following questions:

- How well can we distinguish winner, gullible and traitor nodes by incorporating negative edges (defense edges) between honest- and Sybil regions?
- How resilient is our method when there are negative edges among honest nodes in addition to defense edges?
- How effective is our method to limit the votes collected from Sybil identities to less than the attack edges?
- How scalable are the techniques of the backbone construction and the pathfinder algorithm?

The experimental setup used in the rest of the section is as follows. Given a dataset, let the size of the respective graph be N. We first create a partition of this graph into Sybil- and non-Sybil (honest) regions, along the lines of [15]. We select a small fraction of nodes ($3\%N$) as traitors uniformly at random; the remaining nodes constitute the honest region. For each attack edge to a gullible node in the honest region, a traitor creates a Sybil identity and forms a positive edge with it. We note here that creating more Sybil identities and/or forming edges among themselves is not going to improve their attack capability, since their vote paths to the vote envelope are constrained by attack edges. Next, we construct the backbone of the graph (of size $3\%N$) starting from a random node in the honest region. Subsequently, we build the resistance network by incorporating defense edges ($e_D = 20\%N$).

5.1 Resistance

We examine the resistance of three types of nodes – winner, gullible, and traitor – in the following two scenarios.

5.1.1 Impact of Defense Edges

In the first experiment, we vary the number of defense edges between nodes in the honest- and Sybil regions. For now, we do not consider negative edges within the honest region. We construct the resistance network incorporating these edges, and collect a random sample of 5000 of these nodes.

Figure 4 plots the distribution of resistance versus defense edges, in 5 runs. We make the following observations. First, the resistance of gullible and traitor nodes is overall much higher than that of winner nodes. This enables one to differentiate winner nodes from gullible and traitor nodes with a high probability. In 3 out of the 6 datasets, it is possible to consistently distinguish the three types of nodes, and in 2 other it is possible to distinguish winner and traitor nodes. Second, the resistance of all nodes increase as the number of negative edges increase. Traitor nodes in particular, and gullible nodes to some extent, gain higher resistance when more honest nodes are vigilant. Third, the resistance of gullible nodes is often less than that of traitor nodes, suggesting that the paths along positive edges corresponding to each negative edge are constrained by traitor nodes more than by gullible ones. This is because the number of traitor nodes is typically lesser than the number of gullible nodes.

5.1.2 *Resilience to Negative Edges in Honest Region*

We now add negative edges among honest nodes, in addition to defense edges. In this experiment, we keep the defense edges fixed ($e_D = 20\%N$).

Figure 5 plots the resistance as we vary the number of negative edges among honest nodes. We draw the following conclusions. First, the resistance of traitor nodes still remains higher than that of winner and gullible nodes generally, even after including ($10\%N$) half the number of defense edges. This shows the resilience of our method to negative edges among honest nodes. Second, as expected, the resistance of winner nodes is higher in this experiment than the previous one (see Fig. 4), albeit much lesser than that of gullible and traitor nodes. Third, the resistance of gullible nodes is comparable or higher than that of the traitor nodes only in a few cases (e.g., Flickr and Digg). This is not a major concern since our main aim to reduce the flow across these high resistance nodes, be they traitors or gullible.

5.2 Attack Capability

We examine the robustness of the vote limiter mechanism by varying the number of defense edges. Figure 6 plots the attack capability versus defense edges. The attack capability is 1 when there are no defense edges, like the scenario of SumUp when the vote collector has no distrust relationships. Our method reduces the attack capability by nearly 80% taking into account defense edges, one each from only a quarter of all honest users, indicating the resilience of our mechanism to Sybil attack. Furthermore, the attack capability decreases as defense edges increase, implying that the more vigilant honest nodes the fewer the fake votes collected from Sybil identities.

Let us now consider the scenario of Digg as an example, where we report values averaged over 5 runs. When the traitor nodes constitute $3\%N = 16,264$, the number of attack edges and thus the number of Sybil voters is $e_A = 221,073$. As a result, the number of fake votes collected is equal to e_A, when there are no defense edges ($e_D = 0\%N$). By using $e_D = 25\%N$ defense edges, our mechanism limits the number of fake votes collected by nodes in the honest region to $15\% * e_A = 34,405$, which is nearly double the number of traitors. When $e_D = 100\%N$, fake votes collected by honest nodes is $6\% * e_A = 15,029$, which is similar to the number of the traitor nodes. We note that the number of collected fake votes can be reduced further by a method that decreases the effective capacity more 'aggressively' than the one used in this paper (refer to Sec. 4.3).

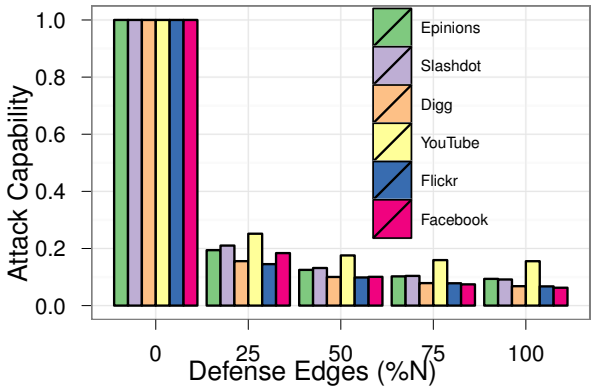


Figure 6: [Best viewed in color.] Attack capability of Sybil identities as the defense edges increase.

5.3 Scalability

We examine the scalability of the two building blocks of the resistance network: backbone and pathfinder algorithm. We perform our experiments on an HP commodity laptop with Intel Core i7 1.6GHz processor and 4GB RAM running Windows 7. The code is written using SNAP library [1], and compiled as a single-threaded Visual C++ application.

The left-half of Table 2 shows the computation times of the backbone using Expander Sampling (XSN) technique as we increase the sample size, averaged over 5 runs. Our dynamic programming-based implementation of XSN is observed to scale to large networks. Even a sample size of 5% in a 1.6-million node network of Flickr takes less than 20 seconds. A 3-million node network of Facebook takes just over a minute to sample over 150,000 nodes for the backbone, whose expansion quality is close to 1 (Fig. 3) implying that every node is nearly just a hop away from the backbone. Although the backbone is not likely to change frequently, the system designers can choose to compute it periodically at a very low cost.

The right-half of Table 2 shows the average query times and path lengths for the pathfinder algorithm based on 1-million queries of random node-pairs. We make the following observations. First, the average query time in these large networks is less than 1-millisecond, even for a 3-million node Facebook network. Second, as expected, finding a path takes longer time as the graph size increases in general. However, the average query time on Facebook network is faster than that on Flickr even though the latter network is half the size of the former. The main reason is that the expansion quality of Facebook is much higher for the same sample size percentage than that of Flickr (Fig. 3), which allows better searching capability.

6. IMPLICATIONS AND FUTURE WORK

Our study opens scope for improvements in existing systems (the top three points below) as well as research in multiple directions (the rest), some of which we discuss here.

Vote envelope creation: Creating the vote envelope around

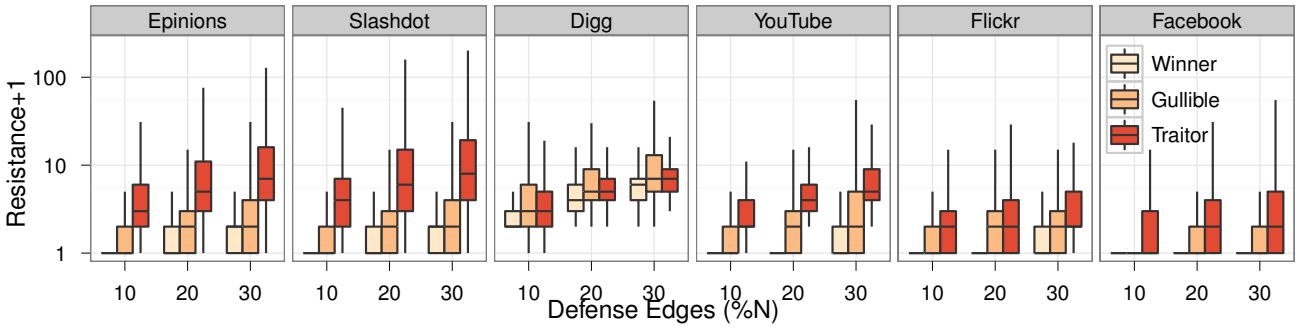


Figure 4: [Box-and-whisker plot.] Resistance of winner, gullible and traitor nodes as the negative edges between honest- and Sybil regions increase. Notice that the vertical axis is in log-scale, and its measure is 'Resistance+1' instead of 'Resistance' so as to avoid plotting logarithm of Resistance=0. Also, note that Resistance+1=1 for winner nodes in some cases is represented by a line, indicating all winner nodes plotted have Resistance=0.

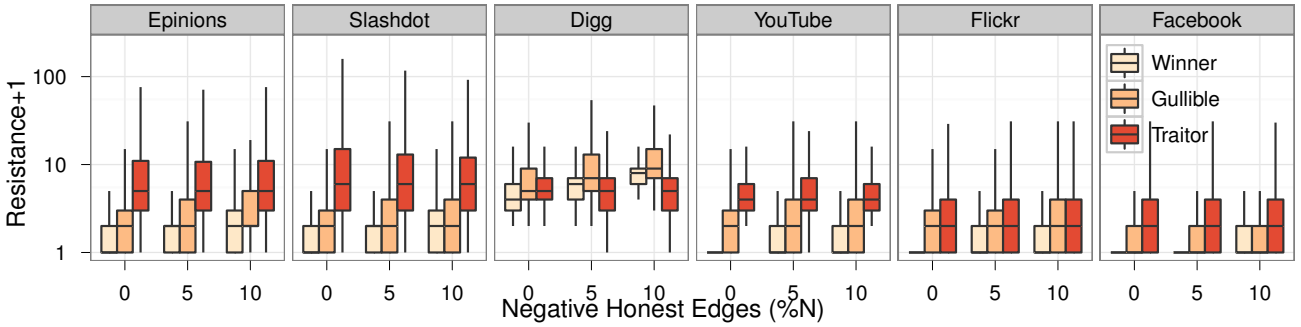


Figure 5: Resistance of winner, gullible and traitor nodes as the negative edges among honest nodes increase.

Table 2: Computation times of backbone (in seconds) and pathfinder algorithm (in μ seconds).

	Backbone			Pathfinder	
OSM	1%	3%	5%	Query/μs	Avg Path
Epinions	0.262	0.324	0.377	89	6.773
Slashdot	0.418	0.496	0.561	104	7.460
Digg	3.160	3.887	5.257	320	6.656
YouTube	3.378	5.447	9.032	409	7.804
Flickr	10.127	13.440	18.202	734	7.413
Facebook	38.394	49.339	70.499	505	13.680

a given vote collector is not the main focus in this paper. Based on our method, a natural extension for envelope creation is to consider both minimal expansion and the resistance network. Starting from the vote collector in the sample S representing vote envelope, at each step, we add a node $v \in N(S)$ in S's neighborhood to S such that it minimizes (i) the expansion of the updated S, thus reducing its conductance [26] as well as its bias toward high-degree nodes [10], and (ii) the resistance of the updated S, thereby including nodes with low resistance and conflict.

Decentralized setting: While our method was discussed in a centralized setting in this paper, we believe that it can also be extended to distributed systems. The relationship between a node and its *blacklist* containing other nodes that have acted maliciously with it in the past can be viewed as a defense edge in our context. One can incorporate such defense edges, for instance, in an existing social network-based Sybil-resilient DHT system such as Whanau [15], in order to further improve its robustness. In Whanau, as the number of attack edges increases, the number of messages required to find an authentic value for a given key also increases significantly. *Failed lookups* can be used as defense edges to further reduce (i) the ability of the adversaries to pollute the DHT routing tables, and (ii) the lookup times and network overhead to find an authentic key-value pair when there are many attack edges.

Multiple-pair short path problem: Our pathfinder algorithm, based on maximum expansion-based search [16,17], shows promise in finding a path between any two nodes in a large network in an efficient manner. While the all-pair shortest-path problem is well studied, the usage of shortest paths in the real-world scenarios is not always desired. Nodes and edges on these shortest paths can get overloaded if paths of many simultaneous queries of node-pairs coincide on a few nodes and edges. Our algorithm offers an alternative particularly in the scenarios where the constraint of finding *the* shortest paths is less strict.

Formal analysis: Our study shows the feasibility of incorporating negative edges, not necessarily from the evaluating node, in the network to defend against Sybil attacks. The formal analysis of why our method performs considerably better than the state-of-art approach – SumUp – which uses mainly positive edges along with negative edges just from vote collector for its robustness is a part of the future work. Can we build generic algorithms taking into account both trust and distrust against Sybil attacks?

Privacy-preserving: Most current SNSD schemes take into account all positive edges in the network for their robustness. Considering the promising results of our approach, a research question that naturally follows is: by incorporating defense edges, can a few positive links be hidden to preserve privacy and still attain a comparable performance to the a priori?

Consequence of negative edges: A downside of using negative edges is that it might antagonize the environment. Competing nodes for popularity, for instance, will use Sybil attacks and negative edges to reduce the 'reputation' of each other. A recent study [6] suggests the use of undirected negative edges instead of the directed ones so that it disincentivizes such 'unethical' behavior.

7. CONCLUSIONS

In the context of voting in online social media, this paper leverages trust and distrust relationships to limit votes from Sybil identities. The fundamental rationale of our approach is that attack edges, gullible- and traitor nodes constrain the paths along positive edges between two endpoints of a defense edge. Based on this rationale, we first build a resistance network to identify such bottlenecks using negative edges, and then employ a vote limiter mechanism to limit the votes from Sybil identities whose paths to honest nodes pass through gullible and traitor nodes.

Our simulation results on large-scale OSMs point to the feasibility and the scalability of this approach. Defense edges from a small fraction (10%) of vigilant nodes to Sybil identities are sufficient to differentiate the winner nodes from the gullible- and traitor nodes with a high probability. Our method is resilient to negative edges among honest users. The attack capability is around 0.2 in our method, compared to 1 in SumUp, which uses one defense edge each from a quarter of all honest users. Finally, the pathfinder algorithm on which our approach is based can find a path given any two nodes in large networks in less than a millisecond on a commodity laptop, showing that the resistance network can be built in a scalable fashion.

Acknowledgments. This work was partially supported by the European Commission's 7th Framework Program through the P2P-Next and QLectives projects (grant no. 216217, 231200), by the CAPES/Nuffic program DRI/CGCI 006/2011, and by CNPq (grant 480855/2010-2).

8. REFERENCES

[1] Snap network analysis library. http://snap.stanford.edu/.
[2] Stat gaming services come to youtube. http://techcrunch.com/2007/08/23/myspace-style-profile-gaming-comes-to-youtube/.
[3] C. Borgs, J. Chayes, A. T. Kalai, A. Malekian, and M. Tennenholtz. A novel approach to propagating distrust. WINE '10.
[4] F. Chung. Spectral graph theory. American Mathematical Society, 1997.
[5] G. Danezis and P. Mittal. Sybilinfer: Detecting sybil nodes using social networks. NDSS, 2009.
[6] C. De Kerchove and P. V. Dooren. The pagetrust algorithm: How to rank web pages when negative links are allowed? SDM '08.
[7] J. R. Douceur. The sybil attack. IPTPS '01.
[8] T. DuBois, J. Golbeck, and A. Srinivasan. Predicting trust and distrust in social networks. SocialCom '11.
[9] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. WWW '04.
[10] A. M. Kakhki, A. Hannak, A. Mislove, and R. Sundaram. Mitigating sybil attacks on content rating systems. SOSP '11.
[11] J. Kunegis, A. Lommatzsch, and C. Bauckhage. The slashdot zoo: mining a social network with negative edges. WWW '09.
[12] J. Kunegis, S. Schmidt, A. Lommatzsch, and J. Lerner. Spectral analysis of signed graphs for clustering, prediction and visualization. SDM '10.
[13] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Predicting positive and negative links in online social networks. WWW '10.
[14] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Signed networks in social media. CHI '10.
[15] C. Lesniewski-Laas and M. F. Kaashoek. Whanau: a sybil-proof distributed hash table. NSDI '10.
[16] A. S. Maiya and T. Y. Berger-Wolf. Expansion and search in networks. CIKM '10.
[17] A. S. Maiya and T. Y. Berger-Wolf. Sampling community structure. WWW '10.
[18] A. Mishra and A. Bhattacharya. Finding the bias and prestige of nodes in networks based on trust scores. WWW '11.
[19] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. IMC '07.
[20] M. Mitzenmacher and E. Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005.
[21] A. Mohaisen, N. Hopper, and Y. Kim. Keep your friends close: Incorporating trust into social network-based sybil defenses. INFOCOM '11.
[22] D. Quercia and S. Hailes. Sybil attacks against mobile users: friends and foes to the rescue. INFOCOM '10.
[23] N. Tran, J. Li, L. Subramanian, and S. S. Chow. Optimal sybil-resilient node admission control. INFOCOM '11.
[24] N. Tran, B. Min, J. Li, and L. Subramanian. Sybil-resilient online content voting. NSDI '09.
[25] B. Viswanath, M. Mondal, A. Clement, P. Druschel, K. P. Gummadi, A. Mislove, and A. Post. Exploring the design space of social network-based sybil defenses. COMSNETS '12.
[26] B. Viswanath, A. Post, K. P. Gummadi, and A. Mislove. An analysis of social network-based sybil defenses. SIGCOMM '10.
[27] C. Wilson, B. Boe, A. Sala, K. P. Puttaswamy, and B. Y. Zhao. User interactions in social networks and their implications. EuroSys '09.
[28] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao. Sybillimit: A near-optimal social network defense against sybil attacks. IEEE Symposium on Security and Privacy '08.
[29] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. Sybilguard: defending against sybil attacks via social networks. SIGCOMM '06.
[30] C.-N. Ziegler and G. Lausen. Propagation models for trust and distrust in social networks. *Information Systems Frontiers, Volume 7, Issue 4-5,*, 2005.