On the Mixing Time of Directed Social Graphs and Security Implications

Abedelaziz Mohaisen Huy Tran Nicholas Hopper Yongdae Kim

Department of Computer Science and Engineering
University of Minnesota – Twin Cities, Minneapolis, MN 55455
{mohaisen, huy, hopper, kyd} @ cs.umn.edu

ABSTRACT

While many social graphs are directed by nature, applications that use social graphs are often evaluated on undirected versions of these graphs. Manipulating a social graph in this manner, however, may change important properties like the mixing time, a critical parameter for applications such as Sybil defense and anonymous communication. In this paper we measure the mixing time and behavior of several directed graphs and their undirected counterparts. Counter-intuitively, we find that some directed graphs are faster mixing than their undirected counterparts, whereas the general pattern is that directed graphs are slower mixing than undirected ones. To relate to the applications suggested in the literature, we measure how directionality of edges in several social graphs impact these applications, and find that evaluation on the undirected graphs always overestimates the security provided by these schemes.

Categories and Subject Descriptors

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

General Terms

Security, Design, Experimentation

Keywords

Social networks, Sybil defenses, Mixing time, Measurement

1. INTRODUCTION

Many graphs in general, and social graphs in particular, are directed by nature. However, applications built on top of social networks, including Sybil defenses [1, 3, 10, 11, 12, 26, 28, 29], information routing and dissemination [5, 13, 14], and anonymous communication [18, 2] require mutual relationships which produce undirected graphs. When undirected graphs are used as testing tools for these applications to bring insight on their usability and potential deployment, directed graphs are converted into undirected graphs by omitting edge directions [1, 12, 26, 17] or by augmenting the underlying social graph [28, 16].

For example, Cai and Jermaine [1], Lesniewski-Laas and Kaashoek [12], Yu et al. [28], Viswanath et al. [26], Mohaisen et al. [17], and Mittal et al. [16], among others, have altered directed social graphs by either omitting edge directions entirely or by considering a connected subgraph in which an edge is established between two nodes if it is symmetric (i.e., an edge that exists in both directions). It has been claimed that the fraction of edges added by this process is small, since the majority of edges among nodes are already symmetric (e.g., see section 9.1 in [12] and section 5.1 in [26]). Also, it has been claimed that the majority of nodes remain in the largest connected component when considering edges in the undirected graph only if they are symmetric in the directed one [28, 16]. Furthermore, in most of these works it has been either explicitly argued [12, 1, 17] or implicitly assumed [28, 16] that converting a directed graph to an undirected graph will not significantly influence the graph structure nor the mixing time, the length of the random walk needed to randomly reach every node in the graph with probability proportional to the degree distribution. The quality of the mixing time has been a crucial property in these systems, where security and performance of them are based on the mixing time. In particular, most of these applications assume "fast-mixing" social graphs for their operation.

Unfortunately, it is not clear how the process of altering these graphs affects the quality of their mixing time. Although, the intuition is that directed social graphs (for which the mixing time is well-defined) would have different—and potentially slower—mixing time than undirected graphs. Motivated by the lack of prior work on this problem, we investigate mathematical tools for measuring the mixing time of directed social graphs and its associated error bounds. We use these tools to measure the mixing time of several benchmarking directed social graphs and to understand the difference in the mixing time quality between directed graphs and their undirected counterparts. We then measure how this difference impacts two applications built on top of social networks: a Sybil defense mechanism and an anonymous communication system.

The property used in both applications addressed in this paper is the mixing time of social graphs, where these graphs are assumed to be "fast-mixing". Characterizing the mixing time by a single parameter using the second largest eigen-

ASIACCS '12, May 2–4, 2012, Seoul, Korea.

value modulus (SLEM) method [22] or by the mixing time's mathematical definition, which is shown in Eq. (2), is insufficient to capture the richer mixing behavior and patterns of social graphs [27, 17]. Since this mixing behavior is the actual property utilized by many applications built on top of social networks [27], we measure it by computing the *mean* of the mixing time for walks originated from several sources in a social graph.[1] To achieve that, we use tools from the fixed point theory [7] to closely estimate the mixing time of directed graphs and its associated error bound. This estimation process is needed because, to the best of our knowledge, the literature provides no method for computing a closed-form expression of the stationary distribution of walks on directed graphs—which is required for computing the mixing time.

In order to make our results relatable to other works in the literature, we experiment with directed graphs that are widely used in this context for evaluating systems built on top of social networks. By converting directed graphs to undirected ones, and measuring the mixing time of both cases, we find that undirected graphs are generally faster mixing than directed graphs. We also find that evaluation of the aforementioned systems on undirected graphs always overestimates the security of these systems.

Finally, while it might be possible to use directed graphs for building certain applications on top of social and other networks, in this work we emphasize that *we do not advocate the use of these graphs for such applications.* We rather examine a method that is widely used in the literature for modifying graphs and using them to verify the effectiveness of certain security and privacy applications. We highlight the difference in the mixing characteristics of both types of graphs, directed and undirected, and explore how this difference influences the performance and security guarantees of these applications in a "what-if" scenario. In particular, we address the following questions. What is the quality of the mixing time property in the directed graphs? What would be the performance of these applications if the directed graphs with their original property are used instead of the altered graphs? What security guarantees do these applications provide using the original property in the directed graphs before altering them? How do these guarantees compare to the guarantees claimed on the undirected counterparts?

1.1 Contributions

In this paper we make two primary contributions. First, we investigate tools for measuring the mixing time of directed graphs, and use these tools to measure the mixing time of four directed social graphs, before and after graph conversion. We show the significant impact of edge directionality on the mixing time. Second, we study the impact of the quality of the mixing in these graphs, with and without edge directionality, on two applications from the literature, namely a Sybil defense and an anonymous communication system. Both applications rely on the mixing time and its quality in social graphs for their operation. In total, both of our contributions are motivated by a widely-used method in the security community for altering graphs without paying attention to how this impacts the underlying property in the original graph and security guarantees of applications built on top of them.

[1]Notice that this abuse of notation is only to simplify the exposition of our results. The reader should keep in mind that the *mixing time* is the *maximum* walk length to reach close to the stationary distribution from any source in the graph; see section 3 for details and (2) for the definition.

1.2 Organization

The rest of this paper is organized as follows. In section 2 we introduce the related work. In section 3, we investigate a method for efficiently computing the mixing pattern in directed graphs, and a method for bounding the error due to estimating the stationary distribution of such graphs. In section 4 we review the datasets and the method used for their preprocessing. In section 5, we outline the first part of our findings by measuring the mixing of several social graphs, before and after omitting directionality of their edges. In section 6, we investigate the impact of the difference in the mixing time on Sybil defenses and anonymous communication systems. Our concluding remarks appear in section 7.

2. RELATED WORK

To the best of our knowledge, there is no prior work on measuring the mixing time of directed graphs. Although, several works measure the mixing time in undirected graphs generated by omitting directions in naturally directed graphs [6, 17]. As such, there is no prior work on examining how the difference in the mixing time in naturally directed graphs and their undirected counterparts affects the performance of social network-based designs. On the other hand, there has been several works on designing systems on top of social graphs that exploit the mixing characteristics and other social network properties. Some of these works alter directed graphs to undirected ones using the methods mentioned in section 1. In the following, we review some of these works.

2.1 Measurements of The Mixing Time

The "fast-mixing" property of social graphs has been utilized for building social network-based Sybil defenses and for reasoning about their security guarantees. However, none of the authors of these defenses considered measuring the mixing time in social graphs directly. To address this issue, Mohaisen et al. [17] and Dellamico et al. [6] measured the mixing time of several social graphs. Mohaisen et al. [17] neglected edge directions entirely by following prior works which argued or assumed that altering graphs in that way will not affect social graphs and their properties [12, 26, 28]. Accordingly, all social graphs including directed ones are considered undirected [17]. Dellamico et al. [6] measured the mixing time of four large social graphs, and while they used some directed graphs it is unclear what techniques they used for computing their mixing time; suggesting that a conversion of the graph as in other works [17, 26, 12] was likely performed before measuring the mixing time.

2.2 Systems Exploiting the Mixing Time

The significance of the mixing time of social graphs is because of two categories of applications that use it for their operation: Sybil defenses, and anonymous communication Systems. Sybil defenses that utilize social networks include SybilGuard [29], SybilLimit [28], SybilInfer [4], SumUp [25], GateKeeper [24], Whanau [12], and X-Vine [16], among others—a summary and comparisons of most of these works can be found in the recent work of Haifeng Yu [27]. All of these defenses require social networks to be "fast-mixing", infor-

mally meaning that a short random walk originated from any honest node in a social graph is sufficient to reach every honest node in the graph with a probability proportional to the node's degree. The fast mixing property is required in all of these designs for both performance and security reasons [27].

Anonymous communication systems on top of social networks include "anonymity in the wild" [18] and Drac [2]. In section 6 we elaborate on "anonymity in the wild" [18] and a Sybil defense (called SybilLimit [28]), which we use to demonstrate our results and findings.

2.3 Prior Work and Graph Preprocessing

The pre-processing methods of directed graphs in the prior literature motivate this work. While some directed social graphs are used to demonstrate Sybil defenses in many of the above works, they are converted to undirected graphs by entirely neglecting edge directions [1, 12, 26] or by considering largest connected component with symmetric edges only [28, 16]. Each method alters the social graphs differently. While the first method is likely to add edges to guarantee symmetry of edges in the resulting graph, the second method is likely to trim nodes to ensure that no asymmetric edges are in the resulting graph—trimmed nodes are connected to other nodes in the graph with asymmetric edges.

We observe that both methods may result in a well mixing graph from a graph that violates the mathematical definition of the mixing time (i.e., a graph for which no mixing time is defined; see section 3 for more details on the mathematical definitions). In both methods, the original directed graph might be weakly connected for which no mixing time is defined, yet adding additional edges or removing nodes would make it mix well. This observation has two implications. First, comparing the mixing time of graphs after processing them to the mixing time of the original graph using either of the two methods might not be possible. Second, by violating the mathematical definition the ultimate result might be obvious and uninteresting: certain nodes in a weakly connected graph will never be visited and the mixing time of the directed graph is likely to be larger than the modified graphs.

To this end, we limit our discussion to the case where the largest strongly connected component (SCC) is used to represent the directed graph. We note that the largest SCC of a graph includes the graph produced using the second preprocessing method. Furthermore, we note that a comparison between directed and undirected graphs produced using the first method only makes sense when the original directed graph has the SCC property. We mainly use the first method of pre-processing, and lightly discuss the second method, when comparing directed and undirected graphs in terms of mixing and performance of applications on social networks.

3. DIRECTED GRAPHS' MIXING TIME

Informally, conditions required for defining the mixing time in undirected and directed social graphs are quite similar. Random walks on a graph have a stationary distribution that is used for bounding these walks' distribution if the graph is connected. In order for the mixing time to be well defined on a directed graph, it needs to be a *strongly connected component* (SCC). In an SCC, there exists a path between every pair of nodes. Computing the SCCs of a graph in linear time is done using Tarjan's algorithm [23], which we use in this work. We use the *largest* SCC (in size) as a representation of each directed graph and measure its mixing time (see justification in section 2). In this section, we present theoretical tools we need for measuring the mixing time of directed graphs. We define the mixing time for directed and undirected graphs in section 3.1, estimate the stationary distribution in section 3.2, and bound the error in our measurements due to the stationary distribution estimation in section 3.3.

3.1 Defining Directed Graphs' Mixing Time

Let $\mathbb{G} = (V, E)$ be a directed and strongly connected graph, where $|V| = n$ and $|E| = m$. Let $\mathbf{A} = [a_{ij}]^{n\times n}$ be the adjacency matrix of $\mathbb{G}$, where $a_{ij} = 1$ if there is an edge from node v_i to node v_j in $\mathbb{G}$ (denoted by $v_i \rightarrow v_j$), and 0 otherwise. Let $\deg(v_i)^-$ be the out-degree of node v_i. We define the transition probability matrix $\mathbf{P} = [p_{ij}]$ where $p_{ij} = 1/\deg(v_i)^-$ iff $v_i \rightarrow v_j$, and 0 otherwise. In a clean matrix form, $\mathbf{P} = (\mathbf{D}^-)^{-1}\mathbf{A}$, where $\mathbf{D}^-$ is a diagonal matrix in which the ii-th element is defined as $\sum_j a_{ij}$. The stationary distribution, π, of random walks on $\mathbb{G}$, is defined formally as a distribution that is invariant to transition probability. That is, $\pi = \pi\mathbf{P}$. Unlike the stationary distribution of an undirected graph, which is characterized in a closed form in terms of the degree distribution (This is, $\pi = [\deg(v_i)/2m]^{1\times n}$ for $1 \leq i \leq n$), the stationary distribution in directed graphs has no closed-form expression.

Formally, π is defined in Theorem 1 (which also applies to the case of undirected graphs as well).

THEOREM 1. *Let $\mathbf{P}$ be the probability transition matrix of a Markov chain that is aperiodic and strongly connected, defined on a graph $\mathbb{G}$. Then,*

$$\lim_{t\to\infty} \mathbf{P}^t = \mathbf{P}^\infty \tag{1}$$

where $\mathbf{P}^\infty$ in (1) is defined as an $(n\times n)$ matrix of identical rows, where each row equals to π, the stationary distribution of every walk on $\mathbb{G}$.

Same as in undirected graphs, the mixing time of a directed graph $\mathbb{G}$ is defined as:

$$T(\epsilon) = \max_i \min\{t : |\pi - \pi_i\mathbf{P}^t|_1 < \epsilon\}, \tag{2}$$

where π_i is the delta distribution of 1 concentrated at the i-th position in a $(1 \times n)$ probability vector representing the probability distribution when starting a walk from node v_i, and $|\cdot|_1$ is the total variation distance defined as $\frac{1}{2}\sum_j |\pi(j) - \pi_i^t(j)|$ (where $\pi_i^t = \pi_i\mathbf{P}^t$). Notice that the mixing time of the entire graph, as defined in (2), is the longest walk starting from the worst initial distribution to reach the worst reachable part of the graph [27] and to achieve a fixed total variation distance from the stationary distribution π.

While the mathematical definition in Eq. (2) is necessary to characterize the mixing time of the entire graph, richer patterns of mixing are expressed by the distribution of the mixing time obtained after t-step walks starting from different initial distribution in the graph. This distribution can be further characterized by the mean or median ϵ at a given walk length t. Both of the mean and median of the mixing time are usually far from the "worst-case scenario" expressed in Eq. (2) [17], and are usually representative to the quality of the mixing property required for the operation of Sybil defenses [27]. In this work, we are interested in

both characterizations of the mixing time: the distribution (mean and median) and the worst case as per the definition (maximum).

3.2 Estimating the Stationary Distribution

Computing π as in Eq. (1) is only theoretically possible, since it requires computing $\mathbf{P}^t$, where $t \to \infty$. Even when setting t to some large number that is an order of magnitude of the size of the graph (e.g., 10^6 for 1000 nodes graph), the process becomes both time and space inefficient for typical social graphs due to their large size. In part, this inefficiency is due to losing sparsity of $\mathbf{P}$ as t grows thus requiring multiplication and storage of a non-sparse large matrix ($\mathbf{P}^t$). This inefficiency turns into infeasibility as the size of the graph grows to a few thousands of nodes, which is smaller than benchmarking directed graphs used in this study. Accordingly, other methods are required for computing π, or a good estimate of it. In the following we devise such method without modifying the structure of $\mathbf{P}$.

We observe that regardless of the initial distribution, every walk on a strongly connected graph ultimately converges to the stationary distribution π. We could further make the total variation distance between the ideal (unknown) stationary distribution and the distribution of a random walk beginning from an arbitrary node in the graph arbitrarily small. In other words, given an arbitrary initial distribution π_i and the transition matrix $\mathbf{P}$ of a strongly connected graph, we can compute $\pi_i^\ell = \pi_i \mathbf{P}^\ell$ such that $|\pi_i^\ell - \pi|_1 < \delta$, where δ is close to 0, for some large walk length ℓ.

The convergence of π_i^ℓ to π is a guaranteed property of aperiodic Markov chains on connected graphs [22]. Without knowing π, one can use $\pi' = \pi_i^\ell$ as an estimate for π, for some large ℓ. Such π' is sufficient to measure a very close estimate of the mixing time of directed graphs, and to serve the purpose of our measurement in understanding the difference between the mixing patterns in directed and undirected graphs. Notice that computing $\pi_i^\ell = \pi_i \mathbf{P}^\ell$ does not require computing $\mathbf{P}^\ell$. We can iteratively compute π_i^ℓ using a vector-matrix multiplications of π_i and $\mathbf{P}$ by observing that $\pi_i^\ell = \pi_i^{\ell-1}\mathbf{P}$, where we can benefit from the sparsity of $\mathbf{P}$. In our measurements, we set π_i to a uniform distribution (i.e., $\pi_i = [1/n]^{1\times n}$) and ℓ to a large number that makes error in our measurements arbitrarily small. In the following subsection, we elaborate on the method we use for setting ℓ.

3.3 Determining the Proper Parameters

While one can argue that setting ℓ to a large number would make the distribution of a random walk of length ℓ starting from an arbitrary distribution close to the stationary distribution—and thus the former distribution can be used as an estimate of the latter distribution, the distance between both distributions is required to identify an upper bound on the error in the measured mixing time. Fortunately, the fixed point theory [7] provides tools to estimate the required length of the walk ℓ in order to achieve a certain distance from the stationary distribution (also known as the "fixed point" distribution). The following definition and theorem outline the main results needed for bounding the distance between the estimated stationary distribution, after ℓ iterations, and the ideal stationary distribution.

DEFINITION 1 (CONTRACTION MAPPING [7]). *Let (X, d) be a non-empty and complete metric space. A mapping $T : X \to X$ is said to be a contraction mapping if there exists a real number $\lambda < 1$ such that $d(T(\pi_x), T(\pi_y)) \leq \lambda d(\pi_x, \pi_y)$ for all $\pi_x, \pi_y \in X$ (where $d(.)$ is a metric, or distance, defined on points in the space X; e.g., $|\cdot|_1$).*

Table 1: The (original) datasets used for deriving directed and undirected graphs and measuring their mixing time with their statistics (number of nodes, number of edges, and the number of strongly connected components).

Dataset	# nodes	# edges	# SCC
Slashdot [9]	77,360	905,468	6,724
Epinion [20]	75,879	508,837	42,176
Wiki-vote [8]	7,115	103,689	5,816
Gnutella [21]	10,876	39,994	6,560

THEOREM 2 (FIXED POINT THEOREM [7]). *Let (X, d) be a non-empty and complete metric space and let T be a contraction mapping defined on X. For that mapping, there exists a fixed point π s.t. $T(\pi) = \pi$. To find the fixed point, an iterative mapping of an initial point π_i would result in a sequence of points $\pi_i^0, \pi_i^1, \pi_i^2 ... \in X$, where the following inequality describes their convergence rate to the fixed point:*

$$d(\pi, \pi_i^{w+1}) \leq [\lambda/(1-\lambda)] d(\pi_i^{w+1}, \pi_i^w) \quad (3)$$

Using Theorem 2 and by recalling that the matrix multiplication, which represents random walks on the graph, is a contraction mapping [7], the following inequality characterizes the walks convergence rate to the stationary distribution (at length $w+1$):

$$|\pi - \pi_i \mathbf{P}^{w+1}|_1 = \delta \leq [\mu/(1-\mu)] |\pi_i \mathbf{P}^{w+1} - \pi_i \mathbf{P}^w|_1, \quad (4)$$

where μ is the second largest eigenvalue of the transition matrix $\mathbf{P}$. By setting $\ell = t + 1$, for large t, we make $\frac{\mu}{1-\mu}|\pi_i \mathbf{P}^\ell - \pi_i \mathbf{P}^{\ell-1}|_1$ arbitrarily small, and thus make the distance between π and $\pi' = \pi_i \mathbf{P}^\ell$ within an acceptable range that does not influence the measured mixing time. Using (4), we bound the error in measurements starting from any node v_j after t steps (using the triangular inequality) as:

$$|\pi - \pi_j \mathbf{P}^t|_1 \leq |\pi' - \pi_j \mathbf{P}^t|_1 + |\pi' - \pi|_1 \quad (5)$$

Eq. (5) tells that we can bound the error in the measured mixing time using the estimated stationary distribution (π') and the bounded distance between it and the stationary distribution in Eq. (4). Since we can make the right hand side in Eq. (4) arbitrarily small (we make it $< 10^{-15}$ in our measurements), and since we are interested in a region where the computed distance is within 10^{-1} to 10^{-12}; c.f. section 5, the error due to estimating the stationary distribution is considered negligible.

4. DATASETS AND DATA PREPROCESSING

4.1 Datasets and Their Prior Uses

We use four datasets in this work as shown in Table 1. Three of the four datasets represent social networks, whereas the fourth represents relationships in a peer-to-peer file sharing system. In Slashdot Zoo [9], a directed edge between two

Table 2: The largest strongly connected component (SCC) of each of the different graphs in Table 1. Note that the percent in parenthesis of each graph correspond to the relative size of the largest SCC in the original graphs.

Dataset	# nodes (%)	# edges (%)
Slashdot	70,355 (90.94)	818,310 (90.37)
Epinion	32,223 (42.47)	443,506 (87.16)
Wiki-vote	1,300 (18.27)	39,456 (38.05)
Gnutella	4,317 (39.69)	18,742 (46.86)

Table 3: The undirected graphs resulting from converting the largest SCCs with statistics in Table 2. Note that the percent in parenthesis correspond to the number of added edges to make the directed graph undirected by having all edges in both directions. The percent is the number of added edges divided by *twice* the total number of edges in the resulting undirected graph.

Dataset	# nodes	# edges	Added edges (%)
Slashdot	70,355	459,620	100,930 (10.98)
Epinion	32,223	342,013	240,520 (35.16)
Wiki-vote	1,300	36,529	33,602 (45.99)
Gnutella	4,317	18,742	18,742 (50.00)

nodes indicates that the first node tags the second node as a friend. In Epinion [20], a "who-trust-whom" online social network, an edge between two nodes indicates that the first node has tagged the second node as a trusted node. In wiki-vote [8], a link between two nodes indicates that the first node has voted for the second node. In Gnutella [21] an edge between two nodes indicates that is first node (host) is connected to second one. Notice that we do not advocate the use of these (interaction) graphs for building Sybil defenses and anonymous communication systems. We use the fact that these graphs are already benchmarks for testing such applications to validate our results based on them.

We note that three of these datasets (Wiki-vote, Epinion, and Slashdot) were previously used for measuring the mixing time [17], two (Wiki-vote and Gnutella) were used for demonstrating the efficiency of a social network-based Sybil defense [1], and one (Wiki-vote) was used for analyzing social network-based Sybil defenses [26].

4.2 Graphs Conversion

In each directed graph, we compute the largest SCC in order to satisfy the connectivity condition required for measuring the mixing time. The SCC of each graph and its relative size compared to the original graph are shown in Table 2. The largest SCC varies in size, and ranges from as low as 18% of total nodes in the original graph (38% of edges, as in Wiki-vote) to as high as 90% of nodes (and edges, as in Slashdot). For each graph, we compute π' as in section 3.3, and make the distance to π a negligible factor.

We convert each SCC to an undirected graph. In each SCC we first exclude self-loops, if any existed. Then, given the adjacency matrix of the self-loops-free SCC, $\mathbf{A}$, we compute the adjacency matrix of the corresponding undirected graph as $\mathbf{A} \vee \mathbf{A}^T$, where $\mathbf{A}^T$ is the transpose of $\mathbf{A}$ and $\vee$ is an bitwise logical *or* operation. The resulting graphs and their statistics are shown in Table 3. While the graphs before and after conversion maintain the same number of nodes, some additional (directed) edges are added to create edge-symmetry and to produce undirected graphs. The number of the added edges in each of the undirected graphs as a percent of the total number of edges ranges from as low as 11%, as in Slashdot, to as high as 50%, as in Gnutella and Wiki-vote.

4.3 Notes on Graph Conversion

We emphasize several aspects of the method used for graph conversion in this paper. First, we note that recent works on social network-based Sybil defenses [12, 1, 26, 16, 28] have altered social graphs in several ways, some which are similar to what we used in this paper (see section 2 for more details). These works and the methods used in them for altering graphs are the main motivation of this paper. Second, obtaining the largest SCC from the directed graph as used this paper is for a mathematical *necessity*, despite potentially trimming a portion of the original directed graph (see Table 1 for details). Third, our method—by first obtaining the largest SCC and then converting the directed graph to undirected—is *milder* than both literature methods explained in section 2.

Compared to the first method in the literature [12, 1, 26], our method ensures that the mixing is well-defined on both graph types, directed and undirected. Compared to the second method [16, 28], our method trims less nodes in the directed graph, and still brings insight on the difference between both graph types when used for these applications.

We note that prior work [15] suggests that the majority of nodes belong to the SCC in large-scale social graphs. Thus, our findings may apply to these graphs when used by omitting directions according to the first method as well.

5. RESULTS AND DISCUSSION

Now we proceed to measure the mixing time of directed graphs shown in Table 2 and undirected graphs shown in Table 3. We consider each graph with and without directions (as explained above) and compute the stationary distribution appropriately. For directed graphs we compute the stationary distribution using the method in section 3.2 and for undirected graphs we compute it using the method in section 3.1.

5.1 Methodology

For each graph in tables 2 and 3, we measure ϵ—the total variation distance between the stationary distribution and the accumulated distribution—after w steps. We initially set w to 1 and increase it for a fixed to 50, with steps of 1. A walk length of 50 is sufficient to characterize the mixing rate of walks on all graphs we used. Using these measurements, we then restrict the walk length to 10. A walk of up to that length is used in the literature for demonstrating the operation of applications of top of social networks.

For each graph, we repeat this process by beginning from 1000 different nodes as sources of initial distributions in order to capture the pattern of mixing in these graphs. Finally, to eliminate bias caused by starting from different initial distributions, we fix each of the 1000 nodes (labels)

used for both measurements. We first randomly select 1000 nodes for experimenting with the directed graphs, and use them again for estimating the mixing time of the undirected graph, after graph conversion.

5.2 Main Results and Discussion

The main results of measuring the mixing time are shown in Figure 1 and Figure 2. Figure 1 plots a comparison of the *mean* mixing characteristics between graphs before and after modifications. We plot the mean ϵ at each walk length, as we increase the random walk length from 1 to 50. The mean is computed over all ϵ's obtained by starting from each of the 1000 initial distributions (nodes). Figure 2 shows the maximum ϵ as we increase t, making t the mixing time for the computed ϵ by definition (see section 3.1).

First, we compare the mixing characteristics across different datasets. We notice that different graphs, and regardless to edge directions, have different mixing characters. For example, whereas slashdot and Epinion have similar mixing pattern, Wiki-vote mixes faster than both of them. Furthermore, we notice that Gnutella mixes faster than all of the three other graphs, regardless to edge direction.

While some of the difference in the mixing characteristics across different datasets is attributed to graph density and size, as shown in Table 2 and Table 3, other crucial factor of determining the mixing time of these graphs is their structure. For example, we notice that a graph like Wiki-vote, shown in Figure 1(b), which has several social hubs [8], mixes faster than other graphs that do not exhibit such hubs clearly (e.g., Slashdot in Figure 1(c) and Epinion in Figure 1(a)). Also related to the structure, we notice that a random graph like Gnutella, shown in Figure 1(d), mixes faster than other graphs. This is initially not surprising, since random graphs are good expanders, which are also fast mixing [19]. The interesting observation, though, is that the same pattern of "fast-mixing" characteristics is preserved in Gnutella even when only considering a modified version of it; a strongly connected component. Furthermore, it is interesting to see how such expander "notion" applies to directed graphs (where the general notion and mixing time studies considered undirected expander graphs only).

The context of the graphs may affect their structure, and thus influence their mixing time. For example, we notice that two graphs with similar social contexts, like Slashdot and Epinion, have similar mixing times for longer walks (at average). This, however, is not shown in any of the other graphs. This similarity in the mixing pattern is exhibited clearer with short random walks, as shown in Figure 3.

In these measurements, and as we increase w, we find all possibilities of comparison exhibited—when comparing directed to undirected graphs: 1) no difference as in Wiki-vote graph in Figure 1(b), 2) undirected graphs mix better than directed graphs as in Slashdot and Epinion graphs in figures 1(c) and 1(a), and 3) directed graphs mix better than undirected graphs as in Gnutella in Figure 1(c).

While the two first cases are anticipated, the third case is surprising. It is unexpected to find that directed graphs mix better than undirected graphs in general. However, given that Gnutella has a special structure (originally a fixed degree graph), we hypothesize that omitting directions would alter the graph structure in unfavorable way to the mixing time. A potential case would be that adding more edges to the graph would divert long random walks on the graph. We further notice that this behavior is reverted when we restrict the random walk length to less than 10 steps.

Now we turn our attention to measuring the mixing time per the definition in Eq. (2)—the results are shown in Figure 2. While the mean computed over all ϵ's for a given random walk length is meaningful for the average node—especially when evaluating systems built on top of social networks, it does not capture the worst case scenario which is of interest to the theoretical guarantees proposed in prior work in the literature that utilizes social graphs [28, 29]. For example, for an anonymous communication system or asocial defense suggested on top of social networks, it is always better to prove guarantees with lower-bounds—which is actually used by Yu et al. in their Sybil defenses [28, 29]. Lower bound guarantees are satisfied by the mixing time in the definition in Eq. (2), and computed as the maximum ϵ among all distributions for a given walk length. For the same experiment above, we plot the maximum ϵ for different walk lengths in Figure 2.

While some of the patterns exhibited are consistent for the maximum ϵ in Figure 2 with the patterns in the average case in Figure 1, we observe that this pattern is switched in Slashdot and Wiki-vote. Particularly, we observe no difference in Slashdot, while the undirected Wiki-vote graph is faster mixing than the directed one, contrary to what is observed in the average case. Mohaisen et al. [17] argue that similar cases happen in other graphs, and may not be representative to the real mixing nature of these graphs. These cases happen when a set of nodes are sparsely connected to the majority of other nodes in the graph, thus slowing the mixing time of the whole graph [27].

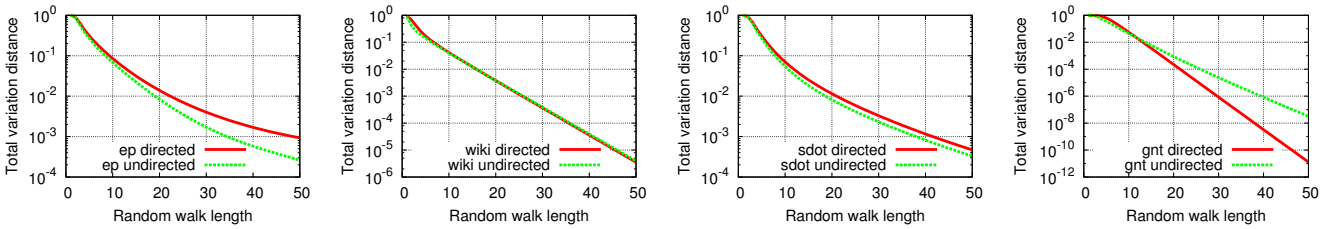
We notice that Sybil defenses and anonymous communication systems built on social networks rely on shorter random walks than those shown in our experiments. Typically, a random walk used in a Sybil defense is in the order of $\log n$, where n is the size of the graph, which translates into 10 to 15 for a graph of $100,000$ nodes [28]. We examine the mixing characteristics of these graphs in the needed range, for a random walk of proper length for the operation of these applications. We consider a walk length from 1 to 10 with 1 step increment and zoom-in the prior measurements discussed earlier for longer random walks (with the same settings as explained before). Interestingly, we find that *all* undirected graphs are faster mixing than directed graphs in that region of walk length, as shown in Figure 3. Surprisingly, we find that for a walk length of 5—the length we use for demonstrating impact of the mixing time on applications on social networks; see section 6—ϵ changes from 0.7 in the directed Gnutella to 0.3 in the undirected one. This difference, and in other graphs in Figure 3, has a great significance on social network-based applications.

6. IMPLICATIONS ON APPLICATIONS

In this section we explore the impact of the difference in the mixing of directed and undirected graphs on potential applications built on top of social networks that exploit the mixing time as their operation property. We consider SybilLimit [28] as a state-of-the-art representative work for Sybil defenses and "anonymity in the wild" [18] as an anonymous communication system.

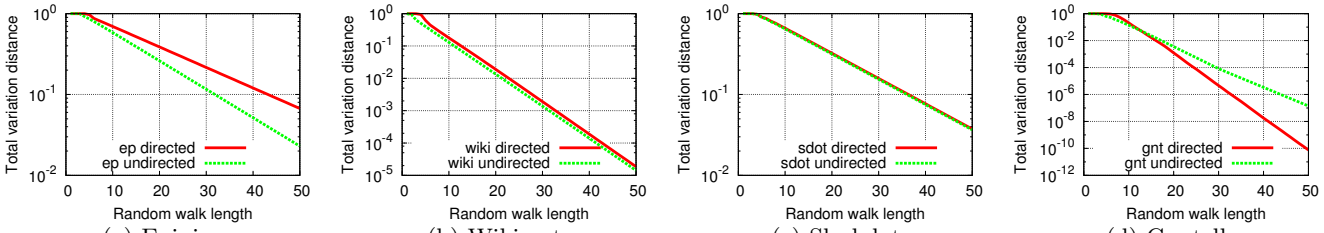
6.1 Sybil Defenses

In the following we examine how the difference in the mix-



(a) Epinion (b) Wiki-vote (c) Slashdot (d) Gnutella

Figure 1: The (mean) mixing time of directed graphs before and after omitting directions of edges. Each of the figures corresponds to the mean of measurements of ϵ that corresponds to the given random walk length for 1000 different initial distributions of sources in each social graphs.



(a) Epinion (b) Wiki-vote (c) Slashdot (d) Gnutella

Figure 2: The (max) mixing time of directed graphs before and after omitting directions. Note that, by definition, these figures correspond to the mixing time of the social graph. While it bounds the mixing of all sources for which the mixing time is measured, it is less strictly representative of the quality needed for applications such Sybil defenses [17, 27].

ing time of directed and undirected social graphs impacts the performance of Sybil defenses operated on top of them. To demonstrate this impact, we choose SybilLimit [28].

6.1.1 SybilLimit

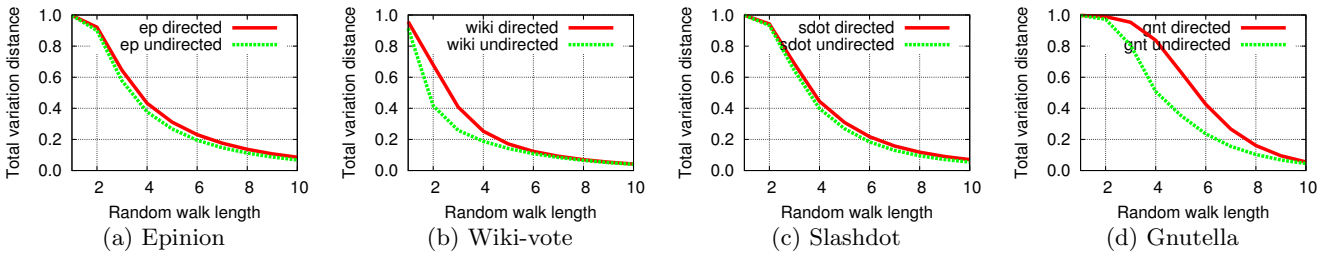
In SybilLimit, each node samples r edges in the graph as "witnesses", where $r = r_0\sqrt{m}$, by running r independent instances of random walks each of length $w = O(\log n)$. Under certain assumptions on the graph's mixing characteristics, there is an overwhelming probability that the sampled subsets of honest nodes in the social graph will have a non-empty intersection, which would be used for suspect verification. Formally, if the social graph is fast mixing—i.e., has a mixing time of $O(\log n)$—then probability of the last node/edge visited in a walk of length $O(\log n)$ drawn from the edge/node stationary distribution is at least $1 - \frac{1}{n}$ (Theorem 1 in [28]). Accordingly, by setting r_0 properly, one can use the birthday paradox to make sure that the intersection between two sampled subsets of edges (by two honest nodes) is non-empty with an overwhelming probability. Furthermore, given that the social graph is fast mixing, and the number of attack edges—edges that connect Sybil with honest nodes—is limited, probability for random walks originated from honest region to dishonest region are limited. The impact of such "escaping tails" on the operation of the defense is further marginalized using a "balance condition" which ensures that accepting a suspect would not cause a spike of the number of accepted suspects via a certain edge in the graph. Chances of dishonest nodes being accepted by sampling honest edges is limited, and bounded by the number of attack edges.

To evaluate the performance of SybilLimit, we use two evaluation metrics. We use acceptance rate of honest nodes, which is governed by the mixing characteristics of the social graph, and the acceptance of dishonest (Sybil) nodes, which is determined by both the mixing characteristic and the number of attack edges in the graph.

6.1.2 Results and Discussion

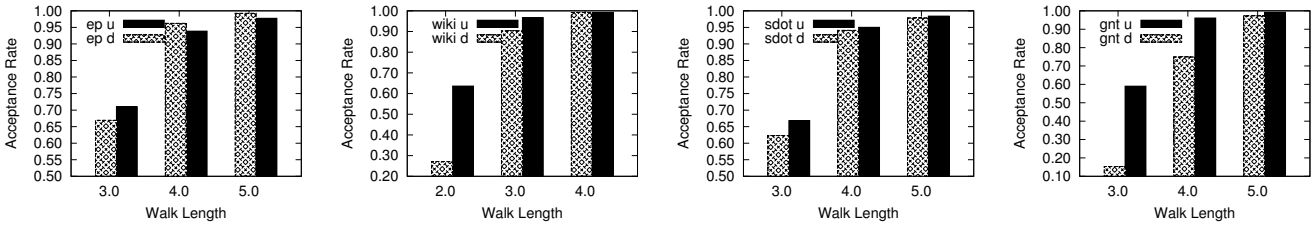
In each directed and undirected graph, we use 1000 random nodes as suspect/verifier pairs (total of 499,500 pairs of suspect/verifier in each graph). We then run SybilLimit and compute the average of both metrics of evaluation for the 1000 random verifiers. To further reduce randomness in the results and increase accuracy, we perform each experiment three times, by changing the 1000 random verifiers, and compute the average of the outcomes. In the following we summarize the main results and findings.

The results are shown in Figure 4 for the average honest nodes acceptance rate and in Figure 5 for the average of the total number of accepted Sybil nodes per the given number of attack edges, as the number of attack edges increases. In each of the figures, we make several observations. First, we observe that both metrics, the number of accepted honest and Sybil nodes are strongly correlated with the mixing characteristics of the different graphs, which is not surprising since the mixing time is the main property used for this application. By comparing Figure 3 to Figure 4 we observe that shorter random walks in faster mixing graphs are sufficient to accept all honest nodes. By comparing Figure 3 to Figure 5 and by fixing the length of attack edges (e.g., 100), we also observe that the number of Sybil identities introduced at average is strongly correlated with the mixing time; faster mixing graphs introduce more Sybil identities



(a) Epinion (b) Wiki-vote (c) Slashdot (d) Gnutella

Figure 3: The (mean) mixing time of directed graphs before and after omitting directions for short walks.



(a) Epinion (b) Wiki-vote (c) Slashdot (d) Gnutella

Figure 4: Acceptance rate of the honest nodes (suspects) by honest verifiers in directed and undirected graphs. Note that all undirected graphs outperform directed, except in Epinion.

for the same number of attack edges.

We find that almost always directed graphs accept less honest nodes (by honest verifiers) as in Figure 4 and more Sybil identities (per attack edge) as in Figure 5. For example, we notice that for a random walk length 3 SybilLimit would accept about 95% of the honest nodes when operated on an undirected graph, where only 90% of these nodes are accepted when operated on the directed graph. Also, where operating SybilLimit on an undirected Gnutella graph with random walk length of 4 would allow accepting 95% of the honest nodes, it would only allow accepting 75% of the honest nodes on the same graph when considered directed.

An observation in favor of directed graphs is made on Epinion (in Figure 6(a)) when using walk length of 4. SybilLimit on a directed Epinion graph would accept roughly 95% honest nodes by honest verifiers (at average), whereas it would accept only 90% in the undirected graph case. We notice that, a lot more Sybils are introduced per attack edge for the same case (thrice the number of Sybils introduced per attack edge in directed graphs).

We notice that introduced Sybils per attack edge are always more in directed graphs than in undirected graphs as shown in Figure 5. In some of these cases, the number of Sybils in directed graphs is three times more than in undirected ones—as in Figure 6(a), whereas other cases have almost no difference—as in Figure 6(a). Sizable, but not too large difference happens in the two other cases.

6.2 Anonymous Communication Systems

The idea of mixers over social links is very simple [18]. In these systems [18, 2], users recruit their social acquaintance to relay their traffic and to provide anonymity to them. In the nutshell, each node (user) forwards her own traffic to her friends, and friends forward that traffic to their friends, and so on, for a certain number of hops, e.g. w. The number of hops w is a system-wide parameter, which is determined by the security level desired in the system. The anonymity is defined for two parties; the sender and the receiver of traffic (we follow the same model in [18] for defining the anonymity of both parties).

6.2.1 Anonymity Measures

For a sender, the anonymity defined in terms of the *anonymity set* is n, thus the entropy of the probability distribution of any node being the sender is $\log_2(n)$—same for both directed and undirected graphs. On the other hand, the anonymity set for a node being the receiver is determined by the probability distribution achieved after the fixed number of hops w used in the system. Let the distribution of the final node selected in a *random walk* after w hops be $\pi_i^w = \pi_i \mathbf{P}^w$, where $\pi_i^w = [\pi_i^w(j)]^{1\times n}$ (π_i is an initial distribution). The anonymity of the receiver of the traffic (the last hop in the walk) is measured by the entropy H_w, which is given as

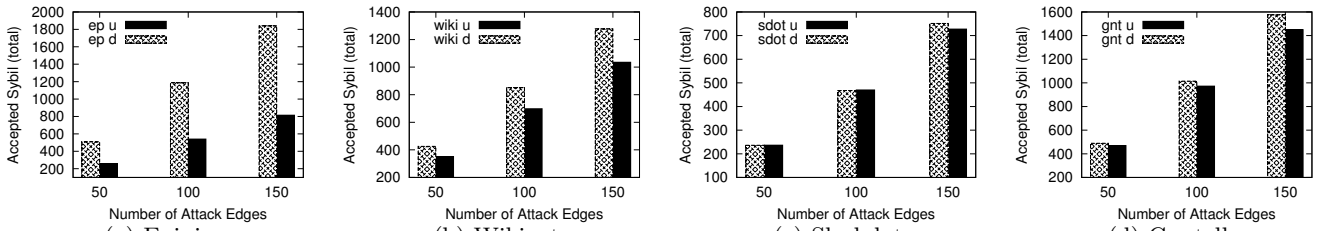
$$H_w = -\sum_{j=1}^{n} \pi_i^w(j) \log_2 \pi_i^w(j) \qquad (6)$$

Using the entropy in Eq. (6), we define the *anonymity set* $A_w = 2^{H_w}$. The maximum entropy and anonymity set for a walk on a graph are achieved with the probability distribution of that walk as it approaches the stationary distribution.

We use $\overline{H_w^d}$ and $\overline{A_w^d}$ for the average entropy and anonymity sets in a directed graph, while $\overline{H_w^u}$ and $\overline{A_w^u}$ are used for the average entropy and anonymity sets in an undirected graph. We define the average entropy and anonymity sets for 1000 random walks starting from different sources (see below).

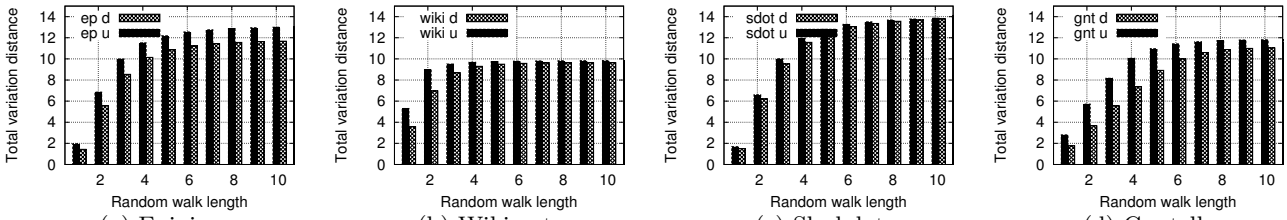
6.2.2 Results and Discussion

Same as when we measured the mixing time and the performance of SybilLimit, we use 1000 initial distributions for each social graph. An initial distribution at node v_i in a graph of n nodes is a $1 \times n$ probability distribution of 1 at the i-th entry and 0 otherwise. We increase the length of



(a) Epinion (b) Wikivote (c) Slashdot (d) Gnutella

Figure 5: Acceptance rate of the dishonest (Sybil) nodes by honest verifiers in directed and undirected graphs. Used walk lengths are 3, 4, 4, and 5 for Wiki-vote, Epinion, Slashdot, and Gnutella, respectively.



(a) Epinion (b) Wiki-vote (c) Slashdot (d) Gnutella

Figure 6: The mean entropy of random walks distributions on graphs before and after omitting directions for varying walk lengths (H_w^u vs. H_w^d for varying w values).

the random walk w from 1 to 10 with step increments of 1. At each time step we compute the probability distribution $\pi_w(i)$ of the random walk as w increases, from which we compute the entropy as in above. We then compute the average entropy for the 1000 initial distributions (denoted in Table 4 as $\overline{H_w^d}$ and $\overline{H_w^u}$ for directed and undirected graphs, respectively). We plot the results of the entropy measurements in Figure 6. We observe a consistent pattern with measurements of the mixing time in section 5 and earlier results of SybilLimit in section 6.1. We find that walks on directed graphs have less entropy (at average) than in undirected graphs, for the same walk length. However, in some cases this difference is not big, indicating that the direction of edges in the graph may have insignificant impact.

If we set the random walk length to 6 in each graph, we obtain the results of average entropy and average anonymity sets shown in Table 4. We observe that while the difference in the entropy is small, one should keep in mind that the entropy is a sensitive measure, and a small difference in it would translate into large difference in the anonymity set. For example, the obtained pairs of entropy for directed and undirected cases for the same set of graphs, respectively, translate to the following pairs of anonymity sets: $(2411, 6013)$, $(764, 884)$, $(8551, 9891)$, and $(1026, 2759)$. The exponentially scaled measures of anonymity demonstrate the difference in both cases, which is related to the underlying graphs and their altered structure.

Table 4: The entropy and anonymity set comparison in directed and undirected graphs for random walk length $w = 6$ and for the different datasets.

Dataset	$\overline{H_w^d}$	$\overline{H_w^u}$	$\overline{A_w^d}$	$\overline{A_w^u}$
Epinion	11.236	12.554	2411	6013
Wiki-Vote	9.579	9.788	764	884
Slashdot	13.062	13.272	8551	9891
Gnutella	10.003	11.430	1026	2759

7. CONCLUSION

In this paper we have investigated mathematical tools and used them for measuring the mixing time of directed social graphs. Our work is mainly motivated by prior work in the literature on building social network-based applications and bringing insight on their operation by neglecting direction of edges in naturally directed social graphs. These applications are aimed to build services to improve security aspects of distributed systems, such as in Sybil defenses and anonymous communication. We show a consistent pattern in which directed graphs are in general slower mixing than undirected graphs for the specific parameters recommended by these applications. Furthermore, we show that these applications perform relatively poorly when the original property in the directed graphs is used.

While applications built on top of social networks may have a great potential in improving the security and reliability of distributed systems, we advocate that researchers have to pay more attention to the methods used for evaluating these applications. After all, it might not be possible, nor needed, to build a Sybil defense that accepts 95% or 99% of the honest nodes and to allow a small fraction of Sybils [27]. We observe that for a gain of 5% in its acceptance rate by undermining edge directions, a Sybil defense may hide a larger unseen cost: three folds the number of reported accepted Sybil identities per attack edge.

Our final recommendation is that, given most of these designs are intended for operation on undirected graphs, it is better to use undirected graphs—plenty of which are already available—for bringing insight on their operation rather than massaging directed graphs. Simple alteration

of graphs may greatly affect the underlying properties used for building such systems.

Acknowledgement—We would like to thank the anonymous reviewers for their valuable input. This work has been supported by NSF grant CNS-0917154, a research grant from Korea Advanced Institute of Science and Technology, and a Doctoral Dissertation Fellowship from the graduate school of the University of Minnesota.

8. REFERENCES

[1] Z. Cai and C. Jermaine. The latent community model for detecting sybil attacks in social networks. In *Proc. of NDSS*, 2012.

[2] G. Danezis, C. Díaz, C. Troncoso, and B. Laurie. Drac: An architecture for anonymous low-volume communications. In *Proc. of PETS*, pages 202–219. Springer, 2010.

[3] G. Danezis, C. Lesniewski-Laas, M. Kaashoek, and R. Anderson. Sybil-resistant DHT routing. *ESORICS*, 2005.

[4] G. Danezis and P. Mittal. SybilInfer: Detecting sybil nodes using social networks. In *Proc. of NDSS*, 2009.

[5] J. Davitz, J. Yu, S. Basu, D. Gutelius, and A. Harris. ilink: Search and routing in social networks. In *Proc. of KDD*, August 2007.

[6] M. Dellamico, , and Y. Roudier. A measurement of mixing time in social networks. In *Proc. of IWSTM*, 2009.

[7] A. Granas and J. Dugundji. *Fixed point theory*. Springer Verlag, 2003.

[8] J. Leskovec, D. P. Huttenlocher, and J. M. Kleinberg. Predicting positive and negative links in online social networks. In *Proc. of WWW*. ACM, 2010.

[9] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Community structure in large networks. *CoRR*, abs/0810.1355, 2008.

[10] C. Lesniewski-Laas. A Sybil-proof one-hop DHT. In *Proceedings of the 1st workshop on Social network systems*, pages 19–24. ACM, 2008.

[11] C. Lesniewski-Laas. *Design and Applications of a Secure and Decentralized Distributed Hash Table*. PhD thesis, Massachusetts Institute of Technology, 2010.

[12] C. Lesniewski-Lass and M. F. Kaashoek. Whānau: A sybil-proof distributed hash table. In *Proc. of NSDI*, 2010.

[13] S. Marti, P. Ganesan, and H. Garcia-Molina. Sprout: P2p routing with social networks. In *Proc. of EDBT Workshops*, pages 425–435. ACM, 2004.

[14] S. Marti, P. Ganesan, and H. Garcia-Molina. DHT routing using social links. In *Proc. of Peer-to-Peer Systems*, pages 100–111. Springer, 2005.

[15] A. Mislove, M. Marcon, P. K. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. In *Proc. of IMC*, pages 29–42, 2007.

[16] P. Mittal, M. Caesar, and N. Borisov. X-Vine: Secure and pseudonymous routing using social networks. In *Proc. of NDSS*, 2012.

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

[18] S. Nagaraja. Anonymity in the wild. In *Proc. of PETS*. Springer, 2007.

[19] I. Pak. Mixing time and long paths in graphs. In *SODA*, pages 321–328, Philadelphia, PA, USA, 2002. Society for Industrial and Applied Mathematics.

[20] M. Richardson, R. Agrawal, and P. Domingos. Trust management for the semantic web. In *Proc. of ISWC*, pages 351–368. Springer, 2003.

[21] M. Ripeanu, A. Iamnitchi, and I. T. Foster. Mapping the gnutella network. *IEEE Internet Computing*, 6(1), 2002.

[22] A. Sinclair. Improved bounds for mixing rates of mc and multicommodity flow. *Comb., Prob. & Comp.*, 1992.

[23] R. Tarjan. Depth-first search and linear graph algorithms. In *Proc. of the 12th Annual Symposium on Switching and Automata Theory, 1971*, pages 114–121. IEEE, 1972.

[24] N. Tran, J. Li, L. Subramanian, and S. S. Chow. Optimal sybil-resilient node admission control. In *Proc. of INFOCOM*. IEEE, 2011.

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

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

[27] H. Yu. Sybil defenses via social networks: a tutorial and survey. *ACM SIGACT News Distributed Comp. Column*, 2011.

[28] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao. SybilLimit: A near-optimal social net defense against sybil attacks. In *Proc. of Symposium on Security and Privacy*. IEEE, 2008.

[29] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. SybilGuard: defending against sybil attacks via social networks. In *Proc. of SIGCOMM*, pages 267–278. ACM, 2006.