Optimal Sybil-resilient node admission control

Nguyen Tran
New York University

Jinyang Li
New York University

Lakshminarayanan Subramanian
New York University

Sherman S.M. Chow*
New York University

***Abstract*—Most existing large-scale networked systems on the Internet such as peer-to-peer systems are vulnerable to Sybil attacks where a single adversary can introduce many bogus identities. One promising defense of Sybil attacks is to perform social-network based admission control to bound the number of Sybil identities admitted. SybilLimit [22], the best known Sybil admission control mechanism, can restrict the number of Sybil identities admitted per attack edge to $O(\log n)$ with high probability assuming $O(n/\log n)$ attack edges.**

In this paper, we propose Gatekeeper, a decentralized Sybil-resilient admission control protocol that significantly improves over SybilLimit. Gatekeeper is optimal for the case of $O(1)$ attack edges and admits only $O(1)$ Sybil identities (with high probability) in a random expander social networks (real-world social networks exhibit expander properties). In the face of $O(k)$ attack edges (for any $k \in O(n/\log n)$), Gatekeeper admits $O(\log k)$ Sybils per attack edge. This result provides a graceful continuum across the spectrum of attack edges. We demonstrate the effectiveness of Gatekeeper experimentally on real-world social networks and synthetic topologies.

I. Introduction

Open systems like Digg, Youtube, Facebook and BitTorrent allow any user on the Internet to join the system easily. Such lack of strong user identity makes these open systems vulnerable to Sybil attacks [8], where an attacker can use a large number of fake identities (Sybils) to pollute the system with bogus information and affect the correct functioning of the system. The only known promising defense against Sybil attacks is to use social networks to perform user admission control and limit the number of bogus identities admitted to the system [22], [23], [7], [18]. A link in the social network between two users represents a real-world trust relationship between the two users. It is reasonable to assume that an attacker usually has few links to honest users since establishing trust links requires significant human efforts. Therefore, **Sybil-resilient admission control** can be stated as follows: Consider a social network $\mathcal{G}$ consisting of n honest users and arbitrarily many Sybils connected to honest nodes via k attack edges (an *attack edge* is a link between an honest and a Sybil node). Given an honest node acting as the admission controller, determine the set of nodes to be admitted so that the vast majority of honest nodes in $\mathcal{G}$ are admitted and few Sybil nodes are admitted.

The knowledge of the social graph $\mathcal{G}$ may reside with a single party or be distributed across all users. Centralized node admission assumes complete knowledge of $\mathcal{G}$ (e.g. SybilInfer [7] and SumUp [18]) while distributed admission control only requires each user/node to be initially aware of only its immediate neighbors in $\mathcal{G}$ and seeks to discover all the other honest users/nodes in $\mathcal{G}$. This paper addresses the distributed node admission control problem.

We make a few important observations about the Sybil-resilient node admission problem. First, the problem is inherently probabilistic in its definition; hence, we seek to admit *most* honest nodes while limiting Sybil nodes. Finding a perfect algorithm that can detect all honest nodes and reject all Sybil nodes is fundamentally impossible. Second, the problem makes no assumption about n, the number of honest nodes in $\mathcal{G}$. As we show in our result, if the social network exhibits expander-graph properties, one does not require the knowledge of n to solve the problem. Third, any distributed admission control protocol can also be run in a centralized setting and hence is more general than centralized admission control.

The distributed admission control problem has been studied in prior work. SybilGuard [23] is the first work to show an admission protocol which limits the number of admitted Sybil identities to be $O(\sqrt{n}\log n)$ per attack edge, where n is the number of honest users in the social network. SybilLimit [22] significantly improves over SybilGuard and limits the number of Sybils admitted per attack edge to $O(\log n)$.

In this paper, we present a distributed Sybil-resilient admission control protocol called Gatekeeper with the following results:

Theorem: *Given a social network $\mathcal{G}$ which exhibits a random expander-graph property, Gatekeeper achieves the following properties with high probability:*

1) *In the face of k attack edges with k up to $O(n/\log n)$, Gatekeeper limits the number of admitted Sybil identities to be $O(\log k)$ per attack edge. This implies that only $O(1)$ Sybil nodes are admitted per attack edge if the attacker has $O(1)$ attack edges.*
2) *Gatekeeper admits almost all honest users.*

To achieve these results, Gatekeeper uses an improved version of the *ticket distribution* algorithm in SumUp [18] to perform node admission control in a decentralized fashion. Gatekeeper executes the ticket distribution algorithm from multiple randomly chosen vantage points and combines the results to perform decentralized admission control. We prove the results under the assumption of random expander graphs, an assumption that holds for many existing social networks. Expander graphs are by nature fast-mixing, a common assumption made in SybilLimit and other related protocols [22], [23], [7], [10].

*Current affiliation: University of Waterloo

Our result establishes optimality and improves over SybilLimit by a factor $\log n$ in the face of $O(1)$ attack edges. Under constraints that attack edges are hard to establish and there is only a constant number of them, Gatekeeper is an optimal decentralized protocol for the Sybil-resilient admission control problem. The general result on admitting $O(\log k)$ Sybils per attack edge in the face of k attack edges for any $k \in O(n/\log n)$ establishes a continuum across the attack capacity spectrum. This provides a graceful degradation with increased number of attack edges. In the worst case when $k = O(n/\log n)$, Gatekeeper achieves the same level of resilience as SybilLimit where both Gatekeeper and SybilLimit admit $O(\log n)$ Sybils per attack edge with high probability.

We have tested our protocol experimentally on real-world social networks and synthetic topologies for varying number of attack edges. Our analysis shows that our protocol is able to drastically limit the number of admitted Sybil identities to a very small number while admitting almost all honest identities. Even when we significantly increase the number of attack edges to cover $\sim 2\%$ of the nodes, the number of admitted Sybil identities per attack edge remains very small.

II. Related Work

Traditionally, open systems rely on a central authority who employs CAPTCHA or computational puzzles to mitigate the Sybil attack [20], [14], [15]. Unfortunately, these solutions can only limit the *rate* with which the attacker can introduce Sybil identities into the system instead of the total number of such identities. Even before the recent surge of interest in social-network-based Sybil defenses, there have been attempts at exploiting the trust graph among users to mitigate the Sybil attack: Advogato [11], Appleseed [24] and SybilProof [4] are the most well-known of these early proposals. However, it is not the goal of these protocols to perform Sybil-resilient node admission. Rather, they aim to calculate the reputation of each user/node in a way that prevents the attacker from boosting its reputation using Sybil identities. Below, we discuss recent work in node admission control and related efforts in Sybil-resilient Distributed Hash Table (DHT) routing.

SybilGuard [23] has pioneered the use of fast-mixing social networks for Sybil-resilient admission control. Using a distributed verification protocol based on random routes, SybilGuard can limit the number of Sybil nodes admitted per attack edge to $O(\sqrt{n}\log n)$. SybilLimit [22] improves this bound to admit no more than $O(\log n)$ Sybils per attack edge with high probability. Yu et al. claim that SybilLimit is nearly optimal in that its security guarantee is only a factor of $O(\log n)$ away from that of any optimal protocol.

SybilGuard and SybilLimit are both designed to work in a distributed setting where each node is initially only aware of its neighbors. By contrast, SybilInfer [7] is a centralized algorithm which assumes complete knowledge of the social graph. SybilInfer uses Bayesian inference to assign each node a probability of being a Sybil. The key observation is that, if the attacker connects more Sybils to its few attack edges, the conductance of graph including the Sybil region becomes smaller to the point that the entire graph is not fast-mixing, thereby causing the detection of the Sybil nodes. Unlike SybilGuard and SybilLimit, SybilInfer has no analytical bound on the number of Sybil nodes admitted per attack edge. SumUp [18] is another centralized admission algorithm which admits nodes by computing max-flow paths from a “vote envelope” to all nodes. SumUp admits $O(\log n)$ Sybil nodes per attack edge on average. In [16], Quercia et al. propose a Sybil-defense mechanism for the mobile setting where a node collects graph information from those nodes that it has previously encountered and analyzes the partial graphs to determine the likelihood of a node being Sybil. Like SybilInfer, there is no formal bound for the algorithm in [16].

Recently, Viswanath et al. [19] has performed a comparative study of SybilGuard, SybilLimit, SybilInfer and SumUp. The study reveals two potential limitations of social-network based admission control. First, many *small* social networks (up to tens of thousands of nodes) exhibit community structure (i.e. not fast-mixing), thus causing existing protocols to falsely reject many honest nodes as Sybils. This finding suggests that Sybil-resilient admission control must be performed on large-scale social networks: the larger the graph, the better connected communities are to each other and the faster the mixing time. Thus, our evaluations use real world social graphs that consist of hundreds of thousands of nodes. Second, given a known admission controller, the attacker can strategically acquire attack edges close to the controller to gain unfair advantage. In Gatekeeper, we address this limitation by having a controller select a few random vantage points for ticket distribution. Viswanath’s work compares all existing schemes in a centralized setting even though SybilGuard and SybilLimit are originally designed to work as a distributed protocol. It is worth pointing out that Sybil defense is more challenging in a distributed setting than in a centralized setting. This is because, in a centralized setting, the attacker must decide upon the graph structure of the Sybil region before the admission algorithm starts to execute. On the other hand, in a distributed setting, the attacker has the freedom to change the Sybil region of the graph arbitrarily during protocol execution to maximize its gain.

A Sybil-resilient DHT [10], [6] ensures that DHT lookups succeed with high probability in the face of an attacker attempting to pollute the routing tables of many nodes using Sybil attacks. Danezis et al. [6] leverage the bootstrap tree of the DHT to defend against the Sybil attack. Two nodes in such a tree share an edge if one node introduced the other one into the DHT. The assumption is that Sybil nodes attach to the tree at a small number of nodes, similar to the few attack edge assumption in SybilGuard and SybilLimit. Wh$\overline{\ }$anau [10] uses social connections between users to build routing tables in order to perform Sybil-resilient lookups. Sybil-resilient node admissions can potentially simplify the construction of distributed Sybil-resilient protocols by bounding the number of Sybil identities admitted in the first place.

III. System Model and Threat Model

We use a similar system model and threat model as those used in previous systems (e.g. SybilLimit [22], SybilGuard [23] and Whānau [10]). The system consists of$n$ honest nodes belonging to n honest users. There exists an undirected social graph among all nodes in the system. A link between two honest users reflects the trust relationship between those users in the real-world. The knowledge of the social graph is distributed among all nodes. In particular, each honest node knows its immediate neighbors on the social graph and may not know the rest of the graph, including the value of n. Each node has a locally generated public/private key pair. A node knows the public-keys of its neighbors, however, there exists *no* public-key infrastructure that allows a node to correctly learn of all other nodes' public-keys.

The system also has one or more malicious users and each malicious user controls a number of malicious Sybil nodes. All Sybil nodes may collude with each other and hence are collectively referred to as the adversary or attacker. Honest nodes behave according to the protocol specification while Sybil nodes are assumed to behave in a Byzantine fashion. The attacker may know the entire social graph and is able to create arbitrary links among his Sybil nodes. We assume the attacker has k links with honest users (attack edges), where k can be up to $O(n/\log n)$.

Distributed admission control: A node acting as an admission controller determines which of the other nodes (suspect nodes) should be admitted into the system. The process can either be creating a list of admitted nodes, or deciding whether a particular suspect node can be admitted or not. In the centralized setting, one typically assumes the existence of a trusted controller that performs admission control on behalf of all nodes. By contrast, in the distributed setting, there exists no centralized source of trust and each node must act as its own controller. Each controller needs to consult other nodes to make its admission decisions. We note that a node acts as its own controller as well as a suspect for other controllers.

Sybil-resilient node admissions: The goal of Sybil-resilient admission is two-fold – it should accept most honest nodes and it should admit few Sybil nodes. The attacker aims to maximize the number of admitted Sybil nodes, and to minimize the number of admitted honest nodes.

It is worth emphasizing that the number of admitted Sybil nodes is ultimately dependent on k, the number of attack edges. Specifically, since attack edges are indistinguishable from honest edges, any protocol that admits most honest nodes would admit approximately one Sybil node per attack edge, resulting in k admitted Sybil nodes. The goal of a Sybil-resilient admission protocol is to approach this lower bound of one admitted Sybil node per attack edge. Separate mechanisms are required to ensure that k is likely to be small. Today's popular online social networks like Facebook do not promise small k. To minimize k, one can use techniques proposed in [2] and [21] to ensure that honest users only establish trust links with their close friends in the real-world so that the attacker is unlikely to possess many links to honest users, resulting in a small k.

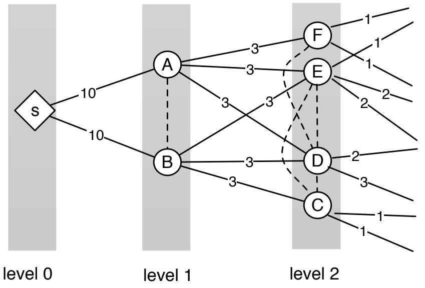

Fig. 1. The ticket distribution process of a particular node S: The number on each link represents the number of tickets propagated via that link. The dotted lines are links between nodes at the same distant to the source.

IV. Design Overview

In this section, we first describe the central component of Gatekeeper, the *ticket distribution process*. We proceed to discuss the challenges involved in using ticket distribution for node admission control and explain how Gatekeeper addresses these challenges.

A. Ticket Distribution

The principle building block of Gatekeeper is the ticket distribution process where each node acting as a ticket source disseminates t "tickets" throughout the social network until a significant portion of the honest nodes receive some tickets. We originally designed the distribution algorithm for SumUp [18], a *centralized* Sybil-resilient vote collection system. SumUp uses ticket distribution to assign link capacities which are needed for its centralized max-flow computation. As we will see later, Gatekeeper uses ticket distribution completely differently.

We illustrate the ticket distribution process using the example of Figure 1 where the ticket source (S) intends to disseminate $t = 20$ tickets. Tickets propagate in a breadth-first-search (BFS) manner: Each node is placed (conceptually) at a BFS-level according to its shortest-path distance from S. S divides the tickets evenly and sends them to its neighbors. Each node keeps one ticket to itself and distributes the rest evenly among its neighbors at the next level. In other words, a node does not send tickets back to neighbors that are at the same or smaller distance to the source. If a node does not have any outgoing links to the next level, it simply destroys all remaining tickets. The process continues until no tickets remain.

We use ticket distribution as a fundamental building block in Gatekeeper because of two considerations. First, since each node only needs knowledge of its immediate neighborhood to propagate tickets, the entire distribution process can be realized in a completely distributed manner. Second, as nodes propagate tickets in a BFS manner from the source, edges further away from the ticket source receive exponentially fewer tickets. Our intuition is that, since the attacker only controls a small number of attack edges, a randomly chosen ticket source is relatively "far away" from most attack edges, resulting in few tickets propagated along an attack edge. As a result, Gatekeeper may be able to directly use a received ticket as a token for a node's admission.

B. *Our approach*

The naïve strategy for applying ticket distribution to admission control works as follows: each node admission controller (S) disseminates n tickets and accepts a suspect node if and only if it has received some tickets from S. Such a strategy has two inherent limitations. First, it is infeasible to reach the vast majority (e.g. $> 99\%$) of honest nodes by distributing n tickets from a single ticket source. For example, the simulation experiments in [18] shows that only $\sim 60\%$ honest nodes receive some tickets. Second, in the presence of a single ticket source, an attacker may be able to strategically acquire some attack edge close to the source, resulting in a large amount of tickets being propagated to Sybil nodes via that attack edge.

The key idea of Gatekeeper is to perform distributed ticket distribution from *multiple* ticket sources. In Gatekeeper, an admission controller explicitly picks m random nodes (using the random walk technique in [22]) to act as ticket sources. Each randomly chosen ticket source distributes t tickets where t is chosen such that $\frac{n}{2}$ nodes receive some tickets. Later in Section VII, we will show that a source only needs to send out $t = \Theta(n)$ tickets. We say that a node is *reachable* from a ticket source if it has received a ticket disseminated by the source. The admission controller admits a suspect node if and only if the node is reachable from at least $f_{admit} \cdot m$ ticket sources, where f_{admit} is a small constant (our evaluations suggest using $f_{admit} = 0.2$).

Multi-source ticket distribution addresses both limitations associated with using a single ticket source. The first limitation is concerned with a single source not being able to reach the vast majority of honest nodes by sending only $t = \Theta(n)$ tickets. In Gatekeeper, an honest node not reachable from one source may be reached by other sources. Ultimately, an honest node is admitted as long as it is reachable by $f_{admit} \cdot m$ sources which is a high probability event. On the other hand, with a small number of attack edges, the attacker cannot appear close-by to many m randomly chosen sources, and thus is unlikely to receive a large number of tickets from as many as $f_{admit} \cdot m$ sources. Therefore, by admitting only nodes reachable by $f_{admit} \cdot m$ sources, Gatekeeper ensures that the number of admitted Sybil nodes per attack edge is small. The second limitation is concerned with an attacker strategically acquires some attack edge close to a known ticket source. Gatekeeper solves this problem because the admission controller explicitly picks m random ticket sources as opposed to acting as the ticket source itself. In Section VII, we present a detailed analysis of these intuitions.

V. GATEKEEPER: THE PROTOCOL

The Gatekeeper protocol consists of two phases: a *bootstrap phase* where each node acts as a ticket source to disseminate $\Theta(n)$ tickets throughout the network and an *admission phase* where a node acting as the admission controller selects m ticket sources and accepts another node if that node possesses tickets from $f_{admit} \cdot m$ of the m chosen sources. Below, we describe the details of these two phases:

A. *Bootstrap: decentralized ticket distribution*

To bootstrap the protocol, every node performs decentralized ticket distribution with the aim of reaching more than half of the honest nodes. Since ticket distribution proceeds in a BFS fashion, a forwarding node needs to know its neighbor's "level" (i.e. the neighbor's shortest path distance to the ticket source) in order to decide whether to forward that neighbor any tickets. In order to establish such shortest path knowledge, all nodes execute a secure path-vector based routing protocol. We adopt a known secure path-vector protocol [9] where a node explicitly advertises its shortest path to each ticket source using a signature chain signed by successive nodes along the path. As a result, Sybil nodes cannot disrupt the shortest path calculation among honest nodes.

The number of tickets a source should disseminate, t, is not a fixed parameter. Rather, each source adapts t iteratively by estimating whether a sufficiently large fraction of nodes receive some tickets under the current value of t. We first describe how a source S disseminates t_j tickets in the j-th iteration and discuss how S adapts t_j later. Each ticket from S consists of the current iteration number j, a sequence number $i \in [1..t_j]$, and a message authentication code (MAC) generated using the private key of S. The MAC is verifiable by the source and is necessary to prevent the forgery and tampering of tickets.

A node Q receiving r tickets consumes one of them and evenly divides the other $r - 1$ tickets to those neighbors at the next BFS "level", i.e. neighbors that are further away from S than Q. Node Q can learn which neighbors are further by requesting and verifying its neighbors' shortest path signatures. If Q has no such neighbor, it simply discards its remaining tickets. When Q sends a ticket to its neighbor R, it explicitly transfers the ownership of that ticket by appending a tuple $\langle Q, R \rangle$ to the ticket and signing the ticket with Q's private key. If Q consumes a ticket, it appends itself $\langle Q* \rangle$ to denote the end of the transfer chain. The use of a signature chain allows a ticket source to detect a "double-spender", i.e. a malicious node that has sent the same ticket to different nodes. The signature chain scheme represents one of many solutions for detecting double-spenders. Alternative mechanism include secure transferable e-cash schemes [5]) which allow a source node to act as a "bank" issuing e-coins as tickets.

In order to help source S determine its reachable nodes, each node that has consumed a ticket from S forwards its ticket in the reverse direction of the ticket's signature chain. Suppose S receives a ticket consumed by Q, S must verify the validity of the signature chain associated with that ticket. In particular, S checks that the chain is not "broken", e.g., $\langle S, A\rangle, \langle A, B\rangle, \langle C, Q\rangle$ is not valid because it misses the link $\langle B, C\rangle$. Additionally, S also checks in its database of received tickets to see if there is any double spending. For example, if S discovers two tickets $(\langle S, A\rangle, \langle A, B\rangle, \langle B, Q\rangle)$ and $(\langle S, A\rangle, \langle A, B'\rangle, \langle B', Q'\rangle)$, it will blacklist node A as a double-spender and ignore both tickets. If Q's ticket passes verification, S records Q in its database of reachable nodes.

Adjust the number of tickets distributed iteratively: After a pre-defined time period, the ticket source terminates the current j-th iteration of ticket distribution, and decides if it needs to proceed with the $(j+1)$-th iteration with increased number of tickets to be distributed. In particular, the ticket source samples a random subset (W) of nodes in the social network by performing a number of random walks. Let R be the set of reachable nodes in the source's database. If less than half of the sampled nodes are within the reachable set, i.e. $\frac{|R \cap W|}{|W|} < 1/2$, the source proceeds to the next iteration $(j+1)$ with twice the amount of tickets, $t_{j+1} = 2 \cdot t_j$.

Intuitively, when the attacker controls up to $O(n/\log n)$ attack edges, only a negligible fraction of nodes ($o(1)$) are Sybils in the sampled set (W) and the reachable set (R). As a result, if the majority of the sampled nodes (W) are not in R, it implies that the amount of tickets distributed in the current iteration is insufficient and the source should distribute more tickets in the next iteration. On the other hand, once the amount of tickets distributed reaches $\Theta(n)$, the majority of honest nodes become reachable, thereby terminating the iterative process.

Our adaptive ticket adjustment process is similar to the benchmarking technique used in SybilLimit [22]. In SybilLimit, each node performs $O(\sqrt{n})$ random walks and benchmarking is used to determine the number of random walks to perform without explicitly estimating n. Similarly, in Gatekeeper, each ticket source adaptively decides on the amount of tickets to distribute ($t = \Theta(n)$) without having to explicitly estimate n.

B. Node admission based on tickets

After all ticket sources have bootstrapped, each node can carry out its own admission control to decide upon a list of nodes to be admitted into the system.

To perform admission control, a controller first selects m random ticket sources by performing m random walks of length $O(\log n)$. In fast-mixing social networks, a random walk of length $O(\log n)$ reaches a destination node drawn from the node-stationary distribution. Because nodes have varying degrees, a forwarding node i picks neighbor j as the random walk's next hop with a probability weight of $\min(\frac{1}{d_i}, \frac{1}{d_j})$, where d_i and d_j are the degree of node i and j, respectively. This ensures that m random walks sample m nodes uniformly at random [7]. It is in the attacker's best interest to claim that Sybil nodes have degree 1 in order to attract random walks into the Sybil region. To protect an unlucky controller who is a friend or a friend-of-a-friend of some Sybil node, we make an exception for honest nodes to forward random walks to its neighbors with *equal* probability during the first two hops of a random walk. We use the same strategy in [22] to estimate the required random walk length without the knowledge of n.

The controller asks each of the m chosen ticket sources for its reachable node list. The controller admits a node if and only if that node has appeared in more than $f_{admit} \cdot m$ reachable lists returned by the m chosen ticket sources. The parameter f_{admit} is set to a fixed value 0.2 in our simulations and we will analyze how to set the appropriate value for f_{admit} in Section VII.

VI. Protocol Message Overhead

We consider the asymptotic message overhead of Gatekeeper when every node acts as a controller and compare to that of SybilLimit. During the bootstrap phase, the number of bits that need to be transferred during the ticket distribution process of a single source is $\Theta(n \log n)$ because the source sends out $\Theta(n)$ tickets and each ticket travels a path of length $\Theta(\log n)$. Therefore, in a network of n ticket sources, the total message overhead is $\Theta(n^2 \log n)$. In the admission phase, each controller obtains m node lists each of size $\Theta(n)$ from m chosen ticket sources. When each node acts as a controller, the total number of bits transferred during the admission phase is $\Theta(n^2)$. Thus, the total message overhead incurred by Gatekeeper is $\Theta(n^2 \log n) + \Theta(n^2) = \Theta(n^2 \log n)$. This overhead is the same as that of SybilLimit if every honest node aims to admit every other honest node. However, we must point out that if each controller only intends to admit a small constant of honest nodes, SybilLimit incurs only $\Theta(n\sqrt{n} \log n)$ total overhead. By contrast, the total overhead in Gatekeeper is always $\Theta(n^2 \log n)$ regardless of the number of honest nodes each controller intends to admit.

In some circumstances, it may be desirable to run Gatekeeper in a centralized setting using a single admission controller. For example, the online content voting site, Digg.com, may run Gatekeeper on its social graph using a single controller to decide upon the list of nodes allowed to cast votes. In these cases, Gatekeeper's overall runtime is $\Theta(n \log n)$, which is much better than that of SybilLimit ($\Theta(n\sqrt{n} \log n)$).

VII. Security Analysis

We show Gatekeeper's Sybil-resilience by proving that, if the attacker possesses $k = O(n/\log n)$ randomly injected attack edges, a controller admits at most $O(\log k)$ Sybil nodes per attack edge and that each controller admits almost all honest nodes. Our proof makes certain assumptions about the social graph formed by honest users, denoted by $\mathcal{G}$. Specifically, we assume that:

1) $\mathcal{G}$ is a fixed degree sequence random graph constructed by the pairing method in [3], [12] with maximum node degree d. It has been shown that the pairing method generates an expander graph with expansion factor α with high probability. In other words, for every set W of vertices with fewer than $n/2$ nodes, $|N(W)| \geq \alpha|W|$ where $N(W)$ denotes the set of vertices adjacent to W but do not belong to W [1]. Compared to previous work

which only assumes fast-mixing graphs [23], [22], [10], expanders represent a stronger assumption. Nevertheless, expander has been commonly used as reasonable model for large-scale social graphs.

2) $\mathcal{G}$ is reasonably balanced. Let $\Delta_{half}(v)$ be the distance such that v is less than $\Delta_{half}(v)$ distance away from more than half of the honest nodes. In other words, $\Delta_{half}(v)$ is the BFS-level when v reaches more than $\frac{n}{2}$ nodes. Let $dist(u,v)$ be the distance between u, v. Define $S(v) = \{u | u \in \mathcal{G}, dist(u,v) \le \Delta_{half}(u)\}$. In other words, $S(v)$ represents the set of ticket sources that deem v as reachable. We say $\mathcal{G}$ is balanced if for almost all v, $\frac{|S(v)|}{n} > f_{th}$ for a constant threshold value $f_{th} < 0.5$. In probabilistic terms, $\Pr(\frac{|S(v)|}{n} < f_{th})$ for any randomly chosen v is $o(1)$ (a function asymptotically lower than a constant). Most real world social graphs satisfy this balance criterion.

A. Gatekeeper admits $O(\log k)$ Sybils per attack edge

For this proof, we proceed in two steps: first, we bound the number of tickets sent to the attacker (via k attack edges) by a randomly chosen ticket source to $O(k \log k)$. Second, we show at most $O(\log k)$ Sybil nodes can receive tickets from more than $f_{admit} \cdot m$ of the m ticket sources using the Chernoff bound.

The more tickets a source distributes, the more tickets that likely end up with the attacker. Therefore, in order to bound the number of tickets received by the attacker, we must bound the number of tickets distributed by a ticket source, as described formally by the following theorem:

Theorem 7.1: Suppose the graph $\mathcal{G}$ is a fixed degree sequence random graph constructed by the pairing method. The expected number of tickets required by a given ticket source to reach more than $n/2$ honest nodes is $E[t] = \Theta(n)$. (see proof in technical report [17])

Given a ticket source u, we order honest nodes from closest to farthest from u according to their BFS level. Let Δ_{small} be the level of the $\frac{\epsilon}{k} \cdot n$-th node, where ϵ is a small constant like 0.01. Let Δ_{big} be the level of the $(1 - \frac{\epsilon}{k}) \cdot n$-th node. In other words, $\Delta_{small}, \Delta_{big}$ are chosen so that the BFS levels of $1 - \frac{2\epsilon}{k}$ fraction of honest nodes fall between $(\Delta_{small}, \Delta_{big}]$. As a result, the probability that all k attack edges are at some distance within the range $(\Delta_{small}, \Delta_{big}]$ is $(1 - \frac{2\epsilon}{k})^k > 1 - 2\epsilon$, which is high because of small ϵ. Next, we will bound the number of tickets received by the attacker for the high probability event that all k attack edges lie within ticket distribution levels $(\Delta_{small}, \Delta_{big}]$.

Lemma 7.1: For a given ticket source u, given that all k randomly injected attack edges are at some distance in the range $(\Delta_{small}, \Delta_{big}]$ from u, the expected number of tickets received by the attacker is $O(k \log k)$.

Proof: Let A_i be the number of u's tickets that are sent from level-i to level-$(i+1)$. $A_0 = t$ is the number of tickets distributed by the source u. Let L_i be the number of nodes at level-i. We can calculate the expected number of tickets that pass though a random node at level $(\Delta_{small}, \Delta_{big}]$ as:

$$\sum_{\Delta_{small}+1}^{\Delta_{big}} \frac{A_{i-1}}{L_{(\Delta_{small}+1)} + \cdots + L_{\Delta_{big}}} \qquad (1)$$

$$\le \quad (\Delta_{big} - \Delta_{small}) \frac{A_0}{L_{(\Delta_{small}+1)} + \cdots + L_{\Delta_{big}}} \qquad (2)$$

By the definition of Δ_{small} and Δ_{big}, we know that $(L_{(\Delta_{small}+1)} + \cdots + L_{\Delta_{big}})$ has greater than $(1 - \frac{2\epsilon}{k})$ fraction of honest nodes. Furthermore, $E(A_0) = E(t) = \Theta(n)$ according to Theorem 7.1. Hence, $\frac{A_0}{L_{\Delta_{small}+1} + \cdots + L_{\Delta_{big}}} = O(1)$.

To show that $(\Delta_{big} - \Delta_{small})$ is $O(\log k)$ we consider the two terms $(\Delta_{half} - \Delta_{small})$ and $(\Delta_{big} - \Delta_{half})$ where Δ_{half} is the level where we reach the $n/2$-th node in the BFS tree of u. Because $\mathcal{G}$ is an expander with expansion factor α across each level, we have $\frac{\epsilon}{k} n \cdot \alpha^{\Delta_{half} - \Delta_{small}} \le n/2$. Hence $\Delta_{half} - \Delta_{small}$ is $O(\log k)$. Similarly, we can bound $\Delta_{big} - \Delta_{half}$ to $O(\log k)$ by expanding from graph from the $\frac{\epsilon}{k} n$ nodes farthest from u to the $\frac{n}{2}$-th node. Summing up the two results, we get $(\Delta_{half} - \Delta_{small})$ as $O(\log k)$. Hence, we can bound the expected number of tickets received by a random node within the level range $(\Delta_{small}, \Delta_{big}]$ to be $O(\log k)$. Since an attack edge is connected to a random node at level within the range $(\Delta_{small}, \Delta_{big}]$, the expected number of tickets received by an attack edge is bounded by $O(\log k)$. Hence, with k attack edges all within this range, the expected number of tickets received by the attacker is $O(k \log k)$.

■

Based on Lemma 7.1, a ticket source gives $O(k \log k)$ tickets to the attacker with k attack edges. However, the $O(k \log k)$ bound is only in expectation and some ticket sources may give much more than the expected number of tickets to the attacker. By requiring each admitted node to receive tickets from at least $f_{admit} \cdot m$ of m randomly chosen sources, we can prove the following theorem:

Theorem 7.2: Gatekeeper admits $O(\log k)$ Sybils per attack edge with high probability.

Proof: Let $T_1, T_2, \cdots, T_m$ be the random variables representing the total number of tickets received by the attacker via k attack edges from each of the m ticket sources. Since $E(T_i) = O(\log k)$, according to Markov's inequality, there exist constants, $\beta > 1$ and $\tau < \frac{f_{admit}}{2}$, such that $\Pr(T_i > \beta k \log k) \le \tau$. In other words, the probability that any ticket source reaches more than $\beta k \log k$ Sybil nodes is bounded by τ.

We define a new random variable, Z_i, as follows:

$$Z_i = \begin{cases} 1 & \text{if } T_i \ge \beta k \log k \\ 0 & \text{if } T_i < \beta k \log k \end{cases}$$

Let $z = Z_1 + Z_2 + \cdots + Z_m$. Since $\Pr(Z_i = 1) < \tau$, using Chernoff bound, we can show that

$$\Pr(z \ge \frac{m f_{admit}}{2}) \le e^{-m \cdot D(\tau, \frac{f_{admit}}{2})}$$

where $D(\tau, \frac{f_{admit}}{2})$ is the Kullback-Leibler divergence function that decreases exponentially with m. Hence, with high probability, $z \leq \frac{m f_{admit}}{2}$. We refer to the i-th source as type-A if $Z_i = 1$ or as type-B if $Z_i = 0$. Among the m sources, there are z type-A sources and $m - z$ type-B sources.

Suppose $\bar{s}$ Sybil nodes are finally admitted. In order to be admitted, each of the $\bar{s}$ Sybils can present at most z tickets from type-A sources. Additionally, *all* $\bar{s}$ Sybils can use at most $(m - z)\beta k \log k$ tickets from type-B sources. Hence, the total number of tickets that can be used for the admission of $\bar{s}$ Sybils is at most $\bar{s}z + (m - z)\beta k \log k$. Since $\bar{s}$ Sybils need at least $\bar{s} f_{admit} \cdot m$ tickets for admission, we arrive at the following inequality:

$$\begin{aligned} \bar{s} \cdot f_{admit} \cdot m &\leq \bar{s} \cdot z + (m - z) \cdot \beta k \log k \\ \bar{s} &\leq \frac{(m - z)}{f_{admit} \cdot (m - z)} \beta k \log k \\ \bar{s} &\leq \frac{2 - f_{admit}}{f_{admit}} \beta k \log k \\ \frac{\bar{s}}{k} &= O(\log k) \quad \square \end{aligned}$$

■

B. Gatekeeper admits most honest nodes

Theorem 7.3: Gatekeeper admits any honest node with high probability.

Proof: Recall our earlier definition of $S(v)$, which represents the set of potential ticket sources that deem v as reachable. Since $\mathcal{G}$ is balanced, the probability that a randomly chosen ticket source can reach v is at least f_{th}. Since the events that v is reachable from randomly chosen ticket sources are independent, we can apply the Chernoff bound to show that the probability v is reachable from less than $f_{admit} \cdot m$ ticket sources is bounded by $e^{-m \cdot D(f_{admit}, f_{th})}$ where $D(\cdot)$ is the Kullback-Leibler divergence function. Thus, when choosing f_{admit} such that $f_{admit} < f_{th}$, the probability that an honest node is not admitted decreases exponentially with m. Hence, Gatekeeper admits an honest node with high probability. ■

Note that we have proved both Theorem 7.2 and Theorem 7.3 for the case when *all* m ticket sources are honest. A Sybil node may be chosen as a source if a random walk escapes to the Sybil region of the graph. Let f_{esc} be the fraction of m sources in the Sybil region. When the attacker controls up to $O(n/\log n)$ attack edges, with high probability, f_{esc} is asymptotically smaller than a constant, i.e. $f_{esc} = o(1)$. Our earlier proofs can be extended to handle $f_{esc} = o(1)$. Next, we analyze the worst case scenario when f_{esc} is non-negligible.

C. Worst Case Analysis

The worst case scenario applies to those few unlucky controllers that are extremely close to some attack edge, resulting in a non-negligible f_{esc}. Let m' be the number of honest sources, i.e. $m' = (1 - f_{esc}) \cdot m$. We adjust the proof for Theorem 7.2 to handle the case when only m' sources are honest. For each of the $\bar{s}$ Sybils to be admitted, it can use at most z tickets from type-A ticket sources and at most $(m' - z) \cdot \beta k \log k$ from type-B sources. Additionally, $\bar{s}$ Sybils can use $\bar{s} f_{esc} \cdot m$ from those ticket sources in the Sybil region. Recall that $z < \frac{f_{admit} m'}{2}$, we have:

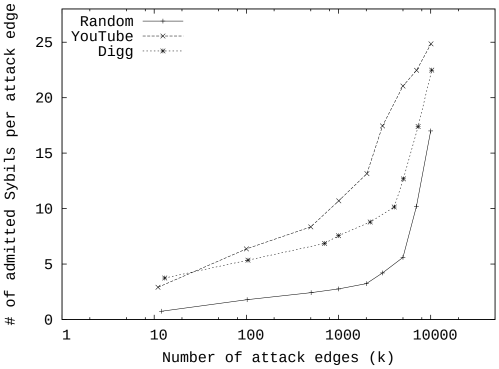

Fig. 2. The number of Sybil nodes accepted per attack edge as a function of the number of attack edges (k).

$$\begin{aligned} \bar{s} \cdot f_{admit} \cdot m &\leq \bar{s} f_{esc} m + \bar{s} z + (m' - z)\beta k \log k \\ \bar{s}((f_{admit} - f_{esc})m - z) &\leq (m' - z)\beta k \log k \\ \Rightarrow \frac{\bar{s}}{k} &\leq \frac{(1 - f_{esc}) \cdot m - z}{(f_{admit} - f_{esc})m - z} \beta \log k \end{aligned}$$

Therefore, to admit at most $O(\log k)$ Sybils per attack edge (i.e. $\frac{\bar{s}}{k} = O(\log k)$), the escape probability f_{esc} must be small enough such that $(f_{admit} - f_{esc}) \cdot m - z > 0$. Since $z < \frac{f_{admit} m'}{2}$, we obtain that $f_{esc} < \frac{f_{admit}}{2 - f_{admit}}$.

We adjust the proof of Theorem 7.3 similarly. In order for an honest node to be admitted, it must possess tickets from $f_{admit} \cdot m$ nodes out of the m' honest sources. Therefore, we require $f_{admit} < (1 - f_{esc}) f_{th}$, i.e. $f_{esc} < 1 - \frac{f_{admit}}{f_{th}}$. In summary, to satisfy both Theorem 7.2 and 7.3, we require that $f_{admit} < \min(\frac{f_{admit}}{2 - f_{admit}}, \frac{1 - f_{admit}}{f_{th}})$.

We use $f_{admit} = 0.2$ in our evaluations. Therefore, a controller admits $O(\log k)$ Sybil nodes per attack edge as long as $f_{esc} < 0.11$. As a concrete example, let us consider a controller with degree d who is immediately adjacent to the attacker. In this case, $f_{esc} = 1/d$. Hence if d is bigger than 9, f_{esc} will be small enough to satisfy both Theorem 7.2 and 7.3. If d is smaller than 9, the controller must be more than 1-hop away from the attacker to ensure that f_{esc} is small enough.

VIII. Evaluation

We evaluate the effectiveness of Gatekeeper in both synthetic graphs and real-world social network topologies. Specifically, we show that Gatekeeper admits most honest nodes ($> 90\%$ across different topologies) and significantly limits the number of Sybils admitted per attack edge to a small value even in the face of a large number of attack edges ($k \approx 0.02 \cdot n$).

A. Experimental Methodology

For real-world social topologies, we use the YouTube [13] and Digg [18] graph. For synthetic graphs, we generate random graphs with average node degree of 6. Table I summarizes the basic graph statistics. To model the Sybil attack, we

Data set	Synthetic	YouTube [13]	Digg [18]
Nodes	varying	$446,181$	$539,242$
Undirected edges	varying	$1,728,948$	$4,035,247$
Average:median degree	6 : 6	7.7 : 2	15 : 2

TABLE I
SOCIAL GRAPH STATISTICS

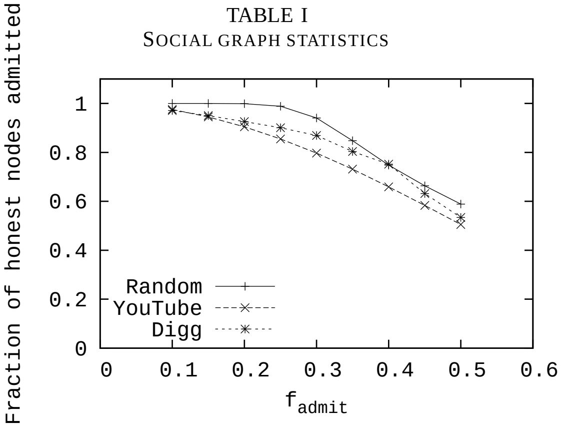

Fig. 3. Fraction of honest nodes admitted under varying f_{admit}

randomly choose a fraction of nodes to collude with the attacker so that all the edges of these nodes as attack edges. The attacker optimally allocates tickets to Sybils to maximize the number of Sybils admitted. In each simulation run, we randomly select a controller to perform admission control and measure the number of Sybils admitted per attack edge and the number of honest nodes admitted. We repeat each experiment for 2000 runs and compute the average and the deviation. Unless otherwise mentioned, a controller uses $m = 100$ ticket sources and admits another node if it has received tickets from at least $f_{admit} = 0.2$ fraction of the m sources.

B. Number of Sybils admitted

We first measure the number of Sybil nodes admitted per attack edge as a function of the number of attack edges (k). Figure 2 shows the number of admitted Sybil nodes as a function of k for a random graph with $500,000$ nodes, the YouTube graph and the Digg graph. Our theoretical result shows that Gatekeeper admits $O(\log k)$ Sybils per attack edge. Figure 2 confirms our analysis showing that the number of Sybils admitted per attack edge increases very slowly with k; even when k reaches 2% of the network size (i.e. $k = 10,000$), the number of Sybils nodes accepted per attack edge remains smaller than 25.

Unlike SybilLimit, Gatekeeper's bound on Sybils admitted per attack edge ($O(\log k)$) is independent of the network size n for a given k. We have verified this property by running Gatekeeper on random graphs with different network sizes.

Comparison with SybilLimit: We compare the performance of Gatekeeper and SybilLimit under both synthetic and real graph topologies with $k = 60$ attack edges. In separate experiments, we find the parameter values so that both Gatekeeper and SybilLimit admit $> 95\%$ honest nodes and use these values in our comparison.

Table II summarizes the parameter values used in each

	SybilLimit		
Dataset	Synthetic ($n = 500,000$)	YouTube	Digg
Parameter w	12	15	14
Parameter r	3200	3400	5100
Sybils admitted per attack edge	40.3	49.1	45.1
	Gatekeeper		
f_{admit}	0.2	0.15	0.15
Sybils admitted per attack edge	1.5	4.9	7.1

TABLE II
COMPARISON WITH SYBILLIMIT

protocol and the number of Sybils admitted per attack edge. As we can see, SybilLimit admits $40 - 50$ Sybils per attack edge across all the three topologies, while Gatekeeper admits only $1-7$ Sybils nodes per attack edge. Therefore, Gatekeeper represents a significant improvement over SybilLimit in practical settings.

Compared to the random graph case, Gatekeeper accepts more Sybil nodes on the YouTube and Digg graphs because real-world graphs can exhibit certain asymmetries that are not present in a random graph. Because of this asymmetry, more tickets are dropped at some node with no neighbors at the next BFS-level. Having more ticket drops in turn causes a ticket source to send more tickets in order to reach more than half of honest nodes. As a result, attack edges also receive more tickets, thereby causing more Sybils to be admitted.

C. Admitting honest nodes

The parameters f_{admit} and m affect the fraction of honest nodes admitted by Gatekeeper. Choosing the appropriate f_{admit} is dependent on the balance properties of the graph. Figure 3 measures the fraction of honest nodes admitted for different values of f_{admit} under various topologies. We can see that larger f_{admit} results in fewer honest nodes being admitted. On the other hand, smaller f_{admit} will increase the number of Sybils admitted by a constant factor. Since synthetically generated random graphs are more balanced than YouTube and Digg graphs, Gatekeeper admits higher fraction of honest nodes in the random graph than in YouTube and Digg graph for the same value of f_{admit}. When $f_{admit} = 0.2$, Gatekeeper can admit more than 90% honest nodes in all three graphs. Hence, we use 0.2 as the default value for f_{admit}. We have also experimented with varying m and found that $m = 100$ was sufficient to admit most honest nodes across different topologies. Setting m to be bigger than 100 yields diminishing returns.

D. Worst case scenario with a close-by attacker edge

The worst case scenario happens for controllers that are extremely close to some attack edge such that a significant fraction of the m random walks escape into the Sybil region, causing the controller to use many Sybil nodes as ticket sources. To evaluate such worst case scenario, we ran Gatekeeper from different controllers with varying distances to some attack edge and recorded the fraction of the chosen ticket sources that turn out to be Sybil nodes, f_{esc}.

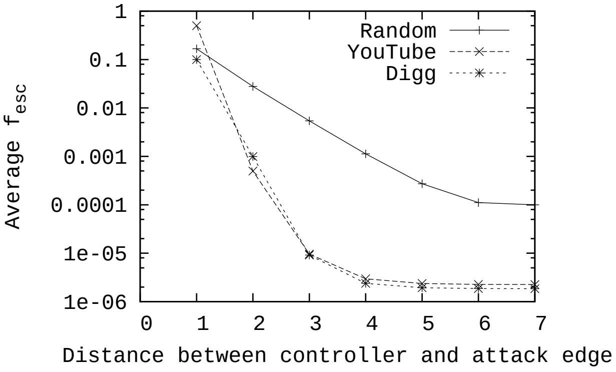

Fig. 4. The average random walk escape probability, f_{esc}, as a function of the distance between the controller and the closest attack edge

Figure 4 shows f_{esc} as a function of the distance between the controller and the closest attack edge under various graph topologies. We can see that f_{esc} drops off quickly to a negligible value as long as the controller is more than 2 hops away from the attacker. The worst case comes when the controller is the immediate neighbor of some attack edges. We first note that if $f_{esc} > f_{admit}$, the controller may accept arbitrarily many Sybil nodes because the $f_{esc} \cdot m$ sources can give infinitely many tickets to Sybils. As we have discussed in Section VII-C, our theoretical bound only holds when $f_{esc} < \frac{f_{admit}}{2-f_{admit}}$. Specifically, with a default value of $f_{admit} = 0.2$, f_{esc} must be smaller than 0.11. When the controller is immediately adjacent to some Sybil node, the escape probability is $1/d$ where d is the controller's node degree. Hence, only those controllers with more than 9 neighbors can afford to be-friend the attacker while still satisfying $f_{esc} < 0.11$ and achieving our proven bound.

IX. Conclusion

Gatekeeper is an optimal decentralized admission control protocol based on social networks that admits $O(\log k)$ Sybil nodes per attack edge with high probability. Our protocol improves over SybilLimit, the best known Sybil-resilient node admission protocol by a factor of $O(\log n)$ on random expander graphs when the attacker controls only $O(1)$ attack edges. Simulation results demonstrate that Gatekeeper works well on real-world social networks. Even in the face of a large number of attack edges, Gatekeeper can significantly limit the number of admitted Sybil nodes per attack edge.

Acknowledgments

We thank C. Thach Nguyen and Van H. Vu for helpful discussions. This work is supported by NSF CAREER Award (CNS-0747052). Nguyen Tran is also supported by a Google Ph.D. fellowship in distributed systems.

References

[1] Alon, N., and Spencer, J. H. *The probabilistic method.* John Wiley and Sons, 2008.

[2] Baden, R., Spring, N., and Bhattacharjee, B. Identifying close friends on the Internet. In *HotNets* (2009).

[3] Bollobas, B. *Random Graphs*. Cambridge University Press, 2001.

[4] Cheng, A., and Friedman, E. Sybilproof reputation mechanisms. In *P2PECON '05: Proceedings of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems* (2005), ACM, pp. 128–132.

[5] Chow, S. S. M. Running on Karma - P2P reputation and currency systems. In *CANS* (2007), pp. 146–158.

[6] Danezis, G., Lesniewski-Laas, C., Kaashoek, M. F., and Anderson, R. Sybil-resistant DHT routing. In *European Symposium On Research In Computer Security* (2008).

[7] Danezis, G., and Mittal, P. SybilInfer: Detecting sybil nodes using social networks. In *NDSS* (2009).

[8] Douceur, J. The Sybil attack. In *1st International Workshop on Peer-to-Peer Systems* (2002).

[9] Hu, Y., Perrig, A., and Sirbu, M. SPV: Secure path vector routing for securing BGP. In *Proc. of ACM SIGCOMM* (2004).

[10] Lesniewski-Laas, C., and Kaashoek, M. F. Whānau: A Sybil-proof distributed hash table. In *NSDI'10: Proceedings of the 7th USENIX Symposium on Networked Systems Design and Implementation* (2010), USENIX Association.

[11] Levien, R., and Aiken, A. Attack-resistant trust metrics for public key certification. In *SSYM'98: Proceedings of the 7th conference on USENIX Security Symposium, 1998* (Berkeley, CA, USA, 1998), USENIX Association, pp. 18–18.

[12] McKay, B. D., and Wormald, N. C. Uniform generation of random regular graphs of moderate degree. Journal of Algorithms.

[13] Mislove, A., Marcon, M., Gummadi, K. P., Druschel, P., and Bhattacharjee, B. Measurement and analysis of online social networks. In *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement* (2007), ACM, pp. 29–42.

[14] Peterson, R. S., and Sirer, E. G. AntFarm: Efficient content distribution with managed swarms. In *Proceedings of the 6th conference on Networked Systems Design & Implementation (NSDI)* (2009), USENIX Association, pp. 1–1.

[15] Piatek, M., Isdal, T., Krishnamurthy, A., and Anderson, T. One hop reputations for peer to peer file sharing workloads. In *NSDI'08: Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation* (2008), USENIX Association, pp. 1–14.

[16] Quercia, D., and Hailes, S. Sybil attacks against mobile users: Friends and foes to the rescue. In *IEEE INFOCOM* (2010).

[17] Tran, N., Li, J., Subramanian, L., and Chow, S. Optimal Sybil-resilient node admission control. Tech. rep., New York University, 2010. Available at http://www.news.cs.nyu.edu/∼trandinh/publications/GateKeeper_TR.pdf.

[18] Tran, N., Min, B., Li, J., and Subramanian, L. Sybil-resilient online content voting. In *NSDI'09: Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation* (2009), USENIX Association, pp. 15–28.

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

[20] Walsh, K., and Sirer, E. G. Experience with an object reputation system for peer-to-peer filesharing. In *NSDI'06: Proceedings of the 3rd conference on 3rd Symposium on Networked Systems Design & Implementation* (2006), USENIX Association, pp. 1–1.

[21] Wilson, C., Boe, B., Sala, A., Puttaswamy, K., and Zhao, B. User interactions in social networks and their implications. In *Proceedings of ACM EuroSys* (2009).

[22] Yu, H., Gibbons, P., Kaminsky, M., and Xiao, F. SybilLimit: A near-optimal social network defense against Sybil attacks. In *IEEE Symposium on Security and Privacy* (2008), IEEE Comoputer Society, pp. 3–17.

[23] Yu, H., Kaminsky, M., Gibbons, P. B., and Flaxman, A. SybilGuard: defending against Sybil attacks via social networks. In *SIGCOMM '06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications* (2006), ACM, pp. 267–278.

[24] Ziegler, C.-N., and Lausen, G. Propagation models for trust and distrust in social networks. *Information Systems Frontiers 7*, 4-5 (2005), 337–358.