Brief Announcement: Toward an Optimal Social Network Defense Against Sybil Attacks

Haifeng Yu
National Univ. of Singapore
haifeng@comp.nus.edu.sg

Phillip B. Gibbons
Intel Research Pittsburgh
phillip.b.gibbons@intel.com

Michael Kaminsky
Intel Research Pittsburgh
michael.e.kaminsky@intel.com

Categories and Subject Descriptors: C.2.4 [**Computer-Communication Networks**]: Distributed Systems – *distributed applications*

General Terms: Algorithms, Design, Security

Keywords: Sybil attack, social network, SybilGuard, SybilLimit

1. INTRODUCTION

Distributed systems are particularly vulnerable to *sybil attacks* [1], where a malicious user pretends to have multiple identities. Among the small number of decentralized solutions, our recently proposed SybilGuard [2] protocol is based on a unique insight on *social networks*. Formally, the system has n *honest users*, and one or more colluding *malicious users*. Each honest user has a single (honest) *identity*, while each malicious user has an arbitrary number of (malicious) *identities*. All identities created by the malicious users are called *sybil identities*. All of these honest and sybil identities are nodes in the social network. An (undirected) edge exists between two nodes if the two corresponding users have strong social connections and trust each other not to launch a sybil attack. For explanatory purposes, we also consider an (undirected) edge as two directed edges. Edges connecting the honest nodes and the sybil nodes are called *attack edges*. With SybilGuard, the number of attack edges is independent of the number of sybil nodes and is limited by the number of trust relation pairs between malicious and honest users. The basic idea in SybilGuard is that if malicious users create too many sybil nodes compared to the number of attack edges, the graph will have a large mixing time. On the other hand, social networks tend to be fast mixing (i.e., $\Theta(\log n)$ mixing time) [2].

SybilGuard is decentralized and enables any honest node V (called the *verifier*) to decide whether or not to *accept* another node S (called the *suspect*). "Accepting" means that V is willing to receive service (e.g., back-up service) from and provide service to S. Throughout this paper, we let ϵ and δ be arbitrary constants between 0 and 1. If the number of attack edges is at most $\Theta(\sqrt{n}/\log n)$, SybilGuard guarantees that i) an honest node accepts at most ϵn sybil nodes with probability at least $1-\delta$, and ii) an honest node accepts another honest node with probability at least $1-\delta$. We say that the *tolerance* of SybilGuard is $\Theta(\sqrt{n}/\log n)$.

This brief announcement first summarizes SybilGuard and then presents our new protocol, SybilLimit, that leverages the same social network but dramatically improves the tolerance from $\Theta(\sqrt{n}/\log n)$ to $\Theta(n/\log n)$. It has been shown [2] experimentally that for $n = 10^6$, SybilGuard's tolerance is around 2500 attack edges. SybilLimit, on the other hand, is expected to tolerate around 2.5×10^6 attack edges. To break SybilLimit, the adversary will need to establish 2.5 trust relations (on average) with *every* honest user in the system. Furthermore, we can show that the $\Theta(n/\log n)$ tolerance is optimal for any protocol based on social network mixing time.

This work is partly supported by NUS grants R-252-050-284-101 and R-252-050-284-133.

PODC'07, August 12–15, 2007, Portland, Oregon, USA.
ACM 978-1-59593-616-5/07/0008.

2. USING SOCIAL NETWORKS: SybilGuard

SybilGuard leverages a special kind of random walk called *random routes*, where each node uses a pre-computed random permutation as a one-to-one mapping from incoming edges to outgoing edges. As a result, two random routes entering an honest node along the same edge will always exit along the same edge (i.e., they *converge*). Furthermore, the outgoing edge uniquely determines the incoming edge as well; thus the random routes can be *back-traced*.

Accepting honest nodes. Each node performs a random route of length $\Theta(\sqrt{n}\log n)$. Unless the social network changes, each user needs to do this only once and the route will be "remembered". The verifier only accepts a suspect whose random route intersects with the verifier's random route. A length-w random walk starting from a uniformly random honest node will stay entirely within the honest region with probability of at least $1 - gw/n$, where g is the number of attack edges [2]. Thus with $g = \Theta(\sqrt{n}/\log n)$, the probability is at least $1-\delta$. Next, with $\Theta(\log n)$ mixing time, a random walk $\Theta(\sqrt{n}\log n)$ long will include $\Theta(\sqrt{n})$ random nodes drawn from the stationary distribution of the graph. It follows from the generalized Birthday Paradox that an honest suspect will have a random route that intersects with the verifier's random route (and thus be accepted) with probability at least $1-\delta$.

Bounding the number of sybil nodes accepted. To intersect with the verifier's random route and be accepted, a sybil node's random route must traverse one of the attack edges. Consider Figure 1 where there is only a single attack edge. Because of the convergence property, all the random routes from sybil nodes must merge completely once they traverse the attack edge. Thus, all of these routes will have the same intersection with the verifier's route; furthermore, they enter the intersection along the same directed edge ($e1$ in the figure). We say that a directed edge e *intersects* with a random route if the ending node of the direct edge is on the route. Obviously, with g attack edges there can be at most g directed edges that intersect with the verifier's route.

Because of convergence and back-traceability, there can be at most $\Theta(\sqrt{n}\log n)$ distinct random routes (of length $\Theta(\sqrt{n}\log n)$) that traverse a certain directed edge, if all the nodes in the random routes are honest nodes. Thus each directed edge conceptually has a *registry table* with $\Theta(\sqrt{n}\log n)$ entries. The ith entry is *registered* with the

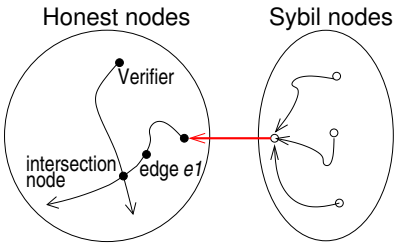

Figure 1: Routes traversing the same edge merge.

node whose random route traverses the directed edge at hop i. The verifier always confirms that the suspect is properly registered before accepting it. Thus the verifier will accept $\Theta(g \cdot \sqrt{n} \log n) = \epsilon n$ sybil nodes when $g = \Theta(\sqrt{n}/\log n)$.

Estimating the needed length of random routes. While the length of the random routes is designed to be $\Theta(\sqrt{n} \log n)$, the value of n is unknown. In SybilGuard, each node locally determines the needed length of the random routes via sampling, as follows. Each node is assumed to know a rough upper bound t on the mixing time. To obtain a sample, a node A first performs a random walk of length t, ending at some node B. Next A and B conceptually both perform random routes to determine how long the routes need to be to intersect. A sample is *bad* (i.e., potentially influenced by the adversary) if any of the three random walks/routes in the process enter the sybil region. Assuming $g = \Theta(\sqrt{n}/\log n)$, the probability of a sample being bad is at most δ.

3. TOWARD OPTIMALITY: SybilLimit

SybilGuard's (suboptimal) tolerance of $g = \Theta(\sqrt{n}/\log n)$ is actually quite fundamental as it is simultaneously needed by the following three properties:

- The number of sybil nodes accepted is at most ϵn with probability $1 - \delta$.
- The verifier's random route stays in the honest region with probability $1 - \delta$.
- The probability of a bad sample (for estimating random route length) is at most δ.

Thus the challenge for our new protocol, SybilLimit, is that we must preserve these three properties (or equivalent properties) when $g = \Theta(n/\log n)$.

Reducing the number of accepted sybil nodes from $\Theta(g\cdot\sqrt{n}\log n)$ to $\Theta(g\cdot\log n)$. We will consider $g = 1$ because the generalization to $g > 1$ is trivial. In SybilLimit, each node performs $\Theta(\sqrt{m})$ random routes of length $\Theta(\log n)$, where m is the total number of edges in the graph. (We will discuss later how nodes estimate $\Theta(\sqrt{m})$.) For each of route, a node conceptually records the termination edge (i.e., the last directed edge traversed). Intersection is performed between the verifier's termination edges and the suspect's termination edges.

Because of the deterministic routing tables, for any fixed w, a degree-d node is limited to at most d different length-w random routes. To overcome this, SybilLimit uses $\Theta(\sqrt{m})$ independent instances of the random route protocol for all verifiers. Each verifier takes one random route in each instance. Another $\Theta(\sqrt{m})$ instances are used for all suspects to ensure independence.

One can easily see that the intersection guarantees between honest nodes are still the same as before. For sybil nodes, in each instance, the adversary can shift the random route (crossing the attack edge) at $\Theta(\log n)$ different positions, and each position allows it to control one registry table entry. With $\Theta(\sqrt{m})$ instances, the adversary controls a total of $\Theta(\sqrt{m}\log n)$ entries. One can show that to maximize the number of sybil nodes accepted, the optimal strategy for the adversary is to register different nodes in different entries. The verifier has $\Theta(\sqrt{m})$ termination edges. For the termination edges in the honest region, we can show that because these edges are independent uniformly random edges, the number of sybil nodes accepted is at most $\Theta(\log n)$ with probability at least $1 - \delta$.

Protecting verifiers with termination edges in the sybil region. As discussed earlier, a length-w random walk enters the sybil region with probability at most gw/n. In SybilLimit, with $g = \Theta(n/\log n)$ and $w = \Theta(\log n)$, gw/n becomes a constant. Thus as many as a constant fraction of a verifier's $\Theta(\sqrt{m})$ random routes may enter the sybil region. The adversary can introduce arbitrary intersections with a termination edge in the sybil region, causing potentially an unlimited number of sybil nodes accepted. We say that a suspect is *accepted by a termination edge* e if e belongs to the intersection.

Our insight here is that because termination edges for verifier routes that remain in the honest region are uniformly random edges, each such termination edge should accept roughly $\Theta(n/\sqrt{m})$ out of the n honest suspects in the system. Making the intuition rigorous requires a careful argument because the termination edges of different suspects are not independent.

The above observation enables a verifier to enforce a limit of $\Theta(n/\sqrt{m})$ on the number of nodes accepted per termination edge, without hurting honest suspects. On the other hand, a Chernoff bound can show that an honest verifier will have $\Theta(\sqrt{m} \cdot g \log n/n)$ termination edges that are in the sybil region with at most probability δ. Because each termination edge will accept only $\Theta(n/\sqrt{m})$ nodes, the number of sybil nodes accepted by these termination edges will be $O(g \log n)$.

Estimating the number of routes needed. Similar as in SybilGuard, we assume we know a rough upper bound T on the mixing time. We need to estimate the number of random routes needed (i.e., $\Theta(\sqrt{m})$). The sampling technique as in SybilGuard now faces the challenge that with $g = \Theta(n/\log n)$, both A and B will quite likely have at least one route entering the sybil region. The adversary can then make them intersect and cause an under-estimation for the number of routes needed. (The adversary cannot, however, cause over-estimation.)

SybilLimit uses a novel and perhaps counter-intuitive design to address this challenge. The verifier maintains two sets of suspects, the *benchmark set* K and the *test set* S. The benchmark set is constructed by repeatedly taking random walks of length T and then adding the ending node to K. The test set contains the nodes that the verifier wants to verify. If $T = \Theta(\log n)$, then at most ϵ fraction of the nodes in K are sybil nodes. The verifier will increase the number of random routes (from 0) until most (e.g., 95%) of the nodes in K are accepted.

As an intuitive correctness argument, notice that the adversary may still cause under-estimation. Under-estimation will not increase the number of sybil nodes accepted. On the other hand, for honest suspects in S, the adversary does not know whether they belong to K or S. Thus, if the adversary manipulates intersections such that most of the suspects in K are accepted, with high probability, most honest suspects in S are accepted as well. This is somewhat counter-intuitive because SybilLimit manages to provide the desired end guarantee despite the under-estimation.

4. REFERENCES

[1] J. Douceur. The Sybil attack. In *IPTPS*, 2002.
[2] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. SybilGuard: Defending against sybil attacks via social networks. In *ACM SIGCOMM*, 2006.