Supplementary file for "SybilDefender: A Defense Mechanism for Sybil Attacks in Large Social Networks"

Wei Wei*, Fengyuan Xu*, Chiu C. Tan†, Qun Li*
*The College of William and Mary, †Temple University
*{wwei, fxu, liqun}@cs.wm.edu, †cctan@temple.edu

TABLE 1
Notations used in the analysis

$G(V, E)$	social graph, V is the set of nodes, E is the set of edges
P	transition matrix of the random walk process
n	number of honest nodes in G
λ	initial state vector of the random walk process
$\overline{\pi}$	stationary distribution of G
l	random walk length
Q_l	accumulated probability distribution of the nodes being traversed by a random walk with length l
t	threshold frequency used in the sybil identification algorithm
R	number of random walks originating from a given node
$\mathcal{D}(d)$	number of nodes with degree d

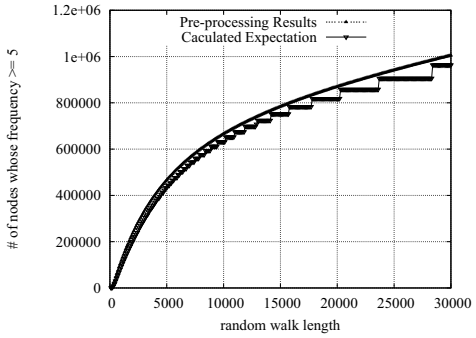

Fig. 1. Pre-processing results and calculated expectation values

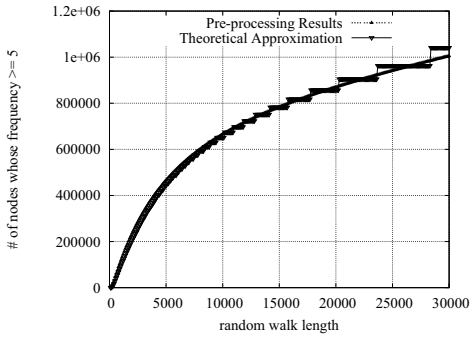

Fig. 2. Pre-processing results and theoretical approximate results

APPENDIX A
ANALYSIS OF THE SYBIL IDENTIFICATION ALGORITHM

In this subsection we investigate the validity of our sybil identification algorithm with theoretical analysis. For the ease of analysis we list the used notations in Table 1. A random walk with length l on an undirected graph G can be modeled as a Markov Chain process. The starting state of the random walk is described as λ, the initial state vector of V. $\lambda_v = 1$ if v is the starting node of the random walk, otherwise $\lambda_v = 0$. As defined in Section 3, P is the transition matrix of the random walk process. Therefore, the probability distribution of the nodes being visited by the i^{th} hop of the random walk is λP^i. Based on our fast-mixing assumption, λP^i converges to the stationary distribution $\overline{\pi}$ of G with $i \geq \Theta(\log n)$. The accumulated probability distribution of nodes being traversed by a random walk with length l is $Q_l = \sum_{i=0}^{l} \lambda P^i$, and $(Q_l)_j$, the j^{th} element in vector Q_l, is the expected number of times node j being traversed by a random walk with length l. Therefore, $R \cdot (Q_l)_j$ is the expected number of times node j being traversed by R random walks with length l originating from the same honest node, i.e., the expected frequency of j.

The pre-processing phase of our sybil identification algorithm sets up the criterion to identify sybil nodes by performing R random walks originating from each judge node with every length value $l \in \{l_{min}, l_{min} + 100, ..., l_{max}\}$, respectively, and records the mean and the standard deviation of the number of traversed nodes with frequency no smaller than t. Define set S_l as $\{j | R \cdot (Q_l)_j \geq t\}$, then $|S_l| = |\{j | \sum_{i=0}^{l} R \cdot (\lambda P^i)_j \geq t\}|$ is the expected number of nodes whose frequency is no smaller than t with R l-hop random walks. With a randomly chosen source node and $R = 2000$, based on our Facebook dataset, we calculate $|S_l|$ for different lengths

and draw the *calculated expectation* curve in Figure 1. To demonstrate the validity of our sybil identification algorithm, we set the number of judge nodes to be 10 and draw the *pre-processing results* curve based on the mean value outputs of the pre-processing phase in the same figure. It shows that even with a small number of judge nodes, the two curves match well when the random walk length is smaller than 10000 hops. As the random walk length increases there shows some horizontal segments in the calculated expectation curve. This is because in a fast-mixing network, $(\lambda P^i)_j$ converges to $\frac{d_j}{2|E|}$ with $i \geq \Theta(\log n)$, where d_j is the degree of node j and E is the set of edges. This means that with $l \geq \Theta(\log n)$ the value of $\sum_{i=0}^{l}(\lambda P^i)_j$ for all the nodes with the same degree increases by the same amount when l increases by 1, and thus their expected frequency, $\sum_{i=0}^{l} R{\cdot}(\lambda P^i)_j$, will reach the threshold t at the same random walk length, which leads to the jumps in the calculated expectation curve. Note that although the calculated expectation curve is divided into horizontal segments when the random walk length is large, its inflection points still match well with the pre-processing results curve. Figure 1 illustrates that *with a small number of judge nodes and limited R*, the results derived from the pre-processing phase of the sybil identification algorithm are already accurate enough to match with the expectation values.

Moreover, we will show that the results derived from the pre-processing phase *starting from a random honest node* are generic enough to serve as the criterion to identify sybil nodes. Since the network is fast mixing, given a random starting node, i.e., a random initial state vector λ, we have

$$\begin{aligned} Q_l &= \sum_{i=0}^{l} \lambda P^i \\ &\approx \lambda + \lambda P + ... + \lambda P^{\Theta(\log n)-1} + \underbrace{\overline{\pi} + ... + \overline{\pi}}_{l-\Theta(\log n)+1} \\ &\approx l\overline{\pi}. \end{aligned}$$

Also, $\overline{\pi}_j = \frac{d_j}{2|E|}$, then we have $(Q_l)_j \approx l\frac{d_j}{2|E|}$. Recall that $R \cdot (Q_l)_j$ is the expected frequency of node j. To make this value no smaller than t, we have

$$R \cdot (Q_l)_j \approx R \cdot l\frac{d_j}{2|E|} \geq t \;\Rightarrow\; d_j \geq \frac{2t|E|}{lR}. \qquad (1)$$

Define $S'_l = \{j | d_j \geq \frac{2t|E|}{lR}\}$. Then $|S'_l|$ is the approximate number of nodes whose frequency is no smaller than t with R l-hop random walks. Let $\mathcal{D}(d)$ be the number of nodes with degree d. Then

$$|S'_l| = \sum_{d=\lceil \frac{2t|E|}{lR} \rceil}^{\max} \mathcal{D}(d). \qquad (2)$$

Following Equation 2 we draw the *theoretical approximation* curve in Figure 2 based on our Facebook dataset, and we compare it with the pre-processing results curve identical to that in Figure 1. Note that Equation 2 is irrelevant to the initial state vector λ, so the shape of the theoretical approximation curve does not rely on the starting node. Figure 2 shows that the pre-processing results also match well with the theoretical approximate results. Similar to the caculated expectation curve, there are horizonal segments in the theoretical approximation curve. This is because node degrees are integers and l needs to increase by a certain amount such that the value of $\frac{2t|E|}{lR}$ reaches the next integer. Nevertheless, the middle point of each horizonal segment still matches with the pre-processing results curve. Figure 2 illustrates that the pre-processing results drawn from a random honest node can be effectively used as the criterion to identify sybil nodes.

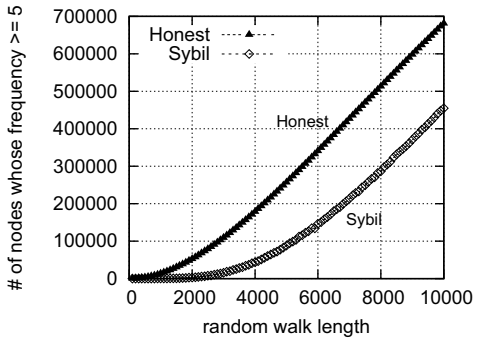

Fig. 3. Difference between the coverage of random walks originating from honest nodes and from sybil nodes

To gain an understanding of the difference between the footprint of random walks originating from an honest node and from a sybil node, assume φ is the expected number of hops for the random walks starting from a sybil node to enter the honest region. If we draw the curve for the number of nodes with frequency no smaller than t based on the random walks starting from that sybil node, it is approximately like moving the pre-processing results curve in Figures 1 and 2 to the right by φ and then raising it by the size of the sybil region. In the evaluation we will show that this difference is large enough to identify sybil nodes.

APPENDIX B
VERIFICATION OF THE INTUITION OF THE SYBIL IDENTIFICATION ALGORITHM

The intuition of our sybil identification algorithm is that, because of the existence of a small cut between the honest region and the sybil region, there is a difference between the coverage of random walks originating from an honest node and from a sybil node. Figure 3 illustrates this difference. Here, we use the PA model to construct the sybil region. We set the size of the sybil region to be 10000 nodes, and the number of attack edges to be 1000. In the experiments we perform 1000 random walks originating from each randomly selected source node. The upper curve in Figure 3 is the number of nodes traversed by random walks originating from an

TABLE 2
False positive and negative rates of the sybil identification algorithm (10000 attack edges)

	10 sybil nodes per attack edge (100000 sybils)								5 sybil nodes per attack edge (50000 sybils)							
	Orkut				Facebook				Orkut				Facebook			
	PA Model		ER Model		PA Model		ER Model		PA Model		ER Model		PA Model		ER Model	
	F^+	F^-	F^+	F^-	F^+	F^-	F^+	F^-	F^+	F^-	F^+	F^-	F^+	F^-	F^+	F^-
1000RWs	0	0.06%	0	0.13%	0.4%	0.76%	0.4%	0.78%	0	0.14%	0	0.23%	0.3%	1.40%	0.2%	1.31%
1500RWs	0	0.03%	0	0.12%	0.6%	0.67%	0.4%	0.68%	0	0.11%	0	0.22%	0.4%	1.31%	0.5%	1.07%
2000RWs	0	0.03%	0	0.12%	0.7%	0.62%	0.7%	0.66%	0	0.04%	0	0.21%	0.5%	1.09%	0.5%	0.97%

honest node no smaller than 5 times, while the lower curve is the number of nodes traversed by random walks originating from a sybil node no smaller than 5 times. Each point in the curves represents the mean value of 20 experiments. It is easy to see that the difference is larger than 200,000 nodes when the random walk length reaches 10000 hops. As described in Algorithm 2, we use $\tau = mean - \alpha * stdDeviation$ as the threshold to identify sybil nodes. In our experiments we observe that $stdDeviation < 1500$, so the sybil nodes can be identified even with a relatively large α, to limit the number of falsely identified honest nodes.

APPENDIX C
MORE EVALUATION RESULTS OF THE SYBIL IDENTIFICATION ALGORITHM

To investigate the performance of our sybil identification algorithm when more sybil nodes are controlled by the adversary, we raise the number of attack edges to 10000 and repeat the experiments. Following the approach mentioned in Section 5.1, creating 10000 attack edges means that on average the adversary needs to compromise 131 honest nodes in the Orkut dataset, or 546 honest nodes in the Facebook dataset. Table 2 lists the experimental results when each attack edge introduces 10 sybil nodes, which leads to a sybil region consisting of 100000 sybil nodes, and the results when each attack edge introduces 5 sybil nodes. It is easy to see that our algorithm still achieves low false positive and negative rates in these scenarios.

Online social networks do have communities, which have denser relationships internally. However, real-world evidence shows that these communities are not separated by small cuts. Instead, they tend to be well connected with each other. One proof is the remarkably small average path lengths and diameters of online social graphs, as shown in Section 5.4 of [1] and Section 4.1 of [2]. Previous sybil defense schemes and our scheme are all built on the assumption that the honest region is fast mixing, i.e., there is no small cut in the honest region. Yu et al. validate this assumption by showing that despite the existence of social communities, even social networks of very large scales tend to mix well within a rather small number of hops (10 to 20 hops) [3]. Our experimental results also validate this assumption: the false positive rate of our sybil identification algorithm is very low, which is slightly larger than or equal to 0. Considering

TABLE 3
False rates of SybilLimit on the Facebook dataset

	PA Model		ER Model	
	F^+	F^-	F^+	F^-
10000 sybils	1.5%	8.55%	0.6%	15.35%
5000 sybils	1.2%	15.16%	1.4%	32.62%
1000 sybils	1.4%	61.3%	0.8%	85.3%

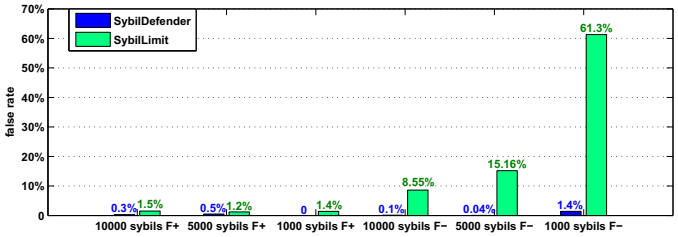

Fig. 4. Comparison between the false positive and negative rates of SybilDefender and those of SybilLimit

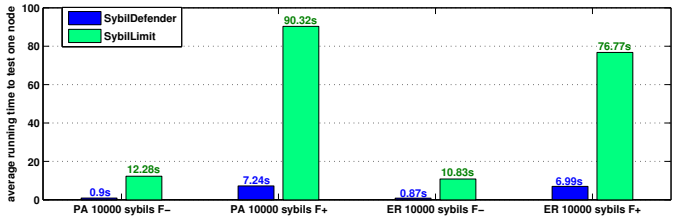

Fig. 5. Comparison between the average running time to test one node by SybilDefender and that by SybilLimit

that our experiments are performed over unmodified online social network samples, the low false positive rate indicates that the number of small cuts in real online social networks is very small (if any). Obviously, the existence of small cuts in the honest region will increase the false positive rate of our sybil identification algorithm. This can be addressed by first locating the community behind the small cut with our algorithms, and then identifying the identity (sybil or honest) of a node in the community through out-of-band methods. If this node is honest then with high probability the detected community is honest.

APPENDIX D
COMPARISON BETWEEN THE SYBIL IDENTIFICATION ALGORITHM AND EXISTING SCHEMES

We fully implemented SybilLimit and evaluated it using our Facebook dataset. We didn't evaluate SybilLimit on

the Orkut dataset as the running time is approximately 4 times longer, since the average degree of the Orkut dataset is about 4 times of the average degree of the Facebook dataset. Following the method in [3], we found the optimal parameters for SybilLimit on the Facebook dataset. We set w, the length of random routes, to be 20 hops, and r, the number of instances of the random route generation protocol, to be 10000. Table 3 lists SybilLimit's false positive and negative rates when each attack edge introduces 10 sybil nodes, 5 sybil nodes, and 1 sybil node, respectively. The results show that with each attack edge introducing 10 sybil nodes, SybilLimit accepts 8.55% of the sybil nodes when the sybil region is constructed by the PA model, and 15.35% of the sybil nodes when the sybil region is constructed by the ER model. In comparison, SybilDefender only accepts 0.1% of the sybil nodes in both cases when $R = 2000$. With the decrease of the number of sybil nodes introduced by each attack edge, the false positive and negative rates of SybilLimit raise significantly. When each attack edge introduces one sybil node, SybilLimit accepts the majority of the sybil nodes.

Figure 4 compares the false positive and negative rates of SybilDefender with those of SybilLimit, when the sybil region is built with the PA model. It is easy to see that in all the three cases the false positive rate of SybilDefender is lower than that of SybilLimit, and the false negative rate of SybilDefender is lower than that of SybilLimit by one to two orders of magnitude. The reason is SybilLimit assumes that almost all the short random routes originating from an honest node will stay within the honest region, and it bounds the number of admitted sybil nodes by the number of attack edges and random route length. When each attack edge introduces few sybil nodes, SybilLimit cannot effectively identify sybil nodes. On the other hand, SybilDefender interprets the small cut between the honest region and the sybil region as a bias in the coverage of the random walks originating from an honest node and from a sybil node. It can effectively identify the sybil nodes even when the number of sybil nodes introduced by each attack edge approaches the theoretical lower bound.

Figure 5 compares the average running time to test one node on one core of an Intel Xeon 2.93GHz processor by SybilDefender with that by SybilLimit. The results show that SybilDefender is faster than SybilLimit by more than 10 times. The reason is that SybilLimit invokes a large number ($r = 10000$ for our Facebook dataset) of instances of the random route generation protocol [3]. Within each instance a random routing table is generated for every node in the social graph. After all the instances are finished, SybilLimit verifies if the intersection condition and the balance condition are satisfied to determine whether to accept each suspect node. By contrast, SybilDefender only relies on performing a limited number of random walks, which can be done in a short time even on large-scale network graphs.

Viswanath et al. proposed using a community detection algorithm [4] as the ranking algorithm to investigate the similarity between different sybil defense schemes. We evaluated their algorithm using our two datasets, and found that the algorithm alone cannot be used to identify sybil nodes. The reason is that the algorithm starts from an honest node and iteratively adds nodes that improves the normalized conductance at each step. In our evaluation the normalized conductance always reaches the first inflection point after adding only several honest nodes. As a result, their algorithm cannot distinguish the sybil nodes from the honest nodes without providing a cutoff point.

We also evaluated Gatekeeper [5] using our datasets, which heavily relies on the assumption that the social networks are random expander. This assumption is stronger than our fast-mixing assumption and has not been validated in previous research, which makes Gatekeeper suffer from high false positive and negative rates on the real-world social topologies that exhibit asymmetries. For example, on our Facebook dataset with a 10000-node sybil region built through the PA model, the average false positive rate of Gatekeeper is 11.7%, and the average false negative rate is 17.2%. When the sybil region is built with the ER model, the average false positive rate is 11.7%, and the average false negative rate is 14.7%. In the evaluation we used the parameters ($m = 100, f_{admit} = 0.2$) recommended by [5] and repeated each experiment 20 times.

TABLE 4
Accuracy of the sybil community detection algorithm (10000 attack edges)

10 sybil nodes per attack edge (100000 sybils)				
	Percentage of found sybil nodes		Number of falsely detected honest nodes	
	Orkut	Facebook	Orkut	Facebook
PA model	99.77%	99.85%	0.2	3.7
ER model	99.90%	99.89%	0.1	3.0
5 sybil nodes per attack edge (50000 sybils)				
	Percentage of found sybil nodes		Number of falsely detected honest nodes	
	Orkut	Facebook	Orkut	Facebook
PA model	99.67%	99.69%	0.4	3.6
ER model	99.79%	99.66%	0.2	3.5
1 sybil node per attack edge (10000 sybils)				
	Percentage of found sybil nodes		Number of falsely detected honest nodes	
	Orkut	Facebook	Orkut	Facebook
PA model	99.28%	98.68%	0.4	4.3
ER model	98.88%	98.38%	0.1	3.7

APPENDIX E
MORE EVALUATION RESULTS OF THE SYBIL COMMUNITY DETECTION ALGORITHM

To investigate the scalability of our sybil community detection algorithm, we raise the number of attack edges to 10000 and repeat the experiments. All the parameters used in the algorithm stay the same. The results are shown in Table 4, which illustrates that with the size

TABLE 5
Accuracy of the sybil community detection algorithm on a weighted social network

	Percentage of found sybil nodes			Number of falsely detected honest nodes		
# of sybils	10000	5000	1000	10000	5000	1000
PA Model	99.79%	99.66%	98.6%	0.4	0.5	0.6
ER Model	99.83%	99.68%	98.4%	0.5	0.8	0.7

of the sybil region increasing by 10 times, our algorithm still achieves similar performance.

We also evaluated the performance of the sybil community detection algorithm on the weighted social network sample used in Section 5.2.1. The modified algorithm runs by performing *weighted partial* random walks. Each weighted partial random walk behaves the same as the partial random walks, i.e., it does not traverse the same node more than once. However, at each intermediate node, the next hop is chosen by considering edge weights, as described in Section 5.2.1. When building the sybil regions, we randomly assign weights to the edges connecting sybil nodes, such that the average weight is equal to the average weight of our dataset. The results in Table 5 show that the number of both undetected sybil nodes and falsely detected honest nodes is very small.

APPENDIX F
SURVEY RESULTS OF OUR FACEBOOK APPLICATION

Figure 6 is the user interface of our Facebook application, "Rate Your Relationships". Users of the application can rate each of their relations on Facebook either as "Friend" or "Stranger", where "Stranger" means the user hardly has any impression about this relation.

To investigate the user experience of our application, we carried out a survey through the Amazon Mechanical Turk platform [6]. In the survey we asked the respondents to use our application to rate all the relations in their Facebook friend lists, either as "Friend" or "Stranger", where "Stranger" means the respondent hardly has any impression about this relation. Then the respondents are asked 4 questions based on the outputs of our application, including "What's the completion code?", "What's the number of relations in your friend list?", "What's the percentage of strangers in your friend list?", and "Will you use this kind of relationship rating applications in online social networks if you know it can help to defend against malicious users?". Note that the completion code, the number of relations, and the percentage of strangers will show only after the respondent has rated all the relations in her Facebook friend list.

We get the results from 214 respondents, whose average number of relations is 118. The average time for the respondents to finish the survey is 249 seconds. This indicates that it does not take long for online social network users to rate their relationships. Figure 7 is

Fig. 6. Our Facebook application: Rate Your Relationships

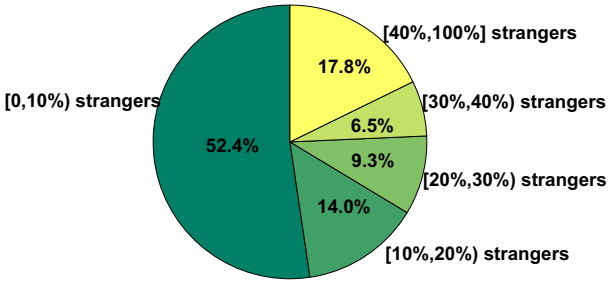

Fig. 7. The proportion of respondents based on the percentage of strangers in their Facebook relations

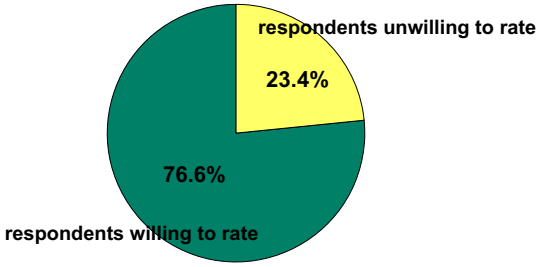

Fig. 8. Respondents' willingness to rate their relationships to defend against malicious users

the pie chart of the proportion of respondents based on the percentage of strangers in their relations. In the survey 47.6% of the respondents indicate that more than 10% of their Facebook relations are strangers, and the average percentage of strangers among all the relations is 19.8%, which shows that the assumption made by previous work that all the links in social networks are trusted does not apply to online social networks. As mentioned in Section 4.4, one way to limit the number of attack edges is to let users rate their relationships, and all the edges rated as stranger are removed from the social network graph when applying sybil defense mechanisms. As shown in Figure 8, in the survey 76.6% of the respondents would like to rate their relationships when they know this can help to defend against malicious users. This shows that relationship rating is a promising way to limit the capacity of the adversary to create attack edges.

REFERENCES

[1] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and analysis of online social networks," in *ACM/USENIX IMC*, 2007.

[2] C. Wilson, B. Boe, A. Sala, K. P. N. Puttaswamy, and B. Y. Zhao, "User interactions in social networks and their implications," in *EuroSys*, 2009.

[3] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao, "Sybillimit: A near-optimal social network defense against sybil attacks," in *IEEE Symposium on Security and Privacy*, 2008.

[4] B. Viswanath, A. Post, K. P. Gummadi, and A. Mislove, "An analysis of social network-based sybil defenses," in *SIGCOMM*, 2010.

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

[6] "Amazon mechanical turk," https://www.mturk.com/mturk.