Sybil-Resistant Trust Value of Social Network Graph

Kyungbaek Kim, *Member, KISM*

***Abstract*—The distributed systems tend to be vulnerable to the Sybil attack. This paper proposes three methods to initialize the Sybil-resistant trust value from a social network graph. The trust value implies the Sybil-resistant property which is embedded in the social network graph, and it can be used for the admission control as well as for the reference value. Through the real social graph-based evaluation, we compare three methods with each other, and we find that Sybil nodes get just about 0.025 trust value while the average of trust value for honest nodes is around 0.9.**

***Index Terms*—Sybil-resistant Trust Value, Social Network Graph, Sybil Attack.**

I. INTRODUCTION

As distributed systems become very popular and the participants inevitably cooperate with strangers, the trust between participants becomes essential to operate distributed systems properly. While some reputation systems [1][2] have been proposed to figure out the trust between participants by referring their past behaviors, it is still hard to judge the trust of new participants which have no past record. According to this, distributed systems are still vulnerable to the ***Sybil attack*** [3], where a single malicious user pretends to have multiple participants which are called ***Sybil participants***. The purpose of creating Sybil participants is not only to subvert distributed systems, but also to increase the creator's profit abnormally. Every new Sybil participant, who does not have any past behavior, can avoid the restriction of those reputation systems for a while. When a Sybil participant is eventually restricted by those reputation systems, a malicious user simply discard the restricted Sybil participant and create other new Sybil participants.

In this paper, we propose methods to initialize the Sybil-resistant trust value from a social network graph, for an individual participant of a distributed system to recognize the Sybil participants. The social network graph is corresponding to the real relationship between identities, and the artificially generated Sybil identities are barely connected to the real identities except their creators or themselves. This Sybil-resistant property can be embedded to the trust value extracted from the social network graph. By referring the trust value, a participant can evaluate the likelihood that the first-met participant is Sybil. The trust value can be assigned to each node in [0,1] where 0 means that the owner of this value most likely a Sybil identity.

From now on, we explore three methods for initializing the Sybil-resistant trust value; Random Route Intersection (RRI), Random Route Tail Intersection (RRTI), and Random Walk Tail Intersection (RWTI).

II. METHODS INITIALIZING SYBIL-RESISTANT TRUST VALUE

A. Assumptions and overall operation

In a social network graph, a node is mapped to the user identity of a participant in a distributed system. While an honest user has a single identity, a malicious user can generate multiple fake identities. Two nodes are connected by an undirected edge only if both of them trust each other. In the given social network graph, there are two big regions: the ***honest region*** and the ***Sybil region***. The honest region is one strongly connected component composed of the honest nodes which collaborate with others honestly. That is, any two honest nodes have a path connecting them in the region. The Sybil region is one another strongly connected component composed of the Sybil nodes which are generated for malicious purposes. While each of regions is an individual fast-mixing graph [6], the whole social network graph is not because of the ***attack edge*** which is the special edge connecting the Sybil region to the honest region. The attack edge can be created if an honest node is deceived by a Sybil node.

We note that the path from the honest region to the Sybil region or vice versa should pass through the attack edges. Also, because of the limited number of attack edges, it is hard that a random walk over the given social network graph crosses the attack edges. Subsequently, if a node in the honest region obtains samples in the manner of the random walk, it is less likely to get samples belonging to the Sybil region.

Let us say that there are a suspect node S, to which we want to assign a trust value, and a verifier node V, which resides in the honest region. We assume that the verifier node is very trustable and hard to be compromised by malicious users. Both S and V get the set of samples in the manner of the random walk. If there is at least one sample contained in both S's and V's set of samples, we say that V accepts S. If S resides in the honest region, V most likely accepts S. Otherwise, if S locates in the Sybil region, V less likely accepts S.

However, using only one verifier node may cause the subjective trust value and be a vulnerable point of attacks by malicious users. We use multiple verifier nodes rather

Kyungbaek Kim is with the Department of Electronics and Computer Engineering, Chonnam National University, 77 Yongbong-ro, Buk-ku, Gwangju, 500-757, South Korea (Email: kyungbaekkim@chonnam.ac.kr)

than single verifier in order to get more objective trust value and make the methods getting the trust value resilient to attacks. Also, we can quantify the trust value as ***(# of accepted verifiers)/l***, where ***l*** is the total number of verifier nodes. That is, the trust value represents the likelihood that verifier nodes accept a suspect node, and the likelihood is mainly affected by the ways how to obtain samples. Hereafter, we describe the three different methods to obtain a set of samples for a suspect/verifier node.

B. Random Route Intersection Method

The ***RRI*** (***Random Route Intersection***) method exploits the ***random route*** [4] to get a set of samples. The random route is a special kind of random walk. Each node prepares a pre-computed random permutation as a one-to-one mapping from incoming edges to outgoing edges. Since every edge can be an incoming edges or outgoing edges, the length of the random permutation is same to the number of edges of a node. A random route determines a next destination by referring the outgoing edge of the permutation corresponding to the incoming edge rather than by picking an outgoing edge randomly. The most promising property of a random route is the ***convergence property***, i.e. two random routes entering a node along the same incoming edge will always exist along the same outgoing edge [4]. In order to obtain samples, a node initiates the random routes starting from its all outgoing edges, and all the visited nodes are added to the set of samples.

The length of a random route, ***w***, is the performance knob of RRI. As ***w*** increases, the size of the set of sample nodes increases, and the probability that a verifier node accepts a suspect node also increases. On the fast-mixing graph, ***w*** should be sufficiently long, such as ***Θ(√(n log(n)))*** [4] (***n*** is the number of nodes in the given social graph). But when ***w*** increases in order to augment the average of the trust value for honest nodes, the trust value of Sybil nodes may increase excessively. That is, as ***w*** increases, the probability that a random route crosses the attack edges increases. Moreover, once a random route initiated by a Sybil node stretches into the honest region, the probability that a verifier node accepts the Sybil node increases significantly along with ***w***.

C. Random Route Tail Intersection Method

While RRI collects all the nodes on a random route as the samples, the ***RRTI*** (***Random Route Tail Intersection***) method obtains a set of samples by gathering a ***tail*** which is the last direct edge of a random route multiple times [5]. That is, a node initiates a random route starting from a random outgoing edge and it adds the tail of the random route to the set of samples. The interesting property of the set of sample tails is that the tails are uniformly distributed on the fast-mixing social graph only if the length of the random route (***w***) is sufficiently long such as ***Θ(√(log(n)))*** [5].

RRTI gets only one tail from one random route. According to the Birthday Paradox, in order to ensure that the two set of random sample tails share at least one common tail with high probability, the required size of the set of samples, ***r***, is approximate to ***Θ(√(m))*** where ***m*** is the number of directed edges of the fast-mixing social graph [5]. Subsequently, a node should initiate a random route ***r*** times to get ***r*** tails. In this case, a node should prepare ***r*** independent random permutations to get ***r*** completely independent tails. For each time, a random route uses the designated random permutation. That is, the random route initiated by using the N_{th} permutation of a node always uses the N_{th} permutation of all the other nodes until it meets the tail.

Because RRTI uses shorter ***w*** such as ***Θ(√(log(n)))*** than RRI (***Θ(√(n log(n)))***), the probability of RRTI that a random route of a Sybil node crosses the attack edges becomes less than RRI. Even if a random route of a Sybil node crosses the attack edges, it can affect only few tails because of the convergence property of the random route.

Despite of the excellent performance of decreasing the probability that a verifier node accepts a Sybil node, RRTI is too much memory bounded method. RRTI should manage ***r*** permutations for all nodes. This huge number of permutations incurs huge runtime memory spaces, and whenever the membership of the given social graph changes all the related tables should be updated.

D. Random Walk Tail Intersection Method

The ***RWTI*** (***Random Walk Tail Intersection***) method simply uses a random walk rather than the random route in order to eliminate the overhead incurred by preparing the huge number of permutations in RRTI. In RWTI, a node performs a random walk for ***r*** times to get ***r*** tails as a set of samples. Since there is no need to prepare huge number of permutations and to look up the designated permutation for each random walk, RWTI performs faster and more efficiently than RRTI. Despite losing the convergence property of the random route, the sufficiently long random walk, such as ***Θ(√(log(n)))***, can hold the same property that the tails are uniformly distributed [7]. The required size of the set of samples, r, is also similar to RRTI, such as ***Θ(√(m)***.

Because the required length of a random walk and the required size of the set of samples for RWTI are almost same to RRTI, the performance of RWTI may be almost similar to RRTI. However, the lack of the convergence property might hamper the performance of RWTI. However, because the portion of attack edges of a node is very small, the probability that a node randomly selects an attack edge as the next destination of a random walk is very small. Moreover, the log-scaled length of the random walk is generally too short for the random walk initiated by a Sybil node to cross an attack edge. Because of these positive properties, the side effect of the random walk of RWTI can be limited.

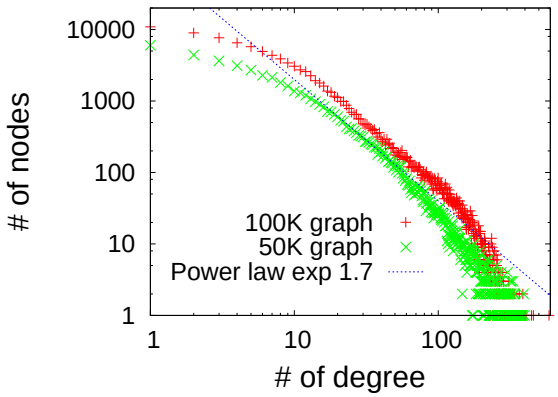

Fig.1. Degree distribution of sampled social network graphs (50K and 100K)

III. EVALUATION

To evaluate the effectiveness of our methods, we used the crawled Facebook social graph [8]. To guarantee our assumptions and to speed up the evaluation process, we extracted the sub social graphs which are strongly connected components from the whole social graph by using forest-fire sampling technique [9]. We got two sub social graphs; 50K and 100K whose total number of directed edges are 905004 and 1861360, respectively. Diameter and radius of both graphs are same to 18 and 6, respectively. Both of the sub social graphs are power-law networks having ~1.7 coefficient like Fig.1. We assume that the sub social graphs represent honest regions and the Sybil region is generated artificially where the average degree of a node is 14. The number of the randomly selected verifier nodes (***l***) set to 100. In this paper, we only show the results for 100K graph because the results for 50K graph is almost identical.

A. Preliminary Results

At first, we explored the effects of parameters of each method and figured out the proper setting for each method to assign reasonably high trust value to honest nodes and low trust value to Sybil nodes. The preliminary results for the proposed methods are shown in Fig.2, Fig.3, and Fig.4. In RRI, as ***w*** increases to get high trust value for honest nodes, the trust value for Sybil nodes increases moderately. Both of RRTI and RWTI is mainly affected by ***r***. As ***r*** increases, the trust value increases; this follows the generalized Birthday Paradox. The length of a random route/walk (***w***) also affects the trust value, but it is very limited. We found that RWTI assigns slightly less trust value to honest nodes than RRTI under the same setting. This is because a random route implying the convergence property provides more uniformly distributed tails than a random walk. If RWTI wants to cope with it, RWTI use longer random walks than random routes of RRTI. According to this, for the rest of results, RRI uses ***w=200***, RRTI uses ***r=2000*** and ***w=15***, and RWTI uses ***r=2000*** and ***w=20***.

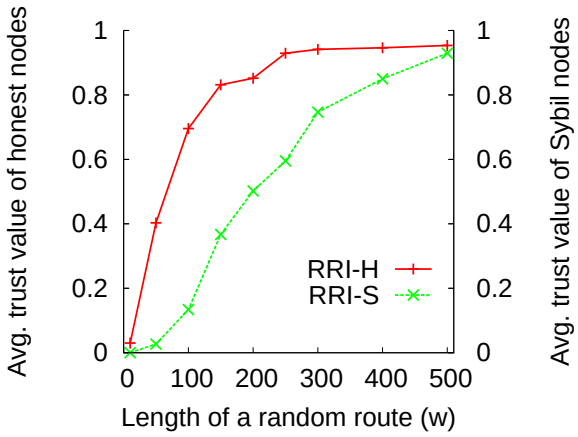

Fig.2. Preliminary results for RRI with 100K graph. “H” and “S” represents honest and Sybil nodes respectively.

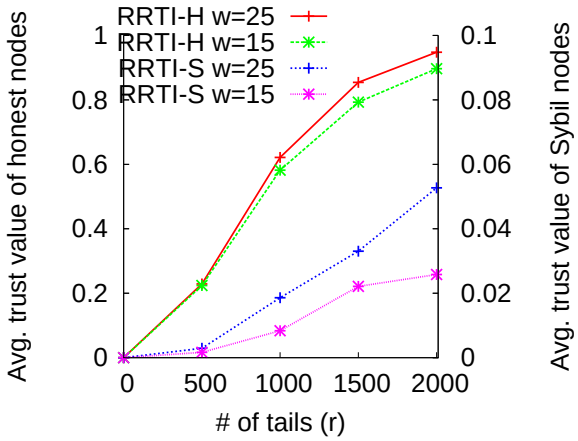

Fig.3. Preliminary results for RRTI with 100K graph. “H” and “S” represents honest and Sybil nodes respectively.

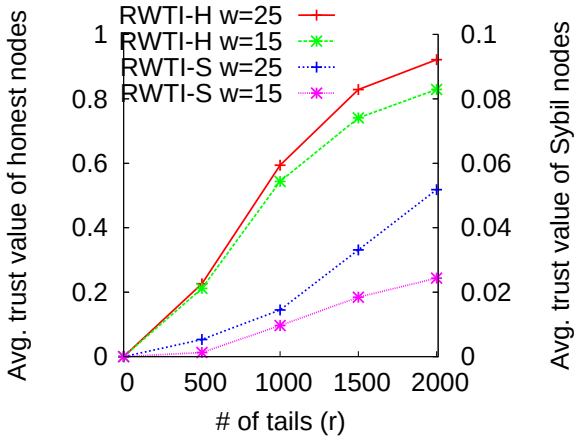

Fig.4. Preliminary results for RWTI with 100K graph. “H” and “S” represents honest and Sybil nodes respectively.

B. Sybil Resistant Trust Value

With the proper settings getting from the preliminary results, we show the CDF of Sybil-resistant trust value for honest nodes in Fig.5. All of three methods have the very

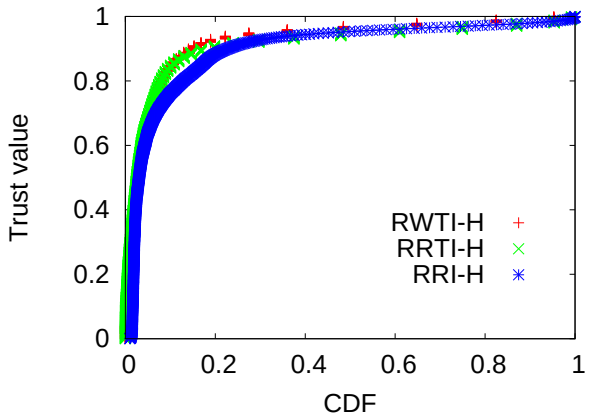

Fig.5. Distribution of Sybil-resistant trust value of honest nodes.

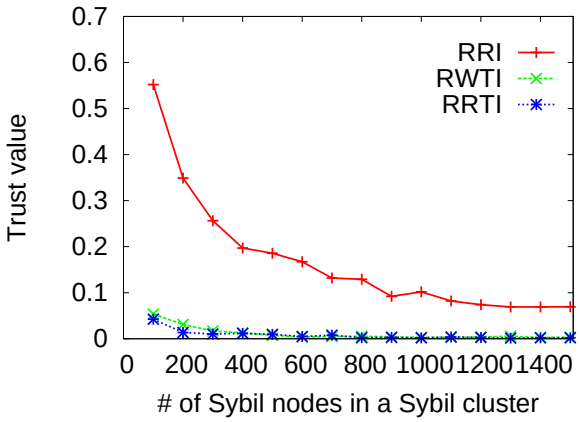

Fig.6. Distribution of Sybil-resistant trust value of Sybil nodes with various size of Sybil clusters.

similar shape; over 90\% honest nodes have higher trust value than 0.8, but around 10\% honest nodes have low trust values. We obtained this distribution of the trust value of honest nodes regardless of the number of verifiers (***l***). That is, despite we increased the number of verifiers from 100 to 1000, there is no enhancement of the portion of honest nodes obtaining high trust value.

Fig.6 shows how our methods are tolerant to the Sybil attack increasing the size of the Sybil cluster which has limited number of attack edges. The assigned trust value for Sybil nodes decreases along with the size of Sybil cluster. As the size of Sybil cluster increases, a random route/walk wanders in the Sybil cluster longer and the probability that a random route/walk crosses the attack edge decreases.

In RRI, some Sybil nodes may have high trust value, especially when the size of Sybil cluster is small. These Sybil nodes usually locate near the ***gateway Sybil node*** which is the Sybil node having an attack edge. The random route/walk initiated by them can cross the attack edge with much higher probability than the other Sybil nodes, and they can get higher trust value. On the other hand, in RRTI and RWTI, because ***w*** is very short, the Sybil nodes near to the gateway Sybil nodes cannot get high trust value.

IV. RELATED WORKS

There are several works to mitigate the Sybil attack by exploiting a social network graph [4][5][12]. They focused on the admission control preventing the access of Sybil identities. Paper [10] proposed the Sybil-resilient voting system and paper [11] presented the Sybil-resilient messaging system. Our paper focuses on evaluating the likelihood of Sybil node as a trust value.

V. CONCLUSION

We proposed three methods (RRI, RRTI, and RWTI) to initialize the Sybil-resistant trust value by using the social network graph. All of them function effectively against the simple Sybil attack which has limited number of attack edges. Even though the Sybil attack becomes more sophisticated, RRTI and RWTI can be still effective. In the aspect of the computation cost, RRI is most efficient among them and RWTI is also faster and lighter than RRTI. Even though RWTI suffers from a slight side effect caused by the lack of convergence property, its performance compares favorably with RRTI, and it can be useful to the distributed systems which concern the memory limitation and the computation cost.

REFERENCES

[1] S.D. Kamvar, M.T. Schlosser and H. Garcia-molina, "The EigenTrust Algorithm for Reputation Management in P2P Networks," *Proc. 12th WWW Conference,* 2003.

[2] A.G.P. Rahbar and O. Yang. "PowerTrust: A Robust and Scalable Reputation System for Trusted Peer-to-Peer Computing." *IEEE Trans. Parallel and Distributed Computing*, Vol.18, Issue.4, April 2007

[3] J. R. Douceur. "The sybil attack," *Proc. IPTPS 2002*, pages 251-260, 2002.

[4] H. Yu, M. Kaminsky, P. B. Gibbons and A. Flaxman. "SybilGuard: Defending Against Sybil Attacks via Social Networks," *Proc. ACM SIGCOMM 2006,* August 2006

[5] H. Yu, P. B. Gibbons, M. Kaminsky and F. Xiao. "SybilLimit: A Near-Optimal Social Network Defense against Sybil Attacks," *Proc IEEE Symposium on Security and Provacy 2008*, pp. 3-17, 2008

[6] D. J. Watts and S. H. Strogatz. "Collective dynamics of small-world networks", *Nature*, 393(6684), 1998.

[7] M. Mitzenmacher and E. Upfal. "Probability and Computing," *Cambridge University Press*, 2005

[8] M. Sirivianos, K. Kim, X. Yang, "SocialFilter Introducing Social Trust to Collaborative Spam Mitigation", *Proc. Infocom 2011*, 2011

[9] J. Leskovec and C. Faloutsos. "Sampling from large graphs," *Proc. ACM SIGKDD 2006*, 2006.

[10] N. Tran, B. Min, J. Li and L. Subramanian. "Sybil-resilient online content voting," *Proc. NSDI 2009*, 2009

[11] A. Mislove, A. Post, P. Druschel and K. P. Gummadi. "Ostra: Leveraging Trust to Thwart Unwanted Communication," *Proc. NSDI 2008*, 2008

[12] G. Danezis, C. Lesniewski-Laas, M. F. Kaashoek, and R. J. Anderson. "Sybil-resistant DHT routing," *Proc. ESORICS 2005*, pages 305-318, 2005.