Trust-Based Collection of Information in Distributed Reputation Networks

Gkorou, D. ; Pouwelse, Johan; Epema, D.H.J.

DOI
10.1145/2695664.2695868

Publication date
2015

Document Version
Accepted author manuscript

Published in
SAC '15 Proceedings of the 30th Annual ACM Symposium on Applied Computing

Trust-based Collection of Information in Distributed Reputation Networks

Dimitra Gkorou
EEMCS, Parallel and
Distributed Systems
Mekelweg 4, 2628 CD
Delft, the Netherlands
d.Gkorou@tudelft.nl

Johan Pouwelse
EEMCS, Parallel and
Distributed Systems
Mekelweg 4, 2628 CD
Delft, the Netherlands
j.a.Pouwelse@tudelft.nl

Dick Epema
EEMCS, Parallel and
Distributed Systems
Mekelweg 4, 2628 CD
Delft, the Netherlands
d.h.j.Epema@tudelft.nl

ABSTRACT

Distributed reputation systems establish trust among strangers in online communities and provide incentives for users to contribute. In these systems, each user monitors the interactions of others and computes the reputations accordingly. Collecting information for computing the reputations is challenging for the users due to their vulnerability to attacks, their limited resources, and the burst of their interactions. The low cost of creating accounts in most reputation systems makes them popular to million of users, but also enables malicious users to boost their reputations by performing Sybil attacks. Furthermore, the burst of user interactions causes an information overload. To avoid the impact of malicious users and information overload, we propose EscapeLimit, a sybil attack-resistant, computationally simple, and fully distributed method for information collection. EscapeLimit leverages user interactions as indicators of trust and similarity between the corresponding users, and collects relevant and trusted information by limiting the escape probability into the Sybil area. We evaluate it by emulating interaction patterns derived from synthetic and real-world networks. Our evaluation shows EscapeLimit's effectiveness in terms of its resilience to Sybil attacks, its scalability, and its ability to provide relevant information to each user.

1. INTRODUCTION

Reputation systems establish trust and provide incentives for cooperation among users in many decentralized networks such as P2P networks [13, 19], distributed social networks [21], and markets on mobile devices [8]. In such systems, each user independently collects and stores the history of user interactions, and aggregates it in one reputation value per user. The collection of the history of user interactions directly affects both the quality and the cost of a reputation system [10]. In this paper, we propose a scalable method based on random walks to collect information in distributed reputation networks.

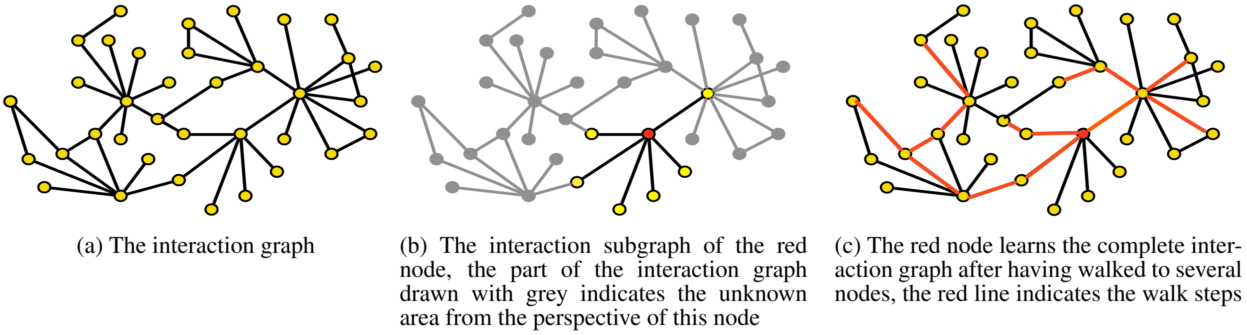
Characterized by their open nature, most distributed systems such as P2P networks are popular to millions of users, yet they are very vulnerable to attacks, scalability issues and information overload. The ease of creating accounts in these systems enables malicious users to create numerous fake identities, their *Sybils*, and to spread false reports about their interactions. Moreover, with the increase of network size, communication load balancing becomes harder. Finally, the large number of interactions between users causes an information overload. By blindly storing and processing information, users easily become victims of Sybil attacks as well as waste their resources for information that contributes little to the reputations. To avoid the impact of malicious nodes and the misuse of their resources, users have to collect *trusted* and *relevant* information.

In this paper, we propose EscapeLimit, a method to collect trusted and relevant information in decentralized reputation systems. While traditional methods against Sybil attacks, such as SybilGuard [29] and SumUp [24], require the existence of a social network, we assume no such network. As a result, our approach is suitable for systems without a social network such as P2P networks and markets on mobile devices [8]. Our only assumption is the existence of interactions between users, such as file sharing in the context of P2P networks, and exchanged messages in the context of social networks. We interpret interactions as indicators of trust and similarity between the corresponding users. EscapeLimit uses random walks with restarts to successively visit nodes to collect interaction reports. In this way, it exploits the transitive flow of positive interactions and guarantees a link between the creator of a report and the user who uses it in his reputation calculations. EscapeLimit reduces the *escape probability*, which is defined as the probability that a random walk initiated by an honest node ends up in a Sybil node.

We evaluate EscapeLimit in terms of its resilience to Sybil attacks, its scalability, and its ability to collect relevant information. In our experimentation, we emulate user interaction patterns derived from networks with different properties. Particularly, we use synthetic power-law networks, and real-world networks derived from the Internet-deployed Bartercast reputation system [11] used in the BitTorrent-based client Tribler [20], and from Facebook [25]. In our evaluation, the reputations of users are simply computed as the ratios of the resources they contribute to the system and the resources they consume. In this way, we can assess the quality of the collection of information independently of the computation of reputations. To further enhance EscapeLimit, we bias random walks with trust-driven properties such as the strength of user interactions and the activity level of users, and we explore their effect on the collection of information. Our evaluation shows that the strength of user interactions guides random walks efficiently in almost any type of network. Finally, we show the ability of EscapeLimit to radically limit the influence of Sybils and to collect relevant information.

2. PROBLEM STATEMENT

In this section, we describe the problem of collecting information in distributed reputation systems and three requirements that a collection method applicable in distributed systems has to fulfil. We model the history of all the interactions in a decentralized reputation system as the (*user*) *interaction graph* with the nodes of the system representing its vertices and the interactions among the nodes of the system representing its edges. As each node in de-



(a) The interaction graph

(b) The interaction subgraph of the red node, the part of the interaction graph drawn with grey indicates the unknown area from the perspective of this node

(c) The red node learns the complete interaction graph after having walked to several nodes, the red line indicates the walk steps

Figure 1: The red node collects information using its interaction subgraph.

centralized reputation systems has a limited view of the system, it builds its own subgraph of the interaction graph, its (*user*) *interaction subgraph*, using its stored interactions. In Figures 1a and 1b, we present an illustration of the interaction graph and the interaction subgraph available at each node. Initially, a node knows only its own *direct interactions* in its interaction subgraph and so, it is able to compute only the reputations of the nodes with which it has previously interacted.

Nodes should periodically contact each other in order to acquire information about the interaction subgraphs of other nodes, since they need to expand their interaction subgraphs to compute the reputations of other nodes in the network. Ideally, the interaction subgraph of a node converges to the interaction graph when it successively contacts other nodes. The information collection is crucial for the quality of the computed reputations, since poor information collecting results in inaccurate reputations [10]. In order to be applicable in decentralized reputation systems, a collection method must satisfy the following three requirements:

Resilience to attacks. From the various self-promoting attacks in decentralized reputation systems, we consider only sybil attacks since most other types (e.g., link farming, collusion, spam), can be seen as special cases of it. In a *sybil attack* [17], the attacker manages to gain a disproportionally large influence on the network, for instance, in order to determine the result of a voting process, to spam other users, to monopolize system resources, or even to sell its sybils to other attackers as has occurred in Facebook [7]. The sybil attack is predominant in reputation systems. It has been reported that in Facebook more than 1.5M fake accounts have been identified during February 2010 [7], that in RenRen more than 660K fake accounts exist [27], and in Tuenti about 180K [7]. Acquiring polluted information results in inaccurate reputations.

Scalability. We define scalability in terms of computational and communication cost at each node, which increases with the size of the network. Due to the limited resources available at nodes, the method of information of collecting information should not add a lot of computational and communication cost. Moreover, it should distribute the communication load evenly across all the nodes. Particularly, in many real world networks where a few nodes have the majority of connections, balancing communication load is challenging.

Relevance of Information. Not all the information about node interactions contributes equally to the computation of reputations at every node. Interactions of high strengths occurring close to a node contribute more in the computation of reputations than interactions of low strengths in the periphery of the network. Furthermore, a node is more likely to interact with highly active nodes close to it, and so, computing the reputations of these nodes accurately is more useful. Therefore, every node must acquire *relevant* information so that the reputations computed from its interaction subgraph are close to the reputations of the interaction graph.

These three requirements cannot be completely satisfied by a single method since they are conflicting to a some extent. For instance, in a perfect attack-resilient solution, every node should only contact nodes with which it has previous successful interactions. However, this results in very poor collection of relevant information since that node fails to compute the reputations of potential new encounters. Furthermore, a node can obtain relevant information fast if it contacts the highly active nodes in the system often, since those nodes keep the network connected and perform the largest number of interactions. However, this results in overloading these nodes. Nevertheless, by not considering all these requirements, a collection mechanism is not applicable to online distributed systems and so, it has to make a trade-off among them.

3. COLLECTING INFORMATION USING RANDOM WALKS

In this section we describe EscapeLimit, the network model, and the proposed trust-driven biases for random walks.

General description. EscapeLimit is based on random walks, which are computationally tractable, and naturally decentralized using only information locally available at each node. Furthermore, random walks are flexible and with the appropriate biases, they are able to quickly detect relevant and trustworthy information in a network. As a result, they have been widely used in distributed systems for search [6], topology maintenance, and computations of reputations such as Eigentrust [16] and SybilRank [7]. In EscapeLimit, we use random walks with restarts [23], where a random walk is directed back towards its initiator with a fixed *restart probability*. A random walk with restarts represents better the inherent trust in a network, since each node trusts itself more than the other nodes and its trust towards the other nodes decays with the increase of their distance [14, 17]. Each node in the network performs its own random walks and requests parts of the interaction subgraphs of the contacted nodes.

The propagation of the interactions subgraphs when using random walks can be implemented with two different strategies, *push* (information dissemination) or *pull* (information collection). In the push strategy, a node sends (pushes) its messages towards other nodes in the network while in a pull strategy, it probes another node for messages that it has not received yet, and then it *fetches* (pulls) the corresponding messages. We perform a *pull-based* strategy since it results in significantly smaller overhead and ensures the delivery of message in sparsely connected areas of the network [12]. From a security perspective, a node collecting information requests messages from selected nodes following the flow of trust in the interaction graph and so, it has more control on the received information.

Unlike previously proposed methods, in order to incorporate trust we use the interaction subgraph available at each node, as regular and successful interactions between nodes are strong indicators of trust relationships and similarity among users [14], [17]. In many proposed systems such as SybilRank [7] and SybilInfer [9], random walks enhance their resilience against sybil attacks by using a social network to incorporate trust. However, in many systems such as P2P networks no social network is available. Furthermore, in social networks many social connections between nodes are superficial or of very low strength and thus, they do not indicate trust [26]. As a result, EscapeLimit is not only useful in systems without a social network available, but it is more effective against sybil attacks as well.

During the collection of information, a node contacting another node requests the part of the latter's interaction subgraph containing only its direct interactions. Collecting only direct information from other nodes enhances security, scalability, and the importance of the collected information since it allows control on the received information and decreases the redundancy of information. Figure 1 presents an illustration of our method.

Network model and definitions. The interaction graph of a network is a weighted directed graph of interactions $G = (V, E)$ whose vertices V correspond to the nodes in the network, and whose edges E correspond to the interactions among nodes. Its adjacency matrix is denoted by $A = \{\alpha_{ij}\}$. A weighted edge $e_{ij} \in E$ connecting $i, j \in V$ in the direction $i \to j$ has a weight w_{ij} that represents the strength of an interaction, for instance, the amount of data transferred across the edges in a P2P network or the number of interactions in Facebook. We denote by r the vector containing the (real) reputations of nodes. Depending on the application, the vector r can be computed by various computations. We use a very simple computation of reputation so that we can investigate the dissemination of nodes. The reputation of a node j is the ratio of the resources it contributes to the network and the resources it consumes, and so, $r(j) = \sum_{k \in N_j} w_{jk} / \sum_{k \in N_j} w_{kj}$ where N_j denotes the set of neighbours of node j in G. Each node i in the network locally stores its interaction subgraph $G_i = (V_i, E_i)$ with $V_i \subseteq V$ and $E_i \subseteq E$.

A random walk on G is defined by its transition matrix $P = \{p_{ij}\}$, and its stationary distribution π is given by the equation $\pi = \pi P$. Its *mixing time* indicates the time (in walk steps) needed for any initial distribution π_0 to approach the stationary distribution π. To measure the mixing time of our graphs we compute the total variation distance between the two distributions $1/2||\pi - \pi_0 P^t||_1$ over consecutive walk steps t, as it is described in [17]. A low mixing time implies that the initial distribution converges to the stationary distribution in $O(\log V)$ steps.

Types of random walks. We use random walks with restart probability α. Then, the transition matrix becomes $P' = (1 - \alpha)P + \alpha\mathbf{1}$, where $\mathbf{1}$ is the matrix with all its entries equal to 0 except for the elements of the column corresponding to the initiator, which are equal to 1.

In *simple RW* (RW) with restarts, the next step of the walk is chosen uniformly at random among the neighbors of the currently visited node. The transition probability at each step is determined by the adjacency matrix $p_{ij} = \alpha_{ij} / \sum_j \alpha_{ij}$. We use three additional biased random walks.

First, we consider a random walk biased towards the strength of interactions assuming that the strength of an interaction reflects both trust and similarity between adjacent nodes. We call this walk *weighted RW* (wRW) where the bias towards node j from node i is denoted by w_{ij} and $p_{ij} = w_{ij} / \sum_j w_{ij}$.

Table 1: The diameter, the average path length (L) and the clustering coefficient (cc) of real-world graphs.

Graph	# Nodes	# Edges	Diameter	L	cc
Power-law	$1,000$	$5,725$	5	2.92	0.067
Bartercast	$1,000$	$4,723$	8	2.64	0.0065
Facebook	$1,000$	$11,596$	9	3.38	0.13

Then, we define random walks biased towards the nodes with the lowest activity level, namely the smallest degree. In RW, high-degree nodes are visited with a higher probability since more paths lead to them. On the contrary, low-activity nodes are rarely visited, and biasing a RW towards them may help with a faster spread of their information. Furthermore, it balances the communication overhead among the nodes in the network. This random walk results in a uniform visiting probability of nodes, and corresponds to the *Metropolis-Hastings Random Walk* (MHRW) [15] for uniform selection of nodes. According to MHRW, the probability of visiting node j from node i when $i \neq j$ is defined as $p_{ij} = (1/d_i)\min(1, (d_i/d_j))$ with d_i representing the degree of node i and $p_{ii} = 1 - \sum_j p_{ij}$.

Finally, we consider random walks biased towards the nodes with the highest activity level, namely the highest degree. Intuitively, highly active nodes are trustworthy, have fresh information, and interact with the other nodes with a higher probability. This type of random walk corresponds to *Maximal Entropy Random Walk* (MERW) [5] and has been introduced and studied in [4]. The probability of visiting node j from node i is equal to $p_{ij} = (\alpha_{ij} u[j])/(\lambda u[i])$, where u is the principal eigenvector of A, $u[i]$ is the i-th entry of u, and λ is the corresponding eigenvalue. This RW requires global information, but recently Sinatra et al. [22] showed that MERW can be accurately approximated by a RW biased towards the degree of nodes in networks without degree correlations. MERW results in fast diffusion of information since it uses the highly connected nodes more often.

4. EXPERIMENT METHODOLOGY

We evaluate EscapeLimit with the different types of random walks in terms of their resilience to attacks, their scalability, and its ability to provide relevant information. Particularly, we integrate EscapeLimit into Tribler, an open-source P2P BitTorrent-based client and we run 1000 clients on a computer cluster. Each client emulates the interaction patterns deriving from traces of synthetic and real-world datasets with different connectivity properties and construction patterns. Simultaneously, each client collects information about the interactions of other nodes using EscapeLimit. In this section, we describe our datasets, the experiment setup, and the model to create the sybil attacks.

Datasets. In order to perform our emulations, we use datasets from synthetic power-law graphs and graphs derived from Bartercast and Facebook networks. In the real world graphs, the creation of edges is defined by timestamps available in the corresponding datasets which are expressed in actual time. In the synthetic graphs, we divide time into time steps during which new edges are added, since no notion of actual time exists.

Power-law graphs are characterized by their degree distribution following a power law. We create a growing directed power-law graph based on the Barabasi-Albert model [3]. We start with a small connected seeding graph, and at each time step we add a new node with 3 edges whose end points are adjacent to already existing nodes with probabilities proportional to their degrees. After having

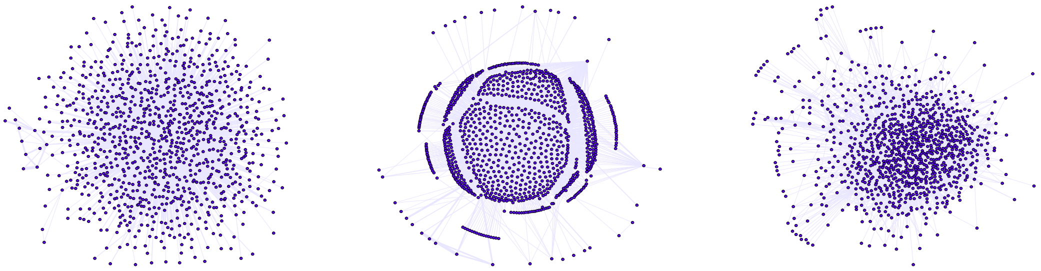


Figure 2: The power-law (left), Bartercast (middle), and Facebook (right) graphs of 1,000 nodes.

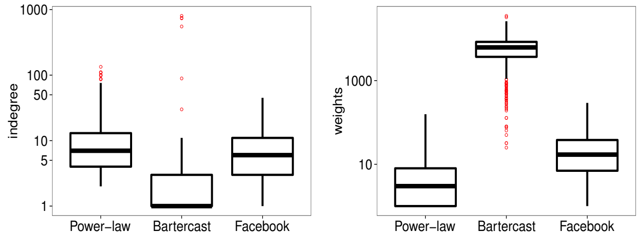


Figure 3: The distributions of nodes degrees and edge weights in the power-law, Bartercast, and Facebook graphs

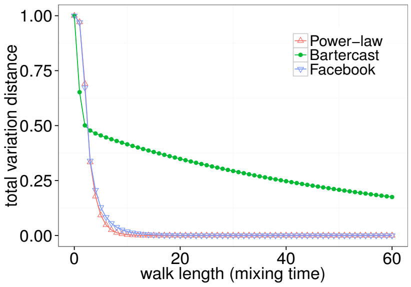


Figure 4: The mixing time of the power-law, Bartercast, and Facebook graphs

created a network with 1000 nodes, we continue adding 3 directed edges at each time step, adjacent to existing nodes again with probabilities proportional to their degrees. We allow the occurrence of multiple edges between a pair of nodes and we consider the number of occurrences of an edge as the weight of that edge.

The **Bartercast graph** is derived from the distributed reputation mechanism called Bartercast [11] of the BitTorrent-based client Tribler [20]. This dataset contains information about 29,716 nodes and their interactions [14]. The weights of the edges represent the amount of data (in KB) transferred between two users. In order to interpret interactions as trust in Bartercast, we reverse the direction of links since when a user downloads from another user, the corresponding trust flows from the former towards the latter.

The **Facebook graph** is derived from the Facebook network in New Orleans with 63,732 users and their interactions [25]. The weights of edges represent the numbers of interactions between the corresponding users.

The main difference between the Bartercast and the Facebook graphs, besides their structural properties, is that the former is derived from a deployed distributed system while the latter is derived from a centralized social network. Due to resource limitations of the computer cluster, we conduct our emulations using a strongly connected component of 1000 nodes. We constructed those strongly connected components from the initial graphs by starting from the node with the highest degree a Breadth First Search (BFS) modified so that it traverses only bidirectional edges. The characteristics of the corresponding subgraphs are presented in Table 1. All the graphs are small-world characterized by small average path length and small diameter. The Facebook graph forms a tightly connected community. In Figure 2, we illustrate the selected strongly connected components (small average path length and high clustering coefficient). On the contrary, the Bartercast graph consists of a few highly connected nodes and many loosely connected nodes (small average path length and small clustering coefficient), since it has a high population turnover. In Figure 3, we show their indegree distribution and the distribution of their weights. We observe that the power-law and Bartercast graphs have a few outlier nodes with high indegree. In this paper, we use Tukey boxplots where the bottom and top of the box depict the first and third quartiles of the distribution, the band inside is the median. The outliers are identified using the interquartile range (IQR), defined as the difference between the third and first quartiles. Outliers fall below 1.5 IQR from the first quartile, and above 1.5 IQR from the third quartile. The whiskers indicate the range of the distribution without the outliers.

To better understand the community structure of the graphs and interpret the evaluation results of the random walks, we estimate their mixing time. The mixing time of our strongly connected graphs is defined since all the nodes will be visited by a random walk. In Figure 4 we present the total variation distance versus the mixing time (walk length) averaged over 1000 initial distributions of a random walk. Facebook and power-law graphs are fast-mixing graphs with tightly connected nodes. On the other hand, Bartercast is slow-mixing because a few highly connected nodes keep the nodes connected by forming clusters around them, as we see in Figure 2. Note that the clustering coefficient of a graph indicates its clustering on a local level (the fraction of closed triangles among its nodes), while its mixing time indicates its clustering into large communities.

The restart probability. During a random walk, the value of the restart probability α determines its expected length l and as a result, its resilience against sybil attacks and its ability to collect relevant information fast. Even though a large value of α allows the discovery of new nodes at a large depth in the network, it draws the walk away from trusted and relevant nodes. The appropriate value of α depends on the characteristics of the graph. In our graphs, we observe that the vast majority of nodes interact with other nodes that are only a few hops away. In Figure 5, we present the probability of interaction between two nodes as a function of their distance just before they interact. Our graphs exhibit a high locality

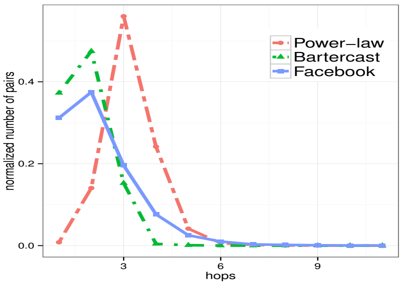


Figure 5: The distance (in hops) between two nodes just before they interact

of interaction, which implies that a node does not need to perform long walks in order to acquire relevant information. Particularly in real-world graphs, we observe that more than 90% of pairs of interacting nodes have a distance of at most 3 hops just before they interact. Therefore, for these graphs we use random walks with an expected length l of 3 hops. In synthetic graphs, we use random walks of an expected length 4 hops, since for power-law graphs the majority of pairs of interacting nodes have a distance of at most 4 hops just before they interact.The restart parameter is computed as a function of the desired expected length as $\alpha = 1/(l+1)$ [2]. So, for real world graphs we use $\alpha = 0.25$ and for synthetic graphs, $\alpha = 0.2$.

Experiment setup We integrated the proposed random walker in Tribler and we can evaluate it on a computer cluster, the DAS-4 supercomputer [1] available at Delft University of Technology. Particularly, we run 1000 clients distributed evenly on 20 nodes of DAS-4. In Tribler, the dissemination of information is based on epidemics [11]. We modified only its dissemination component integrating the proposed random walker.

Each Tribler client has its own local database keeping its own locally stored history of interactions and it interacts with other clients following the interactions in the previously described datasets. At the same time it performs its own random walks towards other nodes. We divide our datasets into two parts: a small part used for initialization and the main part used for emulation. In Facebook, this initialization part consists of the interaction occurred during the first week in the corresponding dataset, in Bartercast during the first day, and in power-law graphs during the first 1000 steps.

After the end of the initialization process, nodes emulate the interactions, compute the reputations of nodes and walk towards each other collecting information. The time of emulated interactions has been mapped to the duration of our emulation. A node emulates an interaction with another node, by creating a record with the details of this interaction and it sends it to the other interacting node. Then, both nodes store this record in their database and they include it in the history of interactions they distribute. All nodes perform walk steps with the same period about every 30 secs so that all nodes perform a similar number of random walks during the experiment. Each experiment lasts 2 hours.

Sybil Attack Model

Most of the random walk-based methods proposed in literature against sybil attacks [7], [9] assume fast mixing graphs, since they have tight trust relationships between the nodes. On the contrary, in a slow-mixing graph multiple communities exist and so, sybil nodes might be incorrectly recognised as honest. We use both fast-mixing graphs such as Facebook where honest nodes form one well connected community, and slow-mixing graphs such as Bartercast, where more than one communities are present.

In our experiments, we take as the honest region our initial datasets and we create a power-law graph of 100 nodes as a sybil region. Then, we randomly chose some sybil nodes and some victim nodes from and connect them through the corresponding attack edges. The evaluation metric is the escape probability which does not depend on the number of Sybil nodes nor on the topological characteristics of the sybil region [28] but on the number of attack edges over the number of honest nodes. To each attack edge, we assign probabilistically a weight in the range of the weights of edges among honest nodes so that, attack edges with small weights are more common, since it is more costly for an attacker to create an attack edge with a large weight than an attack edge of a low weight. For the timestamp of the attack edges, we assume that they are uniformly distributed over time. The nodes in the sybil region can claim any values for the properties of their edges.

5. EVALUATION

In this section, we present the results of the evaluation of EscapeLimit with the different biased random walks in terms of its resilience to attacks, its scalability and its ability to acquire relevant information fast. We use a set of metrics associated with our requirements. All the presented results are the average of 10 experiments.

Resilience against Sybil Attacks A random walker escaping into the sybil area will be trapped there till it restarts. Therefore, we evaluate the fraction of walks escaping into the sybil region when starting at any node in the honest region, which is called the *escape probability* [28]. Lower values of escape probability indicate higher resilience against sybil attacks. This probability depends on the number of attack edges, since in order to escape to the sybil region, the random walk has to traverse an attack edge. We note that the escape probability does not depend on the number of Sybil nodes nor on the topological characteristics of the sybil region [28].

In Figure 6 we present the escape probability of the different random walks depending on the average number of attack edges per honest node for the different datasets. Independently of the characteristics of the graphs, all biased RWs exhibit a smaller escape probability into the sybil region than simple RW, indicating that the strength of interactions, and the activity level of nodes are accurate indicators of trust. The fast-mixing graphs, power-law and Facebook, have smaller escape probability when using RWs and MERW than Bartercast. Being fast-mixing, those graphs have a tightly connected honest region and a random walk does not escape into the sybil region with high probability. In power-law graphs, wRW exhibits the highest escape probability in comparison with the other graphs since in power-law graphs the range of the weights is smaller.

Nevertheless, independently of the characteristics of the graph, wRW and MHRW exhibit the lowest escape probability, which increases very slowly with the increase of the number of attack edges per honest node. Highly weighted attack edges are more rare due to high engineering cost required for their creation and as a result, wRW traverses with low probability the attack edges. Hence, it exhibits low escape probability for all the examined graphs. Furthermore, MHRW tends to visit low degree nodes at the periphery of the network and so, it rarely escapes into the sybil region. On the contrary, MERW visits more often high degree nodes, and as a result it has a similar escape probability to RW.

Scalability In Section 2, we define scalability in terms of computational and communication overhead. EscapeLimit has low computational cost since at each step only the transition probabilities are computed and each node maintains connectivity information only about its neighbors. Therefore, we have to evaluate the communication overhead imposed by EscapeLimit at each node. In

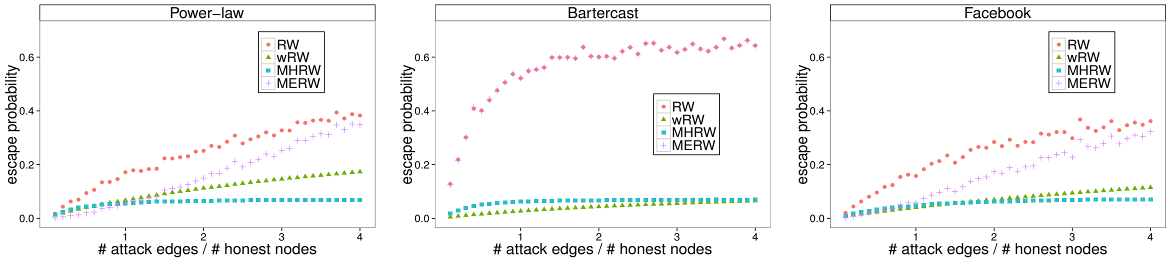


Figure 6: Resilience against sybil attacks: the escape probability of the different random walks in the power-law, Bartercast, and Facebook graphs.

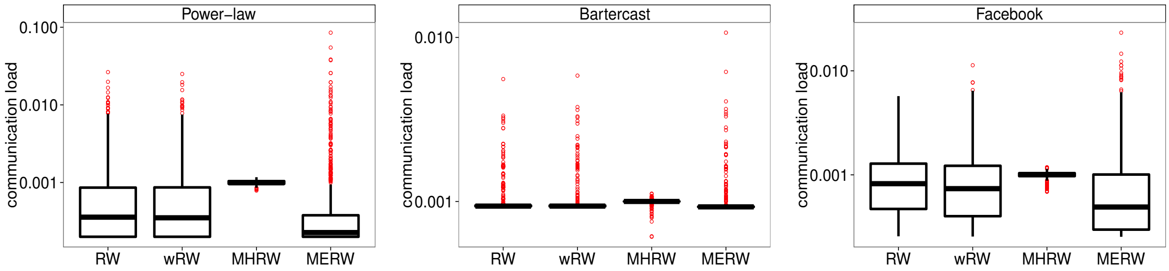


Figure 7: Scalability: the distribution of the communication load of nodes for the different random walks in the power-law, Bartercast, and Facebook graphs.

Table 2: Scalability: the correlation between the indegree and the distribution of the communication load of the walks

	power-law	Bartercast	Facebook
RW	0.939	0.88	0.77
bRW	0.94	0.75	0.78
MHRW	-0.82	-0.6	-0.51
MERW	0.95	0.89	0.817

EscapeLimit, each node sends one introduction request to another node in the system. Thus, the communication overhead at each node during a random walk depends on the visit ratio of that node at each step, namely the fraction of introduction requests it receives from other nodes at each walk step. To evaluate this overhead at each node, we define the *distribution of communication load* in terms of the average visit ratios of the nodes per step. This ratio is proportional to indegree of a node, as is presented in Table 2. Due to the highly skewed indegree distributions of real world networks, a few highly connected nodes receive the majority of introduction requests. As a results, those highly connected nodes may be overloaded.

In Figure 7, we present the distribution of communication load of nodes in the power-law, Bartercast and Facebook graphs when we integrate different biases in the random-walk based collection method. Since we ran the experiment for 1000 nodes, the optimal communication load value for a node is 0.001, meaning that it has been visited exactly once during a walk step. In this experiment, we do not include sybil nodes, since sybils will have no impact on the communication load of honest nodes.

For all the examined graphs, MHRW distributes the load evenly to almost all the nodes independently of the indegree distribution of the corresponding graph. In a random walk, the high degree nodes are visited more often, but this property is counterbalanced by the bias of MHRW towards the low-degree nodes and so, MHRW achieves an almost uniform load distribution. Conversely, MERW intensifies the selection of the highly connected nodes and as a result, in all graphs it exhibits the most skewed distribution of communication load. Particularly in the power-law graph, a few highly connected nodes have a communication load value close to 0.100 implying that those nodes receive 100 introduction requests during a walk step. Those nodes cannot reply to all those request and as a result, MERW is not scalable in that graph. Furthermore, wRW has a load distribution similar to RW.

Relevance of Information. In order to capture different characteristics of the relevance of the acquired information at each step of the walk, we use two metrics. The first metric is the relative size of the interaction subgraph G_i at node i with respect to the size of G and it is defined as $RE(G_i, G) = |E_i|/|E|$.

According to the second metric, the *ranking similarity* (RS), the interaction subgraph G_i at node i is similar to G if it produces similar reputation rankings of the most highly reputed nodes. Ranking similarity is a modification of Spearman coefficient and the vertex ranking metric proposed in [18], so that it can be applied to lists of different lengths and takes into account the reputation of each node. If we denote by r (and r_i) the reputation vector produced at G (and G_i), the ranking similarity is defined as:

$$RS(G, G_i) = 1 - \frac{\sum_{u \in V_i} r(u)(\sigma(r(u)) - \sigma(r_i(u)))^2}{D}$$

where $\sigma(r(u))$ (and $\sigma(r_i(u))$) is the rank of the reputation of node u in $r(u)$ (and $r_i(u)$) when only vertices in V_i are considered and r (and r_i) is ordered in decreasing order. The normalization factor D is equal to $\sum_{u \in V_i} r(u)(\sigma(r(u)) - \sigma(r_w(u)))^2$ where r_w is the sequence containing the nodes in V_i in reverse order from r. The ranking similarity between the two graphs is equal to 1 if their reputation vectors produce exactly the same ranking. On the contrary, a ranking similarity equal to 0 indicates that the ranking derived from r_i is the reverse of the ranking deriving by r.

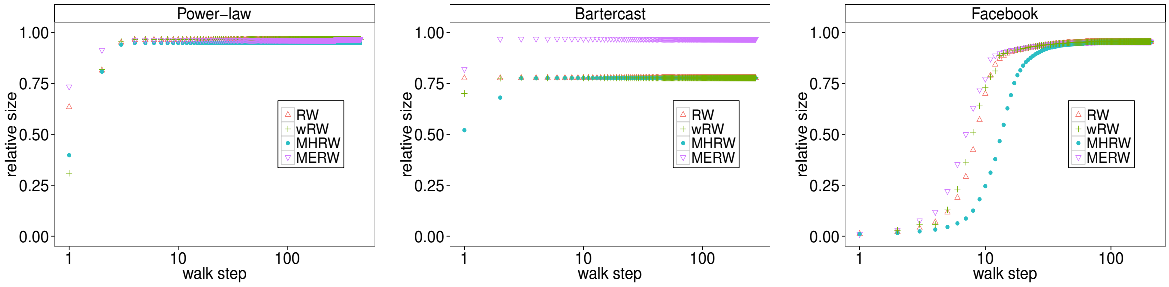


Figure 8: Relevance: the relative size between the interaction subgraphs of nodes and the interaction graphs of the power-law, Bartercast, and Facebook networks over consecutive steps of the different random walks.

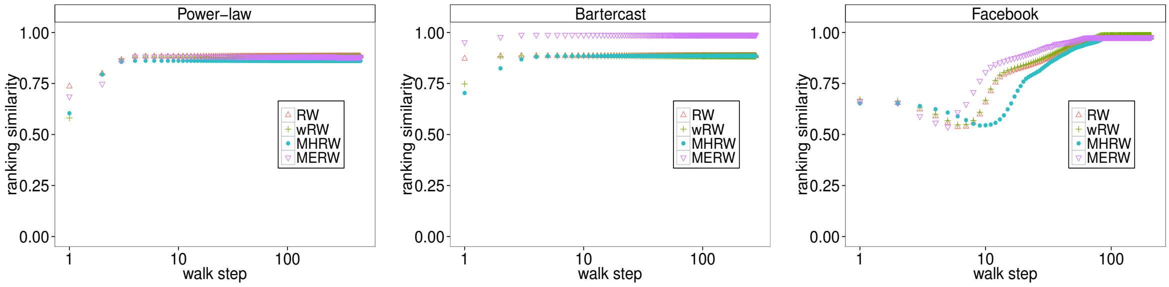


Figure 9: Relevance: the average ranking similarity between the interaction subgraphs of nodes and the interaction graphs of the power-law, Bartercast, and Facebook networks over consecutive step of the different random walks.

In Figure 8, we present the relative size between the interaction graph and the interaction subgraphs over consecutive steps of the different random walks for all the datasets. For all the different graphs, MERW achieves the largest relative size faster. According to MERW, each nodes visits with higher probability the highest degree nodes. These nodes have the majority of information since they perform the majority of interactions. On the contrary, MHRW visits mostly low degree nodes which perform very few interactions and so, results in very small relative size.

In fast-mixing graphs after some steps all the random walks achieve almost perfect relative size. On the contrary, in Bartercast only MERW achieves perfect relative size. As we can see from its indegree distribution in Figure 3, Bartercast has 3 hubs and most nodes are not connected with each other but only with the hubs. As a result, all the RWs but MERW do not manage to visit all the hubs and visit other nodes in the periphery since their expected length is small. In graphs with skewed degree distribution, such as power-law graph and Bartercast, all the different random walks achieve most of the information after a small number of walks steps since those hubs are visited with higher probability. Piatek et al [19] based their dissemination scheme on a similar observation. However, in graphs with a more symmetric degree distribution, such as Facebook, the collection of information is much slower. Particularly, MHRW on Facebook needs more than 40 steps to achieve most of the information.

In Figure 9, we present the ranking similarity between the interaction graph and the interaction subgraphs over consecutive steps of the different random walks for all the datasets. In accordance with the results for the relative size, MERW achieves faster a high ranking similarity while MHRW is the slowest. In power-law and Bartercast graphs, ranking similarity follows the patterns of the relative size due to the skewed degree distribution. In these graphs, the hubs not only have a high degree but a high reputation as well. In Facebook, there is an instability in the ranking similarity in the first steps of random walks. From Figure 8, we observe that during those steps nodes collect about 75% of the information. Afterwards, the ranking similarity increases.

Table 3: The ability of random walks to satisfy the requirements of a collection mechanism applicable to distributed reputation systems

Method	Resilience to Attacks	Scalability	Relevance of information
RW	fair	good	good
wRW	very good	good	good
MHRW	very good	very good	poor
MERW	good	poor	very good

Discussion Our evaluation indicates that properly biased random walks satisfy the requirements of an applicable collection method since they achieve resilience to sybil attacks, good load balancing and provides relevant information. The bias of a random walk determines the extent to which each requirement is satisfied. In Table 3, we summarize the experimental results for all the different random walks.

In fast-mixing networks, simple RW achieves good resilience against attacks while in slow-mixing networks it escapes with high probability into the sybil region even if the number of attack edges per honest node is small. Furthermore, it distributes the communication load across the nodes of a network with a preference towards nodes with high degree nodes. RW is suitable only for fast-mixing networks. Adding the appropriate biases by using richer information about the interactions of nodes further improves its quality.

Through our experimental results, we have shown that wRW achieves robustness against sybil attacks and collects relevant information independently of the characteristics of the graph. Therefore, the strength of interactions represent accurately both the trust and the similarity of nodes. Furthermore, the communication load at each node when using wRW is close to RW. This type of walk is suitable for all networks and particularly for networks where the strength of edges has a skewed distribution.

Next, we studied the bias towards the nodes with low activity, namely the nodes with low degree. In MHRW, the bias is towards the low degree nodes. This walk achieves high resilience to attacks since sharing interactions with low degree nodes is a stronger indication of trust than interactions with high degree nodes. Furthermore, MHRW has excellent load balancing properties independently of the degree distribution of the network. However, a node visiting the low degree nodes cannot obtain fast relevant information. This type of walk can be used when for a network the main concerns are security and load balancing.

Besides the strength of interactions, the activity level of a node as represented by its degree is another indicator of trust and similarity among nodes. In MERW, the resilience against sybil attacks is similar to that of RW indicating that sharing interactions with high degree nodes is not a strong indication of trust. Furthermore, the bias towards the high degree nodes results in overloading those nodes. Nevertheless, MERW achieves fast relevant information since it visits more often the hubs even if the network is slow-mixing. Through the hubs, MERW manages to visit the different communities in a slow-mixing network. This type of walk is suitable for slow-mixing graphs when the fast acquisition of relevant information is relevant.

6. CONCLUSION

In this paper, we propose a method to collect information in distributed reputation systems based on random walks. EscapeLimit collects only relevant and trusted information as well as reduces the escape probability of an honest node to the Sybil area. EscapeLimit uses the observation that user interactions require real effort and so, they reflect trust and similarity among users. We guide random walks in EscapeLimit with three different trust-driven user properties and through experimental evaluation we show their effectiveness in terms of resilience to attacks, scalability and the ability to provide each user with relevant information. Our evaluation suggests that the strength of user interactions guides random walks efficiently in almost any type of network. As future work, we would like to investigate the performance of EscapeLimit in networks with high population turnover.

7. ACKNOWLEDGEMENT

The authors would like to thank Boudewijn Schoon for his help on the integration of random walks into Bartercast and on DAS-4 experiments.

8. REFERENCES

[1] The distributed asci supercomputer 4. http://www.cs.vu.nl/das4/, 2014.

[2] C. Ballester and M. Vorsatz. Random walk based segregation measures. *Review of Economics and Statistics*, 2011.

[3] A. L. Barabási and R. Albert. Emergence of Scaling in Random Networks. *Science*, pages 509–512, 1999.

[4] Z. Burda, J. Duda, J. M. Luck, and B. Waclaw. Localization of the Maximal Entropy Random Walk. *Physical Review Letters*, 102(16):160602–4, 2009.

[5] Z. Burda, J. Duda, J. M. Luck, and B. Waclaw. The various facets of random walk entropy. *Acta Phys. Polon. B*, 2010.

[6] G. C., M. M., and S. A. Random walks in peer-to-peer networks: algorithms and evaluation. *Perform. Eval.*, 2006.

[7] Q. Cao, M. Sirivianos, X. Yang, and T. Pregueiro. Aiding the detection of fake accounts in large scale social online services. In *NSDI*, 2012.

[8] R. Chakravorty, S. Agarwal, and S. Banerjee. Mob: A mobile bazaar for wide-area wireless services. In *ACM MobiCom*, 2005.

[9] G. Danezis and P. Mittal. SybilInfer: Detecting Sybil Nodes using Social Networks. In *NDSS*, 2009.

[10] R. Delaviz, J. Pouwelse, and D. Epema. Targeted and scalable information dissemination in a distributed reputation mechanism. In *Proceedings of the seventh ACM Workshop on Scalable Trusted Computing (ACM STC)*, 2012.

[11] R. Delaviz, N. Zeilemaker, J. Pouwelse, and D. Epema. A network science perspective of a distributed reputation mechanism. *IFIP Networking*, 2013.

[12] P. Felber, A.-M. Kermarrec, L. Leonini, É. Rivière, and S. Voulagris. PULP: an Adaptive Gossip-Based Dissemination Protocol for Multi-Source Message Streams. *Peer-to-Peer Networking and Applications, Springer*, 2011.

[13] M. Feldman, K. Lai, I. Stoica, and J. Chuang. Robust incentive techniques for peer-to-peer networks. In *ACM EC*, 2004.

[14] D. Gkorou, T. Vinkó, J. Pouwelse, and D. Epema. Leveraging node properties in random walks for robust reputations in decentralized networks. In *IEEE P2P Computing*, 2013.

[15] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[16] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *WWW*, 2003.

[17] A. Mohaisen, N. Hopper, and Y. Kim. Keep your friends close: Incorporating trust into social network-based sybil defenses. In *INFOCOM*, 2011.

[18] P. Papadimitriou, A. Dasdan, and H. Garcia-Molina. Web graph similarity for anomaly detection. *J. Internet Services and Applications*, 2010.

[19] M. Piatek, T. Isdal, A. Krishnamurthy, and T. Anderson. One hop reputations for peer to peer file sharing workloads. In *NSDI'08*, 2008.

[20] J. A. Pouwelse, P. Garbacki, J. Wang, A. Bakker, J. Yang, A. Iosup, D. H. J. Epema, M. Reinders, M. R. van Steen, and H. J. Sips. Tribler: a social-based peer-to-peer system. *Concurr. Comput.: Pract. Exper.*, 2008.

[21] D. Quercia and S. Hailes. Sybil attacks against mobile users: friends and foes to the rescue. In *INFOCOM*, 2010.

[22] R. Sinatra, J. Gómez-Gardeñes, R. Lambiotte, V. Nicosia, and V. Latora. Maximal-entropy random walks in complex networks with limited information. In *Phys. Rev. E*, 2011.

[23] H. Tong, C. Faloutsos, and J.-Y. Pan. Fast random walk with restart and its applications. In *ICDM*, 2006.

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

[25] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi. On the evolution of user interaction in facebook. In *ACM WOSN*, 2009.

[26] C. Wilson, B. Boe, A. Sala, K. P. Puttaswamy, and B. Y. Zhao. User interactions in social networks and their implications. In *Proceedings of the 4th ACM European Conference on Computer Systems*, EuroSys, 2009.

[27] Z. Yang, C. Wilson, X. Wang, T. Gao, B. Y. Zhao, and Y. Dai. Uncovering social network sybils in the wild. *IMC*, 2011.

[28] 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.

[29] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. Sybilguard: defending against sybil attacks via social networks. In *SIGCOMM*, 2006.