Decentralized and Sybil-resistant Pseudonym Registration using Social Graphs

Sebastian Friebe, Martin Florian
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
{friebe,florian}@kit.edu

Ingmar Baumgart
FZI Research Center for Information Technology
Karlsruhe, Germany
baumgart@fzi.de

***Abstract*—A registration of identities is necessary in a wide array of systems, from online forums to smart environments. While pseudonyms are, in most cases, sufficient, mechanisms must be put in place to prevent malicious adversaries from registering great numbers of sybil identities. Preventing such sybil attacks becomes an especially significant challenge when the existence of a trusted party cannot be assumed. Several countermeasures against sybil attacks on decentralized systems have been proposed that are based on leveraging information from the social graph between participating users. While promising, existing solutions typically require knowledge of the complete social graph, which is a privacy issue, or are tailored towards specific applications like distributed hash tables. In this paper, we propose an approach for registering general-purpose pseudonyms in a completely decentralized manner while keeping social relationships private. Joining users collect confirmations from a fraction of already registered users while being aware only of their own neighbors in the social graph. Using the presented *SybilHedge* algorithm, sybil attackers are limited in the number of confirmations they can collect. We present an evaluation of the algorithm and discuss its practical application.**

I. Introduction

During a *sybil attack* [1] an adversary creates a large number of virtual identities to take over a system. Consider, as an example, an online forum with a voting system. By registering multiple pseudonyms, or *sybil identities*, an adversary can gain the majority of votes, allowing him to control the outcome of any vote. Sybil attacks are similarly threatening to crowdsensing systems, peer-to-peer networks, and cooperative vehicular route planning services. For counteracting sybil attacks in such systems, a carefully designed registration mechanism is necessary to prevent adversaries from registering unlimited numbers of pseudonyms. Such a mechanism is also required as a base for decentralized sybil-free pseudonym change [2] and anonymous credentials [3].

Mitigating sybil attacks is especially challenging in decentralized systems where no single party can be trusted to decide about the non-sybilness of pseudonyms. Multiple existing works deal with this challenge. The more promising of them leverage trust relationships embedded in the social graph between users. Social graph-based sybil defense mechanisms typically require the connections of the social graph to be completely visible, which implies publicly visible friendship-relations between individual users. This is undesirable from a privacy standpoint. Amongst other things, it can enable the identification of participating users [4]. Another drawback of many existing approaches is that they do not result in a permanent proof of non-sybilness for users. When a user wants to test whether another user is a sybil, he has to execute the sybil verification procedure again, resulting in increased overhead.

In this paper, we propose a novel approach towards the decentralized and yet sybil-resistant registration of virtual identities, i.e., pseudonyms. New pseudonyms can be registered only if sufficient holders of already registered pseudonyms approve. We introduce *SybilHedge*, a novel algorithm for the decentralized validation of pseudonym registration requests. Sybil resistance is achieved by using the trust relations between users in a social graph. To preserve the privacy of users, SybilHedge works without disclosing the relationships to other participants. The algorithm does not require the manual participation of users or the disclosure of personal information. After a pseudonym has been registered, a reusable proof can be constructed for convincing communication partners of its non-sybilness. Our specific contributions are the following:

- A novel decentralized registration approach based on confirmations given by nodes connected in a social graph.
- SybilHedge, a specific algorithm for collecting confirmations while protecting against sybil attacks.
- An analytical and simulative evaluation of SybilHedge.
- A discussion of practicability and approaches towards a complete registration system.

II. Related Work

Several decentralized sybil defense approaches have been evaluated in the past. Most approaches are based on the assumption that adversaries cannot control a larger amount of a specific resource than honest users. Often used resources include communication addresses, computation power (called proof-of-work), or human attention (for example by solving CAPTCHAs [5]). While easy to implement, these approaches can be circumvented by a determined adversary [1]. A promising alternative approach is based on the usage of social graphs, which describe the relationships between humans. Some of the existing approaches are described in the following.

The most important property of these graphs is the existence of a *community structure*. As previous work has shown most users in social graphs are tightly interconnected and are forming a large community [6]. When an adversary starts to add sybil nodes to the graph a second community develops. The small number of edges between sybil nodes and non-sybil nodes form a *sparse cut* which is exploited by most sybil defense algorithms [7].

One way to employ the sparse cut is by using *random walks*. These walks are send between nodes, locally selecting the next node at random out of the neighbors. With a high probability the walks stay in the current community and do not cross the sparse cut to the sybil nodes [7]. This is used in Whanau [8], a sybil-resistant decentralized hash table, where the last node of the random walk on the social graph is selected as neighbor in an overlay graph. A similar approach is used by Pisces [9], a decentralized anonymity network. Random walks are used to select the next node to send a message to. Both algorithms achieve their respective goals but are not applicable in connection with the desired privacy of the friendship edges.

Two sybil defense algorithms are SybilGuard [10] and its successor SybilLimit [11]. For a node to be registered as non-sybil a number of random walks have to intersect, which only rarely happens when the nodes are separated by a sparse cut. A problem with these approaches is the publicly visible social graph. The visibility cannot be avoided since it is a requirement for checking on the graph intersections.

Gatekeeper [12] uses another approach to thwart sybil attacks without using random walks. A node *A* in Gatekeeper distributes tickets in the graph. A second node *B* can later on ask a random selection of nodes about tickets from node *A*, accepting it as non-sybil when its tickets reached enough nodes. Sybil nodes are obstructed by the sparse cut, preventing sufficient distribution of tickets. Gatekeeper can offer good guaranties with regard to the number of permitted sybil identities, but cannot deal with frequent joins of further nodes.

X-Vine [13] provides a decentralized hash table and uses *trails*, permanently stored random walks, to block access by sybil nodes. Each node is required to create a number of trails to become accepted. At the same time, only a limited number of trails per edge in the graph is permitted. When the edges allowing access to sybil nodes reach their capacity, further trails are blocked. While X-Vine is not directly applicable to the goals of our work an idea similar to permanent trails is used.

III. General Approach and Adversary Model

This section first describes the main assumption of our approach, namely the existence of a trust-based social graph between users and an overlay network based on the social graph. Following we introduce our pseudonym registration approach. Lastly, we describe the abilities and aims of an assumed sybil attacker.

A. Social graph

Our approach uses the trust relationships embedded in a social graph. The used graph should not be a normal friendship graph from popular online social networks as Facebook[1] since research has shown that it is quite easy for an adversary to create friendship-edges to honest users [14]. Instead, the used social graph is assumed to consists only of strong trust edges where the connected users know each other. These edges are always symmetrical and represent mutual trust. As a central property required by our approach, social graphs are assumed to have a community structure.

B. Overlay network

To protect the privacy of the users the graph is stored in a decentralized fashion. More specifically, this is realized by maintaining a darknet-type overlay network between participating users. Only users that trust each other form overlay links, so that links in the overlay correspond to links in the social graph. Consequently, a user's node in the overlay communicates only with nodes of that user's social graph neighbors. The existence of such a social graph-based overlay network is assumed in this work. A similar overlay structure is also used in [13] and studied in, for example, [15].

In the scope of this paper, the term *user* is used to denote a human and a node in the assumed social graph. If no additional clarifications are given, *node* is used as *overlay node* (i.e., an actual device participating in the network on behalf of a user). While the social graph between users is assumed to be unchanging, the social graph-based overlay network between users can grow when further users in the underlying social graph start participating in the overlay.

In a real implementation, overlay links have to be able to deal with churn and delay tolerant message delivery. In the scope of this paper, for simplicity, it is assumed that all nodes are always available. This is, however, no general restriction implied by the proposed techniques, as it mainly affects the latency of request handling.

C. Registering pseudonyms

It is unknown to existing nodes whether new overlay nodes are controlled by honest users or are sybil nodes controlled by an adversary. Achieving such a distinction in a decentralized manner is the main goal of this work. In the proposed approach, nodes, respectively the pseudonyms associated with them, become *registered* once they have collected sufficient *confirmations* from already registered nodes. Once registered, they can prove to other nodes that they have been registered and can be considered an honest, i.e., non-sybil, node. When used with, e.g., a voting system, only registered nodes are permitted to vote. For bootstrapping the algorithm, a group of trustworthy preregistered nodes needs to be defined.

Confirmations are solicited by sending a *request* through the overlay network. In order to realize the process of collecting confirmations in a way that puts sybil attackers at a disadvantage, the structure of the social graph can be leveraged during confirmation collection. Section IV describes a specific solution for collecting confirmations based on random walks. Confirmations can be realized using standard cryptographic signatures. More sophisticated and effective approaches are

[1] *https://www.facebook.com*

possible as well, e.g., aggregatable signatures [16] can be used for reducing the storage size of registration proofs.

Lastly, it is assumed that the list of currently registered identities is stored so that interested parties can securely retrieve it. While not required for the proposed algorithm itself, a validated list of previously registered nodes is required to verify a proof presented by an unknown node. Possibilities for a decentralized realization of this storage are discussed in Section VI.

D. Adversary model

The main goal of the adversary we consider is the realization of a sybil attack, i.e., the registration of a significant number of sybil identities.

In general, the adversary can deviate from the proposed protocol in any way. The decentralized storage of the social graph enables the adversary to add as many sybil nodes to the overlay as he wishes. To get them registered he requires the support of already registered nodes in the graph. This results in a secondary goal for the adversary: Having enough nodes registered so that he can register further nodes by himself without requiring confirmation from non-sybil nodes. If this state is reached, the algorithm is *broken* and can no longer offer any protection against sybil attacks. The adversary's *attack* consists of sending as many requests as possible.

It is assumed that edges in the social graph only exist when two users know and trust each other. This does not exclude an adversary from the graph entirely since he might still be able to trick some honest users into trusting him. As he is required to maintain real human relationships to the other users to make them trust him, only a bounded number of edges to non-sybil nodes can be created. Furthermore, he is not able to create edges between multiple of his sybil nodes and a single non-sybil node.

IV. SybilHedge

This section explains the design of SybilHedge, a specific instantiation of our pseudonym registration approach. The first subsection describes the general design without the presence of an adversary. The second subsection details the proposed sybil defense. Afterwards two parameters of SybilHedge are presented. Possible optimizations are introduced in section VI.

A. General design

The algorithm has to allow non-sybil nodes to create a *proof* that they are honest. To do so, a node can send a *request* randomly through the overlay graph. Already registered nodes will confirm this request and relay it. When enough confirmations are present the node currently processing the request will send a *response* back over the same path.

Figure 1 shows an example of this. Node 11 send a request to node 5. There the request is confirmed and relayed to node 7. When the request contains all required confirmations (13 in this example) it is send back over the same path. To do so, every node stores the input- and output-edge of a processed request.

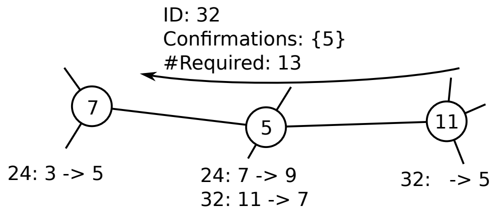

Fig. 1. Relaying a request with local state. Each node stores the ID of the request, the local source, and the next node of the path. Some state from another walk is still stored.

When a node already processed a request and receives it again, it will be send back immediately. In this case the previous node on the path selects another successor until enough confirmations have been collected. When no further successor exists the partially finished request is send back in direction of the origin until a successor can be found.

After the response has been received by the original source node it is stored. Later on, the response can be used to prove that the node has been confirmed as a non-sybil node by already registered nodes.

B. Securing against sybil attacks

The previously presented procedure allows non-sybil nodes to create a proof but it also allows adversaries to create as many proofs as desired. Nodes which receive a request have no possibility to check whether the original sender of the request is a non-sybil node or controlled by the adversary. Additionally, the structure of the overlay graph is kept hidden so it cannot be checked if a lot of requests originate from the same region of the graph. Since only local information about the graph and the requests are known the sybil defense has to work locally, too.

To do so, the edges in the overlay graph are extended with directed *trust values*. These values describe how much the local node trusts the respective neighbor to not be part of an active attack. The trust is relocated along edges when passing messages between nodes. Due to this it can be considered as a kind of currency that is passed between two neighbors in the graph.

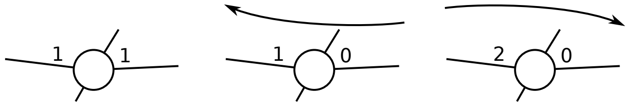

Fig. 2. Local trust values of two edges of a node. The initial state is displayed on the left. When a request is received the trust value of the receiving edge is reduced (middle). When the response is received the trust value of the receiving edge is increased (right).

When a new edge is created the trust value is set to an initial value of t. For how t might be chosen see section IV-C1. How the value is modified, when a message is received from a neighbor, is displayed in Figure 2 and described in the following:

- **Receiving a request** When the trust value on the receiving edge is already 0, the edge is *depleted*. In that case the request is sent back without confirming it. In this way, nodes which are sending too many requests are blocked. If the trust value is greater than 0, it is reduced by 1 and the request is confirmed. In case the request is finished now, a response is send back, otherwise the request is sent to another neighbor.
- **Receiving a response** Instead of reducing the trust value, the trust value is increased by 1 when a response is received. This way the neighbor node is rewarded for sending the response and working on it. This allows the neighbor node to send another request himself.
- **Lost requests** When a request has been sent in the past, but no response has been received for some time, it is assumed that the request has been lost. This might happen due to node failures but also when the request cannot collect enough confirmations to get finished. In this case the previous reduction of the trust value associated with the request is undone to avoid losing trust permanently.

An adversary is heavily restricted by these modifications. As long as his initial trust value is not depleted, he can send requests and obtain proofs for his sybil nodes. Afterwards, he has to receive a request and respond to it, increasing his trust value, before he can register further sybil nodes.

The reductions and increases of the local trust values for one edge result in a relocation of the trust along that edge. This is an important property of the algorithm since it keeps the absolute amount of trust on the edge and in the overlay graph constant and avoids the permanent isolation of nodes due to depleted trust values on both sides of the edges.

C. Parameters

SybilHedge utilizes two parameters which can be adapted: The initial trust value on edges and the amount of required confirmations for a request. Both are detailed in the following.

1) Initial trust value: The initial trust value t restricts the number of requests that can be sent by joining nodes. After this number of request has been sent over an edge, no further requests are allowed until a response has been sent.

With regard to the adversary, a high value results in high numbers of requests he can immediately send and, consequently, in more sybil nodes he can register. At the same time, a high value is advantageous for honest users since their requests have a higher chance of succeeding.

Assuming that nodes join the overlay graph in random positions, requests will traverse the edges approximately equally in both directions. Based on this assumption, small initial trust values should be sufficient to deal with (temporarily) unequal load while still placing the adversary at a disadvantage.

2) Confirmation threshold: Another parameter of the algorithm is the confirmation threshold: the fraction of registered nodes which have to confirm a request. As such, the value has a strong influence on the usability of SybilHedge. When all nodes of an overlay graph with thousands of nodes have to confirm a request, the overhead becomes too large.

At the same time too small fractions favor the adversary. The fraction is required to be higher than the percentage of nodes the adversary can control, otherwise he can register sybil nodes himself.

V. Evaluation

In the following, we evaluate whether sybil attacks are successfully limited by using SybilHedge. Additionally, the usability for honest users is examined.

The evaluation is split into two parts. First, the possibilities of the adversary are analyzed from a theoretical perspective. Afterwards, a simulation-based study is presented and discussed. Effects of churn or permanently unavailable nodes are not considered since they have no impact on the security of the algorithm.

A. Theoretical analysis

The percentage of nodes controlled by the adversary can be analyzed when he initially joins the social graph and how this percentage varies when further nodes join the overlay graph. Furthermore, a possible attack is regarded.

1) Initial percentage of sybil nodes: The maximal initial percentage of sybil nodes controlled by the adversary is based on several factors:

- the number e of edges from the adversary-node to non-sybil nodes of the social graph (assuming all his friends are already registered)
- the initial trust value t assigned to the edges
- the number of already registered nodes r

The first two factors influence the number of nodes the adversary can register, while the last factor determines the percentage of the social graph which is controlled by the adversary. With this, the maximal initial number of nodes N the adversary can register resolves to:

$$N = e * t$$

This is the maximal amount of nodes an adversary can register initially. His edges allow him to send up to N requests into the non-sybil graph. Not all of these requests have to succeed since there might be some edges in the graph that do not allow all requests of the adversary to pass. Assuming that all requests succeed, the percentage of the sybil nodes on the graph resolves to:

$$P = \frac{N}{r} = \frac{e * t}{r}$$

Consequently, the initial percentage of sybil nodes is highly dependent on the number of already registered nodes in the graph, as the number of edges and the initial trust values are bounded.

2) Percentage of sybil nodes over time: When the initial trust on his edges have been spend on requests, the adversary is temporarily blocked from sending any more requests. To continue doing so, the adversary has to increase his trust values again. As described in section IV the trust value is increased when a response is sent by the adversary.

Before a response can be send, a request has to be received. This means that a (probably non-sybil) node in the social graph has to create a request which has to traverse the graph until the adversary is reached.

Simply confirming and relaying the request of another node is not enough to raise the trust value. When the adversary cannot finish the request and send the response himself, he has to send the request to another node which requires trust on the respective edge. Even when he is able to relay the response later on, the trust is only moved from one edge to another without allowing the adversary to send further requests himself.

In order to be able to send further requests, the adversary has to finish a received request himself and return a response. For this the adversary needs to have at least as many registered sybil nodes as there are missing confirmations on the request. Whether this is the case depends on the confirmation threshold, how many nodes have already confirmed the request, and how many sybil nodes the adversary has been able to register until now.

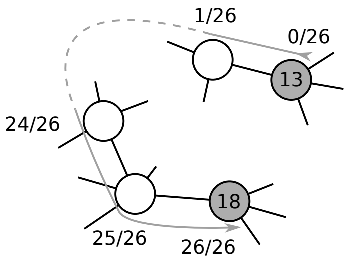

Fig. 3. Possible attack with multiple identities. The request has to be nearly finished by the non-sybil graph so the sybil node 18 can finish it. Afterwards node 18 can send a new response himself.

3) Multiple identities: To increase his chance to capture requests, the adversary could try to create a separate sybil node for each of his friends. This does not increase his chances however, since the number of edges which connect his sybil nodes to the non-sybil graph stays the same.

Splitting his identity by connecting multiple sybil nodes to the graph instead of only using one node allows another attack, displayed in Figure 3. When he sends a request from his only node this node can no longer receive the request over another path to contribute confirmations. When the adversary creates connections over multiple sybil nodes a request might leave his region by one edge and enter it again over another edge. If he is able to finish the request at this time, the response will leave his region, traverse the non-sybil graph, and enter his region again. This way a sybil node gets registered without the adversary spending trust for it. He has to pay trust when sending the request but also gains trust when the response is sent by his second node.

For this attack to work the response has to collect some confirmations in the non-sybil graph and reach another sybil node when the response is nearly finished. Depending on the confirmation threshold and the size of the social graph the probability of this happening becomes very small.

Additionally, the number of tries of this attack is limited. With a high probability the request of the adversary will succeed. When it succeeds but does not reach another of his sybil nodes, he will have to pay the trust, reducing the number of times he can try this attack.

Altogether, the risk through this attack becomes negligible.

4) Conclusion: The initial percentage of sybil nodes is influenced by the delay before attacking, the number of friends, and the initial trust value. Before the adversary can gain further trust, another node has to join the overlay graph and send a request. Because of this, the adversary can only add as many new sybil nodes as non-sybil nodes are joining the graph. Furthermore, the request has to reach the adversary and has to be in a nearly finished state. This results in the adversary registering new nodes even less frequently.

B. *Simulation-based study*

To evaluate the effect of algorithm parameters quantitatively simulations have been used. In this section the setup for and the results of the simulations will be presented.

1) Simulation setup: SybilHedge requires a social graph representing trust relations among users. Since such a graph is not publicly available, an extract of the Facebook wall posts graph [17] has been used. This graph contains edges when two users interacted by posting on each other's profile pages. Being an interaction graph it comes much closer to a trust graph than the normal Facebook friendship graph does. The extract consists of 43953 nodes connected by 182384 edges. On average, nodes in the used graph extract have 8 friends.

As previously mentioned a trusted start group of registered nodes has to exist. For the simulations this group consists of 20 connected nodes. These initial start nodes as well as nodes which try to get registered later on are selected randomly from the graph but are connected to at least one already registered node. Due to the random selection of nodes every run of the simulation has practically a new social graph to work on.

The simulations run until 8000 nodes have been active in the graph. Longer runs would have been possible but do not change the qualitative results. Unless stated otherwise, the following results are based on simulation runs in which the initial trust value was 6, a request had to be confirmed by 50 percent of the registered nodes, and the adversary started his attack after 1000 nodes have been active. Each parametrization has been repeated 20 times.

2) Evaluation of parameters:

a) Initial trust value: As described in section IV-C1 the initial trust value determines how many requests can be send over an edge until the edge is depleted. It can be assumed that already small increments of this value have a noticeable effect on the usability. Doubling the initial trust value doubles the number of registered request per edge which approximately halves the number of required tries to succeed with a request. As can be seen in Figure 4 this effect is most notable with small values. After a certain increase of the initial trust value most requests can be finished when first trying, making further increments of the value unnecessary.

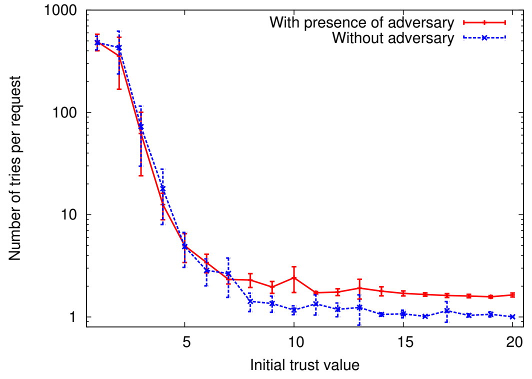

Fig. 4. Number of tries per successful request with different initial trust values. Displayed is the mean over the simulation runs and the 95% confidence interval.

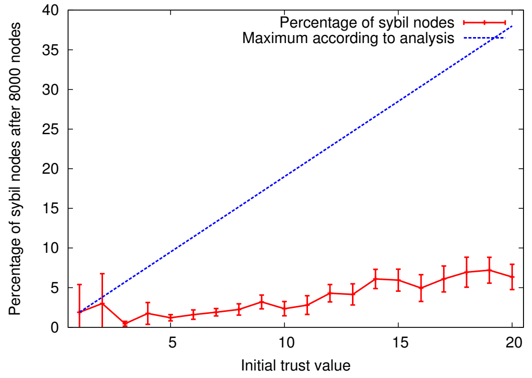

Fig. 5. Percentage of sybil nodes with different initial trust values.

On the contrary, high values should not be used. The percentage of sybil nodes is partially restricted by depleting the initial trust granted to the adversary. Consequently, he benefits from increases of the value as shown in Figure 5. In addition, his theoretically possible strength, based on the number of his friends and the initial trust value, is displayed. It can be seen that the percentage of the sybil nodes grows much slower than the theoretical bound calculated in section V-A1. This is the case since the requests of the adversary does not only have to be send over his direct edges but also through the rest of the graph where they might fail.

b) Time of attack: Section V-A1 details that the initial percentage of sybil nodes is based on the number of his friends, the initial trust value, and the time of his attack. The initial trust value is fixed in a deployed system and the number of friends is assumed to be bounded. As a result, the initial percentage of sybil nodes is mainly based on his time of attack.

When he waits longer before trying to take over the graph the percentage of sybil nodes he can register is diminishing. The absolute number of nodes he can register at once is bounded so a larger number of non-sybil nodes reduces the percentage of adversary controlled sybil nodes in the graph.

This can be seen in Figure 6 as well. The simulation kept the initial trust value constant and the number of friends of the adversary bounded and varied the delay before the adversary started to register sybil nodes. With no delay after the start of the simulation the adversary immediately controls a quarter of the nodes in the social graph. When the attack starts at a later time, when more non-sybil nodes already are part of the social graph, his initial percentage of nodes is significantly less high.

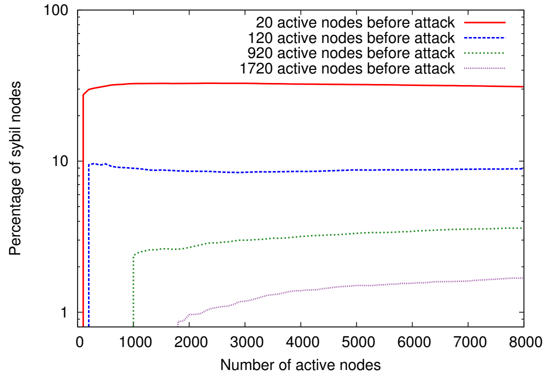

Fig. 6. Percentage of sybil nodes with different times of attack.

The slow rise of the adversary percentage when attacking at a later time, in contrast to the decline when attacking earlier, is caused by the way the simulator works. When attacking earlier, it takes less time until all his friends in the social graph are part of the overlay graph, too, resulting in a maximal percentage early on followed by the decline. When attacking at a later time, the same happens at a much slower pace.

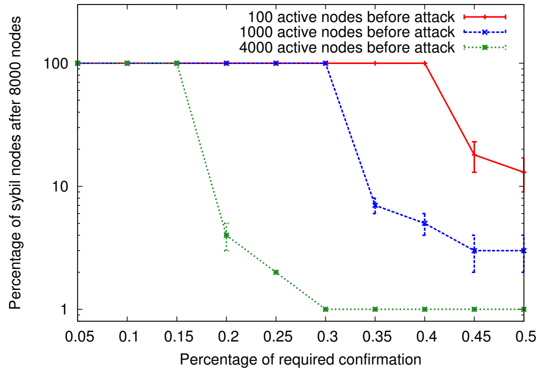

Fig. 7. Percentage of sybil nodes with different confirmation thresholds.

c) Confirmation threshold: To restrict the adversary the required fraction of confirmations is required to be always greater than the percentage of nodes the adversary might control. As set out in the previous section the percentage of sybil nodes is dependent on the delay before the attack starts. In Figure 7 the simulation results of different times of attack are shown. For example, when the attack starts after 4000 nodes already have been registered, 20 percent of the nodes have to confirm a new node to restrict the adversary.

For the honest users of the system a high fraction is disadvantageous. Higher values mean more requests they have to work on and longer delays until their requests can be finished. The latter is especially a problem when the graph becomes larger and more nodes have to confirm the requests.

3) Simulations on different graphs: To evaluate the influence of the social graph on the effectiveness of SybilHedge some simulations were run on different social graphs. The used graphs were:

Facebook Wall Posts This is the default graph used for the simulations. It is an extract of the real social graph on Facebook representing publicly visible interactions between users.

Facebook Friendships Another extract of the Facebook graph representing the friendship graph between users. Since many users register everyone as a "friend" it does not represent real-world friendship or mutual trust.

Watts & Strogatz A graph generator based on the small world model of Watts and Strogatz [18]. It generates a graph similar to real social graphs.

Torus Real social graphs contain communities of interconnected users with sparse cuts between the communities. This generator creates a completely regular graph without any structure, especially without the sparse cuts SybilHedge assumes.

Generator	Friends	Cluster Factor
FB Wall Posts	8	0.114948
FB Friends	25	0.221807
Watts & Strogatz	8	0.081832
Torus	8	0.428571

Fig. 8. Properties of the tested graphs. The values are averages over multiple nodes and simulation runs.

Some properties of the graphs are displayed in Figure 8. The row titled "Friends" lists the average number of contacts a node in the respective graph has. The cluster factor is calculated by measuring how many friends of a node are connected directly. High values indicate a highly connected graph and thereby a graph with only one community.

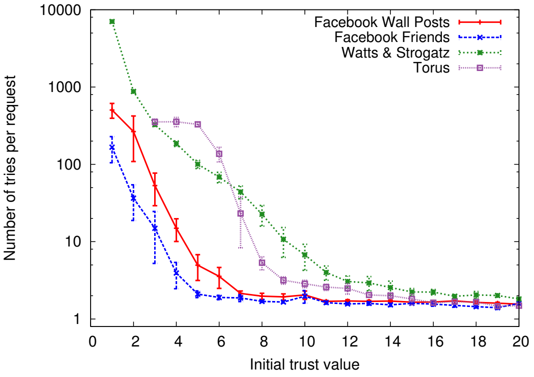

Fig. 9. Required numbers of tries with different initial trust values and graphs. The torus-graph starts with 3 since requests with lower values fail.

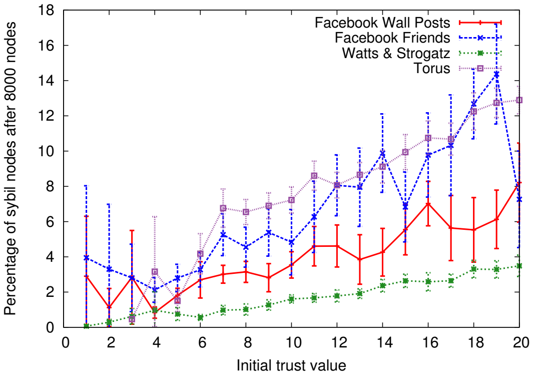

Fig. 10. Percentage of sybil nodes with different initial trust values and on different graphs.

The simulation results are displayed in Figure 9 and Figure 10. Nodes in the Facebook friendships graph have more friends on average and as expected require less tries per successful request. The adversary percentage increases since he can finish requests more easily as well. Similar the graph generated by the Watts and Strogatz generator requires more tries per successful request resulting in less registered sybil nodes. The increased number of tries is based on a lower cluster factor, indicating a higher number of smaller clusters. The trust values of the edges between the clusters are quickly depleted, blocking further requests.

The torus graph has a higher cluster factor since the graph is completely regular without any clusters. Nevertheless, a high number of tries is required per successful request. At the same time the adversary reaches a higher percentage of nodes as on the other graphs. A graph without communities allows low numbers of tries since a lot of path between the nodes exist. The same, however, is true for the adversary resulting in a lot of registered sybil nodes until parts of the graph are depleted, blocking the adversary and non-sybil nodes alike.

C. Conclusion

The evaluation of SybilHedge has shown that the amount of requests send by the adversary is successfully bounded. At the same time, most honest users can be registered on their first request.

A performance issue of the proposed algorithm is a quite high effort to confirm a request. In most cases it is not known at which time an attack will begin, forcing the usage of a high confirmation threshold. To avoid this overhead, the required fraction can be reduced when enough nodes are part of the overlay network.

VI. Performance and Practicability

The main design goal behind SybilHedge is to offer protection against sybil attacks without relying on centrally-controlled system components. In order to achieve practical adoption of the approach, however, several performance-related concerns must be investigated. This section discusses such concerns and proposes adaptations towards a complete identity registration system.

The main variable influencing performance, in addition to the size of the user population, is the number of confirmations required for registering a new identity. Reducing it reduces the expected duration of the registration process as well as scalability concerns related to the storage of registered identities. However, as demonstrated in Section V-B2c, a high confirmation threshold is important for effectively containing sybil attackers. In a practical system, the parameter can be adapted taking into account the size of the registered user base. Given a large number of registered users, for example, an absolute or sublinearly growing registration threshold might be sufficient. Additionally, registration requests can be coupled with additional challenges, e.g., the supplying of a photograph-based proof-of-identity, to further deter sybil attackers.

As SybilHedge is based on random walks, confirmations are collected in a linear fashion, implying linearly growing registration delays. Based on the positive results concerning the effectiveness of this approach against sybil attacks, more complex schemes can be investigated. For example, SybilHedge can be extended towards partial flooding by occasionally splitting random walks. A ticket-based approach, where confirmations are distributed opportunistically without a preceding request, is another possibility for improving registration delays.

Another potential concern is the overhead for verifying the validity of an identity's registration status. Recall that this validity is based on the number of collected confirmations. In a naive instantiation of our approach, each confirmation is realized through a cryptographic signature. This leads to linearly growing proof messages and computation times for validating registrations. The applicability of aggregatable signatures [16] and similar advanced cryptographic schemes can be investigated for alleviating these concerns. Additionally, for verifying confirmations, a consensus must be established over the set of currently registered identities. An adaptation of the blockchain paradigm [19] is promising here for establishing such a consistent common view. Registered identities, including their proofs of registration, can be stored in a distributed hash table, with commitments to registrations posted to an existing blockchain for ensuring consistency.

VII. Conclusion

We propose a decentralized approach for the sybil-resistant registration of pseudonyms that leverages the social interconnectedness of human users. Compared to already existing works, our solution allows the registration of general-purpose pseudonyms with proofs that can be verified without excessive communication overhead. We furthermore propose SybilHedge, a specific proposal for collecting confirmations about the non-sybilness of registration requests. To do so, the confirming nodes are only required to possess knowledge about their direct neighbors in the social graph. This way, the privacy protection is higher than in earlier systems. As shown in the evaluation, SybilHedge ensures that the amount of sybil nodes an adversary can introduce into the system is significantly reduced: The number of sybil nodes controlled by an adversary is constrained by his initial percentage of nodes and cannot be increased arbitrarily later on.

Acknowledgment

This work was supported by the German Ministry for Education and Research (BMBF) through funding for the Center of Excellence for Applied Security Technology (KASTEL).

References

[1] J. R. Douceur, "The sybil attack," in *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, ser. IPTPS '01. London, UK: Springer-Verlag, 2002, pp. 251–260.

[2] M. Florian, J. Walter, and I. Baumgart, "Sybil-resistant pseudonymization and pseudonym change without trusted third parties," in *Proceedings of the 14th ACM Workshop on Privacy in the Electronic Society*, ser. WPES '15. New York, NY, USA: ACM, 2015, pp. 65–74.

[3] C. Garman, I. Miers, and M. Green, "Decentralized Anonymous Credentials," in *Proceedings of the 22nd Annual Network and Distributed System Security Symposium (NDSS)*. USENIX, Feb. 2014.

[4] A. Narayanan and V. Shmatikov, "De-anonymizing social networks," in *2009 30th IEEE Symposium on Security and Privacy*, May 2009.

[5] L. Von Ahn, M. Blum, N. J. Hopper, and J. Langford, "CAPTCHA: Using hard AI problems for security," in *Advances in Cryptology - EUROCRYPT 2003*. Springer, May 2003, pp. 294–311.

[6] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, "Statistical properties of community structure in large social and information networks," in *Proceedings of the 17th International Conference on World Wide Web*, ser. WWW '08, New York, NY, USA, 2008, pp. 695–704.

[7] B. Viswanath, A. Post, K. P. Gummadi, and A. Mislove, "An analysis of social network-based sybil defenses," *SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 4, pp. 363–374, Aug. 2010.

[8] C. Lesniewski-Laas and M. F. Kaashoek, "Whanau: A sybil-proof distributed hash table," in *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation*, ser. NSDI'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 8–8.

[9] P. Mittal, M. K. Wright, and N. Borisov, "Pisces: Anonymous communication using social networks," in *20th Annual Network and Distributed System Security Symposium, NDSS 2013, San Diego, California, USA, February 24-27, 2013*, 2013.

[10] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman, "Sybilguard: Defending against sybil attacks via social networks," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 4, pp. 267–278, Aug. 2006.

[11] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao, "Sybillimit: A near-optimal social network defense against sybil attacks," in *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, ser. SP '08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 3–17.

[12] N. Tran, J. Li, L. Subramanian, and S. S. Chow, "Optimal sybil-resilient node admission control," in *The 30th IEEE International Conference on Computer Communications (INFOCOM 2011)*, 4 2011.

[13] P. Mittal, M. Caesar, and N. Borisov, "X-vine: Secure and pseudonymous routing in dhts using social networks," in *19th Annual Network and Distributed System Security Symposium, NDSS 2012, San Diego, California, USA, February 5-8, 2012*, 2012.

[14] D. Koll, J. Li, J. Stein, and X. Fu, "On the state of osn-based sybil defenses," in *Networking Conference, 2014 IFIP*, June 2014, pp. 1–9.

[15] S. Roos and T. Strufe, "A contribution to analyzing and enhancing darknet routing," in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013.

[16] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2001, pp. 514–532.

[17] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi, "On the evolution of user interaction in facebook," in *Proceedings of the 2nd ACM SIGCOMM Workshop on Social Networks (WOSN'09)*, 2009.

[18] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks." *Nature*, vol. 393, no. 6684, pp. 409–10, 1998.

[19] F. Tschorsch and B. Scheuermann, "Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies." *IEEE Communications Surveys & Tutorials*, Mar. 2016.