A Sybil-Resistant Admission Control Coupling SybilGuard with Distributed Certification

François Lesueur Ludovic Mé Valérie Viet Triem Tong
SUPELEC, SSIR Group (EA 4039)
Avenue de la Boulaie - CS 47601 - 35576 Cesson-Sévigné cedex - France
E-mail: firstname.lastname@supelec.fr

Abstract

Structured P2P networks are vulnerable to the sybil attack. In this attack, a misbehaving person creates many node identifiers and possibly chooses some of them in order to disrupt availability or integrity in the P2P network. In this paper, we propose a sybil-resistant distributed admission control system which combines SybilGuard with distributed certification. A new node can join the network if, using SybilGuard, a fixed ratio of the nodes think this new node is genuine and so participate in its distributed certification. This fully distributed system tackles each described aspect of the sybil attack, preventing users from creating many identifiers and enforcing the use of truly random identifiers.

Keywords: *P2P, Sybil Attack, Security*

Introduction

In structured P2P networks, each node has a unique identifier and the distribution of the all identifiers must be uniform and random in order to allow efficient and robust services.

In the sybil attack [4], an attacker creates a huge amount of node identifiers. This attacker then picks a specific subset of identifiers in order to execute one of the following attacks: (i) alter the performance of the network; (ii) take control of a resource and all its replicas; or (iii) control every node in the routing table of a victim. The problem is thus not only to enforce truly random identifiers but also to prevent users from creating many identifiers from which they can extract a specific subset.

There are mainly three types of sybil protections: computation-based [2], certification-based [5] and social-based [11] (*SybilGuard*). Computation-based protection relies on the difficulty for one person to compute huge calculations and so can be defeated by attackers with high computation power; certification-based protection is based on the difficulty for a real person to prove multiple identities but is centralized, which is opposed to P2P principles, and usually linked to an entry fee which can dissuade honest members from joining while being vulnerable to rich attackers; social-based protection relies on the difficulty to create social links but does not enforce users to have a random identifier.

In this paper, we propose to couple SybilGuard with a distributed certification scheme to obtain a social-based sybil-resistant admission control system for structured P2P networks. This mechanism provides the strengths of SybilGuard (social-based sybil protection, easy deployment) combined with those of a certification authority (truly random node identifiers) and allows to mitigate weaknesses of both. In order to achieve that, each node uses SybilGuard locally and the whole nodes cooperatively manage a distributed certification authority. A new node can join the network if, using SybilGuard, a fixed ratio t of the nodes think this new node is genuine and so participate in its distributed certification.

In Section 1, we detail related work on sybil protection and distributed certification, since our proposition is at the crossing of these two fields. Then, in Section 2, we present our sybil resistant distributed admission control system. In Section 3, we analyze our proposition and compare it to SybilGuard alone. In Section 4, we present preliminary simulations. Finally, we conclude and suggest some future work.

1 Related Work

In this section, we first analyze SybilGuard and certification authorities and we show that these propositions are complementary. Thus, we propose to combine these two approaches by distributing the certification among nodes running SybilGuard, and so we also present related work on distributed certification.

1.1 Sybil Protections

We detail here the two protections against the sybil attack we are combining in this paper: SybilGuard and certification authorities.

1.1.1 SybilGuard

In [11] Yu *et al.* present SybilGuard, a sybil nodes detection system based on social relationships. Each member manually creates connections to other members he knows in the real life and then registers automatically in the generated social graph one finite random route per edge. A random route is a special random walk in which each involved node uses a pre-computed random permutation mapping incoming edges to outgoing edges, instead of randomly choosing an outgoing edge for each walk. The consequence is that two different routes entering through the same edge also leave through the same edge; especially, all routes coming from the same attacker (hence coming from sybil nodes created by this attacker) converge to an unique route in the graph.

A node N then considers a node M genuine (resp. sybil) if the majority of N's random routes intersect one of M (resp. do not intersect). If a random route of an honest node traverses an attacker node, i.e., enters the *sybil region*, then this attacker can capture this random route. This attacker can then make this random route traverse only sybil nodes he has created and so intersect any route from any sybil node he creates. Thus, if the majority of random routes of an honest node traverses an attacker node, then this honest node is not protected by SybilGuard anymore and accepts an unbounded number of sybil nodes created by this attacker.

SybilGuard is based on the fact that all the sybil nodes created by a given physical attacker are only sparsely connected to the real social network: typically, one attacker with a few edges to the honest part of the social network creates several sybil nodes, all these nodes being connected to the social network only through this attacker. In SybilGuard, each protected node accepts $\Theta(gw)$ sybil nodes with $w = \Theta(\sqrt{n}\log n)$ the length of random routes, g being the number of edges between attackers and the honest part of the social network and n being the size of the network. More recently, in [10], Yu *et al.* announced working on reducing this order to $\Theta(g \log n)$, but simulations results are still unpublished to our best knowledge.

However, there is no control on the node identifiers used in SybilGuard. Each member has a public/secret key pair which is used to certify edges in the trust graph and each node is identified by the hash of his public key. An attacker can thus generate many key pairs and use one of them which hashes to a specific identifier [3].

SybilGuard is easy to deploy since each member has only to manually create connections to other members he knows in the real life. However, SybilGuard only allows for limiting the number of sybil nodes without enforcing randomly chosen identifiers, which seems useful for unstructured networks but insufficient for structured ones which rely on the uniform distribution of the node identifiers. To the contrary, certification authorities do not suffer from this weakness.

1.1.2 Certification Authorities

In [5] Druschel *et al.* propose to mitigate the sybil attack by using smartcards issued by a trusted third party, i.e., a *certification authority*. A certification authority can ensure that node identifiers are truly random and that each member has only one identifier. However, such a certification authority creates a central point of trust and failure in the network [6], which is opposed to P2P principles. Moreover, in order to give only one certificate to each member, this certification authority needs to check the real identities of the members and to maintain records of already delivered certificates. Each new member has thus to prove his identity through an external channel, which can be annoying.

A certification authority can enforce randomly chosen node identifiers and limit each person to have only one node identifier. However, such an authority is opposed to P2P principles and seems harder to deploy than SybilGuard. It appears then that coupling SybilGuard with a certification authority can compensate weaknesses of both. In this paper, we propose to distribute the certification among all the nodes in the network, all these nodes running SybilGuard.

1.2 Distributed Certification

Distributing a certification process can be achieved through threshold cryptography, based on Shamir secret sharing [9]. Threshold cryptography consists in sharing a secret key among different entities. A (t, n)-threshold cryptography scheme allows for ciphering a message through the collaboration of any t entities chosen from n, each entity having one share of the secret key. t shares are needed to cipher a message, but $t-1$ shares hold *no* information on the secret key. An attacker must thus obtain t shares of the secret key to be able to recover the full key.

In [8], we propose a distributed certification scheme in which the threshold t is a fixed ratio of the number of nodes instead of a fixed number of nodes, which is mandatory in varying size P2P networks. This ratio is enforced using a fully distributed scheme. This scheme complies with the P2P basics and allows to tolerate misbehaving nodes in the network.

The network is characterized by an RSA public/secret key pair (P, S), $P = (d, m)$ being publicly known and

$S = (e, m)$ being shared among the nodes (no node knows S entirely). This key pair is originally generated by founding members using a distributed algorithm as Boneh and Franklin proposed in [1]. The network is decomposed in s *sharing groups*, each group being formed by g_{min} to g_{max} members. Each share of the network secret key is affected to a specific sharing group and replicated on all its members. Signing a certificate requires then every share and thus the collaboration of one node of each sharing group. Given g_{min} and g_{max}, the ratio t of nodes needed to sign a certificate verifies $\frac{1}{g_{max}} < t < \frac{1}{g_{min}}$ and thus, sharing groups can split (resp. merge) when nodes join (resp. leave) with only local knowledge to enforce this ratio t.

2 Sybil-Resistant Admission Control

In this section, we present our sybil-resistant distributed admission control mechanism in an open structured P2P network. Membership is proved by a certificate signed with the network secret key S through distributed certification.

Each member A of the P2P network has a self-generated public/secret key pair (P_A, S_A) and a share of S. When a new node N wants to join the network, it has to obtain its certificate $Cert_N$ which consists of the hash of its public key P_N signed with the network secret key S. N needs the agreement and cooperation of a *fixed* ratio t of the nodes already members of the network to obtain this certificate.

Each of the asked nodes uses SybilGuard to guess whether N is genuine or sybil. In this paper, SybilGuard is viewed as a black-box present on each node and deciding for each other node if it is genuine or sybil. Such decisions are local to each node and so different nodes can take different decisions for the same genuine or sybil node. N obtains its certificate if and only if one node of each sharing group think N is genuine, hence if a ratio t of the nodes accept N.

Finally, N receives a share of S to be able to participate in future admissions of new nodes.

In the P2P network, each node A is uniquely identified by $nodeId_A = h(Cert_A)$. Since $Cert_A$ contains a signature with S which is not known by anyone, an attacker cannot predict $nodeId_A$ before the certification process.

However, in SybilGuard, any member can revoke his public key and create a new one. Such a possibility would allow here an attacker to obtain sequentially several certificates and finally choose a specific one which identifier allows him to take control over some resource. We thus slightly alter SybilGuard design by not allowing any member to revoke or change his public key. Still, when two nodes add a new edge between them in the social graph, these two nodes should reshuffle their routing tables (used to map incoming edges to outgoing edges) to maintain a random permutation among their edges. However, this reshuffling deviates some established routes and uses the key revocation mechanism to update route information on succeeding nodes: this reshuffling is thus no more available. With our alteration, new edges are mapped definitively to other new edges as soon as there are enough free edges and constant mappings prevent key revocation. The routing tables are thus not random permutations anymore, and we will have to study and perhaps mitigate the impacts.

Consequently, each member can present only one key to his friends (at the SybilGuard level) and can obtain the associated $nodeId$ only after having presented this key: an attacker has no control on his node identifier and thus cannot launch a targeted sybil attack to control a resource or proxy a victim node.

3 Analysis

In this section, we compare our proposition to SybilGuard alone (note that our proposition also allows to enforce truly random node identifiers). The network is composed of $n = 1,000,000$ nodes and each node has the same degree $d = 24$. Simulations of Section 4 show that giving the same degree to each node provides a good approximation even if it will be interesting to further elaborate on distribution of degrees in social networks. The results are based on those presented in [11].

We first calculate the optimal length of the random routes to allow each genuine node to successfully join the network with a probability of 0.999. Then, we estimate an upper bound of the number of sybil nodes a physical attacker can create, which corresponds to dw (d is the degree of nodes and w is the length of random routes). It is worth to be noted that, in fact, these dw nodes have decreasing probabilities of successfully joining the network, since their random routes have only from w to 1 steps in the honest part of the social graph: we only calculate an upper bound here. Finally, we calculate both when 25% of the nodes are sybil (we consider the network broken) and when attackers can insert as many sybil nodes as they want with a probability of 0.001.

P_c is the probability of collision between two random routes and the reversed birthday paradox in [12] gives the needed length of random routes w to achieve a given probability of collision P_c in a $1,000,000$ node network through the formula

$$w = \frac{\sqrt{ln(\frac{1}{1-P_c})}}{\sqrt{ln(\frac{1}{0.5})}} \times \frac{1906}{2.07892}$$

3.1 SybilGuard Alone, Without Distributed Certification

We consider that a node successfully joins the network when it is accepted by every honest node. The probability for a route to intersect at least 1 out of the d routes of another node is $1-(1-P_c)^d$. The probability to have i routes intersecting at least 1 out of the d routes of another node is $(1-(1-P_c)^d)^i((1-P_c)^d)^{d-i}C_d^i$. Probability to have more than half of the routes intersecting is

$$P_{inter} = \sum_{i=\frac{d}{2}}^{d}(1-(1-P_c)^d)^i((1-P_c)^d)^{d-i}C_d^i$$

.

Probability that each node accepts a given honest node is thus P_{inter}^n. Numerical resolution gives us $P_c = 0.1053$ to achieve 0.999 as probability of acceptation of a genuine node. $P_c = 0.1053$ yields $w = 368$, which allows each attacker to create a maximum of $dw = 8832$ sybil nodes.

Let k be the ratio of attackers and $w = 368$ the length of the random routes. In order to own $250,000$ nodes (25%), 28 attackers have to collude which corresponds to a ratio of attackers $k = 0.000028$. We calculate now the ratio of attackers after which those attackers can create an unlimited number of sybils in the view of a single honest node, which corresponds to this honest node having more than half of his random routes entering the sybil region. We consider a uniform repartition of sybil edges. The probability for a route not to enter the sybil region is $(1-k)^w$. Such route enters the sybil region with $1-(1-k)^w$. The probability that i routes enter the sybil region is $(1-(1-k)^w)^i((1-k)^w)^{d-i}C_d^i$. Probability to have more than half of the routes entering the sybil region is

$$P_{sybil} = \sum_{i=\frac{d}{2}}^{d}(1-(1-k)^w)^i((1-k)^w)^{d-i}C_d^i$$

.

Numerical resolution with $w = 368$ gives us that a ratio $k = 0.00061$ of physical attackers have a probability of 0.001 to introduce an unlimited number of sybil nodes in the view of one honest node.

3.2 SybilGuard With Distributed Certification

With distributed certification, a new node needs to be accepted at the SybilGuard level by one node of each sharing group to join the network. We have the size of a group g_i and the number of shares s ($n = \sum_{i=1}^{s} g_i$ is thus the size of the network). The probability to have more than half of the routes intersecting is P_{inter}. Probability that there is one node in a given group accepting a given node is $1-(1-P_{inter})^{g_i}$. Probability that there is one node in each group accepting a given node is $\prod_{i=1}^{s} 1-(1-P_{inter})^{g_i}$. If we consider sharing groups composed of $g_{min} = 20$ to $g_{max} = 40$ members, numerical resolution gives us the bounds $P_c = 0.0286$ (resp. $P_c = 0.0239$) for $g_* = 20$ (resp. $g_* = 40$) and $s = 50000$ (resp. $s = 25000$) to achieve 0.999 as probability of acceptating a genuine node.

$P_c = 0.0286$ needs $w = 188$ and $P_c = 0.0239$ needs $w = 172$. We use the worst case $w = 188$ in the following, which allows each attacker to create a maximum of $dw = 4512$ sybil nodes.

Let k be the ratio of attackers and $w = 188$ the length of the random routes. In order to own 25% of the nodes, 55 attackers have to collude which corresponds to a ratio of attackers $k = 0.000055$.

We calculate now the ratio of attackers after which those attackers can create an unlimited number of sybil nodes in the network. To break the system, attackers need either to break the distributed certification scheme or to break SybilGuard for one node of each sharing group. According to [8], distributed certification in a network composed of $1,000,000$ nodes with $g_{min} = 20$ and $g_{max} = 40$ breaks with probability 0.001 when 36% of the nodes are corrupted. If each attacker can create 4512 nodes, then a ratio of $k = 0.00008$ of attackers can break the system. Breaking SybilGuard for one node in each sharing group corresponds to have one node of each sharing group which accepts an unlimited number of sybil nodes, i.e., which has the majority of its random routes entering the sybil region. Probability that there is at least one node in a given group having more than half of its routes entering the sybil region is $1-(1-P_{sybil})^{g_i}$. Probability that there is one node in each group having more than half of its routes entering the sybil region is $\prod_{i=1}^{s} 1-(1-P_{sybil})^{g_i}$. Numerical resolution with $w = 188$ gives us that a ratio $k = 0.00311$ of physical attackers have a probability of 0.001 to break the system.

4 Preliminary Simulations

In this section, we present preliminary simulations in a $10,000$ nodes network. We analyze the needed length of random routes and the number of sybil nodes an attacker can insert into the network, in both cases of Sybilguard alone or using Distributed Certification. As in [11], we model the social network using a Kleinberg graph [7]. In a Kleinberg graph, nodes are arranged in a lattice and are connected to their p closer neighbors and to q randomly chosen long-range contacts, the probability of having each other node as a long-range contact being inversely proportional to the distance to this node according to a parameter r. In this section, we use $p = q = 8$ and $r = 1.9$, as in [11]. Each node has thus 8 local neighbors, 8 long-range contacts and, on average, 8 other nodes have this node as a long-range contact: the average degree of nodes is 24.

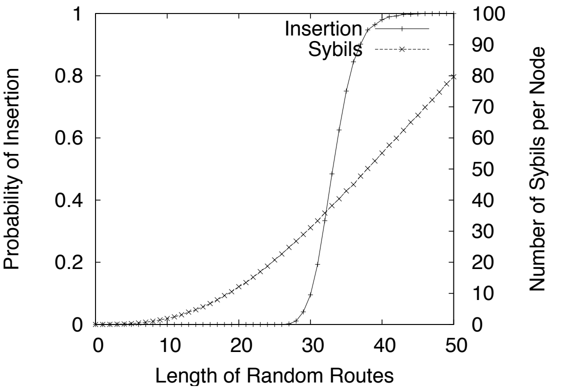

Figure 1. SybilGuard alone with $10,000$ nodes. The *Insertion* curve represents the probability of insertion for a new node (left axis) and the *Sybils* curve represents the number of sybil nodes an attacker can create (right axis), both in function of the length of random routes.

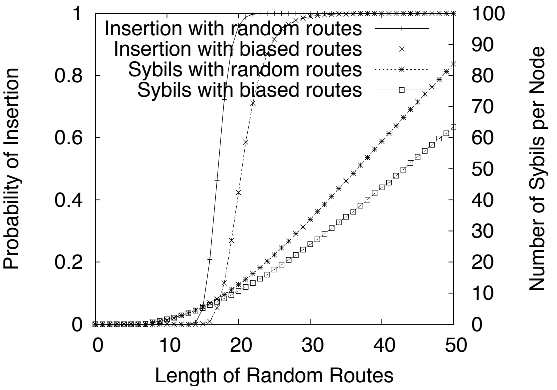

Figure 2. SybilGuard with Distributed Certification. The *random* routes represent the optimal case where routing tables are truly random (allowing key revocation) and the *biased* routes represent the experimental case in which there is no key revocation.

4.1 SybilGuard alone

With SybilGuard alone in a $10,000$ nodes network, Section 3.1 gives us the route length $w = 38$ to have 0.999 as probability of insertion. In Figure 1, this probability is obtained for $w = 45$. Using other simulations, it seems that this difference is explained by the fact that, in Section 3.1, we considered that all nodes had the same degree whereas in the simulations the degrees vary between 16 and 39. The formula of Section 3.1 using the degrees obtained in the simulation seems more precise since it gives $w = 47$. So, even if results of Section 3.1 already provide approximate results, it will be interesting to further investigate distribution of degrees in the theoretical analysis. For $w = 45$, each malicious node can on average create 67 sybil nodes.

4.2 SybilGuard with Distributed Certification

We now present results using SybilGuard with Distributed Certification in a $10,000$ nodes network. As stated in Section 2, users are not allowed to revoke their public keys and the reshuffling of routing tables when new edges are added is thus not possible anymore. In the presented simulations, we use a simple but not optimal algorithm: when a node, already inserted into the network, adds a new edge to a friend, this edge is not mapped immediately to another edge; then, when another edge is added, these 2 edges are mapped in both directions. The random routes generated are thus not truly random and we expect the results to be badly impacted.

Using SybilGuard with Distributed Certification in a $10,000$ nodes network, Section 3.2 gives us the route length $w = 19$ to have 0.999 as probability of insertion. In Figure 2, this probability is obtained for $w = 37$. This difference is explained in part by the distribution of the degrees and in other part by the biasing of the random routes due to not revocating keys. Simulations with random routes (allowing key revocation) give an optimal $w = 23$ to have 0.999 as probability of insertion: it will be interesting to approach this bound with a better algorithm to generate the routing tables. For $w = 37$, each malicious node can on average create 38 sybil nodes, which is nearly half of the ones allowed with SybilGuard alone.

Conclusion and Future Work

We proposed here a sybil-resistant distributed admission control system for structured P2P networks. This mechanism is based on the coupling of SybilGuard with a distributed certification. The use of SybilGuard limits the number of sybil nodes and provides an easy deployment, while the distributed certification prevents an attacker from choosing his node identifier. This coupling is thus able to tackle each described aspect of the sybil attack.

Based on properties from [11], we sketched a performance evaluation of the proposed system. Our system allows less sybil nodes than SybilGuard alone (lower w) and resists 55 physical attackers in a $1,000,000$ node network (28 for SybilGuard alone). These figures are still quite low but already better than for SybilGuard alone. It should also be noted that since SybilGuard relies on trust relationships, attackers would probably have fewer relationships than honest users in a real graph, which is not modeled here where all nodes have the same degree. Moreover, our system provides truly random node identifiers.

We also presented simulation results in a $10,000$ node network. These simulations show that the theoretical evaluation provides a good approximation of the performance, although it will be interesting to further elaborate on the distribution of degrees. Moreover, whereas the theoretical analysis only provided upper bounds on the number of sybil nodes an attacker can create, simulations show that this number is much lower on average and that the use of distributed certification halves this number.

Simulating such a system is a challenging task. In this paper, we followed [11] in which authors generate small world graphs using Kleinberg's model. Indeed, Kleinberg's model generates small world graphs, but it seems not sure whether they model social networks. In particular, the disparities between the degrees of the nodes in Kleinberg's model are not so high (degrees are comprised between 16 and 39 in our simulations) although they are in social networks. Moreover, Kleinberg's graphs generation algorithm inputs three parameters, which we can sum up as number of close successors in the graph, number of far successors, and the distribution of these far successors: to our best knowledge, there are no well-accepted values for these parameters to model social networks.

When using a real graph, the problem is that there are only a few real social networks data available (they are rarely public because they usually contain private information). In parallel to Kleinberg graphs, we thus plan on using a part of the PGP graph, which contains many nodes, usually linked through social relationships.

Finally, SybilGuard provides an heuristic to locally estimate a suitable value for w, which is dependent on the network size: we will also have to address this problem, since our w should be different.

References

[1] Boneh and Franklin. Efficient generation of shared RSA keys. In *Proceedings of the 17th Annual International Cryptology Conference (CRYPTO)*, volume 1294 of *Lecture Notes in Computer Science*. Springer-Verlag, 1997.

[2] Nikita Borisov. Computational puzzles as sybil defenses. In *Proceedings of the 6th IEEE International Conference on Peer-to-Peer Computing (P2P)*, volume 0, pages 171–176. IEEE Computer Society, 2006.

[3] Miguel Castro, Peter Druschel, Ayalvadi J. Ganesh, Antony I. T. Rowstron, and Dan S. Wallach. Secure routing for structured peer-to-peer overlay networks. In *Proceedings of the 5th ACM Symposium on Operating System Design and Implementation (OSDI)*, Operating Systems Review, pages 299–314. ACM Press, 2002.

[4] John R. Douceur. The sybil attack. In Peter Druschel, M. Frans Kaashoek, and Antony I. T. Rowstron, editors, *Proceedings of the International Workshop on Peer-to-Peer Systems (IPTPS)*, volume 2429 of *Lecture Notes in Computer Science*, pages 251–260. Springer-Verlag, 2002.

[5] Peter Druschel and Antony I. T. Rowstron. PAST: A large-scale, persistent peer-to-peer storage utility. In *Proceedings of the 8th IEEE Workshop on Hot Topics in Operating Systems (HotOS)*, pages 75–80. IEEE Computer Society, 2001.

[6] C. Ellison and B. Schneier. Ten risks of PKI: What you're not being told about public-key infrastructure. *Computer Security Journal*, 16(1):1–7, 2000.

[7] Jon Kleinberg. The small-world phenomenon: an algorithmic perspective. In *Proceedings of the 32nd annual ACM Symposium on Theory of Computing (STOC)*, pages 163–170. ACM Press, 2000.

[8] François Lesueur, Ludovic Mé, and Valérie Viet Triem Tong. A Distributed certification system for structured P2P networks. In *Proceedings of the 2nd International Conference on Autonomous Infrastructure, Management and Security (AIMS)*, volume 5127 of *Lecture Notes in Computer Science*, pages 40–52. Springer-Verlag, 2008.

[9] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11), 1979.

[10] Haifeng Yu, Phillip B. Gibbons, and Michael Kaminsky. Brief announcement: Toward an optimal social network defense against sybil attacks. In *Proceedings of the 26th ACM Symposium on Principles of Distributed Computing (PODC'07)*, 2007.

[11] Haifeng Yu, Michael Kaminsky, Phillip B. Gibbons, and Abraham Flaxman. Sybilguard: defending against sybil attacks via social networks. In Luigi Rizzo, Tom Anderson, and Nick McKeown, editors, *Proceedings of the ACM SIGCOMM Conference (SIGCOMM)*, pages 267–278. ACM Press, 2006.

[12] Haifeng Yu, Michael Kaminsky, Phillip B. Gibbons, and Abraham Flaxman. Sybilguard: defending against sybil attacks via social networks. Technical Report IRP-TR-06-01, Intel Research Pittsburgh, 2006. Also available at `http://www.comp.nus.edu.sg/~yuhf/sybilguard-tr.pdf`.