The Sybil Attack - Theory and Practice

Kelong Cong
Delft University of Technology
k.cong@student.tudelft.nl

ABSTRACT

TODO

1. INTRODUCTION

Electronic commerce and online social networks are common phenomenons at the present time. They allow us to orchestrate many aspects of our lives in the comfort of our homes, behind the monitors of our devices. An online identity is often required to use such services, for examples we must create an account to tweet[1] our friend, who must also have an account. In this scenario, users can choose to remain psudonymous if they are careful, where their real-life identity is uncorrelated with their online identity.

While creating pseudonyms is useful for protecting users' privacy, it also opens an alleyway for attackers. The Sybil attack, first described by Douceur[19], is an attack where an entity can assume multiple identities or Sybils, and then attack either another entity or undermine the whole system. For example, a malicious Twitter user can create many fake identities and have the fake identities follow his real identity, thus creating a false reputation. It is one of the most important attacks because it leads to a large number of consequences including but not limited to spreading false information, identity theift[5] and ballot stuffing[4]. Furthermore, to the best of our knowledge, there is no general solution for preventing the Sybil attack.

In this work, we survey various aspects of the Sybil attack. But in contrast with previous surveys, we include both the theoretical and practical aspects. First, we describe the Sybil attack in more detail and and illustrate its importance by looking at how researchers and black-hat hackers mounted the attack on real-world e-commerce and online social network systems in section 3. Since there is a large variety of Sybil attack defence mechanisms, from using trusted-third-party to exploiting the graph characteristics in online social networks, thus we classify these mechanisms by their "main idea" in section 5. Finally we present the related work and conclude in section 6 and section 7.

[1]A message sent using Twitter is a tweet.

© 2016 ACM. ISBN 123-4567-24-567/08/06.
DOI: 10.475/123_4

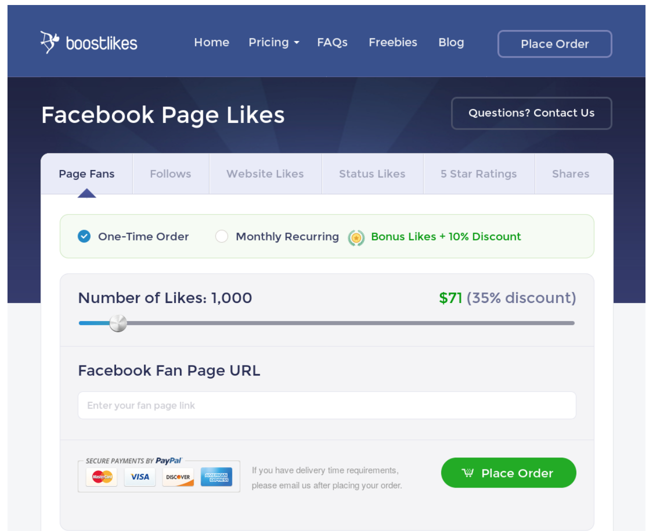

Figure 1: Screenshot of the Facebook likes service page of boostlikes.com.

2. MOTIVATION

We begin our survey by showing some alarming Sybil attacks happening in the real-world. Social network and micro-blogging websites are popular platforms for organisations to improve public relations and their reputation, but they are also platforms to spread propoganda. A recent article in the Atlantic described how Twitter bots (Sybils) are shaping the 2016 US presidential election[26]. Over a third of pro-Trump tweets and almost a fifth of pro-Clinton tweets, totalling at about 1 million, came from bots. The article questions whether the bots are a threat to democracy because opinions of real users are eclipsed by spam of bots.

Using Sybils to manipulate public opinion is not only accessible to campaigners with a large budget. There are marketplaces where anybody can purchase reputation scores such as Twitter followers. BoostLikes shown on Figure 1 is a professionally presented website, it offers a large range of services including Facebook likes, Twitter followers, Instagram followers and YouTube views. SocialFormulae (Figure 2) is a similar service but at a much lower price point, one thousand Twitter followers is only $9.99. There can be little

Figure 2: Screenshot of the main banner on socialformulae.com.

doubt that those companies use automated bots to provide their services.

SadBotTrue and its related website Socialpuncher publishes studies on social media fraud. Two of their studies is particularly useful for demonstrating the scale of the Sybil attack on Twitter. Firstly, there exist a botnet that consist of 3 million accounts. Since their creation, they generated 2.6 billion tweets. Surprisingly, all of the 3 million accounts were created on the same day and the account names are simply numbered sequentially[63]. Such an obvious activity should be easily detectable by Twitter, but these accounts are still not closed at the time of writing. Secondly, the top-100 Twitter users have 523 million unique followers between them, but 310 million are bots, that is almost 60%[71]. Suppose the bots all belong to the same attacker, then they can effectively suppress the opinions of the real users.

Clearly, the defence mechanisms employed by social network and micro-blogging websites are not adaquate to combat the Sybil attack. If the Sybils infiltrate even more of our cyberspace, then it may become a form of censorship. In that case, can we still be considered to have the right to freedom of speech?

TODO Tor https://blog.torproject.org/blog/tor-security-advisory-relay-early-traffic-confirmation-attack

3. THE SYBIL ATTACK

The Sybil attack is coined by Douceur[19] in 2002 in the context of peer-to-peer systems. But people were well aware of it before 2002. For instance in 2001, Friedman and Resnick used the term "cheap pseudonyms" to describe Sybils [61]. In this section, we introduce the key theoretical results and the definitions used in the remainder of this survey.

3.1 Theoretical Results

Douceur defined the Sybil attack as forging multiple identities under the same entity[19]. An entity can be for example a physical user of the system and identities are how entities present themselves to the system. Thus a local entity has no direct knowledge of remote entities, only their identities. The forged identities do not necessarily follow the protocol specified by the underlying network, i.e. they assume the characteristics of Byzantine fault[41].

The author modelled the system as a general distributed computing environment where there is no constraint on the topology, every node has limited computational resources and messages are guaranteed to be delivered. Under this model, the author proved that the Sybil attack is always possible without a central, trusted authority.

Cheng and Friedman proved a similar result in the context of reputation systems[11]. Reputation systems are commonly used in e-commerce websites and the internet in general, where identities are rewarded by their good behaviour or usefulness. Google's PageRank[57] is an example of a reputation system, where a large number of links to a website makes it more reputable. It was formally proven that peer-to-peer reputation systems cannot be made to prevent the Sybil attack, it is only possible prevent it by using trusted parties.

3.2 Model and Definitions

One of the common models, especially in the context of online social networks, is shown in Figure 3. It is first introduced by the authors of SybilGuard[89]. Nodes inside the left region are identities created by honest entities, the edges connecting those nodes are real-world trust relationships. The right region contains the Sybils and they are connected with fake relationships. The edges connecting the two regions are called *attack edges*. These can be created by tricking an honest user to befriend a Sybil, stealing an honest user's account and so on. If malicious users create too many Sybils, then the graph begins to have certain properties, i.e. it will have a quotient cut. Effectively, removing the attack edges will disconnect a large number of nodes from the social graph. Many Sybil defence mechanisms rely on the fact that attack edges are difficult to create as we will describe in section 5.

4. SYBIL ATTACK IN VARIOUS APPLICATIONS

Sybil attacks can be mounted in different applications and cause a large array of consequences. This section categorises the attacks by the goal for four common applications. (1) P2P (peer-to-peer) file sharing networks such as BitTorrent, (2) OSN (online social networks) such as Twitter and Facebook, (3) reputation systems such as eBay and (4) WANET (wireless ad-hoc networks) such as sensor networks. We hope this section further illuminates the alarming consequences of the Sybil attack.

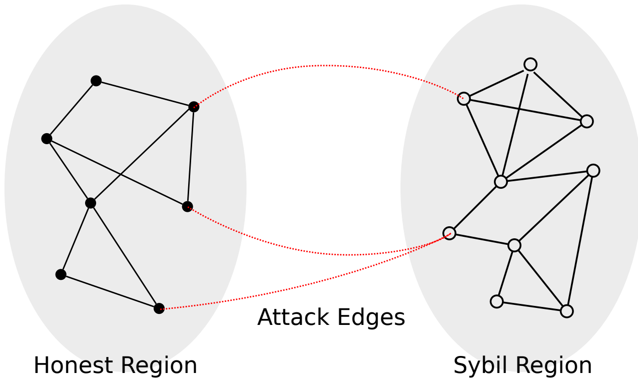

Figure 3: The model in many Sybil defence mechanisms can be seen as a social graph that is partitioned into a Sybil region and an honest region. The two regions are connected by *attack edges*. Note that in general there may be multiple honest regions multiple Sybil region.

4.1 The Sybil Attack in P2P FIle Sharing Networks

P2P (peer-to-peer) file sharing networks are distributed computer networks that are built for discovering and sharing files. BitTorrent[13] is likely the most popular P2P network at the time of writing. Due to their open and distributed nature, they are vulnerable to the Sybil attack.

4.1.1 Index Poisoning

P2P networks often implement a DHT (distributed hash table). The DHT in BitTorrent is called Mainline-DHT, based on Kademlia[48]. Keys are the infohashes (file identifiers) and values are the metadata of the files, these are distributed across all the participating peers. Every node stores a routing table and requests are routed iteratively to the node responsible for a particular key[45]. The goal of index poisoning is to corrupt routing table so that honest peers fail to find the values they want. It can be mounted by injecting Sybils into the DHT that do not follow the protocol. Wang and Kangasharju created honeypots in the BitTorrent network and detected as many as 300,000 Sybils[84]. Similar attacks are possible in other P2P networks such as Overnet[43].

4.1.2 Eclipse Attack

Steiner et al. mounted an Eclipse attack on a Kademlia based DHT for P2P file sharing known as KAD, it is used in Overnet and eMule[74]. The Eclipse Attack[70] is a special form of targeted Sybil attack. Sybils are arranged in the network such that they eclipse the the victim from the rest of the network. The victim can either an identity or an object such as a key in DHT. The authors mounted the latter variation. The authors show that it is possible to eclipse a key using only 32 Sybils in a DHT with 1.5 million users and 42,000 key.

4.1.3 Denial of Service

By exploiting vulnerabilities in the BitTorrent network, denial of service attack can be directed at any machine connected to the internet, not just machines in the network[69]. The main idea is to report the victim as the tracker (a server that coordinates the peers). El Defrawy, Gjoka and Markopoulou created a small scale proof-of-concept attack. Using only one machine, they could generate enough traffic to cripple small organisations and home users. The authors suggested that if Sybils are created to perform the same attack aimed at a single victim, then it could easily throttle links with much higher bandwidth[20].

Steiner et al. also succeeded in mounting a DDoS attack but using their in the context of the aforementioned KAD DHT[74]. Instead of replying the correct list of peers to DHT queries, the Sybils always respond with the IP address of the target peer in an attempt to overwhelm the target. The authors show evidence that real-world malicious DDoS attacks involving more than 300,000 peers are mounted using P2P networks.

4.1.4 Spying

Many authors have used Sybils to monitor or spy a P2P file sharing network that uses DHT[31, 74]. In essence, the authors created a lot of light-weight Sybils and trick all the honest peers to store them in their routing table, a form of index poisoning. The Sybils are light-weight because do not follow the DHT protocol and perform much simpler operations, thus a single machine can have thousands of Sybils running simultaneously. Finally, DHT requests would “pass through” the Sybils due to the poisoned routing table, and the requests can be stored in a database for further analysis.

4.2 The Sybil Attack in OSN

OSN (online social networks) are vulnerable to the Sybil attack even when most of them use a central, trusted authority. In OSN, users create profiles and form relationships with friends. In contrast with real world relationships, it is much easier to create relationships in OSN even with strangers. In 2008, Sophos conducted an experiment where they created a Facebook profile and send friend requests to 200 random users, and 41% of the users accepted the friend request[72].

Many OSN in fact have a large number of Sybils. A report by Facebook at the end of 2011 stated 5-6% of their accounts are fake[58]. Jiang et el. analysed data from Renren[2], they discovered almost 1 million Sybils from 2440 Sybil groups[34].

Attackers leverage the ability to fool users into becoming one of their Sybil’s friend and the ability to create a large number of Sybils to mount a large variety of attacks on the user. We outline the different types of attack in this section. Note that online social networks often have a reputation aspect as well, for example a Facebook page with a lot of fans may be considered to be more reputable than others. We discuss attacks specific to OSN in this section and attacks on reputation in subsection 4.3

4.2.1 Identity Theft

Authors of [5] created two attacks - profile cloning and cross-site profile cloning, targeting five social network sites including Facebook and LinkedIn. The iCloner system was created to automate these attacks.

In profile cloning, iCloner uses publicly available information to automatically create clones of the victim’s profiles, effectively creating Sybils. iCloner then sent friend requests from the cloned profile to the friends of the victim. The fact that the victim may have many friends that they do not

[2]One of the largest social network in China.

contact very often, e.g. friend from primary school living in another country, makes this attack highly effective. The authors found that the acceptance rate for cloned profiles was over 60%. Much higher than the acceptance rate of 30% for fictitious profiles. Once the friendship is established, it is possible to extract private information that is not available publicly and perform identity theft.

The idea of cross-site profile cloning is similar, except the cloned profile is created on another social network site that the victim does not yet use. Once the cloned profile is created, iCloner attempts to identify friends of the victim and begins sending friend requests. Similarly, 56% of the friend requests were accepted.

A more recent study created SbN (Socialbot Network) targeting Facebook[8]. Each socialbot is a Sybil created by the attack, it controls a forged profile and minic human behaviour to avoid detection. The attacker is the botmaster who coordinates the socialbots to achieve a common objective such as infiltrating the target OSN by creating friend relationships with real users. The authors found that infiltration success rate was as high as 80% and the FIS[73] (Facebook Immune System) was not sufficient to prevent the attack. Once the relationships are established, the botmaster can command the socialbots to start gathering private information which can then be used for identity theft.

These examples demonstrate that the carelessness of users and the ability to create Sybils makes OSN vulnerable to identity theft. Moreover, identity theft is only an entry point. Once trust relationships are established, the attacker can perform many other types of attacks such as spamming, phishing or astroturfing to gain advantage.

4.2.2 *Astroturf*

Astroturfing is an act of creating grassroot movement that are in reality carried out by a single entity, effectively spreading misinformation to legitimate users. It relies on the ability to create Sybils in the underlying social network. This type of attack is especially effective in social networks such as Twitter where a lot of the social interaction such as sending messages happen in the public.

In the 2010 Massachusetts senate race, Mustafaraj and Metaxas found evidence that Republican campaingners created fake Twitter accounts and used them to send spam. The spam caused Google real-time search results to tip in their favour thus causing a spread of misinformation[54]. Ratkiewicz et el. suggest that this type of attack can be mounted cheaply and may have a larger influence than traditional adversiting[60].

The Truthy system[60] is a web service that perform real-time analysis of Twitter to detect political astroturfing. In the 2010 U.S. midterm election, the authors found accounts which generated a lot of retweets but no original tweets. More importantly, they uncovered a network of bot accounts that injected thousands of tweets to smear the Democratic candidate.

In 2012, Wang et el. investigated two of the largeset crowdturfing[3] platforms in China that brings together buyers and sellers - Zhubajie and Sandaha. One of their services is perform astroturfing on Weibo (The Chinese Twitter). The authors found that the 5364 sellers collectively own 14151 Weibo accounts and the top 1% of the sellers own over 100 accounts. Furthermore, the business is growing and more than $4 million have been spent on these two platforms over five years[83].

4.2.3 *Spam*

Spamming, much like in the context of email, is the act of sending unsolicited or undesire messages (spam). The goal of the attacker varies from advertisement to phishing and spreading malicious software[28, 76]. Many studies have characterised the behaviour of the spammers and found that many spammers are in fact Sybils [75, 87, 25, 34]. More importantly, spamming is possibly the most common attack in OSN. Jiang et el. analysed the malicious activities of the Sybils and found propogating advertisement (at 32.8%) is the most common one[34]. Some authors have worked with the service provide to close the spam accounts, but it is clearly not sufficient as we described in section 2.

4.3 The Sybil Attack in Reputation Systems

Reputation systems cultivate collaborative behaviour by allowing entities to trust each other based on community feedback, usually in the form of a reputation score. Entities decide whom to trust based on the reputation scores, thus entities are also incentivised to behave honestly. Reputation systems are found in many context. In e-commerce, namely eBay, researchers found that the merchant's reputation "is a statistically and economically significant determinant of auction prices"[32], and "buyers are willing to pay 8.1% more" for goods sold buy a reputable merchant[62]. The file sharing peer-to-peer network BitTorrent uses tit-for-tat as an ephemeral reputation system to encourage peers to upload in exchange for better download speeds[12]. The aforementioned PageRank[57] is also a reputation system, used for ranking reputable websites higher in Google's search results.

Unfortunately, reputation systems are also vulnerable to the Sybil attack. Worryingly, there appears to an industry built around it, and their products are easily accessible in the clearnet. In this section, we describe practical attacks on reputation systems.

4.3.1 *Self-promoting*

In self-promotion, the goal of the attacker is to illegitimately raise its own reputation. A common way to perform self-promition is to create Sybils and have them create positive reputation for the attacker's main identity.

Dini and Spagnolo studied the economics of buying reputation on eBay. The authors discovered many cheap items (around €0.7) for sell are simply there to boost feedback. For example, one of the item is titled "Apple Cranberry Crisp Recipe + 100% Positive Feedback". The authors successfully boosted their feedback by purchasing such items. But they made an unsuccessful attempted to place a bid on their own good with a fake account[18].

De Cristofaro et el. performed an emperical study on Facebook page promotion using like farms[15]. Some of the farms such as `SocialFormulae.com` are clearly operated by bots and the operator does not attempt to hide it, others such as `BoostLikes.com` tries to minic human users. The authors purchased the "1000 likes" service on their empty Facebook pages. In under a month, many empty pages have accumulated almost 1000 likes as promised by the like farms. The authors empty accounts were not terminated. Only a small number of the liker's account were terminated.

SEOClerks and MyCheapJobs are also evidences of mar-

[3] Crowdsourced astroturfing.

ketplaces for self-promotion. Some of the top services include "1 million Twitter followers" at $849, "1000+ Instagram followers" at $10 and so on. The revenues of those two marketplaces are estimated to be at $1.3 million and $116 thousand, respectively[21]. Although the authors did not investigate the properties of the fake followers, there is little doubt that many of accounts used in these services are Sybils.

4.3.2 *Slandering*

The goal of a slandering attack is to illegitimately produce negative feedback to undermine the reputation of the target. It is easy to imagine the improvement in effectiveness when using multiple Sybils. From the best of our knowledge, there are no published studies on real-world slandering. But research has shown having a negative feedback may harm the target's ability to do business[3].

4.3.3 *Whitewashing*

In whitewashing, attackers abuse the reputation system for temporary gain and then escape the consequences by joining the reputation system under a new identity to shed their bad reputation. Clearly, whitewashing is only possible when the Sybil attack is possible. Again, there are no studies on whitewashing in the real-world. But many have suggested that it is feasible attack[30, 47].

4.4 The Sybil Attack in WANET

WANET (wireless ad-hoc networks) is a dynamic, self-configuring, self-healing wireless network. Ad-hoc in this case means it does not rely on existing infrastructure for the network to function. Each node in the network is responsible for some general tasks such as routing, and some application specific tasks such as gathering data from its sensors in the case of a sensor network.

Akin to the other applications, an attacker in a WANET may own a single physical node, but it may behave as if it were a large number of nodes. Many WANET designs involve a reputation system[24, 9], thus the same attacks from subsection 4.3 applies here. In this section we describe the WANET specific attacks. From the best of our knowledge WANET are not widely deployed in practice, thus there is little research on real-world attacks.

4.4.1 *Unfair Resource Allocation*

Nodes in WANET often have limited resources such as bandwidth of the radio channels. Resources such as these must be shared between the neighbours using time slices. When the neighbours are Sybils, then the attacker can receive an unfair amount of resource allocation and denies resources for the honest nodes[56]. In contrast with the other attacks, this works even when the Sybils are not behaving malicioiusly.

4.4.2 *Routing Disruption*

An important routing technique is multipath routing, data is routed using multiple paths in the network for better fault-tolerance and bandwidth. However, if Sybils are present in the network, then the different paths may in fact go through the Sybils owned by the same attacker. Another technique geographic routing, nodes route data depending on the geographic location of their neighbours. Sybils in the network can be in more than one place at a time, thus significant distrupting the routing algorithm[37].

4.4.3 *Spreading False Information*

Nodes often need to exchange information with each other to satisfy the underlying requirements of the application. Some of the common tasks include data aggregation, voting. With enough Sybils, it is possible to manipulate the aggregated data or the poll to benefit the attacker. For example, sensor networks may use a bollot to detect misbehaving nodes, the attack could use its Sybils to claim that a honest node is misbehaving and have the other nodes expel it from the network[56].

4.5 TODO

a test bed for sybil attacks[33]
Quantifying Sybil attack[46]

5. DEFENCES

In this section we categorise various defence techniques against the sybil-attack. Many of them are independent of the application, thus we classify them on their main idea, and state explicitly when the mechanism is application specific.

5.1 Certificate Authority

CA (certificate authorities) check the users' identities and then issues certificates to honest users. The certificate can be tangible (trusted hardware[56]) or non-tangible (public key certificate) depending on the application. When an identity wishes to user the application, the CA must verify the validity of its certificate to ensure one-to-one correspondence. This mechanism prevents the Sybil attack as long as the CA does not make mistakes in the issuance stage.

CA can prevent the Sybil attack but it also has a lot of downsides. (1) Users have different opinions and may not agree on a single CA. (2) Users living in authoritarian regimes may not have access to the CA in use. (3) It is difficult to scale up a CA to meet increasing users demands. (4) Anonymity is difficult to obtain because the CA has complete information of the entities. (5) It is a central point of failure; i.e. if the attacker obtains the private key to create certificates then he or she can easily generate Sybils, if the CA goes offline then the application ceases to function because it can no longer verify identites.

Many existing systems today use a form of CA. X.509 is a standard for certificates and is used in a large variety of applications, for example websites (TLS), email (S/MIME), smart cards and so on. Payment systems such as PayPal verifies identities using credit card billing addresses.

5.2 Resource Testing

Every attacker can create multiple Sybils, but the attack cannot duplicate its resources the same way. In resource testing, the goal is to limit the influence of every identity with respect to the amout of resources it can expend. The resource type can vary depending on the application as we describe below. It may deter casual attackers but its usefulness degrades for resourceful attackers.

In P2P networks, IP address can be used as a resource. In Tarzan[23], neighbours are selected not from all known IP addresses, but from distinct IP prefixes. The effectiveness of the Sybil attack is reduced if the attackers cannot easily create Sybils in a large range of IP prefixes. Another example the self-registration technique [17]. When a peer wish to join the network, it needs to compute a ID which is a cryptographically secure hash of its own IP address and port number. While participating in the nettwork, other peers need to verify that the ID matches the peer's origin.

Aspnes et el. proposed the idea of solving difficult puzzles to limiting the number of Sybils[2]. The puzzle in this case is computing hashes on some input y concatinated by a some string x such that the digest begins with w number of zeros. Essentially, every node acts as a verifier by picking y and then broadcast it to the network to the puzzle solver[4]. x is picked by the puzzle solver. The puzzle solver must compute as many x as possible such that they match the requirement. Sybils are unlikely to produce enough x's so the honest nodes will refuse to interact with them.

In fact, many crypto-currencies use the same idea. For instance *proof-of-work* in the case of Bitcoin [55]. The Bitcoin blockchain is a global ledger and it needs to reach a concensus for its blocks. Nodes in the Bitcoin network essentially "vote" for the the latest block. But the vote is performed not by counting the majority, but by "counting" the amount of CPU power, i.e. one CPU is one vote. Thus an attacker cannot simply create a lot of identities to out-vote the honest nodes. It needs to gather a lot of CPU power which is much more difficult.

5.3 Registration Fee

Friedman and Resnick is one of the first to propose the use of a registration fee[61]. It is similar to resource testing except it only happens on registration. Entities can be charged a fee for creating identities, often facilitated by a central authority. The fee need to be set appropriately so that the cost of creating Sybils outweights the benefits but does not hinder honest entities.

The fee does not need to be monetary. For instance, Friedman and Resnick proposed the idea of a once-in-a-lifetime identity[61]. It uses blind signatures and a central authority, the authority does not know the mapping between the real identity (e.g. real-world identity) and the pseudo identity of entities, but it checks whether there has been previous registrations of the same real identity. In this case, the fee is the real identity, and attackers cannot create an arbitrary number of real identities. CAPTCHA[81] is another form of registration fee. It prevents programs from automatically creating new identities and limits the rate at which identities can be created by asking users to solve a puzzle that is difficult for computers.

Feldman et el. proposed another form of registration fee for P2P networks - the adaptive stranger policy[22]. When new peers join the network, they are treated using a policy that is adapted from previous newcomers. For example, the new peers may be expected to contribute to the network before they are allowed to receive benefits from the "mature" peers. The downside is that the policy may deter honest users from joining the network in the first place.

5.4 Network Flow Based Techniques

Network flow based techniques began with BarterCast[49]. It was initially designed to combat freeriding in P2P file sharing networks, where users are selfish and do not share content, but its idea can be extended combat the Sybil attack. The ideas based on BarterCast do not directly identify Sybils, but they prevent Sybils from doing harm in the P2P network.

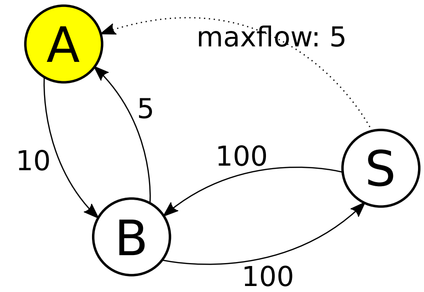

Figure 4: Subjective graph of A. The numbers are the amout of data transfered, they can be seen as the capacity in the context of the maximum flow problem.

The main idea comes from real-world social networks, where the reputation of a person can be from direct experiences, or information obtained frome someone else. The direct experience is always true, but the indirect information may not be, i.e. people can lie about their experiences. Humans solve the problem by treating the indirect information with a grain of salt unless the source of the information is highly trusted.

BarterCast applies this idea in P2P file sharing networks. Peers all maintain a subjective graph which is created by exchanging messages with their neighbours. The direct experiences measured by the number of bytes uploaded and downloaded are represented by the outgoing and incoming edges from the peer, respectively. Indirect experiences are represented by edges that are not directly connected to the peer. For example in Figure 4, A is the subject, it has direct experiences with B and B has told A about S, so it has indirect information about S. But A is unsure about the truthfulness of S's contribution, so it only trusts S as much as it trusts B. This idea is realised using a maximum flow algorithm and the final reputation metric is given in Equation 1.

$$R_i(j) = \frac{\arctan(\text{maxflow}(j,i) - \text{maxflow}(i,j))}{\pi/2} \tag{1}$$

BarterCast does not prevent the Sybil attack by itself. Because attackers can first upload a lot of data to obtain a good reputation in the network. If the attacker now creates Sybils and false report of the Sybils saying that they uploaded a lot. Then the peers who have interacted with the attacker will be tricked to think that the Sybils also have a high reputation. To fix this problem, Delaviz et el. created SybilRes[16]. The main idea is the following. Suppose there are two peers A and B who are sharing data. If A is uploading (represented by an outgoing edge) to B, then it decreases the weight of the incoming edge from B. Vice versa, the weight is increased for the outgoing edge when A is downloading. The rate of change depends on the capacities of the edges and the amout of data transferred after computing the reputation. Using the definition in Figure 3, the attacker cannot built up reputation for its Sybils by uploading to peers in the honest region beforehand, it is now

[4]We simplify the protocol a bit because the original protocol (called Democracy) is made to be Byzantine fault tolerent and is a bit more involved.

forced to keep on uploading to keep its Sybil's reputation which is a much more desirable behaviour.

Seuken et al. provided a formal model of BarterCast. They found that BarterCast is vulnerable to misreporting and proposed a solution called the DropEdge mechanism[66, 67]. DropEdge, like the name implies, drops some edges in the subjective graph that satisfies the following constraints. Suppose peer A wishes to download from peers in set C (the choice set). Then any reports received by A from $p \in C$ is dropped. Also, edges with both end points in C are also dropped from A's subjective graph. Intuitively, peers in C cannot misreport their contribution. The authors formally prove this property in their work. They also prove that it is robust against weakly benefician Sybils, that is Sybils that do not perform actual work for honest peers.

SumUp[77] is a defence mechanism specific for the vote aggregation problem. For example, in social news aggregation websites such as Reddit, users vote on the submitted content to determine its ranking; the problem occurs when Sybils can out-vote honest users. It is a centralised approach that fits the architecture of most websites that perform vote aggregation. SumUp consist of three stages. Firstly, pruning is performed to limit the number of incoming edges of every node, this is to reduce the number of attack edges available and reduce the computational cost in later stages, the threshold is a system parameter. Secondly, it uses a ticket source (the central component) distributes tickets in a breadth-first search manner equally to its neighbours, every node keeps one ticket and distributes the remaining tickets the same way. The number of tickets distributed across an edge plus one is the capacity of the edge. Effectively, edges closer to the ticket source have a high capacity. This idea keeps the capacities in the Sybil region low so that they do not have a large influence on the outcome. Finally, the maximum flow is computed where the source is simply the ticket source and the sink is an imaginary node with edges of capacity one that is connected to every voter. SumUp offers a better guarantee than SybilLimit where it only accepts $1 + o(n)$ votes per attack edge. An improved version of SumUp - GateKeeper is discussed in subsection 5.5.

Conversely, maximum flow is dual to minimum cut, so the problem of finding Sybil can also be formulated as finding sparse cuts[5]. Kurve and Kesidis devised an algorithm for finding sparse cuts to detect Sybils[40]. Unlike the aforementioned techniques, it relies on the presence of trusted nodes.

5.5 Random Walk Based Techniques

Another family, possibly the largest, Sybil defence mechanism is based on random walks, first proposed in SybilGuard[89]. The key assumptions in these techniques is that the honest region is *fast mixing*[6], and the attack edges are difficult to form and are independent of the number of Sybils.

Before explaining the techniques, we define the terms *random walk* and *random route* in the context of social graphs. In random walk, the social graph is traversed such that outgoing edges are selected uniformly at random on every hop of the walk. Random route is a modified form of a random walk. Every node maintains a static routing table that contains a uniformly random one-to-one mapping between incoming edges and outgoing edges, initialised at start-up. Thus the route is determined by the tables on every node. An important property of random route is that if two routes enter the same edge, then they will always exit at the same edge, so their route after exiting will be exactly the same. The number of hops for a random route (the mixing time) should be just right, so that the fast mixing property is achieved only in the honest region.

Every node acts as a verifier in SybilGuard[89] and performs a single random route of a fixed length, determined by the mixing time. The verifier treats every other nodes as suspects initially. The suspect is labeled as an honest node if its random route intersects with the verificer's random route. The number accepted nodes for every intersection is limited by a quota. The process is visualsed in Figure 6. Intuitively, the random route from an honest node is unlikely to escape into the Sybil region because the number of attack edges is limited. Due to the random route property, the number of overlapping random routes from the Sybils is bounded by the number of attack edges. Recall that the number of attack edges is independent of the number of Sybils, thus they are unlikely to intersect with many honest nodes.

SybilLimit[88] is the continuation of SybilGuard and it is an improvement on many fronts while keeping the same or better guarantees. In Sasm as before, every honest node acts as a verifier V and initially treats all other nodes as suspects S. The verification process begins by performing multiple independent random routes instead of a single one as in SybilGuard. V labels S as an honest node if and only if they share at least one tail (the final edge in the route). For each tail of V, there is a quota for the number of node that it labels. The authors prove that SybilLimit bounds the

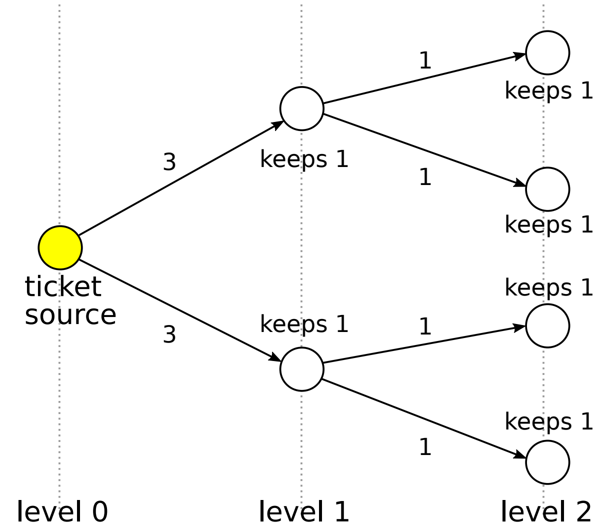

Figure 5: Visualisation of the ticket distribution step of SumUp. Note that tickets are not distributed to nodes on the same level of breadth-first search, or nodes that already hold a ticket. The same process is also applied for GateKeeper, discussed in at the end of subsection 5.5.

[5]The sparse cut problem is to find a partition such that the ratio between the number in the cut and the number of vertices in the smaller partition is minimised. This problem is related to minimum cut.

[6]In a graph, if a random walk of length $O(\log N)$ reaches a stationary distribution of nodes, then the graph is fast mixing.

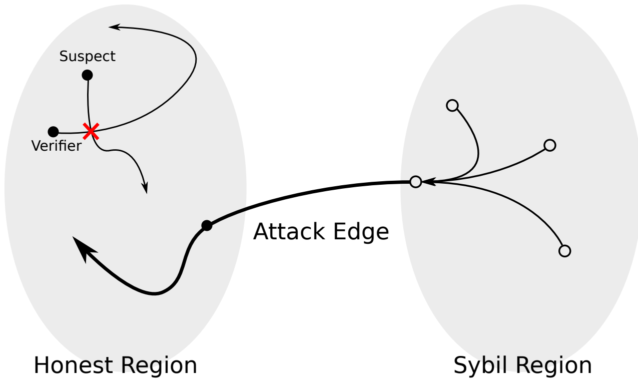

Figure 6: Visualisation of SybilGuard. The verifier accepts the suspect because their random routes intersect. The Sybils' random routes all come from a single attack edges, thus their routes are equivalent after entering the honest region.

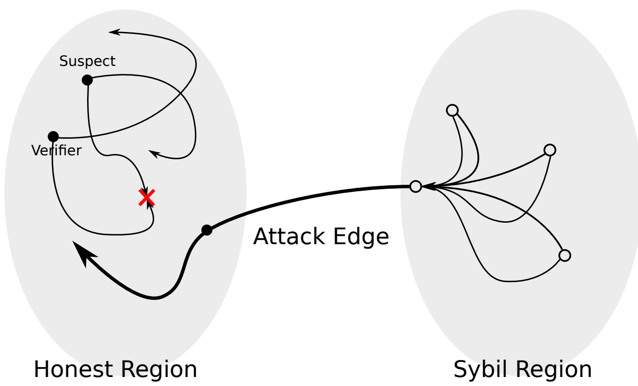

Figure 7: Visualisation of SybilLimit. The verifier accepts the suspect because one of the tails of their random routes meet. Similarly, the Sybils' random route are all from a single attack edges, so their routes are equivalent in the honest region.

number of accepted Sybils (false positives) at $O(\log n)$, an improvement from $O(\sqrt{n}\log n)$ of SybilGuard. The process is visualised in Figure 7.

Let us consider the following three scenarios to intuitively show why SybilLimit works. The same intuition applies to SybilGuard. Suppose S is not a Sybil, and if V and S perform enough random routes, each with enough hops for fast mixing, then due to the Birthday Paradox, S and V will have an intersecting tail with high probability. Next, suppose some of the tails of V are in the Sybil region so they may intersect with a large number of Sybils, but crossing the attack edges is improbable and accepting a lot of Sybils is also difficult due to the aforementioned quota mechanism, thus V has a small probability of accepting a large number of Sybils. Finally, consider there is only one attack edge and suppose a Sybil has tails in the honest region, due to the random route property, the route of the Sybils in the honest region will be equivalent (overlapping), so accepting the Sybils in this scenario is also low due to the quota mechanism.

SybilGuard and SybilLimit inspired many other defence mechanisms. SybilInfer[14] assumes trusted nodes, which create traces by doing random walks in the graph. Based on the traces, a probability model that descries the likelihood a trace T was generated by a specific set of honest nodes X, i.e. $\Pr[T|X = \text{honest}]$. Then using Bayesian inference, $\Pr[X = \text{honest}|T]$ can be computed, that is effectively assigning a "score" to every node. Sybil are the nodes with a low "score". SybilInfer outperforms SybilLimit regarding the number of false positives, but its drawbacks are its high computational cost and reliance on trusted nodes.

SybilDefender[85] can be seen as a two step process. It assumes the size of the Sybil region is smaller than the honest region and the nodes in the Sybil region are well connected. The first step is to perform random walk to detect Sybils. The second step is to detect a complete Sybil region around the detected Sybils. The algorithm in the second step employs *partial* random walk, where the random walk is not allowed to traverse the same node more than once. The property of the partial random walks is that they are likely to "die" (all the neighbour nodes have already been traversed) upon reaching the edge of the Sybil region, thus they are likely to stay in the Sybil region. The Sybil region is detected by examining the nodes traversed by the partial random walk.

SybilRank, in contrast of the aforementioned techniques, is designed to be integrated with real-world OSN and is deployed on Tuenti (an OSN with 11 million users)[10]. SybilRank uses short random walks that begins on trusted nodes in the honest region. The trusted nodes is choosen manually, this allows SybilRank to adapt to different graph structures. A novelty in SybilRank is that it uses power iterations, an efficient technique for computing the landing probability of random walks. Intuitively, the landing probability decreases for nodes that are far away from the trusted nodes (since it is using short random walks), especially for nodes in the Sybil region. The probabilities are normalised by the degree of the node and then ranked. The potential Sybils are the nodes that are under a some threshold. Finally, various actions can be performed to to verify the potential Sybils, e.g. using CAPTCHA puzzles.

SybilShield[68] makes use of multiple communities. It begins the same way as SybilGuard/SybilLimit, i.e. V performs random route to determine whether suspect S is a Sybil. But to reduce the possibility that S is in fact an honest node but labeled as a Sybil, V searches for agents A that are from another community. This is also done using random routes and relies on the assumption that inter-community edges are rare. First, V performs a random route and picks a *candidate* A, then V and the candidate perform random routes simultaneously, if they do not intersect then A is considered to be in another community, otherwise V repeats the process until it finds a suitable A. When a number of suitable A is found, they all perform random route and decides whether S is actually a Sybil and then relay the information back to V. If a large majority of A say S is honest, then V knows that it has made a mistake, otherwise S is indeed a Sybil.

GateKeeper[78] combines ideas from SumUp (discussed in subsection 5.4) and SybilLimit. GateKeeper assumes a admission controller that is honest, the admission controller performs random walks to select n ticket sources. The ticket sources act the same way as SumUp where it distributes ticket in a breadth-first search manner. For a node to be

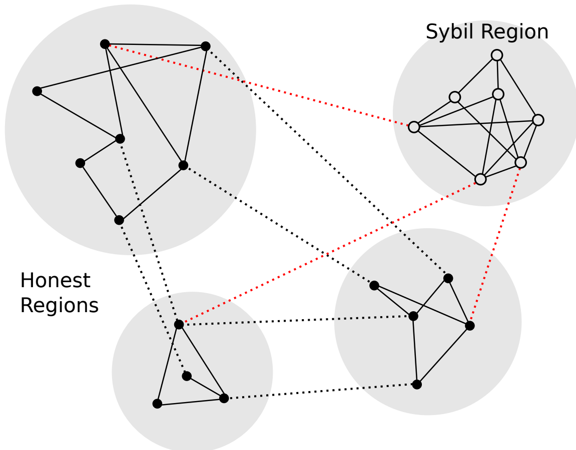

Figure 8: Visualisation of communities in a social graph. Solid lines represent intra-community edges. Dotted lines represent inter-community edges. The red dotted lines are inter-community edges coming from the Sybils, in other words the attack edges. The idea of making use of communities is used in SybilShield (this section) as well as SybilExposer (subsection 5.6).

labeled honest, it must obtain fn tickets, where f is a system parameter (0.2 is shown to be a good value experimentally). This idea works becauses if the ticket sources are evenly distributed and Sybils ony have a few attack edges, then it is unlikely that they will receive a large number of tickets.

5.6 Community Detection

In thi section we discuss techniques that leverage existing community detection algorithm. The work started with Viswanath et al., who realised many of the mechanisms mentioned in subsection 5.5 such as SybilGuard, SybilLimit and SybilInfer, are in fact performing local community detection (i.e. detecting clusters of nodes) which is a more developed field[80]. The authors also argue that social graphs are not always fast mixing, which may result in poor results for techniques that uses the fast mixing assumption. Using synthetic social graphs, the authors show that applying Mislove's algorithm[51] achieve similar results as SybilLimit and SybilInfer. But using a Facebook social graph, Mislove's algorithm performs better.

The authors of SybilExposer[52] argue that the number of attack edges may not be that small which may render the random walk based method ineffective. They proposed a solution that relies on the ratio between the number of inter-community (inter-cluster) edges and the number of intra-community (intra-cluster) edges. The idea is that this ratio is different between honest communities and Sybil communities, namely the Sybil communities have a lower ratio because they are well connected between themselves but not with other honest communities. SybilExposer operates in two stages, first communities are extracted using community detection algorithm (a modified version of the Louvain method[6]), then the communities are ranked based on the ratio and communities with a low ratio are likely to be Sybils.

5.7 Content Based and Machine Learning

In some application domains such as OSN it is possible to leverage content or user feedback to detect Sybils. These techniques work well in practice. But often depend on uninformed attackers that do not try to mimic the behaviour of honest nodes.

Ostra[50] is a system for limiting spam in social networks. In the simplest form, every undirected edge in the social graph is considered as two directed edges, each of them has a credit values. When a user wants to send a message, it needs to find a path with enough credits in the social graph from itself to the receiver. The edge traversed by the message will have its credit deducted, and the opposite edge will have its credit added. The receiver then decides whether the message is a spam, if it's not a spam then the credit operations are reversed. Effectively, only spam messages will have an effect on the credits. If a path cannot be found, i.e. all possible paths have run out of credit, then the message is blocked. Naturally, spams from the Sybils must use the attack edges, if enough honest users mark those messages as spam then the credit on the attack edges will run out and the Sybils can no longer send messages.

Stringhini et el. devised a machine learning technique to classify bot accounts in Twitter[75]. User feedback is incoperated into the features. For example one of features is "FF Ratio", that is the ratio between number of users that the account is following and the number of followers. Honest users typically do not follow bots and this can be considered as a form of feedback. Other features include "URL Ratio", "Message Similarity" and so on. The authors collected data on "honey-profiles", trained a classifier after analysing those data and collaborated with Twitter to delete tens of thousands of spam accounts.

VoteTrust[86] leverages the distrust relationship, i.e. friend request rejection, to detect Sybils in OSN. Suppose A sends a friend request to B, if B accepts/rejects the request then it is considered as a positive/negative vote on A by B. The first step is to use PageRank combined with human scrutiny to select a number of trust seeds in the honest region. Then the trusted seeds distributes *vote capacity*, that is the number of votes each node can cast. Initially only the trusted seeds have a positive vote capacity and other nodes have 0. When a node receives a positive vote from a trusted seed, it also receives some vote capacity. Then it can repeat the same process on nodes it votes on, thus distributing the vote capacity. The vote capacity decreases as it goes further away from the trusted seeds. This technique is comparable to the ticket distribution technique used in SumUp and GateKeeper. Finally, the votes are aggregated to compute a global ration for every node in the graph. Naturally, the Sybils are likely to have a low vote because their vote capacity is slow and many of their friend requests would be rejected.

Integro[7] is a hybrid between random walk and content based approaches. It begin by training a machine learning algorithm (random forest learning algorithm) to identify potential *victim accounts*, that is honest accounts that have accepted Sybils as their friends. Then in the social graph, edges connecting the potential victim accounts will have its weight reduced depending on the likelihood of it being a victim. Finally, a biased short random walk is perfrom starting from some known honest account and the landing probabilities are calculated for every node. Biased in a sense that

that the walk is a higher probability of using a path with a higher weight. Sybils are the nodes with a low landing probabilty. This technique works because, victims are easier to detect than Sybils due to the fact that Sybils can arbitrarily modify their account information to avoid detection. Once the victims are detected, they effectively form a "border-line" between the honest region and the Sybil region. Finally, it is unlikely that the random walk will traverse into the Sybil region due to its bias, so the Sybil will have a low landing probability and be detected.

5.8 Other

Trust transfer[64] is a Sybil defence mechanism for reputation systems that transfers the reputation score from a recommender to a recommended identity. This method discourages self-recommendation behaviour because the attacker would need to lower the reputation of its Sybils to recommend him or herself. The Sybils cannot gain reputation from honest identities because if they do not interact with them. It may be strange to lose reputation when recommending an identity, but the authors argue that in certain scenarios where there are a lot of interactions and the overall trustworthieness is high, then there is no major effect to transfer a little bit of reputation to a recommended identity.

Yu et el. of DSybil[90] argue that defending Sybils in reputation or recommendation systems is a lot more difficult than in social networks because only a very small percentage of the user will vote for an object (e.g. news article in Reddit), so a small number of Sybils and attack edges can easily out-vote honest users. Their proposed solution is DSybil, a distributed algorithm for diminishing the influence of Sybils in recommendation systems using historical data. Suppose Alice is an identity that runs the algorithm, and every identity begins with the same trust score from Alice's perspective. The algorithm runs in rounds. In every round, Alice picks an object to consume (e.g. reads the new article on Reddit) and then makes a binary (good or bad) feedback on the object. Then Alice computes whether the object is *overwhelming*, namely whether the sum of the trust scores of the voters of the object exceeds some threshold. If Alice voted for good and the object is not overwhelming, then she would increase the trust scores by a factor for all the voters of that object. Otherwise she decreases the trust score by a factor. When Alice needs a recommendation, a uniformly random overwhelming object is returned. Trust scores for identities that have the same interest as Alice grow exponentially when Alice consumes a good non-overwhelming object. Conversely the trust scores decreases exponentially for identities that are recommending bad objects, making Sybils ineffective.

SyMon[36] or *Sybil Monitor* assumes that any two sufficiently random nodes in the network cannot both be Sybils with a high probability. Then the nodes are paired together to monitor each other's transactions. For instance, a transaction could be reporting the number bytes transferred in a P2P file file sharing network. Cheating occurs when a node reports some bytes transferred that does not match its network traffic. The authors provide four methods for pairing nodes. Suppose the nodes are identified by a cryptographically secure hash of their RSA public key, then it is difficult to create identities deterministically. Then nodes can be matched by the closeness of their identities when they are in a DHT. The downside of this approach is that it sacrifices a lot of privacy, every action that a node makes is monitored by some other node.

6. RELATED WORK

Reputation Surveys: [47] [35] ? [30] [39] [65] ? [29]
Sybil Surveys: [42] [53] [59] [27] [38] Sok[1] but also some contribution
Other: [82] DHT seucrity survey[79]

7. SUMMARY

Temporal dynamics[44]

8. REFERENCES

[1] L. Alvisi, A. Clement, A. Epasto, S. Lattanzi, and A. Panconesi. Sok: The evolution of sybil defense via social networks. In *Security and Privacy (SP), 2013 IEEE Symposium on*, pages 382–396. IEEE, 2013.

[2] J. Aspnes, C. Jackson, and A. Krishnamurthy. Exposing computationally-challenged Byzantine impostors. *Department of Computer Science, Yale University, New Haven, CT, Tech. Rep*, 2005.

[3] S. Ba and P. A. Pavlou. Evidence of the effect of trust building technology in electronic markets: Price premiums and buyer behavior. *MIS quarterly*, pages 243–268, 2002.

[4] R. Bhattacharjee and A. Goel. Avoiding ballot stuffing in ebay-like reputation systems. In *Proceedings of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems*, pages 133–137. ACM, 2005.

[5] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda. All your contacts are belong to us: automated identity theft attacks on social networks. In *Proceedings of the 18th international conference on World wide web*, pages 551–560. ACM, 2009.

[6] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[7] Y. Boshmaf, D. Logothetis, G. Siganos, J. Lería, J. Lorenzo, M. Ripeanu, and K. Beznosov. Integro: Leveraging Victim Prediction for Robust Fake Account Detection in OSNs. In *NDSS*, volume 15, pages 8–11, 2015.

[8] Y. Boshmaf, I. Muslukhov, K. Beznosov, and M. Ripeanu. The socialbot network: when bots socialize for fame and money. In *Proceedings of the 27th Annual Computer Security Applications Conference*, pages 93–102. ACM, 2011.

[9] S. Buchegger and J.-Y. Le Boudec. A robust reputation system for mobile ad-hoc networks. Technical report, 2003.

[10] Q. Cao, M. Sirivianos, X. Yang, and T. Pregueiro. Aiding the detection of fake accounts in large scale social online services. In *Presented as part of the 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*, pages 197–210, 2012.

[11] A. Cheng and E. Friedman. Sybilproof reputation mechanisms. In *Proceedings of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems*, pages 128–132. ACM, 2005.

[12] B. Cohen. Incentives build robustness in BitTorrent. In *Workshop on Economics of Peer-to-Peer systems*, volume 6, pages 68–72, 2003.

[13] B. Cohen. Bep 3: The bittorrent protocol specification. http://www.bittorrent.org/beps/bep_0003.html, Jan 2008. Accessed: 2016-10-20.

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

[15] E. De Cristofaro, A. Friedman, G. Jourjon, M. A. Kaafar, and M. Z. Shafiq. Paying for likes?: Understanding facebook like fraud using honeypots. In *Proceedings of the 2014 Conference on Internet Measurement Conference*, pages 129–136. ACM, 2014.

[16] R. Delaviz, N. Andrade, J. A. Pouwelse, and D. H. Epema. SybilRes: A sybil-resilient flow-based decentralized reputation mechanism. In *Distributed Computing Systems (ICDCS), 2012 IEEE 32nd International Conference on*, pages 203–213. IEEE, 2012.

[17] J. Dinger and H. Hartenstein. Defending the sybil attack in p2p networks: Taxonomy, challenges, and a proposal for self-registration. In *First International Conference on Availability, Reliability and Security (ARES'06)*, pages 8–pp. IEEE, 2006.

[18] F. Dini and G. Spagnolo. Buying reputation on eBay: Do recent changes help? *International Journal of Electronic Business*, 7(6):581–598, 2009.

[19] J. R. Douceur. The sybil attack. In *International Workshop on Peer-to-Peer Systems*, pages 251–260. Springer, 2002.

[20] K. El Defrawy, M. Gjoka, and A. Markopoulou. BotTorrent: Misusing BitTorrent to Launch DDoS Attacks. *SRUTI*, 7:1–6, 2007.

[21] S. Farooqi, M. Ikram, G. Irfan, E. De Cristofaro, A. Friedman, G. Jourjon, M. A. Kaafar, M. Z. Shafiq, and F. Zaffar. Characterizing Seller-Driven Black-Hat Marketplaces. *arXiv preprint arXiv:1505.01637*, 2015.

[22] M. Feldman, K. Lai, I. Stoica, and J. Chuang. Robust incentive techniques for peer-to-peer networks. In *Proceedings of the 5th ACM conference on Electronic commerce*, pages 102–111. ACM, 2004.

[23] M. J. Freedman and R. Morris. Tarzan: A peer-to-peer anonymizing network layer. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 193–206. ACM, 2002.

[24] S. Ganeriwal, L. K. Balzano, and M. B. Srivastava. Reputation-based framework for high integrity sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 4(3):15, 2008.

[25] C. Grier, K. Thomas, V. Paxson, and M. Zhang. @ spam: the underground on 140 characters or less. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 27–37. ACM, 2010.

[26] D. Guilbeault and S. Woolley. How twitter bots are shaping the election. http://www.theatlantic.com/technology/archive/2016/11/election-bots/506072/, 11 2016.

[27] R. Gunturu. Survey of Sybil attacks in social networks. *arXiv preprint arXiv:1504.05522*, 2015.

[28] Help Net Security. Twitter accounts spreading malicious code. https://www.helpnetsecurity.com/2010/12/03/twitter-accounts-spreading-malicious-code/, 12 2010. Accessed: 2016-11-2.

[29] F. Hendrikx, K. Bubendorfer, and R. Chard. Reputation systems: A survey and taxonomy. *Journal of Parallel and Distributed Computing*, 75:184–197, 2015.

[30] K. Hoffman, D. Zage, and C. Nita-Rotaru. A survey of attack and defense techniques for reputation systems. *ACM Computing Surveys (CSUR)*, 42(1):1, 2009.

[31] T. Holz, M. Steiner, F. Dahl, E. Biersack, and F. C. Freiling. Measurements and Mitigation of Peer-to-Peer-based Botnets: A Case Study on Storm Worm. *LEET*, 8(1):1–9, 2008.

[32] D. Houser and J. Wooders. Reputation in auctions: Theory, and evidence from eBay. *Journal of Economics & Management Strategy*, 15(2):353–369, 2006.

[33] A. A. Irissappane, S. Jiang, and J. Zhang. Towards a comprehensive testbed to evaluate the robustness of reputation systems against unfair rating attack. In *UMAP Workshops*, volume 12, 2012.

[34] J. Jiang, Z.-F. Shan, X. Wang, L. Zhang, and Y.-F. Dai. Understanding Sybil Groups in the Wild. *Journal of Computer Science and Technology*, 30(6):1344–1357, 2015.

[35] A. Jøsang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision support systems*, 43(2):618–644, 2007.

[36] B. Jyothi and J. Dharanipragada. Symon: Defending large structured p2p systems against sybil attack. In *2009 IEEE Ninth International Conference on Peer-to-Peer Computing*, pages 21–30. IEEE, 2009.

[37] C. Karlof and D. Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. *Ad hoc networks*, 1(2):293–315, 2003.

[38] D. Koll, J. Li, J. Stein, and X. Fu. On the state of OSN-based Sybil defenses. In *Networking Conference, 2014 IFIP*, pages 1–9. IEEE, 2014.

[39] E. Koutrouli and A. Tsalgatidou. Taxonomy of attacks and defense mechanisms in P2P reputation systems-Lessons for reputation system designers. *Computer Science Review*, 6(2):47–70, 2012.

[40] A. Kurve and G. Kesidis. Sybil detection via distributed sparse cut monitoring. In *2011 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2011.

[41] L. Lamport, R. Shostak, and M. Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 4(3):382–401, 1982.

[42] B. N. Levine, C. Shields, and N. B. Margolin. A survey of solutions to the sybil attack. *University of Massachusetts Amherst, Amherst, MA*, 7, 2006.

[43] J. Liang, N. Naoumov, and K. W. Ross. The Index Poisoning Attack in P2P File Sharing Systems. In *INFOCOM*, pages 1–12. Citeseer, 2006.

[44] C. Liu, P. Gao, M. Wright, and P. Mittal. Exploiting temporal dynamics in Sybil defenses. In *Proceedings of*

the 22nd ACM SIGSAC Conference on Computer and Communications Security, pages 805–816. ACM, 2015.

[45] A. Loewenstern and A. Norberg. Bep 5: Dht protocol. http://www.bittorrent.org/beps/bep_0005.html, Jan 2008. Accessed: 2016-10-20.

[46] N. B. Margolin and B. N. Levine. Quantifying resistance to the sybil attack. In *International Conference on Financial Cryptography and Data Security*, pages 1–15. Springer, 2008.

[47] S. Marti and H. Garcia-Molina. Taxonomy of trust: Categorizing P2P reputation systems. *Computer Networks*, 50(4):472–484, 2006.

[48] P. Maymounkov and D. Mazieres. Kademlia: A peer-to-peer information system based on the xor metric. In *International Workshop on Peer-to-Peer Systems*, pages 53–65. Springer, 2002.

[49] M. Meulpolder, J. A. Pouwelse, D. H. Epema, and H. J. Sips. Bartercast: A practical approach to prevent lazy freeriding in p2p networks. In *Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*, pages 1–8. IEEE, 2009.

[50] A. Mislove, A. Post, P. Druschel, and P. K. Gummadi. Ostra: Leveraging Trust to Thwart Unwanted Communication. In *NSDI*, volume 8, pages 15–30, 2008.

[51] A. Mislove, B. Viswanath, K. P. Gummadi, and P. Druschel. You are who you know: inferring user profiles in online social networks. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 251–260. ACM, 2010.

[52] S. Misra, A. S. M. Tayeen, and W. Xu. SybilExposer: An effective scheme to detect Sybil communities in online social networks. In *2016 IEEE International Conference on Communications (ICC)*, pages 1–6, May 2016.

[53] A. Mohaisen and J. Kim. The Sybil attacks and defenses: a survey. *arXiv preprint arXiv:1312.6349*, 2013.

[54] E. Mustafaraj and P. T. Metaxas. From obscurity to prominence in minutes: Political speech and real-time search. 2010.

[55] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

[56] J. Newsome, E. Shi, D. Song, and A. Perrig. The sybil attack in sensor networks: analysis & defenses. In *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 259–268. ACM, 2004.

[57] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: bringing order to the web. 1999.

[58] E. Protalinski. Facebook: 5-6% of accounts are fake. https://web.archive.org/web/20160422121639/http://www.zdnet.com/article/facebook-5-6-of-accounts-are-fake/, 2012. Accessed: 2016-10-20.

[59] G. Rakesh, S. Rangaswamy, V. Hegde, and G. Shoba. A survey of techniques to defend against sybil attacks in social networks. *International Journal of Advanced Research in Computer and Communication Engineering*, 3(5), 2014.

[60] J. Ratkiewicz, M. Conover, M. Meiss, B. Gonçalves, S. Patil, A. Flammini, and F. Menczer. Truthy: mapping the spread of astroturf in microblog streams. In *Proceedings of the 20th international conference companion on World wide web*, pages 249–252. ACM, 2011.

[61] P. Resnick et al. The social cost of cheap pseudonyms. *Journal of Economics & Management Strategy*, 10(2):173–199, 2001.

[62] P. Resnick, R. Zeckhauser, J. Swanson, and K. Lockwood. The value of reputation on eBay: A controlled experiment. *Experimental economics*, 9(2):79–101, 2006.

[63] SadBotTrue. Chapter 32. the stealth botnet, 6 2016. Accessed: 2016-11-2.

[64] J.-M. Seigneur, A. Gray, and C. D. Jensen. Trust transfer: Encouraging self-recommendations without sybil attack. In *International Conference on Trust Management*, pages 321–337. Springer, 2005.

[65] C. Selvaraj and S. Anand. A survey on security issues of reputation management systems for peer-to-peer networks. *Computer Science Review*, 6(4):145–160, 2012.

[66] S. Seuken and D. C. Parkes. On the Sybil-proofness of accounting mechanisms. 2011.

[67] S. Seuken and D. C. Parkes. Sybil-proof accounting mechanisms with transitive trust. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 205–212. International Foundation for Autonomous Agents and Multiagent Systems, 2014.

[68] L. Shi, S. Yu, W. Lou, and Y. T. Hou. Sybilshield: An agent-aided social network-based sybil defense among multiple communities. In *INFOCOM, 2013 Proceedings IEEE*, pages 1034–1042. IEEE, 2013.

[69] K. C. Sia. DDoS vulnerability analysis of BitTorrent protocol. *UCLA: Techical Report*, 2006.

[70] A. Singh et al. Eclipse attacks on overlay networks: Threats and defenses. In *In IEEE INFOCOM*. Citeseer, 2006.

[71] Socialpuncher. How many primitive bots follow top-100? http://socialpuncher.com/top-100/how-many-primitive-bots-follow-top-100/, 9 2016. Accessed: 2016-11-2.

[72] Sophos. Sophos facebook id probe shows 41% of users happy to reveal all to potential identity thieves. https://web.archive.org/web/20140926063331/http://www.sophos.com/en-us/press-office/press-releases/2007/08/facebook.aspx, 2007. Accessed: 2016-10-30.

[73] T. Stein, E. Chen, and K. Mangla. Facebook immune system. In *Proceedings of the 4th Workshop on Social Network Systems*, page 8. ACM, 2011.

[74] M. Steiner, T. En-Najjary, and E. W. Biersack. Exploiting KAD: possible uses and misuses. *ACM SIGCOMM Computer Communication Review*, 37(5):65–70, 2007.

[75] G. Stringhini, C. Kruegel, and G. Vigna. Detecting spammers on social networks. In *Proceedings of the 26th Annual Computer Security Applications Conference*, pages 1–9. ACM, 2010.

[76] D. Tamir. Twitter malware: Spreading more than just

ideas. https://securityintelligence.com/twitter-malware-spreading-more-than-just-ideas/, 4 2013. Accessed: 2016-11-2.

[77] D. N. Tran, B. Min, J. Li, and L. Subramanian. Sybil-Resilient Online Content Voting. In *NSDI*, volume 9, pages 15–28, 2009.

[78] N. Tran, J. Li, L. Subramanian, and S. S. Chow. Optimal sybil-resilient node admission control. In *INFOCOM, 2011 Proceedings IEEE*, pages 3218–3226. IEEE, 2011.

[79] G. Urdaneta, G. Pierre, and M. V. Steen. A survey of DHT security techniques. *ACM Computing Surveys (CSUR)*, 43(2):8, 2011.

[80] B. Viswanath, A. Post, K. P. Gummadi, and A. Mislove. An analysis of social network-based sybil defenses. *ACM SIGCOMM Computer Communication Review*, 40(4):363–374, 2010.

[81] L. Von Ahn, M. Blum, N. J. Hopper, and J. Langford. Captcha: Using hard ai problems for security. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 294–311. Springer, 2003.

[82] D. S. Wallach. A survey of peer-to-peer security issues. In *Software Security-Theories and Systems*, pages 42–57. Springer, 2003.

[83] G. Wang, C. Wilson, X. Zhao, Y. Zhu, M. Mohanlal, H. Zheng, and B. Y. Zhao. Serf and turf: crowdturfing for fun and profit. In *Proceedings of the 21st international conference on World Wide Web*, pages 679–688. ACM, 2012.

[84] L. Wang and J. Kangasharju. Real-world sybil attacks in BitTorrent mainline DHT. In *Global Communications Conference (GLOBECOM), 2012 IEEE*, pages 826–832. IEEE, 2012.

[85] W. Wei, F. Xu, C. C. Tan, and Q. Li. Sybildefender: Defend against sybil attacks in large social networks. In *INFOCOM, 2012 Proceedings IEEE*, pages 1951–1959. IEEE, 2012.

[86] J. Xue, Z. Yang, X. Yang, X. Wang, L. Chen, and Y. Dai. Votetrust: Leveraging friend invitation graph to defend against social network sybils. In *INFOCOM, 2013 Proceedings IEEE*, pages 2400–2408. IEEE, 2013.

[87] C. Yang, R. Harkreader, J. Zhang, S. Shin, and G. Gu. Analyzing spammers' social networks for fun and profit: a case study of cyber criminal ecosystem on twitter. In *Proceedings of the 21st international conference on World Wide Web*, pages 71–80. ACM, 2012.

[88] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao. Sybillimit: A near-optimal social network defense against sybil attacks. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 3–17. IEEE, 2008.

[89] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. Sybilguard: defending against sybil attacks via social networks. In *ACM SIGCOMM Computer Communication Review*, volume 36, pages 267–278. ACM, 2006.

[90] H. Yu, C. Shi, M. Kaminsky, P. B. Gibbons, and F. Xiao. Dsybil: Optimal sybil-resistance for recommendation systems. In *2009 30th IEEE Symposium on Security and Privacy*, pages 283–298. IEEE, 2009.