The Sybil Attack—Theory and Practice

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

ABSTRACT

The sybil attack is an attack where an entity can assume multiple identities. It is one of the most important attacks because it leads to a large number of consequences such as identity theft and astroturfing. In this work, we survey the practical and the theoretical aspects of the sybil attack. On the practical side, we demonstrate the severity of the sybil attack using real-world examples, it includes an experimental study to gain first-hand knowledge of properties of the sybils. Furthermore, we describe five of the application domains that are vulnerable to the sybil attack—online social networks, file sharing networks, reputation systems, wireless ad-hoc networks and the Bitcoin network. The theoretical side of this work describe the defence mechanisms. We describe their underlying principals and categorise them according to the main idea. From our findings, we realise the sybil attack is still an unsolved problem. Many of the defence mechanisms do not generalise well into scenarios in the real-world, and they break down when their assumptions are not satisfied. We hope this work serves as a cornerstone for the future sybil defence mechanisms

1. INTRODUCTION

Electronic commerce and online social networks allow us to orchestrate many aspects of our lives in the comfort of our homes, behind the monitors of our devices. With such services, online identity is often required. For example in Twitter[1], users must create accounts to tweet a friend, who must also have an account. In this scenario, users can choose to remain pseudonymous if they are careful, where their real-life identity is uncorrelated with their online identity. While that is useful for protecting the users' privacy, it also opens an alleyway for attackers.

The sybil attack, first described by Douceur[25], is an attack where an entity can assume multiple identities or sybils[2], and then attack either another entity or undermine the whole system. For example, a malicious Twitter user can create many fake identities and have them follow his real identity, thus creating a false reputation. It is one of the most important attacks because it leads to numerous consequences including but not limited to spreading false information, identity theft[6] and ballot stuffing[5]. Furthermore, to the best of our knowledge, there is no general solution for preventing the sybil attack.

There has been over a decade of work on the sybil attack from both perspective—attackers and defenders. Many previous surveys focus only on the defender's perspective or a particular class of defence mechanisms. The purpose of this work is to survey and provide a broad view on all aspects of the sybil attack. First, we illustrate the importance of the sybil attack by looking at how researchers and black-hat hackers mounted the attack in the real-world; we also experiment with fake Twitter followers and look at their properties (section 2). Then we define the sybil attack give some important theoretical results (section 3). With the fundamentals in place, we describe the various types of attacks (section 4) and defence mechanisms (section 5). 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". Finally, we present the related work and conclude in section 6 and section 7.

2. REAL-WORLD ATTACKS

We begin our survey by showing some alarming sybil attacks happening in the real-world. Social network and microblogging websites are popular platforms for organisations to improve public relations and their reputation, but they are also platforms to spread propaganda. A recent article in the Atlantic described how Twitter bots (sybils) are shaping the 2016 US presidential election[33]. 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 the 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 false 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[3] likes, Twitter follow-

[1]Twitter (`https://twitter.com/`) is a social network website where users can send and read short messages—called "tweets", users can also "follow" each other.

[2]In this work we use "sybil" as a noun to mean a fake identity.

[3]`https://www.facebook.com/`

DOI: 10.475/123_4

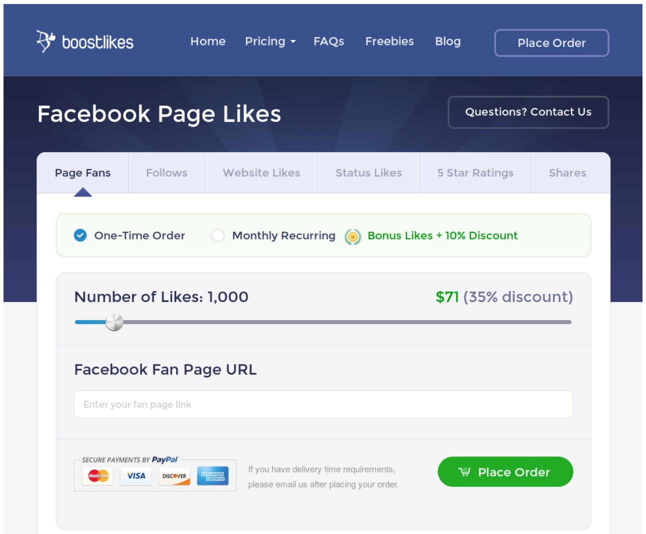

Figure 1: Screenshot of the Facebook likes service page of boostlikes.com.

ers, Instagram[4] followers and YouTube[5] 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 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 and the obliviousness of Twitter. Firstly, there exist a botnet that consist of 3 million accounts. Since their creation, they generated 2.6 billion tweets. Surprisingly, all the 3 million accounts were created on the same day—22/10/2013, and the account names are simply numbered sequentially[75]. 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%[83]. 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 adequate to combat the sybil attack. If the sybils infiltrate even more of our cyberspace, then it may become a form of censorship. Effectively taking away our right to freedom of speech.

Speaking of censorship, around a million [68] people use Tor (The Onion Router) [24] to access the uncensored internet when living in authoritarian regimes such as China, or uphold their privacy from illegal mass surveillance by intelligence agencies. Unfortunately, Tor suffered a sybil attack. In January 2014, 115 bogus relays joined the Tor network. Six months later, it was discovered that those relays were using a protocol vulnerability to deanonymise users and find the location of hidden services. It is unclear to the Tor developers which users are affected or what information was retrieved, thus it is assumed that users who used Tor between that period are all affected[23]. In fact, Tor depends on the fact that majority of the relays are good to guarantee anonymity with a high probability. If the network is infiltrated by numerous of sybils then users can be easily deanonymised.

These example demonstrate a big problem with in the popular social network websites and anonymous communication tool we use today. A lot of sybils controlled by an attacker can censor content and track user behaviour. In the next section, we zoom in on the practical attacks and grouped them by the underlying application.

2.1 Experiment

We crawled Twitter to gain first-hand knowledge on the real-world sybils. Twitter was selected because it has an easy-to-use API and it is one of the major targets of sybils as mentioned at the top of this section.

2.1.1 Setup

A Twitter account was created on 25th of November 2016. We purchased the "1,000 Followers" product from CoinCrack[6] for $9. We made sure our account has 0 tweets and is not advertised in any other medium, this guarantees that all of its followers are sybils. Followers started coming in almost immediately after we made our purchase. No more than 1 hour later, our brand new account with 0 tweets have accumulated 1,300 followers. To find their relationships, we crawled the followers of sybils for 72 hours using `tweetf0rm`[7] recursively for a depth of three—more than enough for our analysis below.

2.1.2 Results and Analysis

We obtained 3 million nodes and 6 million edges by the end of crawling period. We only consider the known sybils (the followers of our account) for two reasons. (1) There is no guarantee how many of those 3 million nodes are actually sybils. (2) Graph visualisation software cannot handle this volume of data under a reasonable amount of time and computational resources. Figure 3 shows the result visualised using Gephi [4]. In particular, we use the OpenOrd layout algorithm [56] to capture the overall structure of the graph. It structures the layout using hierarchical clustering so that we can clearly see the communities. We observe the following properties.

1. Many sybils are connected with each other, and many of them form a large community. This is most likely because the sybils need to be made to look like humans so that they can avoid detection. An account that follows thousands of users but has 0 followers would look suspicious.

2. There exist 4 "super sybils" (large purple nodes at the centre of the graph) and they each have over 1000 followers, so they are connected to almost all the other sybils. One of them is `@gf1av`[8], it also joined in November 2016 and has over 174K followers at the time of writing. We do not know the exact reason for this property, it may be another strategy to avoid detection.

[4] `https://www.instagram.com/`
[5] `https://www.youtube.com/`
[6] `https://coincrack.com/`
[7] `https://github.com/bianjiang/tweetf0rm`
[8] `https://twitter.com/gf1av`

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

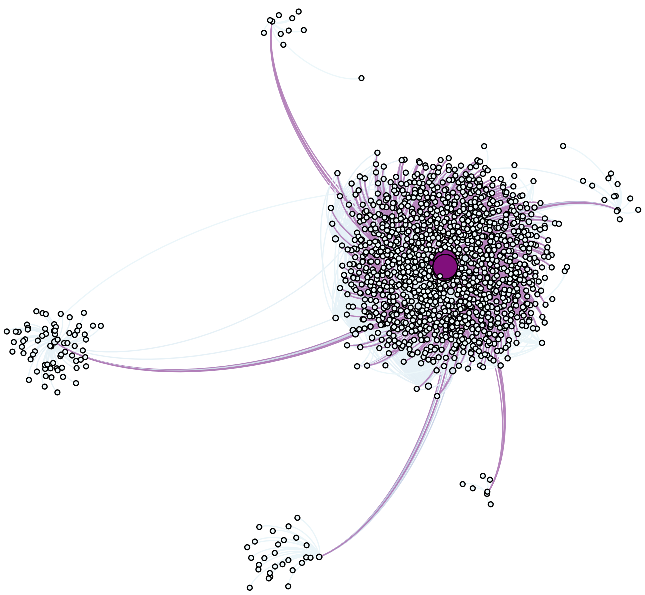

Figure 3: Visualisation of the relationships of the sybils. Dark/light nodes are nodes with a high/low degree.

3. The "1,000 Followers" service is the cheapest Twitter service, so the attacker has many more sybils than what is pictured. Hence, the smaller communities may be in fact parts of a larger community.

From the results we see the sybils do not have the same characteristics, and more importantly they form communities within themselves. In section 5, we describe how to leverage the community property to detect sybils or create sybil-resistant applications.

3. THE SYBIL ATTACK

The sybil attack is coined by Douceur[25] in 2002 in the context of P2P (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 [73]. 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[25]. 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[48]. In this work, we use identity, node and peer interchangeably.

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[14]. 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[67] is an example of a reputation system, where many 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 4. It is first introduced by the authors of SybilGuard[101]. 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. We call the nodes in the honest region that has one or more attack edges *gullible nodes* or *victim nodes*. 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.

A less common model is the aforementioned P2P network

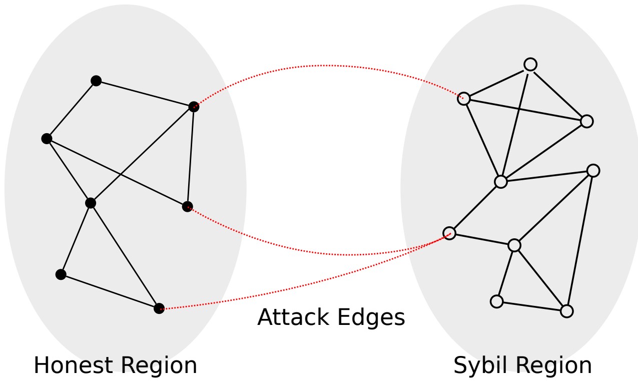

Figure 4: The model in many sybil defence mechanisms can be seen as a social graph that is partitioned into a sybil region and an honest region. The two regions are connected by *attack edges*. Note that in general there may be multiple honest regions multiple sybil region.

model [25]. It does not have a notion of friendship, so sybils are free to attach themselves with any node in the network. Only some techniques are able to defend against sybils under this model, we describe them in subsection 5.5.

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 five common applications. Some attacks are general, so we give them a general description when they first occur, and then give an application specific description under each subsection. We hope this section further illuminates the alarming consequences of the sybil attack.

4.1 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. 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 [84].

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 [69]. Jiang et el. analysed data from Renren[9], they discovered 2440 sybil groups which totals to about 1 million sybils [41].

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 pages with a lower number of fans. We discuss attacks specific to OSN in this section and attacks on reputation in subsection 4.3

[9]One of the largest social network in China (`http://renren.com/`).

4.1.1 *Identity Theft*

Identity theft is the intentional user of the victim's identity to gain an advantage. Authors of [6] 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. iCloner then sends 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 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 profile is cloned, iCloner attempts to identify friends of the victim and begins sending friend requests. Similarly, 56% of the friend requests were accepted.

More recently, Boshmaf et el. created SbN (Socialbot Network), which targets Facebook [11]. Each socialbot is a sybil created by the attacker, it controls a forged profile and mimics 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 (Facebook Immune System [86]) 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.

4.1.2 *Astroturfing*

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 campaigners 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 [63]. Ratkiewicz et el. suggest that this type of attack can be mounted cheaply and may have a larger influence than traditional advertising [72].

The Truthy system [72] 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 largest crowdturfing (crowdsourced astroturfing) platforms in China—Zhubajie and Sandaha. One of their services is to perform astroturfing on Weibo[10]. 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[94].

4.1.3 Spamming

Spamming, much like in the context of email, is the act of sending unsolicited or undesired messages (spam). The goal of the attacker varies from advertisement to phishing and spreading malicious software [36, 89]. Many studies have characterised the behaviour of the spammers and found that many spammers are in fact automated sybils [88, 98, 32, 41]. More importantly, spamming is possibly the most common attack in OSN. Jiang et el. analysed the malicious activities of the sybils on Renren and found propagating advertisement (at 32.8%) is the most common one [41]. 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.

4.2 The Sybil Attack in File Sharing Networks

File sharing networks are often P2P, where peers connect to other peers to download and/or upload files without a central server (for the most part). BitTorrent [17] is likely the most popular P2P network at the time of writing. Unlike social networks where there is a trust relationship between peers, P2P networks are much more open and distributed, making them much more vulnerable to the sybil attack.

P2P networks often implement a DHT (distributed hash table). The DHT in BitTorrent is called Mainline-DHT, based on Kademlia [57]. Keys are the infohashes (file identifiers) and values are the metadata of the files, these are distributed across all the participating peers. Every node also stores a routing table, the entries in the table is precise for the closer nodes and coarse for node that are further away. Requests to search for a key are routed iteratively, every iteration the request gets closer to the destination node. Finally, the value is returned if it is found [54].

4.2.1 Index Poisoning

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 in 2012 [95]. Similar attacks are possible in other P2P networks such as Overnet [51].

4.2.2 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 P2P networks Overnet and eMule [87]. The Eclipse Attack [82] is a special form of targeted sybil attack. Sybils are arranged in the network such that they *eclipse* the victim from the rest of the network. The victim can either be an identity or an object such as a key in DHT. The authors mounted the latter variation, and they show it is possible to eclipse a key using only 32 sybils in a DHT with 1.5 million users and 42,000 key.

4.2.3 Denial of Service

By using sybils and 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 [81]. 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 [26].

Steiner et al. also succeeded in mounting a DDoS attack but in the context of the aforementioned KAD DHT [87]. 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.

4.2.4 Spying

Many authors have used sybils to monitor a P2P file sharing network that uses DHT [38, 87], but such techniques can be directly applied to spy on users. In essence, the authors created a lot of light-weight sybils and tricked all the honest peers to store them in their routing table, a form of index poisoning. The sybils are light-weight because they do not follow the DHT protocol and perform much simpler operations, a single machine can have thousands of sybils running simultaneously. Finally, DHT requests would "traverse through" the sybils due to the poisoned routing table, and the requests are stored in a database for further analysis.

4.2.5 Whitewashing

Whitewashing is the act of joining the network under a new identity, usually for when the previous identity have acquired a lot of negative reports. It can be performed very easily if it is cheap to create new sybils. In file sharing networks that involve a central component for policing the users, whitewashing is often a problem. Many studies have found penalising selfish behaviours (free-riding) encourages whitewashing [29, 99].

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 contexts. In e-commerce, namely eBay, researchers found that the merchant's reputation "is a statistically and economically significant determinant of auction prices" [39], and "buyers are willing to pay 8.1% more"

[10]The Chinese Twitter (`http://weibo.com/`.)

for goods sold buy a reputable merchant [74]. 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 [16]. PageRank [67] is also a reputation system, used for ranking websites in Google's search results.

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-promotion is to create sybils and have them create positive reputation for the attacker's main identity.

De Cristofaro et el. performed an empirical study on Facebook page promotion using like-farms [20]. Some 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 mimic 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. A month later, the authors empty accounts were not terminated, only a few of the liker's account were terminated.

SEOClerks and MyCheapJobs are also evidences of marketplaces for self-promotion. Farooqi et el. found 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 [27]. 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. Many have suggested it is a feasible attack [37, 46], but from the best of our knowledge, there are no published studies on real-world slandering. On the other hand, a mailing list post by Mike Hearn suggest attackers would submit false reports on advertisements provided by AdWords to gain an advantage over their competitors [34]. Research has shown having a negative feedback may harm the target's ability to do business [2].

4.3.3 Whitewashing

In whitewashing for reputation systems, 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[37, 55].

4.4 The Sybil Attack in the Bitcoin Network

Bitcoin is a cryptocurrency. It uses a global ledger that is replicated across all nodes (miners). The ledger is created using blocks, chained together using hash pointers to create a notion of ordering. Each block contains transactions. Miners reach consensus using *proof-of-work* [64]. This section describes how the Bitcoin network suffers from the sybil attack. The consequences are fairly general and may apply to other blockchain networks that use a global ledger.

4.4.1 Eclipse Attack

Attacker can eclipse Bitcoin nodes to gain various unfair advantages [35]. Nodes receive unsolicited messages for IP addresses of other nodes, the IP addresses are stored in an internal table. The attacker essentially sends bogus messages to the target that only contains the IP address of the attacker's sybils or "rubbish" addresses. This technique eclipses the target because it can only connect to the attacker's IP address. There are many consequences. (1) Engineering block races, when two nodes discover the next block at the same time, the eclipsed block will not be able to receive the reward. (2) Eclipsing a large portion of the network can cause a 51% attack, where a single party gain complete control of the network. (3) Selfish-mining is when the attacker do not publish the latest block immediately after discovery but aims to publish 2 or more blocks at the same time to gain a lead; this results in many orphaned blocks for the other miners and the attacker can use the eclipse attack to drop newly discovered blocks by other miners to make the attack more effective. (4) Double spending, the eclipse attack allows the attacker to double spend his Bitcoins to eclipse nodes because they don't have an accurate view of the whole network.

4.4.2 Spamming

In Bitcoin, the spammer's goal is to waste system resources rather than advertisements such as in social networks. Block sizes is fixed to 1 MB and block generation takes about 10 minutes. Thus, there is an upper bound for the number of transactions that the Bitcoin network can handle per second (approximately 7 at the time of writing [7]). Flooding the network with useless transactions will cause its performance to drop. Decreasing the scalability even more. Fortunately, Bitcoin incorporates a transaction fee, so spammers cannot abuse the network indefinitely [8].

4.5 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 many nodes. Many WANET designs involve a reputation system[31, 12], 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.

4.5.1 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[66]. In contrast with the other attacks, this works even when the sybils are not behaving maliciously.

4.5.2 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 a single attacker, nullifying the advantages. Another technique is geographic routing, nodes route data depending on the geographic location of their neighbours. Sybils in the network can pretend to be in more than one place at a time, thus significant disrupting the routing algorithm[44].

4.5.3 Spreading False Information

Nodes often need to exchange information with each other to satisfy the underlying requirements of the application. Some common tasks are data aggregation and 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 ballot to detect misbehaving nodes, the attack could use its sybils to claim that an honest node is misbehaving and have the other nodes expel it from the network[66].

5. SYBIL DEFENCE MECHANISMS

The sybil attack clearly has a lot of consequences, some of them are very severe and may cause financial losses. In this section we categorise various defence techniques against the sybil-attack. We classify them on their main idea, and state explicitly when the mechanism is application specific.

5.1 Reputation Systems

Some reputation systems exhibit the ability to resist sybils. The canonical reputation system is PageRank [67]. In the context of world wide web, it assigns a score to every web page depending on the number of links pointing to it from other web pages. Suppose an attacker wants to boost the score of his web page, he would create sybil pages and create link to his main page. PageRank prevent this type of manipulation because the sybil pages do not have a high score so they cannot influence the attacker's page by a large amount [3]. On the other hand, the initial score for the sybil pages are not zero; Cheng and Friedman found that for about 300,000 pages with a median score of 0.3, it requires 500 sybils to rise a median node to the top 100 [15].

Many other reputation systems exist other than PageRank [43, 85, 65]. However, many of them do not factor the sybil attack into their design and suggest using an independent defence mechanism which we describe in the following sections.

5.2 Certificate Authority

CA (certificate authorities) check the users' real identities and then issues certificates to honest users. The certificate can be tangible (trusted hardware [66]) 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.

Many existing systems today use a form of CA. X.509 [40]—a standard for certificates, it is used in a large variety of applications. For example, emails can be encrypted and signed using S/MIME[71] certificates which are based on X.509. Attackers can still create many sybils and send emails, but the receiver would reject the emails because they are not correctly signed.

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 necessary CA. (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 identities.

5.3 Resource Testing

Resource testing makes attacks costly. That is to say every attacker can create multiple sybils, but the attack cannot duplicate its resources the same way. The resource type varies depending on the application, and we give a few examples 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 [30], 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 [22]. 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, and then broadcast it to the network. While participating in the network, 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 [1]. The puzzle in this case is computing hashes on some input y concatenated 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[11]. 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 Bitcoin [64]. The Bitcoin blockchain is a global ledger and it needs to reach a consensus for its blocks. Nodes in the Bitcoin network essentially "vote" for 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

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

difficult.

5.4 Registration Fee

Friedman and Resnick is one of the first to propose the use of a registration fee [73]. 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 needs to be set appropriately so that the cost of creating sybils outweighs 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[73]. It uses blind signatures and a central authority, the authority does not know the mapping between the real identity (e.g. driving licence) 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 [93] 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[28]. 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.

5.5 Network Flow

Network flow based techniques began with BarterCast[58]. 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.

The main idea comes from human interactions, where the reputation of a person can be from direct experiences, or information obtained from someone else. As we know, our direct experiences are 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 5, 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 the Ford-Fulkerson maximum flow algorithm [18] as shown in Figure 5, A only have 5 units of trust for S even when S apparently contributed a lot to B.

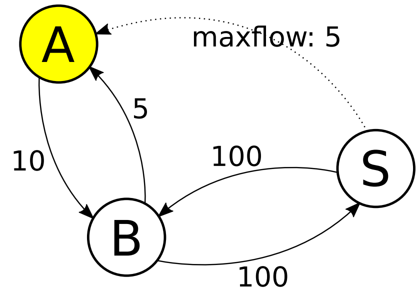

Figure 5: Subjective graph of A. The numbers are the amount of data transferred, they can be seen as the capacity in the context of the maximum flow problem.

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 [21]. 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 amount of data transferred after computing the reputation. Using the definition in Figure 4, the attacker cannot built up reputation for its sybils by uploading to peers in the honest region beforehand, it is now 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 [78, 79]. 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. Peers in C cannot misreport their contribution since all the necessary edges are dropped. The authors prove that it is robust against weakly beneficial sybils, that is sybils that do not perform actual work for honest peers. The authors also prove that no mechanism can prevent strongly beneficial sybils, i.e. sybils that interact with honest users.

SumUp[90] is a defence mechanism specific for the vote aggregation problem. For example, in social news aggregation websites such as Reddit[12], 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,

[12] `https://www.reddit.com/`

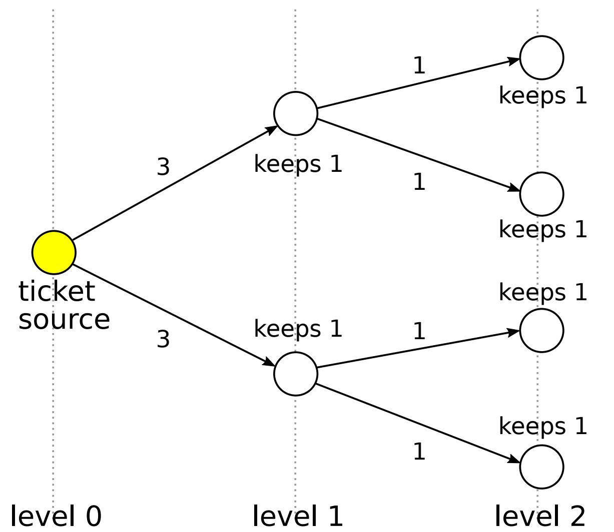

Figure 6: Visualisation of the ticket distribution step of SumUp. Note that tickets are not distributed to nodes on the same level of breadth-first search, or nodes that already hold a ticket. The same process is also applied for GateKeeper, discussed in at the end of subsection 5.6.

the threshold for the number of edges to prune is a system parameter. Secondly, it uses a ticket source (the central component) to distribute 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 (subsection 5.6) where it only accepts $1 + o(n)$ votes per attack edge. Unlike the aforementioned techniques in this section, SumUp requires a social network so it does not work in a generic P2P setting. An improved version of SumUp - GateKeeper is discussed in subsection 5.6.

Conversely, maximum flow is dual to minimum cut, so the problem of finding sybil can also be formulated as finding sparse cuts. 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. Kurve and Kesidis devised an algorithm for finding sparse cuts to detect sybils [47]. It relies on the presence of trusted nodes.

5.6 Random Walk and OSN

Another family, possibly the largest, sybil defence mechanism is based on random walks in OSN (online social network), first proposed in SybilGuard[101]. The key assumptions in these techniques is that the honest region is *fast mixing*[13], and the attack edges are difficult to form and are independent of the number of sybils.

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

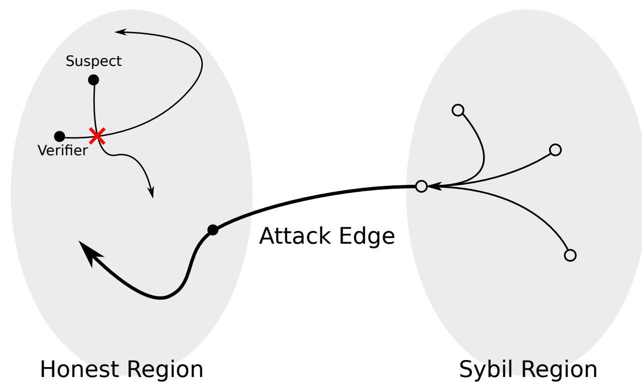

Figure 7: Visualisation of SybilGuard. The verifier accepts the suspect because their random routes intersect. The Sybils' random routes all come from a single attack edge, thus their routes are equivalent after entering the honest region.

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, a 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. In most cases, the number of hops for a random route should be just right, so that the fast mixing property is achieved in the honest region, this is known as the *mixing time*.

In SybilGuard [101] (visualised in Figure 7), every node acts as a verifier 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 labelled as an honest node if its random route intersects with the verifier's random route. The number of accepted nodes for every intersection is limited by a quota. Intuitively, the random route from an honest node is unlikely to escape into the sybil region because the number of attack edges is limited. The number of overlapping random routes from the sybils is bounded by the number of attack edges due to the random route property. 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 [100] is the continuation of SybilGuard and it is an improvement on many fronts while keeping the same or better guarantees. Same 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 number of accepted sybils (false positives) at $O(\log n)$, an

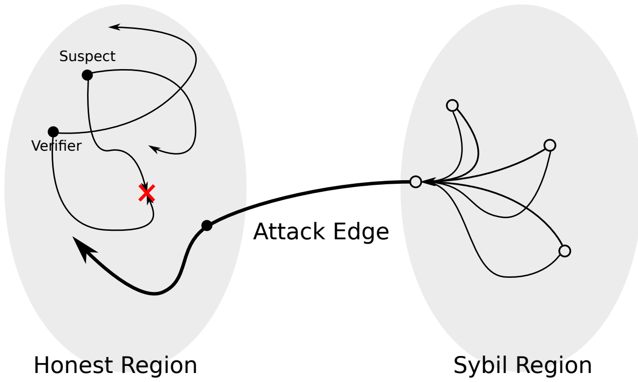

Figure 8: Visualisation of SybilLimit. The verifier accepts the suspect because one of the tails of their random routes meet. Similarly, the Sybils' random route are all from a single attack edges, so their routes are equivalent in the honest region.

improvement from $O(\sqrt{n}\log n)$ of SybilGuard. The process is visualised in Figure 8.

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 tails of V are in the sybil region so they may intersect with many sybils, but crossing the attack edge 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[19] 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. Sybils 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.

Lesniewski et el. created Whānau, a sybil-resistant DHT, it is also inspired by SybilLimit[49, 50]. Suppose all nodes in the DHT belong to a social network and a node, say Alice (who is honest), wish to join the DHT. Alice performs multiple random walks and it inserts the node on the tails of her random walks into her own routing table. Again, the random walk properties guarantee that there cannot be a large proportion of sybils in the routing table with a high probability. The number of entries in the routing table needs to be high enough such that the entries cover the whole key spaces. If Alice wants to find a key, she broadcasts the request to all the entries. Since most of the entries are honest, Alice can retrieve the required value with high probability.

SybilDefender [96] 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. It performs a *partial* random walk, where the random walk is not allowed to traverse the same node more than once. A property of 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 [13], 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). SybilRank uses short random walks that begins on trusted nodes in the honest region. The trusted nodes is chosen 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 some threshold—a system parameter. Finally, various actions can be performed to verify the potential sybils, e.g. using CAPTCHA puzzles.

SybilShield [80] makes use of multiple communities (Figure 9). 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 labelled 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. To do this, 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.

All the techniques so far use fixed length random walks. Recently, Liu et el. argue that fixed length random walks is not adequate because social graphs can have communities of different sizes, nodes in a large community that is well connected may have a lower mixing time than nodes in a smaller community. A longer than desired random walk length will result in a growing number of false negatives. To solve this problem, the authors created SmartWalk[53] which uses adaptive random walks. There are two steps to SmartWalk. The first step uses machine learning to predict the local mixing time given a node. The second step performs the random walk using the local mixing time as the starting length, but on every hop the number of hops remaining is also updated depending on the intermediate node's local mixing time. Finally, the sybils can be detected using the same method as SybilLimit, i.e. node is accepted when their tails meet. The authors evaluated SmartWalk us-

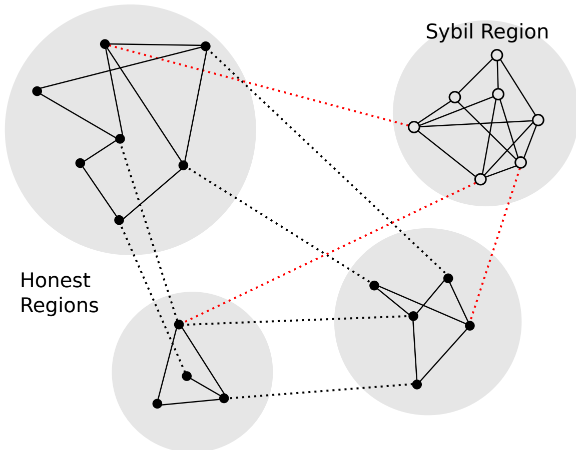

Figure 9: Visualisation of communities in a social graph. Solid lines represent intra-community edges. Dotted lines represent inter-community edges. The red dotted lines are inter-community edges coming from the sybils, in other words the attack edges. The idea of making use of communities is used in SybilShield as well as SybilExposer (subsection 5.7).

ing real-world social graphs. In particular, the false positive rate for a Twitter social graph can be reduced by two orders of magnitude when compared to the standard SybilLimit.

GateKeeper [91] combines ideas from SumUp (discussed in subsection 5.5) and SybilLimit. GateKeeper assumes an 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 (Figure 6). For a node to be labelled 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 because if the ticket sources are evenly distributed and sybils only have a few attack edges, then it is unlikely that they will receive many tickets.

5.7 Community Detection

In this 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 5.6 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 [92]. The authors also argue that social graphs are not always fast mixing, which may result in poor results for techniques that use the fast mixing assumption. Using synthetic social graphs, the authors show that applying Mislove's algorithm [60] achieve similar results as SybilLimit and SybilInfer. But using a Facebook social graph, Mislove's algorithm performs better.

The authors of SybilExposer [61] 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. This is visualised in Figure 9 where the sybil region has fewer inter-community edges than the honest region. 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 [9]), then the communities are ranked based on the ratio and communities with a low ratio are likely to be sybils.

5.8 Content Based and Machine Learning

In some application domains such as OSN it is possible to leverage user 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 [59] is a system for limit 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 [88]. User feedback is incorporated 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. 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 [97] 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 low and many of their friend requests would be rejected.

Integro[10] is a hybrid between random walk and content based approaches. It begins by training a machine 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, perform biased short random walk starting from some known honest account to compute the landing probabilities for every node. Biased in a sense that the walk is a higher probability of using a path with a higher weight. Sybils are the nodes with a low landing probability. 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.

5.9 Other

There are many defence mechanisms that unique in their own right. We cover some of them in this section.

Trust transfer[76] 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 trustworthiness is high, then there is no major effect to transfer a little reputation to a recommended identity.

Yu et el. of DSybil[102] 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 few 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 some factor for all the voters of that object. Otherwise, she decreases the trust score by some 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[42] 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 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. Nodes can be matched by the closeness of their identities. 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

Many surveys on the sybil attack exist in the literature. We attribute much of the initial findings to these surveys. In contrast to the existing work, we also try to cover the possible attacks that can be mounted using sybils as well as a wider range of defence mechanisms.

To the best of our knowledge, Levine et el. published the first survey of the defence mechanisms[55] in 2006. They found that the most popular defence mechanism (in terms of the number of published work) at that time is to use a certificate authority. The surveyed defence mechanisms approximately cover sections 5.2, 5.3 and 5.4 in this work.

Many more surveys were published after the introduction of OSN based sybil defences beginning with SybilGuard. Mohaisen and Kim surveyed certificate authority, resource testing and OSN (random walk) based approaches[62]. They also compare the assumptions, performance and many other properties. Rakesh et el. made a similar survey[70], they discuss six types of attacks in addition to describing the defence mechanisms. Our work can be seen as an extended work of these surveys—in terms of the possible attacks, and the defence mechanisms.

Koll et el. surveyed 8 defence mechanisms for OSN and analysed them in much more depth[45]. The authors experimentally show that increasing the number of attack edges indeed makes the defence mechanisms less effective. More interestingly, sybil tolerance systems[14] such as Ostra (discussed in subsection 5.8) and SumUp (discussed in subsection 5.5) can still be effective when the number of attack edges increase. The authors advise future defence mechanism designers to use information in addition to simply the graph structure to detect or tolerate sybils. In comparison, our work do not go into the same level of depth, but provide a much broader spectrum of defence mechanisms.

Surveys of reputation often cover sybil attacks too [55, 37, 46, 77]. Although they do not cover sybil attack in depth, these surveys provide a lot of insight from a different perspective, especially for the possible attacks in reputation systems.

7. CONCLUSION

In this work, we survey both the practical and the theoretical aspects of the sybil attack. We demonstrate the severeness of the sybil attack by showing the harm it is causing in the real world. Social networks are flooded with sybils making real news and fake news almost indistinguishable. Users of Tor are monitored by sybils hidden in the network trying to reveal their real identity. Then we define the sybil attack using its original definition from Douceur [25] and introduce one of the most common models—the social graph. Next, we

[14]Sybil tolerance systems are those that are designed to limit the effectiveness of the sybils rather than detecting them.

zoom into four different systems—P2P file sharing networks, online social networks, reputation systems and wireless ad-hoc networks—and look the various types of attacks that can be mounted. There are a number of alarming attacks, for instance automated identity theft, self-promotion and so on. The main part of our work summarises the defence mechanisms. Earlier defence mechanisms primarily work by limiting the number of identities or the rate at which they are created. The introduction of BarterCast inspired many network-flow based techniques for limiting the influence of sybils. Similarly, the introduction of SybilGuard stimulated a lot of work on random-walk based techniques for identifying sybils. Hybrids are also available, for instance GateKeeper is a hybrid of the two aforementioned techniques. Finally, we compare and contrast our work with existing surveys.

We hope this work demonstrates the alarming consequences of the sybil attack and many ingenious ways to defend against it. However, there does not exist a general solution and many defence mechanisms must satisfy their own set of assumptions in order to perform well. When the assumptions are violated, which can be the case due to the dynamic structure of real networks, they become ineffective (demonstrated in [53]). Moreover, almost no defence mechanisms considers the temporal dynamics [52], i.e. the attacker may modify the attack edges or its the social graph in the sybil region over time. Lin et el. show the attacker can "greatly undermine the security guarantees" of many defence mechanisms [52].

Without a doubt, much work still needs to be done in order for the cyberspace to be free of the sybil attack. We hope this work serves as a cornerstone for the future defence mechanisms.

8. REFERENCES

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

[2] 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.

[3] R. A. Baeza-Yates, C. Castillo, and V. López. Pagerank Increase under Different Collusion Topologies. In *AIRWeb*, volume 5, pages 25–32, 2005.

[4] M. Bastian, S. Heymann, M. Jacomy, et al. Gephi: an open source software for exploring and manipulating networks. *ICWSM*, 8:361–362, 2009.

[5] 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.

[6] 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.

[7] Bitcoinwiki. Scalability. https://en.bitcoin.it/wiki/Scalability, 2016. Accessed: 2016-11-23.

[8] Bitcoinwiki. Spam transactions. https://en.bitcoin.it/wiki/Spam_transactions, 2016. Accessed: 2016-11-23.

[9] 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.

[10] 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.

[11] 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.

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

[13] 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.

[14] 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.

[15] A. Cheng and E. Friedman. Manipulability of PageRank under sybil strategies, 2006.

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

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

[18] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms*, volume 6. MIT press Cambridge, 2001.

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

[20] 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.

[21] 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.

[22] 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.

[23] R. Dingledine. Tor security advisory: "relay early" traffic confirmation attack.

https://blog.torproject.org/blog/tor-security-advisory-relay-early-traffic-confirmation-attack, 2014. Accessed: 2016-11-16.

[24] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. Technical report, DTIC Document, 2004.

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

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

[27] 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.

[28] 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.

[29] M. Feldman, C. Papadimitriou, J. Chuang, and I. Stoica. Free-riding and whitewashing in peer-to-peer systems. In *Proceedings of the ACM SIGCOMM workshop on Practice and theory of incentives in networked systems*, pages 228–236. ACM, 2004.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] M. Hearn. Modern anti-spam and e2e crypto. https://moderncrypto.org/mail-archive/messaging/2014/000780.html, 2014. Accessed: 2016-11-23.

[35] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg. Eclipse attacks on Bitcoin's peer-to-peer network. In *24th USENIX Security Symposium (USENIX Security 15)*, pages 129–144, 2015.

[36] 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.

[37] 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.

[38] 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.

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

[40] R. Housley, W. Polk, W. Ford, and D. Solo. Internet x. 509 public key infrastructure certificate and certificate revocation list (crl) profile. Technical report, 2002.

[41] 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.

[42] 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.

[43] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the 12th international conference on World Wide Web*, pages 640–651. ACM, 2003.

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

[45] 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.

[46] 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.

[47] 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.

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

[49] C. Lesniewski-Laas. A sybil-proof one-hop dht. In *Proceedings of the 1st workshop on Social network systems*, pages 19–24. ACM, 2008.

[50] C. Lesniewski-Lass and M. F. Kaashoek. Whanau: A sybil-proof distributed hash table. NSDI, 2010.

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

[52] 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.

[53] Y. Liu, S. Ji, and P. Mittal. Smartwalk: Enhancing social network security via adaptive random walks. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pages 492–503, New York, NY, USA, 2016. ACM.

[54] A. Loewenstern and A. Norberg. Bep 5: Dht protocol.

http://www.bittorrent.org/beps/bep_0005.html, Jan 2008. Accessed: 2016-10-20.

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

[56] S. Martin, W. M. Brown, R. Klavans, and K. W. Boyack. Openord: an open-source toolbox for large graph layout. In *IS&T/SPIE Electronic Imaging*, pages 786806–786806. International Society for Optics and Photonics, 2011.

[57] 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.

[58] 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.

[59] 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.

[60] 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.

[61] 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.

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

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

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

[65] A. Nandi, T.-W. J. Ngan, A. Singh, P. Druschel, and D. S. Wallach. Scrivener: Providing incentives in cooperative content distribution systems. In *Proceedings of the ACM/IFIP/USENIX 2005 International Conference on Middleware*, pages 270–291. Springer-Verlag New York, Inc., 2005.

[66] 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.

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

[68] T. T. Project. Top-10 countries by directly connecting users. https://metrics.torproject.org/userstats-relay-table.html, 2016. Accessed: 2016-11-20.

[69] 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.

[70] 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.

[71] B. Ramsdell and S. Turner. Secure/multipurpose internet mail extensions (s/mime) version 3.2 message specification. Technical report, 2010.

[72] 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.

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

[74] 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.

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

[76] 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.

[77] 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.

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

[79] 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.

[80] 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.

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

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

[83] 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.

[84] 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.

[85] M. Srivatsa, L. Xiong, and L. Liu. TrustGuard: countering vulnerabilities in reputation management

for decentralized overlay networks. In *Proceedings of the 14th international conference on World Wide Web*, pages 422–431. ACM, 2005.

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

[87] 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.

[88] 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.

[89] 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.

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

[91] 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.

[92] 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.

[93] 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.

[94] 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.

[95] 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.

[96] 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.

[97] 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.

[98] 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.

[99] M. Yang, Z. Zhang, X. Li, and Y. Dai. An empirical study of free-riding behavior in the maze p2p file-sharing system. In *International Workshop on Peer-to-Peer Systems*, pages 182–192. Springer, 2005.

[100] 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.

[101] 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.

[102] 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.