The darknet's smaller than we thought: The life cycle of Tor Hidden Services

Gareth Owenson[a], Sarah Cortes[b], Andrew Lewman[c]

[a] *University of Portsmounth, UK*
[b] *Northeastern University, USA*
[c] *Laxdaela Technology, USA*

Abstract

The Tor Darknet is a pseudo-anonymous place to host content online frequently used by criminals to sell narcotics and to distribute illicit material. Many studies have attempted to estimate the size of the darknet, but this paper will show that previous estimates on size are inaccurate due to hidden service lifecycle. The first examination of its kind will be presented on the differences between short-lived and long-lived hidden services. Finally, in light of a new Tor protocol for the darknet which will prevent the running of relays to learning darknet sites, an analysis is presented of the use of crawling and whether this is an effective mechanism to discover sites for law enforcement.

Keywords: darknet, cryptography, distributed systems, tor, zeronet.

1. Introduction

Tor is a tool for providing anonymity and privacy when using the Internet, developed by the Tor Project. It does this by encapsulating the user's traffic in layers of encryption and routing it through three intermediate nodes (onion routers (ORs)) such that an attacker cannot resolve all three of: source, destination and content at the same network location [1]. In academic literature, this type of network is often called a mix network and means that at the entry point to the network, the user's identity is known but their traffic is encrypted, and at the exit point their traffic is readable but their identity is unknown.

The Tor Darknet is a feature that Tor provides where two nodes (e.g. a client and a server) may communicate with each other without knowing each other's identity. An anonymous server, or *hidden service*, may offer any ordinary TCP-based Internet service. A user wishing to contact a hidden service will first lookup information via a node (HSDir) in a Distributed Hash Table (DHT) to find *introduction points (IPs)* which will relay a message to the hidden service. The user will then ask the introduction nodes to relay a message detailing a rendezvous point (RP) they have chosen at random, and both parties

Email addresses: sarah.cortes@post.harvard.edu (Sarah Cortes), andrew@laxdaela.is (Andrew Lewman)
URL: gareth.owenson@port.ac.uk (Gareth Owenson)

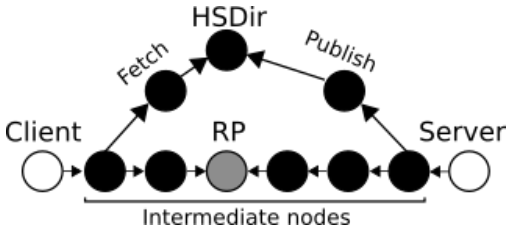

Figure 1: Hidden services network path

will build a three-hop circuit to it. As both the introduction points and the rendezvous point have three intermediate hops between them and the user, and between them and the server, neither knows the identity of each party. The hidden service and the user are now connected via 6-hops through the rendezvous point, and so do not each other's identity (see Fig. 1).

Whilst many believe that Tor hidden services provide them with absolute anonymity, the reality is that there are many long published deanonymization attacks which use simple traffic correlation and do not require significant resources [2]. Additionally, law enforcement and others have had success in targeting and locating criminals acting on the darknet by exploiting vulnerabilities in the Tor software.

Whilst the Tor Project and other stakeholders frequently describe hidden services as an example of privacy and anonymity for political dissidents, the academic literature paints a considerably different picture where the majority of hidden services facilitate criminal activity (e.g. child abuse and drugs) [3, 2]. Other authors have specifically examined the political uses of Tor hidden services and concluded that much of the discourse is banal and of little interest to anyone [4]. Their analysis also concluded that the unethical uses of Tor outweighed the ethical ones particularly due to the harm caused.

Given the criminally orientated content on the Tor darknet, many law enforcement agencies and cyber-security firms have legitimate interests in crawling and collecting information on hidden services. Therefore, in this paper we set out to determine effective strategies for studying the darknet with respect to crawling and monitoring hidden services. One key issue is the size of the darknet, which we define as the total count of concurrently available hidden services. We show that existing estimates of the darknet's size is a gross over-estimate and that crawling strategy can significantly affect results obtained in any study. We also evaluate the impact of methodology on the estimation of which services are available (e.g. by port) and the impact a new hidden service protocol will have on scientific studies of the darknet.

2. Related Work

As described above, Tor uses a DHT to publish information used to contact hidden services. The Tor DHT is similar in design to the Chord DHT [5] in that DHT participant nodes are mapped onto a circle along with data for storage by use of a hash function. In the case of Tor, it's hash function $H : X \rightarrow \{0,1\}^{160}$ is the SHA-1 pseudo-random one-way function mapping the input set X to the set of bit strings of length 160. The

use of a hash function exhibiting strong pseudo-random characteristics is important to ensure even distribution around the circle (see Fig. 2). Each OR is mapped onto the circle by using $H(PK_{OR})$ where PK_{OR} is the ASN.1 encoding of the OR's public key.

Each of the Tor Hidden Services is mapped onto the circle by the use of a unique descriptor ID as defined in equation 1, where $P = H(PK_{onion})$ and $P[a:b]$ denotes bytes a through $b-1$ of P, d is an optional descriptor cookie (shared secret) used to provide client-side authentication for hidden services which are not accessible to all users. Finally, $r \in \{0,1\}^8$ is defined as the replica value and may be 0 or 1. The replica value provides a degree of redundancy, by hashing first with a value of 0, and then again with a value of 1, it gives two distinct (with high probability) locations on the DHT for publication of the descriptor ($\|$ denotes concatenation of bit-strings).

$$descid = H(P[0:10] \parallel H[t_p \parallel d \parallel r]) \tag{1}$$

The time period t_p is defined in equation 2,given the time t in UNIX time (seconds since 00:00 on 1st January 1970). The effect of this is that t_p changes once per day in any one of 256 intervals defined by the first byte of P. This ensures that all hidden services do not attempt to change their publication servers at the same time.

$$t_p = \left\lfloor \frac{t + P[0:1] \cdot \frac{86400}{256}}{86400} \right\rfloor \tag{2}$$

Each Hidden Service, after mapping all ORs and itself onto the DHT circle, publishes its descriptor to the three ORs to the right of its descriptor-id on the DHT. As the Hidden Service is mapped onto the circle at two locations, a total of six ORs receive a copy of the descriptor. The Hidden Service publishes the text document by building a circuit to each of the designated ORs and establishing a HTTP connection to its directory port.

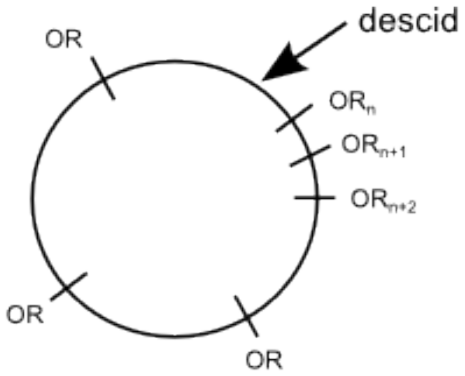

Figure 2: Tor DHT

Due to the open-source nature of Tor, we are able to modify a Tor relay to log requests for hidden services (e.g. visitors) and publications to build a list of hidden services. The effect of the use of eq. 2 is that by running a number of static nodes, over time we will observe the whole DHT.

Pustagarov [2] presented one of the first papers utilising information from the Tor DHT to crawl hidden services and classify content. They collected a sample of hidden

services present on a single day and were able to reach 60,000 hidden services. They then used a bayes naive classifier to classify content into simple categories.

Owen & Savage [3] presented a more recent paper which collected data from the Tor DHT over a period of six months during 2014 rather than a single day. They estimated that there were approximately 45,000 hidden services at the time but that there was considerable churn. They classified content manually to avoid classification errors and found that the majority of content was criminal in nature. Notably, they found the majority of visits to hidden services went to those hosting the most egregious content.

In early 2015, the Tor Project began publishing metrics from Tor relays to estimate the number of hidden services that there were [6]. In the same way as Pustagarov, and Owen and Savage, they record the total number of hidden services published to each relay in a 24 hour period; however, to provide some obfuscation of the raw information they applied additive noise following a laplace distribution. This data is then aggregated from all of the relays and the noise averaged out, leaving a remaining error of approximately $\pm 5\%$ according to their simulations. The principle weakness with this approach, and the subject of part of this paper is that the Tor Project record the cumulative total of hidden services in any 24-hour period, rather than a point sample, and so it grossly overestimates the number.

It is expected that shortly the Tor Project will deploy version 3 of the Hidden Services protocol [7], whose principle goal is to prevent harvesting of hidden service addresses as described above. We will examine the impact this will have on law enforcement and academic studies in Section 5.

3. Measuring the darknet and darkweb size

In this section we seek to understand the true size of the darknet. Our hypothesis is that many hidden services are short-lived, and therefore daily cumulative statistics on the number of hidden services, such as those published by Tor [6], are an overestimate.

To obtain an exact measure to the number of hidden services, one would need to be able to see all onions every day. This is not possible because we cannot observe the entire Tor DHT without controlling all nodes. However, one can observe part of the DHT and then extrapolate global figures. Sampling the DHT is made easier because of two Tor design decisions. Firstly, hidden services publish to six places in the DHT, meaning one relay observes as many as six times more publications than if a hidden service published to just one place. Secondly, because the hidden service publishes to different parts of the DHT each day, and this place is randomised by the use of a pseudo-random function, one has a perfect randomised sampling mechanism. Therefore, it is reliable to make generalisations about the larger population from a small sample.

To sample Tor onions, we run six relays over a period of six months. The relays are configured to obtain the HSDir flag so that they will participate in the distributed hash table for publication and requests for hidden services. The set of tests is as follows: Upon publication, we immediately test the reachability of the onion and then port scan it ($n = 352$). This is then repeated every two hours, recording the time from publication to the up-test.

Recently the Tor Project have actively tried to stop study of the darknet in this way and use a technique similar to honey onions [8, 9]. In this case, they publish honeypot

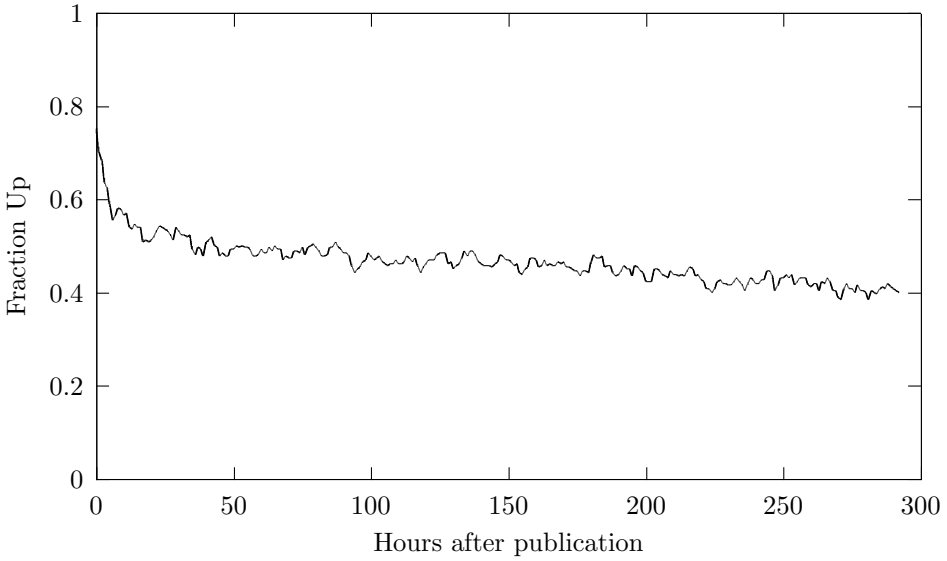

Figure 3: Fraction of onions up following their publication

onions each day to selected sets of relays and see which get visited to identify those which are harvesting data. This detection mechanism is trivially defeated, as described in the original paper; however, deploying the technique in this paper would have adversely affected accurate result correction.

The results for reachability some time after publication are shown in Figure 3. Surprisingly we see a rapid fall off in the first few days of those onions which are reachable, after which it enters a slow decline over time. Notably, after 24 hours following publication, fewer than half of the observed onions are reachable. Furthermore, approximately 30% are never reachable, that is, we were unable to connect to them at all. The reasons for this are unclear, but if the server has an incorrect clock, then it will publish to incorrect parts of the DHT thus making it unreachable. Alternative explanations are that the onions are simply launched for a short test and then stopped.

Figure 4 shows a sample of hidden services and their reachability at particular points in time. Every line on the Y axis represents one hidden service, and every pixel on the X axis represents a two hour window. White indicates that the HS was reachable in that window and black indicates that it was not. Immediately one can see that a large number of HSes were not reachable for each point in time, but also there any a small number which appear to go up and down. Crawling therefore cannot be a one-off event because otherwise these services will be missed - it should be repeated as frequently as possible.

It seems quite reasonable for those crawling the darknet using HSDirs as seeds to find that the majority of onions are unreachable - this is not indicator of a failure in methodology. Furthermore, we can say that the Tor metrics data [6] overestimates the size of the darknet by a factor of two or more. This is because counting publications is not sufficient to establish that an onion is up for the entire day or that it was reachable at

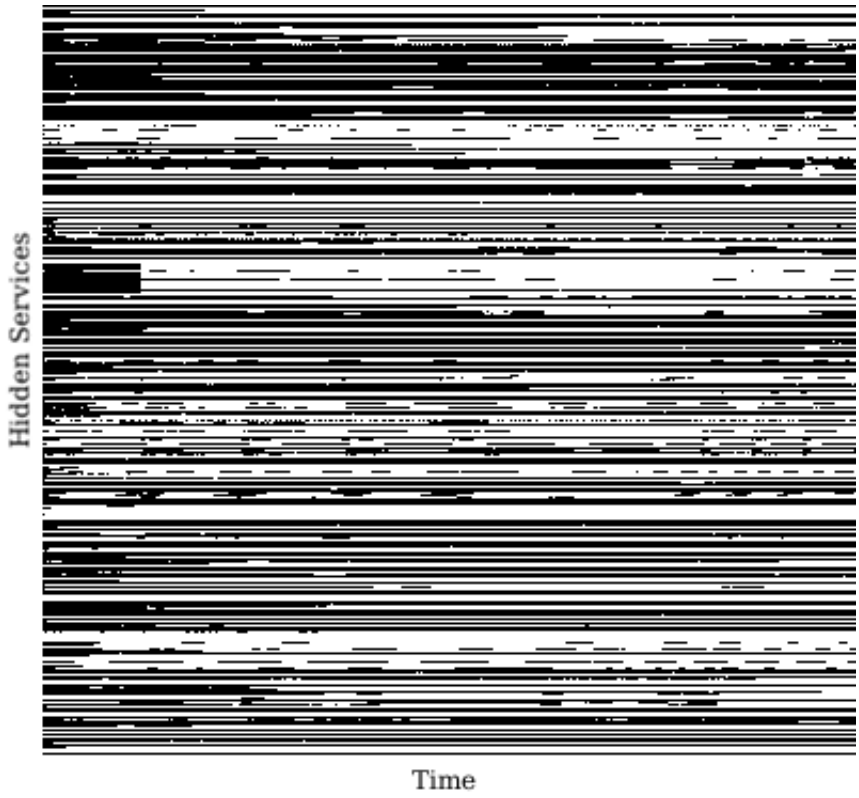

Figure 4: Hidden Service reachability map (time = 25 days)

all. Instead, we conclude that the Tor darknet is approximately half the size previously thought.

4. Services on offer

Next, we study the types of services offered by Tor onions. Given evidence in [3] of a large turnover in hidden services, we sought to understand more clearly the reasons for the turnover. Whilst long-lived services have been studied many time, we are not aware of any papers which have looked at those onions which are only up for a short period of time.

To establish the services offered by a hidden service we test which ports are open. We first fetch the descriptor then build a circuit to the hidden service and send a RELAY_BEGIN cell with a variety of destination ports, and observe the response. If the circuit is closed or a rejection comes back then the port is closed, otherwise it is open. In more recent versions of Tor, the circuit is closed on attempting an incorrect port which considerably slows down port scanning (following a previous academic study [2]). Of course, it is still possible to identify open and closed ports, but in our experience it took around 24 hours to scan 10,000 of the possible 65535 ports for just one onion. Therefore, we adopted an approach of scanning the most likely to be open ports which were determined through a combination of building a list on prior knowledge, collecting ports from existing public/private darknet port scanners (by running hidden services and recording ongoing scans taking place) and scanning a handful of hidden services in full. The final list is as follows: 22 (ssh), 23 (telnet), 25 (smtp), 53 (dns), 80 (http), 81, 110 (pop3), 113 (ident), 135 (smb), 161, 443 (https), 445 (smb), 1337, 1433 (mssql), 3306 (mysql), 4444, 5222 (xmpp), 5223 (xmpp), 5901 (vnc), 6667 (irc), 6668 (irc), 6669 (irc), 6697 (irc), 8060 (onioncat), 8080, 8081, 8333 (bitcoin), 9051 (tor), 9200 (elasticsearch), 9500, 9878 (ricochet.im), 10000, 11009 (torchat), 15441 (zeronet), 17993, 22222, 27017 (mongodb) and 31337.

We adopt the following two approaches to port scanning given that the previous section showed that a large number of services were only active for a short period of time.

1. Port-scan a hidden service as soon as we receive a publication at the HSDir.
2. Count those from the last point which were up less than 24 hours after publication.
3. Port-scan all hidden services learned which are up on a particular day - representing a snapshot of the darknet at a particular point in time.

Table 1 shows the results of our port scans under these two scenarios. Interestingly, web-based services make up the majority of hidden services in both cases, but Zeronet is represented considerably more in the instant scan and those onions up less than 24 hours, indicating that these nodes tend to have a short life-span. Also noteworthy from the results is the small number of ricochet.im (instant messaging) services seen, because this is frequently cited anecdotally within the Tor community as being the cause of spikes in the number of hidden services. SSH is also offered on a surprisingly large number of services which is likely due to one of two reasons: 1) operators are using Tor to access their servers anonymously to decrease the risk of identification; or 2) many users are using Tor to punch through firewalls to access infrastructure. Some Tor hosting providers

Port	On-publication	Snapshot	Up < 24hr
22 (ssh)	12.1%	10.59%	-
23 (Telnet)	0.6%	0.09%	-
25/110 (Mail)	1.1%	4.13%	-
53 (DNS)	-	0.05%	-
80 (http)	54.6%	74.16%	51.4%
443 (https)	2.0%	3.61%	-
IRC (all)	1.2%	1.36%	-
3306 (MySQL)	-	0.08%	-
XMPP (all)	-	1.36%	-
8060 (OnionCat)	-	0.89%	-
8080	0.9%	0.41%	-
8333 (Bitcoin)	0.3%	1.0%	-
9878 (Riccochet)	1.1%	0.67%	-
11009 (TorChat)	-	0.37%	-
15441 (Zeronet)	26.1%	0.77%	48.6%

Table 1: Port-scans of long-lived ($n = 14972$) and short-lived services ($n = 352$)

Port	Min. tpd	Max tpd	Avg. tpd
22	2.6	17	6
80	2.1	17.2	5.5
15441	2.1	52.5	12.6

Table 2: Ports offered against state transitions per day (tpd)

launch two onions per service offered, one for the public facing component and one for SSH remote access - reducing linkability, although there is some prior work on using ssh keyscans to link services together [10].

We also look at differences between those Hidden Services which are frequently going up and down against the more stable services. To do this, we calculate the number of state transitions (from on to off and vice versus) and normalize them over time. For those services which are online all of the time, they will have zero transitions. Those which stay online for a short period of time and then disappear forever will have one transition, whilst those that go up and down will have many more. To normalize this over time, for each service, we divide its measured lifetime by the number of state transitions we recorded to produce transitions per day (denoted tpd). We then assign this number to the ports the service offered and present the data in Table 2. Only those hidden services for which we observed sufficient state changes are included.

One can observe from the data that web, ssh and Zeronet account for most services which exhibit frequent up and down behaviour. Notably however, Zeronet accounted for considerably more of these services that the other two types, with up to 12 transitions per day (e.g. disappearing for 6 periods).

In conclusion, we can say that a big contributor to the high turnover of hidden

Seed Source	Total onions	Onions up	Reachable	Reachable (%)
Reddit	601	272	3047	44.6%
Hidden Wiki	1852	482	3038	44.5%
Combined	2240	580	3047	44.6%

Table 3: Learning seed onions from Reddit

services is Zeronet servers, another darknet which can use Tor to provide some anonymity. Zeronet nodes seem to operate for a relatively short period of time.

5. Will we go dark?

When Tor hidden services publish to the DHT, they sign the documents with a key which is represented by their domain name (xxxx.onion). It is due to this reason that it is possible to harvest hidden service addresses at the DHT nodes. Tor announced in January 2017 the imminent release of the version 3 Hidden Service protocol, which, although not yet available as of November 2017 is in advanced stages on the project's github page. Version 3's principle design goal is to use blind signatures [11] to sign documents published to the HSDirs. There is no known way to take this signature and reverse it into the key for the hidden service itself. The effect of this is to end the ability to run HSDirs to harvest hidden service addresses to study the Tor darknet and the principle mechanism of study will be through crawling from known seed sites. It is worth noting however, that if one knows the hidden service address through another mechanism though it will still be possible to derive the blinded key and so measure popularity at the HSDir (by brute force searching).

Using a crawler to learn HS addresses will mean that those hidden services which neither listed nor linked to will not be discoverable. This presents a problem for law enforcement and introduces potential bias during any attempts to study the Tor darknet. In this section, we examine the impact of not being able to learn all addresses will have. To do this, we choose common start points and crawl the darknet, record which hidden services we learn and then compare them with those learnt through the HSDirs.

We choose two common start points for building crawler seed lists: 1) reddit - a popular discussion site; and, 2) Hidden Wiki - a popular starting point for those browsing the Tor darknet. On Reddit, we targeted two sub-reddits known to be used extensively by Tor users, /r/TOR and /r/onions, where we learn 601 unique onions. For the Hidden Wiki, we crawled the most popular wiki in our dataset (hwikis25cffertqe.onion) and the uncensored hidden wiki (mijps*****.onion), learning 1852 unique onions.

To evaluate the effectiveness of these two common sources as seed lists, we use them to seed a crawl simulator. The simulator takes previously crawled data, and revisits every onion reachable from following hyperlinks (and onion mentions) from the seed lists above. The simulator recursively follows all links until we have reached the maximum number of onions just as a crawler would.

The results are shown in Table 3. Just under half of the darkweb is reachable when using common seed lists and a crawler. A key question is: the other half of any importance? Of the remaining 3786 sites that the crawler did not reach, there were 1898

unique sites. The most popular consisted of botnet command and control addresses, Debian OS clones and several child abuse sites. To evaluate their importance, we summed the total number of requests going to sites reachable through the crawler vs those not. The reachable sites accounted for 2,008,437 daily requests whilst the unreachable only received 35,782 requests. Therefore, we can say that whilst using a crawler finds less than half of those sites learnable through HSDirs, it finds the bulk of sites that users are visiting (or 98% of activity).

Therefore, the move to the version 3 protocol will protect those who truly wish not to be found, but with the exception of a few classes of sites, most wish to drive users to their hidden service and will advertise it as widely as possible. There are exceptions and unfortunately these are the most relevant for law enforcement: botnet command and control and secretive child abuse communities. Botnet hidden services can be found through traditional malware analysis channels however and one hopes that traditional law enforcement intelligence techniques will prevail against abuse sites.

Finally, we ask if a researcher can skip the costly exercise of running HSDirs and simply count mentions on reddit to estimate popularity. To evaluate this, we counted the number of unique reddit pages that an onion was mentioned on and compared with the number of requests seen at our HSDirs. Exactly 272 onions mentioned on reddit were seen at our directories and hence we had a measure of their popularity. We calculated the coefficient of determination between the number of mentions and the number of requests at the HSDir which is $R^2 = 0.288$. Therefore we can say that there is a weak correlation but using the reddit measure alone is not sufficient to gauge popularity. For example, the site with the most mentions on reddit was Facebook's darknet onion; however, when measured at the HSDir, a drugs market and a child abuse sites were most frequently visited.

6. Conclusion

In this paper we have shown that the darknet is smaller than previously believed: about half the size. One reason for this is that the Tor Project count cumulative totals of hidden services online in one day when in fact more than half of them disappear within that day.

We found that HTTP was the most commonly offered service in the longer lived onions, but for shorter-lived onions Zeronet was a notable component - an emerging darknet. Also, riccochet.im plays very little part in the number of hidden services available despite common such statements within the Tor community.

Using traditional sources for lists of hidden services to seed a darknet crawler is an effective means of understanding activity on the darknet, with reachable sites (from a public seed list) accounting for 98% of visits. Whilst the introduction of the new hidden service protocol will obscure the activity of those who do not publish their onion anywhere, many activities require publication to attract users.

References

[1] P. F. Syverson, D. M. Goldschlag, M. G. Reed, Anonymous connections and onion routing, in: Proceedings of the 1997 IEEE Symposium on Security and Privacy, SP '97, IEEE Computer Society, Washington, DC, USA, 1997, pp. 44–.
URL http://dl.acm.org/citation.cfm?id=882493.884368

[2] A. Biryukov, I. Pustogarov, P.-P. Weinmann, Detection, measurement and deanonymisation, in: Proceedings of IEEE Symposium on Security and Privacy, 2013, pp. 80–94.
[3] G. Owen, N. Savage, Empirical analysis of tor hidden services, in: IET Journal of Information Security, 2015.
[4] C. Guitton, A review of the available content on tor hidden services: The case against further development, Comput. Hum. Behav. 29 (6) (2013) 2805–2815. doi:10.1016/j.chb.2013.07.031.
URL http://dx.doi.org/10.1016/j.chb.2013.07.031
[5] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, H. Balakrishnan, Chord: A scalable peer-to-peer lookup service for internet applications, in: Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM '01, ACM, New York, NY, USA, 2001, pp. 149–160. doi:10.1145/383059.383071.
URL http://doi.acm.org/10.1145/383059.383071
[6] G. Kadianakis, K. Loesing, Extrapolating network totals from hidden-service statistics (2015).
[7] D. Goulet, G. Kadianakis, N. Mathewson, Next-generation hidden services in tor, Tech. rep. (2013).
URL https://gitweb.torproject.org/torspec.git/tree/proposals/224-rend-spec-ng.txt
[8] A. Sanatinia, G. Noubir, Honey onions: A framework for characterizing and identifying misbehaving tor hsdirs, in: IEEE Conference on Communications and Network Security (CNS), 2016.
[9] P. Winter, R. Ensafi, K. Loesing, N. Feamster, Identifying and characterizing Sybils in the Tor network, in: USENIX Security, USENIX, 2016.
[10] S. Lewis, Excuse me, i think your dark web is leaking!
[11] D. Chaum, Blind signatures for untraceable payments, in: D. Chaum, R. Rivest, A. Sherman (Eds.), Advances in Cryptology Proceedings of Crypto 82, 1983, pp. 199–203.