Tor Fingerprinting Attacks

John Kristoff
jtk@depaul.edu

ABSTRACT

Tor, the anonymity service, is in a constant game of leap frog between Tor users and would be censors. Tor helps not only protect communications through encryption from would be eavesdroppers, but helps limit the identification of the ends of the communications, providing more than just secrecy, but also anonymity. Censors, unless they wish to completely dismantle all communications must be able to detect and limit access to certain information. Some censors not only wish to limit the flow of prohibited communication, but may seek to identify and possibly punish those who would subvert their control. Tor, it is no secret, is no friend to would-be censors and to those who would prefer to identify the users trying to circumvent their controls. Censors seek to attack the Tor system by identifying the components and its users. Affecting the availability of the Tor system begins by fingerprinting its components parts, the relay nodes, network traffic, and end users. This paper examines the available literature on Tor system fingerprinting and contributes some measurement-based perspective as the system is commonly deployed and used.

ACM Reference format:
John Kristoff. 2016. Tor Fingerprinting Attacks. In *Proceedings of ACM Conference, Washington, DC, USA, July 2017 (Conference'17),* 5 pages.
DOI: 10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Tor is an overlay network for the Internet that aims to provide anonymity for users and services.[3] Using a technology known as *onion routing* that grew out of research sponsored in part by the United States Office of Naval Research and DARPA, Tor has become one of the best known and most widely used anonymity services on the the Internet today. The public Tor network consists of approximately 6,600 relay nodes, 100 Gb/s of relayed traffic, and 3 million connected clients.[1]

Access to the Tor network is relatively straightforward. Using free and widely available software, Tor software is available for an array of operating systems and environments. In order to participate in the Tor network a well known list of systems must be published so that relays and users alike can learn how to bootstrap into the system and build circuits through a reasonable set of relays out of a larger available pool. Since this priming list of relay nodes must be widely available and public, identifying relay nodes is straightforward. Likewise, identifying users of the Tor system can be a matter of simply observing traffic between a known target endpoint and the known list of Tor relays. Of course there is still a matter of the traffic being encrypted that may pose an additional challenge for the would be attacker.

Obtaining a fresh copy of the public list of Tor relay nodes would seem to be sufficient to launching a fingerprint-based attack. More sophisticated approaches of identifying components of the Tor system might be unnecessary. However, there are at least four situations where this bootstrapping list is insufficient. First, an attacker may not have the means to inspect traffic of a target system, either at the relay node or where an end user is positioned. Second, when a Tor relay node is co-resident with other, possibly unrelated but widely used services, falsely associating a conversation as Tor communications may occur. Third, a Tor relay node may not be widely known, such as the case with Tor *bridge* nodes or when used by private, unlisted Tor networks. Fourth, there may be the desire to go beyond simple Tor system usage fingerprinting and uncover end-to-end application and content, effectively breaking the anonymity protections Tor was designed to provide. This last attack type is where most of the research activity has been focused, often in the form of traffic analysis attack models, some of which we summarize in the next section. While interesting and useful work, we are also interested in general Tor traffic and system usage fingerprinting covering the first three scenarios. We provide some evaluation and ideas from some ongoing work with our own measurement and analysis infrastructure in §3 and §4.

2 LITERATURE SURVEY

2.1 Early Anonymity Attacks

Much of the anonymity fingerprinting literature is focused on uncovering the *identities* of a pair of communicating parties. Network anonymity systems are typically designed with one or more relays as a means of ensuring at least one end of the conversation is not directly aware of the identify on the other end in a conversation, providing anonymity to the initiating end of the conservation. A series of one or more anonymous relays should be unable and unwilling to disclose an initiating end's identity to the destination. Attacks on these trust relationships date back before the current Tor system was fully deployed. We begin by briefly outlining the first attacks against anonymity, which provide lessons for a bevy of relay-based anonymity systems such as Tor, foreshadowing much of the attack research to come.

Syverson et al., 2001 [15] provide the first known mathematical analysis of compromised relays in an onion routing system. While compromised relays are beyond the scope of this paper, they provide some helpful commentary on the onion routing network model,

[1] Tor relay node statistics from data provided by https:/torstatus.blutmagie.de/ and usage statistics as displayed at https://metrics.torproject.org/, both as listed in December 2017.

Conference'17, Washington, DC, USA
© 2016 ACM. 978-x-xxxx-xxxx-x/YY/MM...$15.00
DOI: 10.1145/nnnnnnn.nnnnnnn

which eventually spawned into the Tor project. They eluded to the possibility of "potentially trackable" changes in link traffic patterns. They assumed that relays would be configured to operate in a symmetric manner, minimizing differences that may lead to reducing the predictability of traffic patterns. It turns out in practice, many Tor relay operators implement a varied set of policies, from disallowing a relay from acting as an *exit* node to limiting which applications may be permitted to transit an exit relay.

Back et al., 2001 [1] described a series of basic traffic analysis attacks against the Freedom network.[2] One of their attacks, anticipating an approach to more advance attacks to come, was a simple packet counting correlation attack. This attack requires being able to observe traffic entering and leaving relays. Even if relays are shared, this attack attempts to correlate a series of packets on ingress to a relay with an equivalent set of packets on egress. If you're able to observe all of the anonymity network relays, you could, at least in theory, uncover the two ends of a conversation through this packet counting correlation. Hintz, 2002 [7] developed a similar early rudimentary attack using traffic byte counting. While packet-based fingerprinting attacks are still a potential threat, the byte-counting approach has been defeated by Tor's use of fixed size *cells*. While the Tor project has long been aware of packet counting attacks, defenses to them have not become an integral part of the Tor system to this day. Packet counting attacks remain an area of ongoing threat research.

In Dingledine et al., 2004 [3] the authors detail the design of the modern Tor system. From design goals to reflections on attacks and defenses, the authors lay out a number of assumptions and decisions that has led to basis for the modern-day Tor system in widespread use. This paper highlights trade offs made, such as those between protecting traffic from analysis and system complexity. The Tor system maintainers have been particularly transparent and forthcoming about the potential threats and weaknesses of the overall system both in this paper and on their blog. For example, see [2]. This paper notes several possible avenues of attack through traffic analysis and relay node manipulation. The general theme of this paper is while there are many avenues of attack, the system is designed to be reasonably resistant to casual attacks through network size, diversity, and adherence to best practices. In other words, a determined attacker is likely able to compromise anonymity given sufficient motivation and resources.

2.2 Website Fingerprinting

In Shi & Matsuura, 2009 [14] a traffic observation attack between an end user is described. The authors classify a stream of related packets into *vectors*. Each vector is a unidirectional packet flow within a time-limited interval. Some characteristics such as packet count, size, and TCP window size can inform the classifier. The vectors must first be preclassified and then matched with an ingress or egress flow of Tor system packets as they enter and exit the overlay network. By classifying web traffic to popular sites and matching preseeded patterns with test traffic they were able to identify traffic endpoints with a high degree of probability when the end user was limited to a small set of known sites. As the variety of sites increased success fell off dramatically. (i.e. after only about 100 distinct sites were evaluated the success rate was limited to approximately 50%). This paper presented a relatively rudimentary experimental approach to traffic classification with only a marginal rate of success. As destination diversity and web page uniqueness increases the attack is likely to fare even worse.

Pachenko et al. 2011 [13] introduced novel machine learning using support vector machine (SVM) techniques to website fingerprinting (WFP) and summarized the state of the art of attacks on anonymization networks up to that point in time. The authors advanced the traffic analysis approach noticeably, going beyond simple byte and packet counting, giving rise to a number of successor approaches using machine learning algorithms. One innovation introduced in this paper was the use of *markers*, which act like boundaries of a Poisson distribution, though the authors make no mention of this statistical intention. Markers delineate size of packets, directional flow, HTML objects and packet counts. Classification using statistical distributions of these marker-based profiles helps Pachenko et al. reach fingerprinting rates of slightly more than 54% on average.

Inevitably traffic fingerprinting and analysis was destined to rise above early attempts that focused on relatively simpler packet and byte counting mechanisms. In Wang & Goldberg, 2013 [16], such advanced techniques were developed. Like most traffic fingerprinting attacks, observation of a target's ingress and egress traffic to the Tor system is required. In this case the authors find that by reducing this traffic to exclude Tor control traffic, specifically `RELAY_SENDME` cells. This seemingly minor optimization helps the authors advance the success rate of traffic identification from 86-87% to 91% of earlier studies, while keeping the false positive rate to a fraction of a percent. While these modest attack gains seem negligible, the authors significantly increased the sophistication of the attack methods over most previous methods.

Wang & Goldberg advanced the start of the art in at least three new ways. First, the authors leveraged the ability to steer Tor circuit construction, choosing the series of relays a path of packets follow. This allows them to study and hone the ability to classify traffic, as well as detect any differences in exit choice. Depending on the exit relay, traffic may be directed to one of a set of destination instances when a web site is globally distributed through load balancing. While many times the destination may be respond with exact replicas of content when distributed, often times their behavior may assume distinct behavior due to presumed *localization* effects. For instances, a web site may present text in a web page in a language associated with the geolocated position of the connecting source address.

Second, the authors consider and evaluate traffic pattern characteristics on the basis of Tor cells, rather than just the underlying encapsulated IP traffic. This important distinction allows the authors to extract entropy based on Tor-specific overlay network behavior that is lost when considering TCP/IP packet detail alone. While Tor may reveal underlying TCP/IP network fingerprinting through analysis, considering Tor's use of cells and control traffic provides a potentially rich avenue for additional insight. As noted above, removing the `RELAY_SENDME` cells enhances the ability to successfully perform traffic fingerprinting.

[2] The Freedom network is a now defunct anonymity network of relays owned and operated by a company then known as Zero Knowledge Systems (ZKS). ZKS built and deployed was could arguably be called an early precursor, very similar in concept and operation, to the modern day Tor network.

Third, the authors further machine learning techniques, using a SVM approach, to classify and fingerprint web traffic. While earlier attack research also used SVMs, Wang & Goldberg refine the the learning algorithms to both widen the breadth of similarity between training data and reduce seemingly unnecessary tight controls of differences in dynamic web characteristics such as advertisements.

Rounding out a look at website fingerprinting attack literature, He et al., 2014 [6] go beyond passive inspection to actively introducing client latency to artificially improve analysis on these more predictable traffic flows. In other words, by isolating and separating what are often multiple website connections to obtain a variety of site objects, He et al. leverage this behavior to enhance a SVM-based classification engine. They are able to achieve a 15% detection advantage over prior methods using this approach. The drawback of this system is the delicacy required to introduce latency without either disrupting communications or being detected when communications are sufficiently changed or degraded. It is also worth noting that this approach must be carefully aligned with expected retransmission timers and traffic profiles in order to remain effective.

2.3 Onion Services Identification

Fingerprint attacks on Tor onion services (formerly known as hidden services) and onion service users has become an active area of research in recent years thanks in part to the rise and fall of the anonymous, online underground market known as *Silk Road*.[3] As Silk Road may have most famously demonstrated, Tor can be used not only to provide anonymity to users of common services such as the world wide web, but Tor can also provide anonymity to web sites and other server applications. Consequently, researchers have sought out attacks to identify and fingerprint these hidden services and their users.

The first known attack analysis on hidden services was done by Øverlier & Syverson, 2006 [11] and predates the extraordinary events of the *Silk Road* market. Unlike most surveyed literature we have examined at this point, this paper is unique in that not only does it prevent noteworthy attack research, but it led to changes in the Tor system by the time of publication. Therefore, we present just a cursory look as this attack where a derivative form is no longer practical. These attacks, as do many of those we look at require an attacker be in control of one of the Tor relay nodes. The first of four attacks takes advantage of publication or non-publication of a hidden service relay. Monitoring the hidden service and the relays listed in the Tor directory can be used to detect which relay is responsible for a hidden service when both the directory entry and the hidden service becomes unavailable. Two other attacks undermine the ability to infer conversations by examining the timing of communications flowing through a monitored relay. The fourth and final attack is aimed at rendezvous points (RPs), intermediary tor relay nodes each end of Tor circuit rely upon to meet and exchange messages through. An attacker controlling a relay node and a RP will give an attacker knowledge an advantage when their relay node is the last node between one of the hidden service endpoints.

Kwon et al., 2015 [8] first realized Tor circuits established for hidden services are setup and maintained separately than for general purpose Tor relay traffic. This leads to an obvious reduction in traffic analysis required to fingerprint hidden services. A relay operator, while presumably unable to decrypt any of the Tor network traffic or identify both endpoints of a typical Tor circuit, nonetheless has access to the forward and reverse next-hop circuit nodes. This allows an attack to classify circuits, particularly circuits used for hidden services. Coupled with traditional website fingerprinting attacks it then becomes possible to fingerprint and identify end-to-end hidden service endpoints. For endpoints that have been trained through prior traffic, fingerprint detection rates can be over 90%! However, this is an optimistic measure for well known or popular endpoints, with actual results in real traffic scenarios being far less in practice.

Overdorf et al., 2017 [10] took web fingerprinting attacks on hidden services and adapted them to the "closed world" of Tor `.onion` sites.[4] While the fingerprint methods are not a significant evolution over previous attacks, they reinforce the success these methods bring to a closed network environment. What is perhaps most perhaps most noteworthy about their results is the overall rate of success is still only approximately 80% at peak against a total of less than 500 `.onion` sites surveyed. It would seem that with such a small pool of candidate sites, it might be possible to obtain nearly perfect fingerprint identification if not with past methods, with tailored site-specific classifiers. This suggests that Tor does a reasonably good job of masking identifying details of system behavior.

Rounding out our literature survey on fingerprint attacks in the hidden services domain is work by Panchenko et al., 2017[12]. Here the authors outline a method to first identify Tor hidden service communications, then once identified, unveil the specific hidden service using a classifier with the highest rate of success compared to other known classifiers in testing. A key contribution of this work is the observation that many websites are loaded from multiple Tor circuits, some of which may use entirely different Tor network entry nodes and consequently entirely unique end-to-end Tor circuit paths. Panchenko et al. are able to achieve approximately twice the rate of fingerprint success compared to previous fingerprinting efforts, a significant improvement.

2.4 Network Services Fingerprinting

While some studies such as in Feamster & Dingledine 2004 [4] and Murdoch et al., 2007 [9] have enumerated network-level threats against the Tor relay network, few studies have examined the ability to or the threat upon which Tor systems can be passively or actively fingerprinted without direct observational access of the overlay traffic. In this section we highlight two such network-based fingerprint attacks that are not on the data path between of a Tor overlay circuit for any particular communicating target. We will refer to these ideas in the next section with our own contributions and measurement work.

[3] The original Silk Road service was shutdown by the FBI in October 2013, and the so-called Silk Road 2.0 successor was also shutdown by the FBI in cooperation with Europol in 2014.

[4] The `.onion` top-level domain is a DNS-like name space specific to the Tor system. Hidden services use the `.onion` domain name preceded by a distributed hash table (DHT) prefix.

In Greschbach et al., 2016 [5] DNS queries associated with Tor overlay traffic is considered in correlation attacks. That is, off-path networks and resolvers either alone or in conjunction with the capability to observe Tor traffic can enhance anonymity attacks. Tor applications will resolve domain names using local resolver policy at whatever exit node an application happens to be using. The DNS resolver policy is operator independent, but is often whatever ISP or network default is in use. Greschbach et al. find that 40% of all exit nodes use Google public DNS. Regardless of the specific DNS resolver in use, these resolvers are by definition a key component of each Tor user's communication path. If the exit relay traffic can be observed or even just the exit resolvers, DNS traffic analysis may help uncover one or both ends of a Tor conversation. The authors use the DNS signaling to enhance traditional website fingerprinting attacks, bringing additional insight to refine end point identification. Unfortunately the authors did not perform correlation attacks on specific Tor usage traffic so we cannot directly compare their results to other fingerprinting attacks. They did find that they are able to classify websites based on DNS requests with over 90% accuracy, but it is unclear if this enhances identification of specific sessions or not.

3 OUR CONTRIBUTIONS

Many of the system fingerprint attacks are already well known threats, but published evaluation of these attacks is relatively sparse. In this section we offer some evaluation of the Tor system through various means of observational insight including off and on-path from the Tor overlay traffic. In the accompanying code repository we provide some sample code, tools, and data to help illustrate this work.[5]

3.1 X.509 Certificate Identification

Practically all running Tor relay nodes use a version of code that includes this from the `tor/tortls.c` source file:

```
    tor_tls_init();
    nickname = crypto_random_hostname(8, 20, "www.", ".net");
  #ifdef DISABLE_V3_LINKPROTO_SERVERSIDE
    nn2 = crypto_random_hostname(8, 20, "www.", ".net");
  #else
    nn2 = crypto_random_hostname(8, 20, "www.", ".com");
  #endif
```

This code snippet is responsible for generating the X.509 certificate used by a Tor relay node for all the TLS-encrypted and encapsulated traffic. The effect of these instructions is that all Tor relay node X.509 certificates will have the *CN* field value set of the form `www.[a-z2-7]{8,20}.net` and the *ISSUER_CN* field a value of the form `www.[a-z2-7]8,20}.com`. The middle label in these generated domain names are Base32 encoded and always between eight and 20 characters in length. This pattern of X.509 certificate attributes is unlikely to be used by any system not based off the Tor relay code. Obtaining the X.509 certificates from every recently reachable Tor node exhibited this certificate attribute pattern. Where this becomes slightly more useful is in detecting Tor nodes not listed in the public directory. The author had recently conducted Internet-wide TCP destination port 443 scans and where X.509 certificates were found, hundreds of X.509 certificates matching these attribute properties that were not listed in the public directory were found over the course of a few months. All of those discovered tested positively as unpublished bridge nodes.

3.2 Network Neighborhood

The Tor relay directory includes BGP route origination information including a source autonomous network name and number. Evaluating the ASNs in use by Tor relays can highlight the propensity for certain types of networks or locations that run relays. It may also suggest those that are shy away from supporting Tor node relays, although this is likely a large field since in total there are less than 7000 publicly listed Tor nodes as of this writing.

Let us consider the following six classes of networks: 1. Broadband, 2. Enterprise, 3. Datacenter/Cloud, 4. Content Delivery Network, 5. Tier 1/2 ISP, and 6. Internet Exchange Point. We can then classify the top locations where Tor relay nodes are located to see if it any particular network type is more popular for hosting Tor relay nodes than another:

count	type	network
563	Datacenter/Cloud	OVH
428	Datacenter/Cloud	ONLINE S.A.S.
339	Datacenter/Cloud	Hetzner
215	Datacenter/Cloud	Linode
182	Tier 1/2 ISP	Deutsche Telekom
109	UPC	Broadband
106	Datacenter/Cloud	DigitalOcean
105	Datacenter/Cloud	Choopa
100	Broadband	Broadband

Table 1: Tor relay host networks

The Datacenter/Cloud network environment appears to be a popular choice. In fact the next two entries would have been sister networks to DigitalOcean and most of the remaining top 20 most popular networks would continue to be dominated by hosting providers. We could also evaluate the network location of relays along other lines such as IP route prefix or geolocation.

4 FUTURE WORK

In the literature we evaluated there was no survey of originating Tor exit node IP address usage. It would be helpful for instance to enumerate through a Tor directory, setting up a circuit with each available exit node, contact a destination we can observe, and evaluate the exit relay source IP address used. With this data we could then correlate if the exit node address selected as part of the circuit construction is the same as seen by our control destination. We would expect this to be the case most of the time, but if not it may present some interesting provisioning and addressing arrangements that are not reflected in the published directories. Conducting these surveys over the course of time, perhaps with some form of traceroute might also confirm the any one exit node has a stable path to our observed destination or not. We would also be interested

[5] https://github.com/jtkristoff/cs587

to see the propensity of Tor exit nodes that support IPv6 by default when we attempt to access an IPv6 only destination.

When we consider the network neighborhood of Tor relay, bridge, and hidden service nodes, we suspect some interesting patterns may arise for some partial sets of nodes. For instance, if we were worried about malicious node relays, do certain networks seem more prone to hosting them than other networks? What about networks where we might consider the trustworthiness of the operator open to question? For instance, a a government official in country X may be suspicious of any relay operated by any organization affiliated with another country's government or military. Uncovering interesting hosting anomalies or groupings may highlight nodes to prefer or avoid.

In the public Tor node directory listed each relay includes basic operating system information, most of which identify as Linux. We have begun cataloging these systems with a more thorough service scan using *nmap* to better evaluate each relay's system profile to see if we can draw any conclusions about listening services and operating system configuration for typical Tor relay nodes. We are particularly interested in other application services that may be co-resident on relay nodes.

Most known Tor relay nodes have or have had one or more assigned associated public domain names. Issuing historical queries for name mappings on a recent Tor node directory list we try to observe trends or patterns in naming schemes.[6]. We have not concluded our analysis using passive DNS data, but we suspect this may be an area that will lead to interesting correlations. Upon a cursory look we find some relay nodes thousands of name mappings to an IP address while others have none. This outliers may be artifacts of the particular relay provider. Other times where we find one or a small number of associated names there may be an obvious association with the Tor system and other times not. it is not yet clear what patterns may emerge.

5 CONCLUSION

Website fingerprinting attacks have been a popular area of interest throughout much of the research literature we have examined. In laboratory settings these attacks often perform well, but rarely are these attacks evaluated at scale. Some Tor nodes see an enormous amount of traffic and the variability of usage throughout the overlay network is high. This poses practical deployment challenges that has not been well studied. It is unclear how well these attacks ultimately fare in the wild outside of controlled conditions.

Our initial exploration of system fingerprinting beyond entry and exit traffic analysis suggests that the overlay network is both imminently knowable and discoverable, perhaps more easily than some would have imagined. Being able to fingerprint and identify the unpublished systems and users of Tor has ramifications for availability since censors are often looking for easy ways to thwart its use. Overall it seems that there way significantly more avenues of fingerprint attacks than defenders can likely keep up with. The availability arms race for anonymity systems seems likely to continue for years to come.

[6] Passive DNS data obtained through a research grant by Farsight Security https://www.farsightsecurity.com/

REFERENCES

[1] Adam Back, Ulf Mller, and Anton Stiglic. 2001. Traffic Analysis Attacks and Trade-Offs in Anonymity Providing Systems. In *Proceedings of Information Hiding Workshop (IH 2001)*, Ira S. Moskowitz (Ed.). Springer-Verlag, LNCS 2137, 245–257.

[2] Roger Dingledine. 2011. Research problems: Ten ways to discover Tor bridges. (Oct. 2011). https://blog.torproject.org/research-problems-ten-ways-discover-tor-bridges

[3] Roger Dingledine, Nick Mathewson, and Paul Syverson. 2004. *Tor: The second-generation onion router.* Technical Report. Naval Research Lab Washington DC.

[4] Nick Feamster and Roger Dingledine. 2004. Location diversity in anonymity networks. In *Proceedings of the 2004 ACM workshop on Privacy in the electronic society.* ACM, 66–76.

[5] Benjamin Greschbach, Tobias Pulls, Laura M Roberts, Philipp Winter, and Nick Feamster. 2016. The Effect of DNS on Tor's Anonymity. *arXiv preprint arXiv:1609.08187* (2016).

[6] Gaofeng He, Ming Yang, Xiaodan Gu, Junzhou Luo, and Yuanyuan Ma. 2014. A novel active website fingerprinting attack against Tor anonymous system. In *Computer Supported Cooperative Work in Design (CSCWD), Proceedings of the 2014 IEEE 18th International Conference on.* IEEE, 112–117.

[7] Andrew Hintz. 2002. Fingerprinting websites using traffic analysis. In *International Workshop on Privacy Enhancing Technologies.* Springer, 171–178.

[8] Albert Kwon, Mashael AlSabah, David Lazar, Marc Dacier, and Srinivas Devadas. 2015. Circuit fingerprinting attacks: Passive deanonymization of tor hidden services. In *USENIX Security*, Vol. 20.

[9] Steven Murdoch and Piotr Zielifflski. 2007. Sampled traffic analysis by internet-exchange-level adversaries. In *Privacy Enhancing Technologies.* Springer, 167–183.

[10] Rebekah Overdorf, Mark Juarez, Gunes Acar, Rachel Greenstadt, and Claudia Diaz. 2017. How Unique is Your. onion? An Analysis of the Fingerprintability of Tor Onion Services. *arXiv preprint arXiv:1708.08475* (2017).

[11] Lasse Overlier and Paul Syverson. 2006. Locating hidden servers. In *Security and Privacy, 2006 IEEE Symposium on.* IEEE, 15–pp.

[12] Andriy Panchenko, Asya Mitseva, Martin Henze, Fabian Lanze, Klaus Wehrle, and Thomas Engel. 2017. Analysis of Fingerprinting Techniques for Tor Hidden Services. In *Proceedings of the 24th ACM Computer and Communications Security (ACM CCS) 16th Workshop on Privacy in the Electronic Society (ACM WPES 2017).*

[13] Andriy Panchenko, Lukas Niessen, Andreas Zinnen, and Thomas Engel. 2011. Website fingerprinting in onion routing based anonymization networks. In *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society.* ACM, 103–114.

[14] Yi Shi and Kanta Matsuura. 2009. Fingerprinting Attack on the Tor Anonymity System.. In *ICICS*, Vol. 5927. Springer, 425–438.

[15] Paul Syverson, Gene Tsudik, Michael Reed, and Carl Landwehr. 2001. Towards an analysis of onion routing security. In *Designing Privacy Enhancing Technologies.* Springer, 96–114.

[16] Tao Wang and Ian Goldberg. 2013. Improved website fingerprinting on tor. In *Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society.* ACM, 201–212.