Thirteen Years of Tor Attacks

B. Evers, J. Hols, E. Kula, J. Schouten, M. den Toom, R.M. van der Laan,
J.A. Pouwelse (course supervisor)
Computer Science, Delft University of Technology, The Netherlands

Abstract

Tor is the largest anonymous communication network. Recent papers discuss the vulnerabilities of Tor's Onion Router design and question the effectiveness of Tor. These vulnerabilities are increasingly exploited by de-anonymizing attacks. Over the years the attacks have grown to be more complex and effective, increasing the need for hybrid attacks that can be deployed at the network layer, protocol layer or application layer. We will discuss published attacks on Tor and categorize them for further analysis. Tor's principles of freedom and privacy have also introduced some ethical vulnerabilities. The cover that the network provides attracts criminal behavior and has led to a bad reputation. This has caused lawyer-based attacks and adjustments on the fourth amendment to be a point of discussion. Additionally, Tor deals with financial insecurities and a dependency on volunteers. To ensure the continuity of Tor, a dynamic ecosystem should be built around the network by stimulating further development and research in anonymous communication services.

KEYWORDS: Tor, onion routing, de-anonymization, attacks, vulnerabilities, lawyer-based attack, fourth amendment

1 Introduction

We live in an information age in which any person with an Internet connection has all the information in the world at their fingertips. While the Internet has extended the possibility to share information, it has also led to many users worrying that their own private information, including their browse activity, may be snooped without their permission and knowledge. With these rising concerns about privacy and security, Internet users seek ways to anonymize their network traffic. To provide an extensive anonymous communication service, researchers developed the onion routing based system Tor [128]. It is the largest anonymous communication network in existence, with more than 7000 distinct server nodes around the world [116]. It provides anonymous communication services for hundreds of thousands of Internet users and carries terabytes of traffic each day [95].

Tor was originally developed for the U.S. Navy to protect government communications [115]. Nowadays it is an open-source project used for a large variety of purposes by the military, journalists, law enforcement agencies, activists, and many others [128]. The anonymity of the Tor network is appealing to anyone who wants to protect their communications from others, search sensitive topics, avoid surveillance, circumvent censorship and protect their privacy from identity thieves [128]. It has become a tool to keep privacy and freedom of expression alive in the Information Age [93]. Despite the Tor Project's good intentions, it has developed a bad reputation. Just as any large, growing city attracts criminals, the growth of Tor and the anonymity it provides has made the network a hideaway for illegal activities called the Dark Web [84]. A well-known example of a hidden service is Silk Road, a site for selling drugs which was shut down by the FBI in 2013 [93]. The administrator Ross Ulbricht was arrested under the charges of being the site's pseudonymous founder "Dread Pirate Roberts" and he was sentenced to life in prison [114] [71].

The dark side of Tor has drawn the attention from government organizations like the NSA and FBI, that consider Tor a target of particular interest [93]. NSA documents that were leaked by former NSA contractor Edward Snowden have revealed that the organization monitors inexperienced people using Tor, who may not be aware of Internet security and through whom the NSA can gain footholds in the Tor network [93]. Now government organizations are even looking for a way to adjust the legal checks of the Fourth Amendment in order to be able to legally hack users connected to Tor [58].

At the same time, the growing popularity of Tor has lead to the development of an increasing number of deanonymizing attacks on the network. These attacks become increasingly more advanced and effective [87]. Among the most notable attacks is the Sybil attack, which is based on the idea that any system that relies on distributed trust entities can impersonate multiple identities [87] [82]. This involved adding about 115 subverted computer servers to Tor and ensuring they became used as entry guard [8]. The servers took over more than 6% of the network's guard capacity [69]. This attack caused a big stir in the Tor network since the information obtained by the adversary was enough to link some users to specific hidden sites [8].

These recent developments raise questions about the anonymity and security of Tor. The fact that Tor is not 100 percent anonymous is no shocker, but it might be far less secure than most people believe. We will analyze the technical, ethical and financial vulnerabilities of the deployed Tor network. In the first part of this survey we will have a look at Tor's network design and communication protocols. Then we will discuss the attacks on Tor that are currently known and make an effort to categorize them for further analysis. This will be concluded with a section on the way Tor can detect attacks more quickly and how it should protect itself against adversaries in the future. In the second part of this survey we will have a look at the ethical concerns surrounding Tor. We will focus on the ethical issues around the misuse of Tor for a wide range of criminal use and illegal content. In the last section we will discuss the financial insecurities of Tor and the dependence of the network's continuity on financial and non-financial volunteers.

Section 2 presents a high-level overview of the Tor network design and highlights a number of vulnerabilities that are embedded into Tor's protocol. Section 3 analyzes these weaknesses and presents them in a threat model. Section 4 gives an overview of Tor attacks that are published, where each attack is categorized based on assumed goal. Section 5 summarizes how attacks can be detected quicker and how Tor can protect itself in the future. Section 6 summarizes the countermeasures that can be taken to prevent a number of attacks. Section 7 introduces a number of ethical vulnerabilities in Tor. We discuss the implications of the Dark Web on Tor. Section 8 analyzes the financial vulnerabilities and continuous resource starvation of Tor.

2 The Onion Router

The Tor network is based on a low-latency onion-routing design, where traffic is forwarded through randomly selected Onion Routers (ORs), wrapping data in multiple layers of encryption (onion skins) to maintain unlinkability [95]. An OR is also called a *relay*, *node* or simply a router in this context. Each Transmission Control Protocol (TCP) stream can be anonymously channeled through the network in a *telescoping fashion*, meaning that each router only knows the previous and the next relay in the path [87]. Only the first relay, the *entry node*, knows the source of the stream. The last relay, the *exit node*, is the only relay that knows the destination of the client. The onion router(s) in between only exchange encrypted information [108]. Data is wrapped in layers using symmetric cryptography and a relay unwraps one layer of encryption and forwards the message to the next relay in the circuit [104]. A circuit usually consists of three relays.

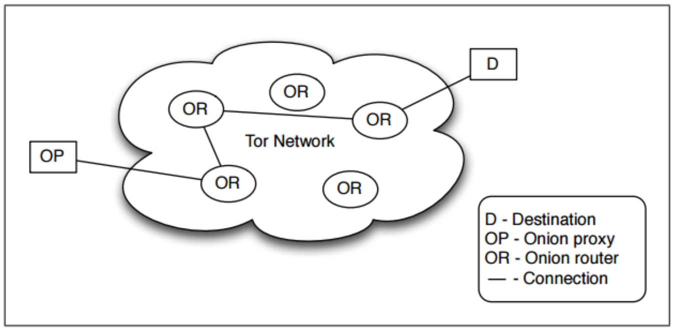

Figure 1: Overview of Tor's Onion Routing Design [108]

2.1 Tor Protocol

A list of trusted and available ORs is advertised on central servers, which are called Directory Server (DS). Furthermore, all relays maintain a Transport Layer Security (TLS) connection to every other relay [40].

A user that wants to connect to the Tor network can use a Tor Bundle. This package contains all necessary components to access the Tor network. A client can connect to the Tor network using an Onion Proxy (OP)[39], which uses the SOCKS protocol [73] to tunnel the client's TCP connections through the Tor network. The TCP streams of the client are sent over the Tor network through *circuits*. Whenever a client wants to create a circuit they can choose a list of ORs and incrementally build a circuit along all those relays. The first relay in the list, the entry node, is contacted and a session key is negotiated. The second relay in the list is contacted via the first relay and the client and the second relay negotiate a session key. This process is repeated until the last relay in the list is reached; the exit node. Session keys are negotiated using a Diffie-Hellman handshake [39].

Sending messages to a server via the Tor network is done through the circuit [39]. The client encrypts the message with the session keys of all relays in the circuit beginning with the last relay and working up to the first relay. The encrypted packets, called *relay cells*, are then sent over the circuit. Each relay along the circuit is able to de-encrypt, or *peel* off, the outermost layer of encryption. At the exit node, the encryption for each relay is peeled off and the raw message is sent to the

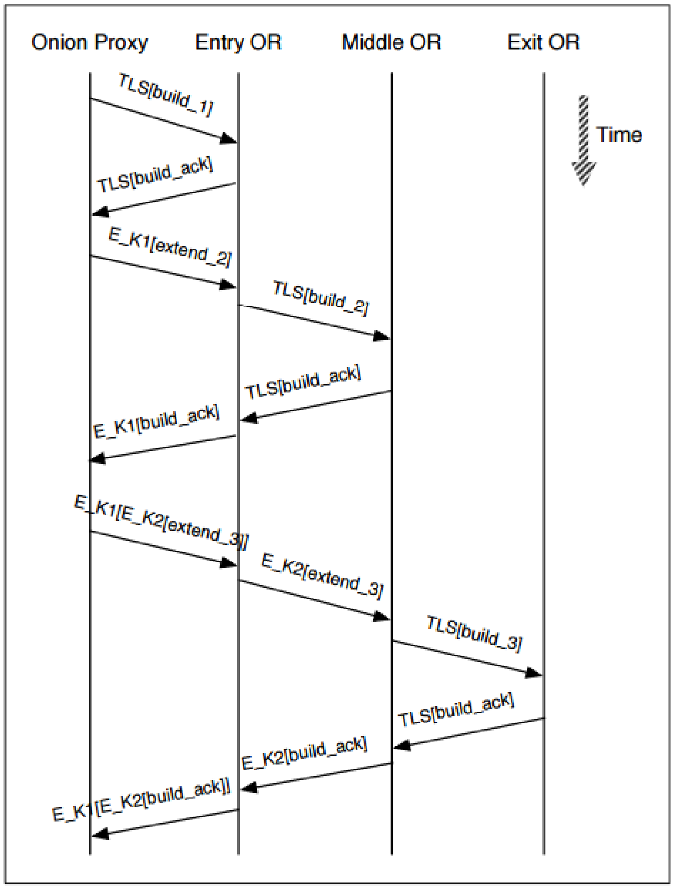

Figure 2: Circuit creation [142]

server. Sending a message backwards works the other way around. Each relay on the circuit encrypts the message it receives with the session key that is negotiated with the client and the client is able to peel off all encryption layers.

Since all relays are listed in the directory servers, access to Tor can easily be blocked by blocking the IP-addresses of all relays. To give access to the Tor network even if all relays are blocked, *bridges* are introduced [85]. A bridge is an OR that is not listed in the directory servers. A Bridge Authority lists all bridges. The Bridge Authority limits access to the bridges' information to prevent the bridges from being blocked.

2.2 Hidden Service Protocol

A Hidden Service (HS) is a network service for which the location of its servers are hidden by the Tor network. In order to connect to a HS, two relays are selected to perform a special task [97]. The *Introduction Point (IP)* is a relay that is tracked by the hidden server for connections to the HS. The *Rendezvous Point (RP)* is a relay that is known to the HS as well as to the client. The details of the basic architecture and the entities can be found in the original paper [39] as well as on the Tor website [129].

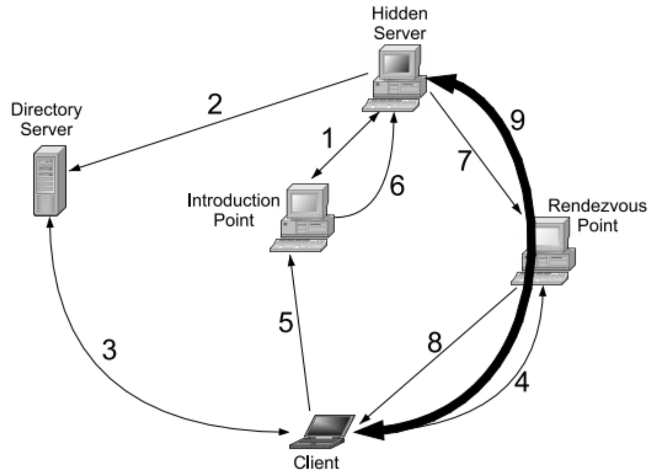

Figure 3: A normal setup of hidden service communication in Tor [95]

A normal arrangement of communication when a client wants to access a resource offered by a hidden server is shown in Figure 3.

First the Hidden Server connects (1) to a relay in the Tor network and asks whether it wants to act as an IP for his service [95]. If the relay allows this, the circuit is kept open. Otherwise, the HS tries another relay until it found an IP. The connections are kept open, until one of the nodes restarts or decides to take it down [2]. There can be multiple IPs per service [95].

Then the Hidden Server contacts (2) the Hidden Servers Directory Servers and requests it to publish the contact information of its HS, including its IPs [95]. The HS is now available to receive connection requests from clients.

In order to use a HS the client contacts (3) a Hidden Servers Directory Server requesting the address of an IP of the HS, which acts a mediator for initial setup [88].

Then the client selects a relay in the network as a RP, connects (4) to it and asks it to listen for connections from a HS [95]. The clients retries this until a RP has accepted and then contacts (5) the IP to request for information about the selected RP [4].

The IP sends (6) the request to the HS which determines whether to connect to the RP or not [95]. If everything is okay, the HS connects (7) to the RP and requests to be connected to the rendezvous circuit [95]. The RP then advances (8) this connection request to the client. Now the RP can start handing over (9) data between the client and the HS [88]. The result is an anonymous data link from the client to the Hidden Server through the RP. All message-flows between these nodes are routed through at least two or more anonymizing relays on their path towards their destination [88].

From the description of the communication between a client and HS we can make the following observations [95]:

- The client does not know the location of the Hidden Server, but knows the location of the RP.
- The HS does not know the location of the client, but knows the location of the RP.
- The RP does not know the location of the client and the HS, and also does not know the content of the service he is offering and the messages transmitted through him.
- There are at least two or more anonymizing nodes between the HS and the RP and between the client and the RP.
- Any node of the network which claims to offer stability can be used by the HS to form an anonymous link to the RP.

3 Threat Model

Most attacks on Tor focus on identifying a relationship between a client and a server that are using the Tor network to communicate [41]. This process is known as de-anonymization [120]. The client has created a circuit in the Tor network to an exit node and the exit node communicates with the server. The attacker wants to confirm that the client and the server are communicating and wants to link a pseudonym (under which a hidden service is being offered) to the operator's real identity, either directly or through some intermediate step (e.g. a physical location or IP address) [101] [88].

The most commonly assumed threat is based on a passive adversary that can observe part of the Tor network and is able to compromise and operate his own onion routers [108][88]. Such an attacker simply observes inputs and outputs of the network and correlates their patterns, so called *traffic analysis* [88]. The attacker tries to measure similarities in the traffic that the client sends and the traffic that the server receives [101]. Traffic analysis is commonly used in attacks on hidden services that try to de-anonymize users [14] [120] [101]. Tor does not protect against a global passive adversary. Its focus is to prevent attacks where an attacker tries to determine in which points in the network a traffic pattern based attack should be executed. By making it difficult for an attacker to determine where to attack, a precision attack is difficult [108].

An active adversary is also a common assumption in Tor's threat model [41]. Such an attacker guesses who is communicating with whom and can analyze individual network links in order to validate this suspicion [88]. They have the ability to inject, delete or modify traffic that is propagated through (compromised) ORs [87]. Since active adversaries are more easily detected, there have been a number of research efforts to develop various countermeasures to defend against these threats. We will discuss these countermeasures and their effectiveness in sections 4 and 6.

In systems like Tor, which is run by volunteers under limited control, it is also a valid concern that an attacker controls a part of the anonymity network [88]. However, it is unrealistic that such a person controls all of the nodes [87]. Therefore this type of attacks is not in the focus of Tor's threat model [108]. The Tor developers are careful, but they still warn their users against using Tor in crucial situations through an announcement upon startup of the Tor client: "This is experimental software. Do not rely on it for strong anonymity." [95]

3.1 Categories of De-Anonymizing Techniques and Attacks

According to existing de-anonymizing techniques on the Tor network, we can sort these techniques into two groups from two different perspectives [148]:

- **Passive and active attacks** The adversary can passively observe the network's traffic or actively manipulate traffic.
- **Single-end and end-to-end attacks** The attacker can impose the network's anonymity by monitoring or controlling Tor circuits at either the enter relay or exit relay side, or at both edges of the circuit.

Based on their method and goal, attacks can be categorized into seven groups:

- **Correlation Attacks** End-to-end Passive Attack
- **Congestion Attacks** End-to-end Active Attack
- **Timing Attacks** End-to-end Active Attack
- **Fingerprinting Attacks** Single-end Passive Attack [38]
- **Denial of Service Attacks** Single-end Active Attack
- **Supportive Attacks** Not classified
- **Revealing Hidden Services Attacks** Not classified

In this case, the label 'not classified' means that the attacks that belong to the corresponding category often combine both types of techniques.

In the following, we give you an overview of attacks on Tor that have been published and discuss them. In Table 1 all attacks are listed in chronological order. Figure 4 shows a mind map that contains the attacks related to their category.

Year	Attack	Category	Paper(s)
2016	Sybil Attack	Supportive Attack	[100] [42] [74]
2015	Guard Selection Attack	Supportive Attack	[76] [45]
2015	RAPTOR Attack	Correlation Attack + Supportive Attack	[120] [62]
2015	Torben Attack	Correlation Attack/ Side-channel Attack	[9] [10]
2015	Circuit Fingerprinting	Fingerprinting Attack	[72]
2014	The Sniper Attack	DoS attack	[65]
2014	BotNet Flooding Attack	DoS Attack	[63] [51]
2014	Relay Early Attack	Correlation Attack	[8]
2013	CellFlood Attack	DoS Attack	[13]
2013	Hidden Service Attack	DoS Attack	[11] [19]
2012	Indirect Rate Reduction Attack	Timing Attack	[54]
2012	HTTPOS Website Fingerprinting	Fingerprinting Attack	[23]
2012	StegoTorus Attack	Supportive Attack	[141]
2011	HTTP-based application-level attack	Correlation Attack	[137] [138] [1]
2011	Packet Size Attack	Supportive Attack	[14]
2011	Bad Apple Attack	Correlation Attack	[21] [106]
2011	Loop Attack	DoS Attack	[75] [92]
2011	Throughput Fingerprinting	Fingerprinting Attack	[86] [29]
2010	Traffic Analysis Attack	Correlation Attack	[154] [68]
2010	Bandwidth Estimation Attack	Correlation Attack + Timing Attack	[27] [28] [26]
2010	Passive Linking Attack	Correlation Attack	[91] [60]
2010	Client Location Attack	Correlation Attack	[91] [60]
2010	Adaptive Surveillance Attack	Correlation Attack	[16]
2009	Cell Counter Based Attack	Correlation Attack	[78] [118]
2009	Protocol-level Attacks	Correlation Attack + Supportive Attack	[52]
2009	FortConsult Security Attack	Correlation Attack	[30]
2009	Bayesian Traffic Analysis Attack	Correlation Attack + Supportive Attack	[130] [90]
2009	Practical Congestion Attack	Congestion Attack	[46]
2009	Website Fingerprinting	Fingerprinting Attack	[113] [67] [135] [134] [98] [64]
2009	Bridge Deanonymization Attack	Supportive Attack	[85]
2009	Link-Based Relay Selection Attack	Supportive Attack	[112]
2009	Tor Authentication Protocol Attack	Supportive Attack	[151]
2009	AS Awareness Attack	Correlation Attack + Supportive Attack	[44]
2008	Route Fingerprinting	Fingerprinting Attack	[36]
2008	Package Spinning Attack	DoS Attack	[132] [37]
2008	Replay Attack	Correlation Attack	[101]
2008	Passive logging Attack	Correlation Attack	[144] [147]
2007	Low Resource Routing Attack	Correlation Attack	[15]
2007	Connection Start Tracking attack	Correlation Attack	[94]
2007	Packet Counting Attack	Correlation Attack	[94]
2007	Stream Correlation Attack	Correlation Attack	[94]
2007	Packet Timing Watermarking Attack	Correlation Attack + Timing Attack	[139]
2006	Clock Skew Attack	Revealing Hidden Services	[88] [150]
2006	First Node Attack	Revealing Hidden Services	[95] [19] [122] [20]
2005	Congestion Attack	Congestion Attack	[89] [142]
2004	Predecessor Attack	Supportive Attack	[145] [146]
2004	Intersection Attack	Correlation Attack	[83]
2003	Active n - 1 Attack	Correlation Attack	[111]
2003	Website Fingerprinting	Fingerprinting Attack	[61]
2003	Robust Watermark Correlation Attack	Correlation Attack + Timing Attack	[140]
2003	Statistical Disclosure Attack	Correlation Attack	[35] [83]

Table 1: Thirteen years of Tor attacks - timeline

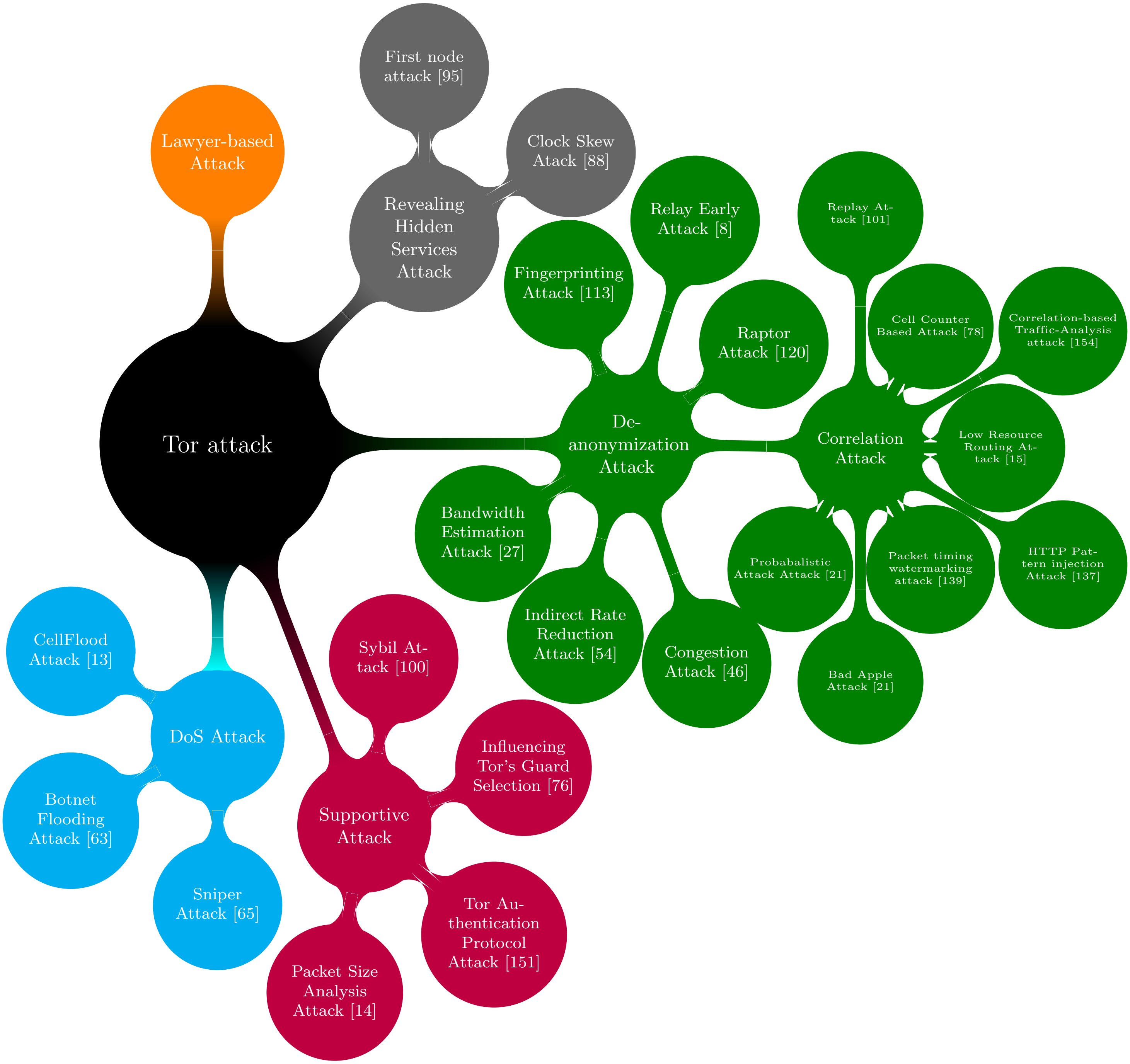
Tor attack
Lawyer-based Attack
Revealing Hidden Services Attack
First node attack [95]
Clock Skew Attack [88]
De-anonymization Attack
Fingerprinting Attack [113]
Relay Early Attack [8]
Raptor Attack [120]
Bandwidth Estimation Attack [27]
Indirect Rate Reduction Attack [54]
Congestion Attack [46]
Correlation Attack
Replay Attack [101]
Cell Counter Based Attack [78]
Correlation-based Traffic-Analysis attack [154]
Low Resource Routing Attack [15]
HTTP Pattern injection Attack [137]
Packet timing watermarking attack [139]
Probabalistic Attack Attack [21]
Bad Apple Attack [21]
DoS Attack
CellFlood Attack [13]
Botnet Flooding Attack [63]
Sniper Attack [65]
Supportive Attack
Sybil Attack [100]
Influencing Tor's Guard Selection [76]
Tor Authentication Protocol Attack [151]
Packet Size Analysis Attack [14]

Figure 4: A mindmap that contains all important attacks on the Tor network that are published.

4 Attacks on Tor

Extensive research is done into the vulnerabilities of Tor. In this section we discuss a number of attacks on Tor that have been published. There is a lot of interest in attacks on Tor. For example, there are rumors that the FBI paid the Carnegie Mellon University (CMU) to develop an attack against the Tor network [7]. Payment from the FBI to the CMU has, however, been denied by the CMU [56]. The attack developed by the CMU was the relay early traffic confirmation attack [7]. This attack will be explained first in this section, followed by other recent or important attacks.

4.1 Correlation Attacks

Correlation attacks are well-known de-anonymization attacks. In this category of attacks it is assumed that the attacker controls both the entry node and the exit node of the circuit between the client and the server. The attacker is looking for a correlation in traffic between the entry node and the exit node, because then he can conclude that the entry node and the exit node participate in the circuit. The entry node knows the client, the exit node knows the server, so the attacker can confirm that the client and the server are communicating.

Relay Early Traffic Confirmation Attack The relay early traffic confirmation attack aims to de-anonymize Tor clients that are using a hidden service [8]. It is known that this attack was actually performed on the real Tor network. Measures against the attackers have been taken.

This attack is a combination of a correlation attack and the Sybil attack. The Sybil attack is explained in Section 4.6. The Sybil attack was used by the malicious ORs to become an entry guard and a hidden service directory. Then a correlation attack is executed to confirm the relation between a client and a hidden service.

For this attack to succeed, the attacker needs control over a hidden service directory relay and the entry node of the client. If the client wants to connect to a hidden service it requests its introduction points at the hidden service directory. The directory relay then sends the name of the hidden service over the circuit encoded in a pattern of relay and relay-early cells. Relay early cells are used to prevent that a client builds long circuits, which can be used in congestion attacks. The entry node can decode the name of the hidden service from the traffic pattern and associate the hidden service with the client.

Replay Attack Another attack that performs an intervention in the communications is the Replay Attack. This attack is described in [101] published in 2008. Assumed is that the length of the circuit from the client to the server is 3.

The attacker selects a cell at the entry router and duplicates this cell. The duplicated cell is also sent to the second node in the circuit. To ensure that the circuit has been created, the duplicated cell should be chosen after the circuit has been established. The selected cell should therefore be a relay cell. The attacker can detect relay cells at the exit node, which he also controls, and notifies the entry node.

Each Tor layer is encrypted with Advanced Encryption Standard (AES) in counter mode. The duplicated cell causes the encryption and decryption counters to go out of sync, resulting in decryption errors. The adversary can detect these decryption errors at the exit node. To confirm that the error is caused by the duplicated cell, the attacker should check that the time for the errors to be detected after the duplicated cell has been sent, is about the same as the time it takes a cell to propagate trough the network. When the exit node detects the error and the timing is correct, the communication between the client and the server is confirmed.

Cell Counter Based Attack A paper by Ling et al. [78] published in 2012 describes an attack in which manipulating the timing of sending relay cells and the cell counter of an enter or exit relay allows the attacker to embed a signal in the traffic of a client or server. This signal can then be recognized by the relay on the other end of the circuit to confirm that a client communicates with a server.

Traffic is sent via cells, which are stored temporarily in a queue, then flushed to the output buffer before entering the network [79]. A signal can be embedded in the traffic by manipulating the cell counter of the output buffer (the amount of cells flushed from the queue to the buffer). For example, three cells means “1” and one cell means “0”. The timing between sending each 'symbol' should be carefully chosen, since waiting too short will cause cells to be combined by other relays in the circuit and waiting too long may look suspicious and will increase the latency which may cause the user to create a new circuit.

Unfortunately, the cell pattern might also be injected at the middle OR(s) due to the natural congestion or delay of the network. Therefore, the amount of cells per symbol should be chosen in such a way that combined cells can still be recognized as symbols at the receiving relay. An advanced recovery mechanism was developed to recover these distorted signals by analyzing the types of combinations and divisions of cells.

This attack is very difficult to detect since the signal can be very short and can have many different properties, which makes it difficult to distinguish from normal

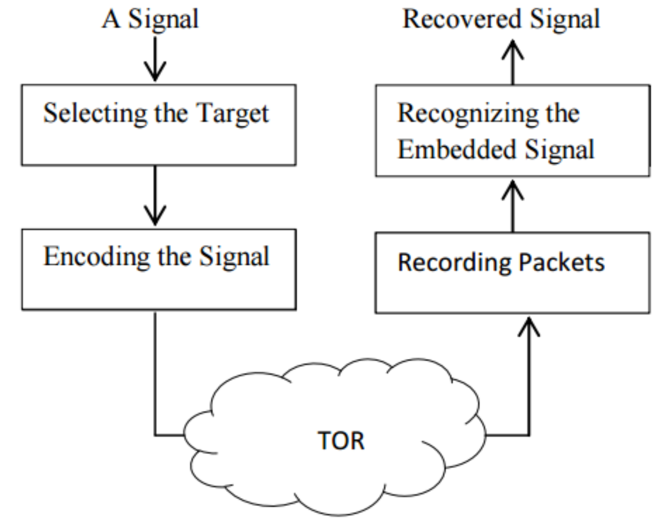

Figure 5: Workflow of Cell Counter Based Attack [118]

traffic. The timing between two symbols can be controlled by a pseudo noise code only known the the attackers. It also has a detection rate close to 100% and can confirm over half of the communication sessions by injecting around 10% malicious onion routes on Tor.

Correlation-based Traffic-analysis Attack In 2010, Zhu et al. [154] introduced an attack that aims to match an input to a mix in a mix network to an output link of that mix. Zhu et al. focused on mix networks that use explicit batching (i.e., an explicit batching algorithm) in their nodes. Tor does not do explicit batching but rather relies on TCP-style feedback-based protocols for perturbing traffic patterns. Although Zhu et al. have not focused on batching methods like the one Tor uses, they do not rule out that the attack could work on systems using such batching methods, so the attack (or a variation thereof) is still relevant.

The aim of the attack will now be stated more precisely. The attack tries to match a known input stream of a mix to one of the output links of the mix. The input stream is assumed to not be further divisible. The challenge lies in finding the input stream in one of the output links that possibly carry multiple streams.

Zhu et al. note that although the attack is described for one mix in their article, the attack could be extended to work on a network of mixes (i.e., a mix network). They describe this in [153]. They also note that if we view a mix network (which could be (a part of) the Tor network) as one super mix, the techniques in their article can be directly applied to the super mix. One could imagine that matching an input stream of the Tor network to an output link of the Tor network could lead to the identification of the communicating parties.

For the attack to be executed, the adversary requires a means to intercept packets flowing through the output links. Malicious ISPs or governments may have the capabilities to intercept packets when they control (part of) a network. Corrupt mixes may also be used, for example.

The attack works as follows. First, the adversary records the packet interarrival times on all output links of the targeted mix.

Second, the interarrival times are transferred into so called pattern vectors that effectively contain the number of packets per batching interval as their elements. This is also done for the known input stream. The transformation process that is used depends on the batching strategy that is used. For example, if packets are sent per specified time-out interval, the number of packets per unit of time for each time-out interval are the elements of the vector.

Third, the distance between the input stream and all the output links is calculated using the pattern vectors. Zhu et al. propose two measures for doing this. The first one is mutual information, which Zhu et al. proposed in [152]. The second one is frequency analysis.

Fourth and last, the output link which has the minimum distance to the known input stream is selected as the output link that corresponds to the known input stream.

One last thing worth noting about this attack is that the amount of available data (i.e., the amount of recorded packet interarrival times) is important. Large amounts of data can lead to detection rates near 100 percent. This is still true when a lot of cross traffic is present.

Related work In 2007, Wang et al. [139] described an attack in which packet streams were also compared based on the inter-arrival times of packets. As opposed to the attack by Zhu et al., they use an "Interval Centroid Based Watermarking Scheme" to influence the inter-arrival times of packets themselves, instead of only recording the inter-arrival times. In this sense, the attack by Wang et al. could be classified as active.

Low-Resource Routing Attack Back in 2007 in a paper by Bauer et al. [15] an attack is described that has the purpose to compromise the users identity by correlating client request to server responses through Tor. The aim is to let clients construct a Tor circuit containing malicious entry and exit nodes. To achieve this the setup procedure of the attack is to enroll or compromise a number of high-bandwidth, high-uptime Tor routers that have a high likelihood of being selected by a client. The paper describes that the resources needed to execute this attack can be significantly reduced by exploiting the fact that a malicious node can report incorrect uptime and bandwidth advertisements to the trusted directory servers as these advertisements are not verified. If only a single malicious node is part of the circuit, it can disrupt the path resulting in the client constructing a new circuit thus increasing the chance to select 2 malicious nodes.

The malicious routers log enough information to correlate client request to server responses. They implemented a circuit linking algorithm that recognizes a circuit request from a Tor proxy. This is were this attack differs from others as it is able to compromise anonymity of a Tor Route before the client starts to transmit any payload data. The researcher validated the attack using an experiment on a realistic test environment network. From the results it is estimated that by contributing less than 1% of the network's aggregate bandwidth they are able to compromise up to 46% of the circuit-building requests for new Tor proxies.

In the paper it is mentioned that the attack can be extended to existing clients. If an attacker can observe a client and make the entry guards unreachable this will result in the selection of new entry guard list with the possibility of selecting a malicious router. An attacker can also perform an denial-of-service attack on a few key stable entry guards, resulting in a large number of clients having to replace the unusable entry guard with a potential malicious one. As defense to this attack it is proposed to verify resource claims such as uptime and bandwidth. To mitigate Sybil attacks they propose limiting the number of routers on a single IP address (which Tor adopted by limiting this to 3). At last alternate routing strategies are proposed to provide adequate load balancing while preserving the networks anonymity.

HTTP-based Application-level Attack In the paper by Wang et al. [138] they present a HTTP-based application level attack against Tor to identify Tor clients. The attack is not specific to web browsing on Tor but rather to the problem of low latency applications based on TCP streams. They assume the attacker can control multiple routers, the entry and the exit router of the circuit. This is possible since Tor is operated in a voluntary manner. They make use of HTTP's vulnerability to man-in-the-middle attacks.

By exaggerating about the resource claims of their routers they can make it likely for a client to select their entry and exit router in a circuit. If the client then issues a HTTP request they can apply their forged web page attack or a targeted web page modification attack. The idea is to let a client's browser initiate malicious web connections to generate a distinctive traffic pattern. This pattern can then be detected by the entry router to expose the client's identity. They also mention that the requirement of a malicious entry router is not necessary in the attack if an adversary can sniff the packets transmitted via the link between the client and the entry router.

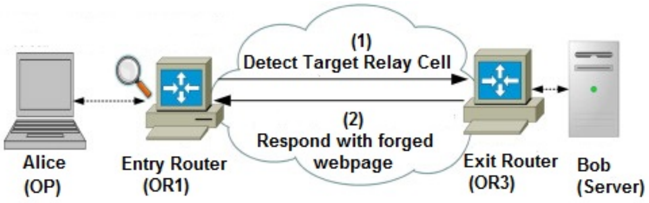

Figure 6: Forged webpage injection attack [138]

As countermeasures they mention minimizing the chance of choosing malicious routers in a circuit. This can be done by increasing the total number of Tor routers and by evolving the circuit construction algorithm to select only fully trusted and dedicated routers through strict authentication and authorization e.g. using a reputation system. Additional countermeasures are abnormal traffic detection using web browser plugins and the use of HTTPS to effectively defend against the attack.

Related work Before the article of Wang et al. [138] described in this section, Wang et al. [137] published an article that is largely the same, except for some rewording and minor additions in [138]. Compared to [137], the notable additions of [138] are another variation of the attack, a more detailed explanation of the experiments and their results and more elaboration regarding the requirements of controlling ORs. They mention that controlling the entry router of the circuit is not strictly necessary for performing the attack.

It is worth noting and also strange that the article from Wang et al. published in 2011 [138] does not reference the article from Wang et al. published in 2009 [137]. In other words, a very large part of the 2009 article [137] has been copied over to the 2011 article [138] without reference.

A paper released in 2015 by Arp et al. [9] describes a similar attack to that of Wang et al. [137]. The biggest difference is in the way of providing web content to users. Where Wang et al. [137] provided content by controlling an exit router and manipulating HTTP results, Arp et al. [9] mainly mentioned providing the side channel content through banner advertisements or cross-site scripting. This makes it unnecessary for the adversary to control an exit router. Additionally Arp et al. [9] assumes that an attacker is able to monitor the encrypted communication between a Tor client and the entry node which is used to recognize the generated network traffic pattern, while Wang et al. is in control of a entry node to achieve this.

A paper released in 2007 by Abbot et al. [1] also describes a very similar attack against Tor. It exploits the same flaws as Wang et al. did in 2011, such as a HTTP man-in-the-middle attack using an malicious exit node to insert javascript code .
The use of man in the middle attacks to the HTTP protocol is used quite often and can be quite a powerful attack technique. A paper by Chaabane et al. from 2010 [24] uses the flaw to rewrite HTTP responses from BitTorrent tracker servers, to let BitTorrent clients connect to their logging client. This way they could estimate the amount of encrypted BitTorrent traffic on the Tor network.

Bad Apple Attack The Bad Apple Attack introduced in [21] is another application-level attack. This attacks requires a malicious application that is installed on the client's computer, to retrieve the IP-address of the user. This application should also use Tor to communicate. How the IP-address of the client is retrieved is not Tor specific and therefore will not be discussed in detail in this paper. The malicious application can just send it's IP-Address via the Tor network to a malicious server, for example.

To correlate traffic from the malicious application with other traffic the attacker should be able to observers the exit nodes of the client's circuits. Since Tor combines multiple streams, possibly from different applications, in one circuit the exit node can correlate traffic from the malicious application with other network traffic. For example, if the malicious exit node first observes traffic from the malicious application, and later observes traffic from website *abc.com* on the same circuit, the attacker can correlate the client with website *abc.com*

Probabilistic Models There have been recent papers on correlation attacks based on probabilistic models. According to Troncoso [130] these attacks have an important advantage: they enable optimal use of all information available about who is talking to whom. They provide an a-posterior probability over all scenarios of interest, whereas most attacks without a probabilistic component only provide the most probable solution. It also means that these models are not concerned with traffic analysis techniques, but effectively assume that the traffic analysis is done. The adversary already has a correct distribution of a user's behavior and communication partners. Most probabilistic models are still research based and have not been deployed to actually attack Tor.

The Bayesian Traffic Analysis of Mix Networks A probabilistic attack on anonymity mix networks presented in [130] casts the traffic analysis in the context of Bayesian inference. The model is based on an Markov Chain Carlo inference engine, that calculates the probabilities of a router being connected to another router given an observation of network traces. The analysis includes conventional aspects of mix networks, e.g. node selection, and complements incomplete observations, erratic users and social network information [108]. In the end it comes down to calculating an a-posterior distribution $Pr[HS|O,C]$ of a set of hidden state user variables HS given an observation O and a set of constraints C based on the user's choice of mixes to relay messages and the user's behavior. However, it is computationally unfeasible to calculate this distribution since the number of possible hidden states is very large. Therefore sample sets $HS_0, ..., HS_n \sim Pr[HS|O,C]$ are used to extract the characteristics of the user variables and to infer the distributions that describe events of interest in the system. The sampling methods estimate the probabilities $Pr[i_x \to o_y|O,C]$ of an incoming message i_x corresponding to any of the outgoing messages o_y as follows:

$$Pr[i_x \to o_y|O,C] \approx \frac{\sum\limits_{j \in N_{MH}} I_{i_x \to o_y}(HS_j)}{N_{MH}}, \qquad (1)$$

where $I_{i_x \to o_x}$ is an indicator expressing if messages i_x and o_y are linked to each other in the hidden state HS_j, and N_{MH} is the number of samples available to the attacker.

This approach enables the attacker to extract information from anonymized traffic traces optimally if he tracks 50 messages of the user he wants to de-anonymize. In all examples, approximately 95% of the samples fall into the confidence interval. The results of the experiments also show that when more messages travel through the network, the attacker is less certain about their destination [90]. An attacker cannot link incoming i_x and outgoing o_y messages with a probability higher than 0.4 when 100 messages have been observed and with a probability higher than 0.1 if more messages are detected.

Danezis [35] presents the **Statistical Disclosure Attack**, an improvement over the original disclosure attack using statistical methods to effectively de-anonymize users of a mix network. The formal model of the disclosure attacks assumes a single mix used by b participants each round, one of them always being Alice, while the $(b-1)$ others are chosen randomly out of a total number of $N-1$ possible ones. The attacker observes the recipient anonymity sets $R_1, ..., R_t$ corresponding to t messages sent out by Alice during t different rounds of mixing. The goal of the attack is to find the set of potential recipients of Alice and in turn to find the recipients of particular messages sent out by Alice. The model defines a vector $\vec{v}^t$ in which the m elements correspond to each potential recipient of messages in the system.

The value of each element is set to $\frac{1}{m}$, meaning that $\vec{v}^t$ is the probability distribution that Alice uses to select a recipient. Next to this $\vec{u}^t$ is defined to be equal to the uniform distribution over all N potential recipients, meaning that $\vec{u}^t$ is the probability distribution that is used by others to select their recipients.

The attacker observes a sequence of vectors $\vec{o}_1^t, ..., \vec{o}_t^t$ expressing the recipient anonymity sets observed in the t messages sent by Alice. Each $\vec{o}_i^t$ represents the probability distribution assigning potential recipients to Alice's message during round i. Using this sequence, the batch size b of the mix and the model $\vec{u}^t$ of other senders the attacker can infer $\vec{v}^t$ and can find an indication on the communication partners of Alice as following:

$$\vec{v}^t = b\frac{\sum\limits_{i=1...t} \vec{o}_i^t}{t} - (b-1)\vec{u}^t \tag{2}$$

The statistical disclosure attack provides important improvements over the original attack. The main bottleneck of the original disclosure attack is its reliance on solving an NP-problem. The statistical disclosure attack only relies on trivial operations on vectors and therefore provides a computational improvement. Next to this, the applicability and effectiveness of the statistical attack are also more predictable because of its closed algebraic form. Another advancement is its extension from being applicable to anonymity systems that create discrete anonymity sets, to probabilistic systems that provide anonymity based on the entropy of the anonymity sets.

Probabilistic Analysis of Onion Routing in a Black Box Model A probabilistic model to evaluate how much an attacker can discover about users by exploiting knowledge of their probabilistic behavior is examined in [47]. The analysis is based on a black-box model of anonymous communication so it could be adapted to anonymous communication networks other than Tor. An active adversary that controls a portion of the network is considered. The abstraction captures the relevant properties of a protocol execution that the adversary can infer from his observations. The model is based on two assumptions. First, a user is responsible for one input and one output. Second, the attacker is able to link network traffic to a user if the input and output are both observable. The mathematical model indicates that user anonymity is worst either when the user shows unique behavior by choosing a destination node other users are unlikely to choose or when other users always visit the user's actual destination. Which case is worse depends on how likely the user was to visit his destination in the first place. This worst-case anonymity with an attacker that observes a fraction b of the network is comparable to the best case anonymity against an attacker that observes a fraction $\sqrt{b}$. In case of common behavior and group joining decisions the anonymity can be kept as the best possible. Feigenbaum expects future research in probabilistic models to focus more on detailed design decisions, such as the impact of entry guards on Tor's anonymity.

Raptor Attack The Raptor attack published in 2015 assumes a powerful adversary [120]. It is assumed that the attackers can use autonomous systems (ASes). There is already evidence that intelligence agencies are cooperating with ASes [110].

The Raptor attack is a combination of three individual attacks and exploits the Border Gateway Protocol (BGP) [105]. First, the Raptor attack uses asymmetric traffic analysis. This means that client and server can be de-anonymized as long as the attacker can observe incoming or outgoing traffic at both the client and the server. Sequence numbers of data packets and/or sequence numbers of acknowledgments can be correlated. This is possible because the TCP headers of the packets are not encrypted at both ends of the client's circuit, and therefore are visible when intercepted by the malicious AS. Asymmetric traffic analysis can be advantageous, because the incoming and outgoing traffic between the client and the entry node or between the exit node and the server might go through different ASes. With asymmetric traffic analysis, only one direction of traffic at both ends of the circuit is needed to correlate the client and the server.

Second, the Raptor attack exploits that BGP paths change due to for example link or router failures. This means that communications between the client and the entry node might go via different ASes over time. Every change in the BGP paths might include a malicious AS into the path between the client and the entry node, which can then perform asymmetric traffic analysis. Asymmetric traffic analysis is only needed once to correlate the client and the server. This means that the chance that the client and the server have been correlated increases over time.

Third, the malicious AS can perform a BGP hijack or BGP interception attack. In a BGP hijack, the malicious AS advertises an IP prefix that does not belong to that AS, as its own. This results in some network traffic intended for that prefix to be captured by the malicious AS. A problem with BGP hijack is that the captured traffic is not forwarded. In a BGP interception attack the malicious AS also advertises an IP prefix that does not belong to that AS. The intercepted traffic is analyzed and then forwarded to the actual destination. BGP Interception might be useful to relate the client with the entry node, when the entry node is known. Paragraph 4.2 describes an attack to retrieve the entry node of a circuit. BGP Interception might then be used with an IP prefix of the entry node to find all the IP addresses that communicate with the entry node. Asymmetric traffic analysis can then be used to find the client that is communicating via the circuit.

Related work A realistic comprehensive analysis was done of the security of Tor against traffic analysis by Johnson et al. [66] for a more generalized attack. It focused on how to make Tor safer for its users, and showed that there are greater risks than previous studies suggested. It discusses how Tor's security can be improved and how users themselves can increase their security against this kind of attack.

4.2 Congestion Attacks

In a congestion attack an adversary tries to determine the identities of the Onion Routers that make up a circuit constructed by a targeted Tor client. To achieve this goal, the adversary congests relays one by one and listens for latency differences in the traffic flow of the target. An adversary could measure latency differences by congesting relays while a client is downloading a large file from a by the adversary controlled website, until the adversary detects the download slowing down. Another method is to inject a script that will periodically perform HTTP requests and start congesting relays until the request frequency decreases.

Congestion Attack by Modulating Traffic Not long after the introduction of Tor, Murdoch et al. [89] introduced a traffic-analysis attack with the goal of uncovering the onion routers (ORs) on a targeted circuit. The attack is not practical at the time of writing [46], because the Tor network has become so large that the attack is not feasible anymore. The Tor network consisted of only 13 ORs at the time of the article of Murdoch et al. compared to 7121 ORs as of 27-03-2016 [103]. The attack will still be discussed here because it is well known.

As stated, the purpose of this attack is to reveal some or all of the ORs that form the circuit established by a client to a corrupt server. In order to achieve this, the adversary needs multiple computers under his control: one corrupt server and one or more corrupt ORs. Hence, the adversary only has a partial view of the Tor network. Note that this attack does not aim to uncover the identity of the client, but only the identities of the ORs that make up the circuit.

To execute the attack, the adversary first needs to get the client to connect to the corrupt server. Murdoch et al. do not specify how they would achieve this. One possibility would be intercepting unencrypted HTTP traffic. When the client has connected to the corrupted server, the server will send modulated traffic to the client (i.e., it will send the traffic using a specific pattern).

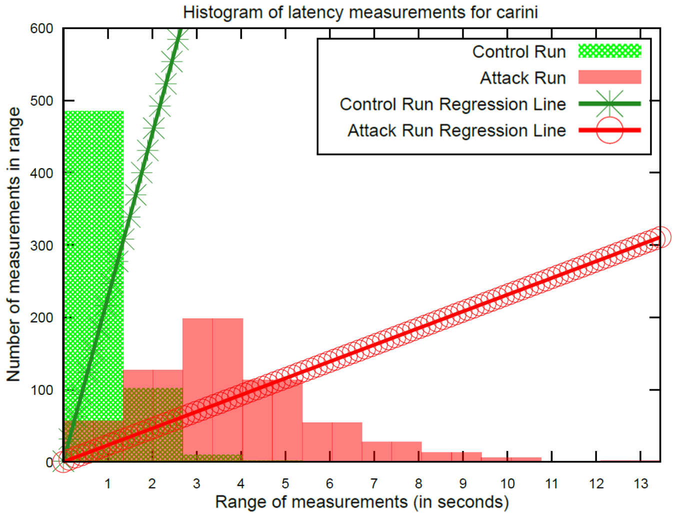

Figure 7: This figure shows that the latency of the cells varies more during the congestion attack, compared to the control measurements [46].

Then the adversary's corrupt ORs will come into play. Each of the ORs will subsequently make connections through legitimate ORs in order to check if they are on the path from the target client to the corrupt server. This check consists of filling the connection through the legit OR with probe traffic and recording the latency of the connection. If the latency pattern that is being recorded matches the pattern that the corrupt server sends to the client, the OR is probably part of the circuit from the client to the corrupt server. If the pattern is not detected there is a high chance that the OR is not in the circuit. Using this technique can lead to the discovery of all ORs on the path from client to server. One can imagine that using more corrupt ORs for probing the legitimate ORs leads to faster results.

There is one additional use of this attack that is worth mentioning. When the ORs on the circuits of two different streams have been discovered and some of those ORs match between the two streams, one can say with quite some confidence that the two circuits originate at the same client. This is because the chance of picking the same ORs is fairly low. The more ORs the streams have in common, the higher the chance of the streams originating at the same client. It is interesting to note that this variant of the attack give better results when a larger pool of ORs in the Tor network is used or when the circuit lengths are greater. The results improve in both situations because in both situations the chance of picking the same ORs randomly when building a circuit decreases.

Related work This attack is used to discover the bridge a client is using in [85].

A Practical Congestion Attack In 2009 a practical congestion attack on the Tor network was introduced [46]. This attack improves the congestion attack introduced in Paragraph 4.2 published in [89], which is no longer reliable because of the growth of the Tor network.

The aim of the attack is to confirm that a node, the entry node, participates in the circuit from the client to the exit node. Assumed is that the attacker controls the exit node. The first step of this attack is to inject some JavaScript code into a HTML response at the exit node. This JavaScript code causes the user's browser to send an HTTP request at regular intervals of 1 second. The HTTP requests contain the time the request was send, so that the attacker at the exit node can correlate the difference in arrival time of the requests, with the difference in send time of the requests. When the attacker does not introduce congestion, he can measure the difference in send time and arrival time of the requests and calculate the average latency of the circuit.

The attacker introduces congestion by creating a circuit from a malicious client to a malicious server. This malicious circuit is a long circuit of length m that repeatedly includes the assumed entry node on its path. Because a relay should not extend a circuit to the previous relay in that circuit, the attacker includes two high bandwidth relays in the malicious circuit and then loops back to the assumed entry node. 24 hops would be effective, according to [46], but as a result of that paper Tor now limits the number of hops to 8. The supposed entry node now has $\frac{m}{3}$ additional circuits to send packets for. Since packets for different circuits are send in round-robin fashion in Tor relays, this causes congestion at this relay. Assuming that the high bandwidth relays are not a bottleneck and the malicious server uses his full bandwidth p to send packets over the circuit, a long path would result in the assumed entry server having to route a bandwidth of $\frac{m \cdot p}{3}$ for the malicious circuit.

The HTTP requests, done by the JavaScript injected at the exit node, will experience delay due to the congestion, which can be measured at the exit node. The exit node will find out that the latency of the circuit varies more. By repeating the process of measuring the average latency and then introduce congestion to measure whether the latency varies more, several times, the attacker's confidence that the entry node participates in the circuit increases.

4.3 Timing Attack

Timing attacks are another form of de-anonymizing attacks. During a timing attack an adversary manipulates both the entry and the exit relay of a targeted client. By correlating flow patterns in traffic flowing from the entry node to traffic flowing to the exit node, the adversary can determine which server a client is communicating with [48].

The Indirect Rate Reduction Attack Gilad and Herzberg introduced this timing attack in 2012 [54]. The attack uses the predictability of the exit nodes that an OP chooses and the congestion control algorithm in the TCP protocol [3] to its advantage. Gilad and Herzberg note that the attack has not been fully tested, but they did some initial experiments which turned out to go well.

The attack aims to check which clients are communicating with a predefined server through the Tor network. The adversary must choose the clients that will be monitored beforehand, because the communication with the entry node of their circuits has to be intercepted. Gilad and Herzberg give example adversaries like governments or employers that control the network their citizens, respectively, their employees are using. Besides having the means to intercept the communication of the targeted clients, the adversary needs a device for sending spoofed TCP packets to the exit nodes of the circuits of the targeted clients.

To execute the attack, the adversary will use the congestion control behavior of TCP [3]. When a device receives three of the same ACK packets, the congestion window of the device will scale down. How much the window will scale down depends on the TCP implementation that is used.

The adversary will also use the observation that which exit node an OP chooses is quite predictable. Gilad and Herzberg found out in an experiment that in 20 percent of the cases, one of the same 7 exit nodes was chosen.

To use those two behaviors to his advantage, the adversary sends three packets with wrong sequence numbers to all exit nodes that are likely to connect to the server and to a lot of different ports. The IP addresses of these packets are spoofed so that it appears they come from the server. This will cause the exit node to send three ACK packets to the server. The behavior described above will cause the server to scale down its congestion window.

Note that three packets are send to all exit nodes that have a high probability of having a connection to the server, to a lot of ports. In other words, packets will be send to exit nodes that do not actually have a connection to the server and to ports that do not have a connection to the server. This is not very efficient. Gilad and Herzberg propose using two other kinds of attacks described in their article, along with this attack, to first identify the exit nodes that actually have a connection to the server, along with the right port numbers. These two additional attacks will not be discussed here.

The scaling down of the congestion window can be detected by the target clients, because their connection to the entry nodes will be scaled down too. Repeating these steps multiple times, the adversary can say with high confidence which clients are communicating with the server.

It is important for the adversary to let the congestion window of the server recover between iterations of the attack. To this end, the adversary has to wait some time before sending the three fake packets to the exit nodes again. Because enough time is needed to execute a sufficient amount of iterations of the attack, the connections that are targeted have to last long enough. According to Gilad and Herzberg, several minutes should be sufficient. Also, the connections under attack should be active, because the congestion window will not recover when no data is being transmitted and data transmission is needed in order to measure the rate reduction caused by scaling down the congestion window.

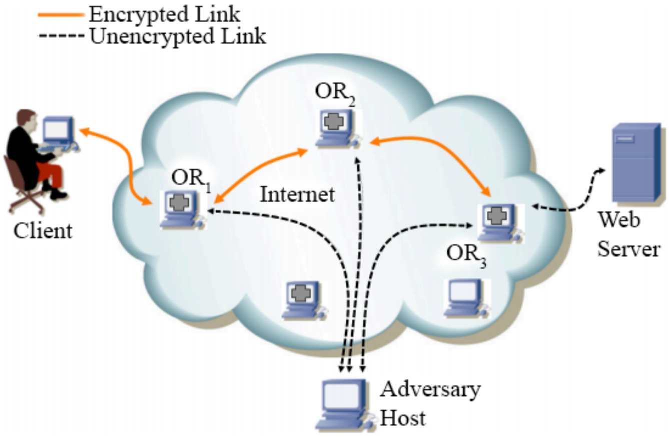

Figure 8: Attacker investigating the fluctuations in available bandwidth of ORs participating in a Tor circuit [27]

Bandwidth Estimation Attack In 2010, Chakravarty et al. [27] introduced a traffic-analysis attack that aims to uncover the identity of users of an onion proxy (OP), the identity of hidden services, or the identity of onion routers (ORs) by using bandwidth estimation techniques. The novelty of the attack is that it does not require any intervention into the Tor network itself. Previous timing attacks, as Chakravarty et al. say, required intervention into the Tor network by using compromised ORs or a global adversary.

It is worth noting that this attack only works well if the Tor network does not influence the bandwidth of its users too much, because the attack depends on tracing a bandwidth pattern. Chakravarty et al. expect that the Tor network will not influence bandwidth much in the future.

The aim of the attack is to uncover the autonomous system (AS) which contain the target and possibly also the precise identity of the target. The target can be either an OP, an OR or a hidden service. The attack will be described for the case of an OP, but the description for the latter two possibilities is very similar.

The adversary needs to follow the following three steps for executing the attack. First, a server is needed that is colluding with the adversary. This server may be controlled by the adversary (which will make the attack easier to execute), but this is not strictly necessary. Second, several network bandwidth probing nodes are required. Third, maps containing ingress and egress routers of ASes are needed. Ingress traffic is traffic in the Tor network that originates outside of it. Egress traffic originates inside the Tor network and is sent outside the network. According to Chakravarty et al., such maps are constructed by several projects and are easy to acquire for an adversary [27].

For discovering the precise location of an OP, internal AS maps are needed. Chakravarty et al. propose using services like Rocketfuel [5] from the University of Washington to acquire such more low-level maps.

The following actions comprise the execution of the attack. First, the adversary places the bandwidth probing nodes close to ingress and egress routers at the boundary of ASes. The most useful location for the probing nodes is determined using the acquired network maps.

Second, the adversary must get the target client to connect to the colluding server. Once the connection is established the server will vary the bandwidth of the connection to the client, leaving a distinct pattern along the path of the connection. This is the pattern that will trace back to the OP of the client.

Third, the bandwidth probing nodes at the AS boundaries will probe ingress and egress routers in order to find the bandwidth pattern used by the colluding server. The routers may be regular routers or ORs. The probing is done using software like LinkWidth [26], which checks the available bandwidth on the probed router. LinkWidth was also introduced by Chakravarty et al. If the pattern is detected on a router, the corresponding AS is probably part of the path from client to server. Using this technique, the adversary can trace the route back to the client.

Fourth and last, when the AS containing the OP of the client has been found, the final part of the path can be traced using the same technique, only this time using the more specific internal AS map.

Chakravarty et al. tested the attack in an emulated environment, an in-lab experiment and on the real Tor network. They executed the attack on 50 circuits on the live Tor network and tried to identify the ORs on the circuit. They detected three, two, one and zero of the ORs 11, 14, 12 and 13 times respectively. Also, 22 of the 150 probed ORs filtered all of the probe traffic.

Using this attack to uncover the identity of a hidden service can be done in a similar way using a colluding OP instead of a server for creating the bandwidth pattern. The identity of an OR can also be discovered using a similar technique.

Related work The variant of this attack that tries to uncover ORs in a Tor circuit was described in more detail, along with practical experiments and its results, by Chakravarty et al. [28]. The experiments described in the article are the first to use an actual implementation of the LinkWidth [26] program, of which a prototype was built especially for this attack.

Another related attack was described by Wang et al. [139] in 2007. The attack is similar to the attack described in this section in the sense that it also tries to impose a specific pattern upon a connection in order to identify that connection later. Wang et al. try to influence the packet inter-arrival times instead of the bandwidth of the connection.

4.4 Fingerprinting Attack

In a fingerprinting attack an adversary utilizes the fact that traffic often has very distinct characteristics. These traffic fingerprints can be used to identify which webpage a client is requesting, whether a client is connecting to a hidden service or to gain knowledge about the path along which traffic is traveling through the network.

Website fingerprinting In 2009 an attack on the anonymity of Tor users was published that uses an adversary that monitors the victim's browsing behavior [113]. This attack is a realistic threat since it requires only the entry point of the victim to be occupied and furthermore it requires very few resources.

A typical webpage consists of many different files that are all downloaded once a browser sends a requests to view a webpage. In most browsers, each file would be downloaded via a separate TCP connection. Since every TCP flow uses a different port, a listening attacker can determine the size of each file being returned to the client by counting the total size of the packets on each port. Not all webpages require the same amount of resource files and besides that, resource files all have different files sizes. Therefore, the set of file sizes of a certain webpage creates a fingerprint that can be used to identify this webpage. The attacker will first build a collection of fingerprints of webpages. Next, he can compare the recorded fingerprint against his fingerprint collection which enables him to monitor the user's browsing behaviour.

This attack was first designed against SafeWeb [61], but can also be used against Tor. However, Tor's design employs two significant characteristics which prevents the fingerprinting attack to some extend.

First, Tor employs fixed sized data cells of 512 bytes. So for an attacker it is difficult to detect the precise size of a file.

Second, Tor uses multiplexing to combine all the TCP streams into one connection which makes it harder for an attacker to distinguish the different files.

This second characteristic proves to be the most troublesome factor. A solution for distinguishing the files is to count the number of incoming packets between 2 out flowing packets. The more packets are received in between 2 out flowing packets, the larger the file is assumed to be. The fingerprint can now be represented by a vector $V = (v_1, v_2, ..., v_n)$ where v_i means "the number of occurrences of i subsequent incoming packets". For stable network conditions, webpages with different files and loading process can be distinguished by using for instance the jaccard or cosine similarity between two fingerprint vectors [77].

Circuit fingerprinting Besides Website Fingerprinting there are also ways for an adversary to detect whether a client is using a hidden service. This type of fingerprinting attack is known as Circuit Fingerprinting. During the circuit construction and communication phase between a client and a hidden service, Tor exhibits traffic patterns that form a fingerprint which enables an adversary to determine whether a circuit is involved in communicating with a hidden service [72]. Therefore, by using circuit fingerprinting an attacker can distinguish the regular from suspicious circuits. Next, the attacker can apply a form of website fingerprinting to gain knowledge about which particular hidden service a suspicious circuit is communicating with [23] [136] [135] [98].

Throughput fingerprinting By observing the throughput of a Tor flow an adversary could learn information about the path via which the traffic has traveled through the Tor network. The throughput of a Tor flow can be used as a fingerprint of the bottleneck relay, which is the relay with the minimal forwarding capacity in a Tor flow. Two circuits that share the same bottleneck relay will have a highly correlated throughput. Based on throughput information, an attacker can identify the bottleneck relay, identify the guard relay(s) and determine whether concurrent TCP connections belong to the same Tor user. Also servers can infer they are communicating with the same client by comparing characteristics of datastreams. Namely streams that are multiplexed over the same circuit have the characteristic that the throughput of those streams repeatedly drop to zero during mutually exclusive periods of time, resulting into a strong negative correlation [86].

4.5 DoS Attacks

Denial of Service (DoS) attacks are not used to deanonymize users, but flood the network resource of a victim resulting in very slow connections or making it unavailable. It can also be used to force honest users to use malicious relays, since the honest relays can be made available. Normally the victim of a Distributed Denial of Service (DDoS) attack is sent a lot of UDP packets from many sources, but since Tor only transports TCP streams, this type of DDoS is not possible on Tor.

Botnet abuse When the command & control (C&C) of a botnet ran as a Tor Hidden Service in August 2013, the number of connected users increased from 1 million to 6 million. This attack was not aimed at specific victims, but had an impact on the whole Tor network. While the amount of traffic did not increase dramatically, it doubled download times for small files because of increased processing loads on relays. The bottleneck was the key exchange protocol that is needed to build encrypted circuits. Since the C&C ran as a hidden service, all systems in control of the botnet were periodically creating circuits to the hidden service.

In 2014, a paper by N. Hopper [63] was published in which some solutions are discussed that limit or discourage the use of Tor for botnets. These solutions mostly serve to encourage more research in this area. In particular, four technical approaches are described, each with its own challenges:

- Resource based throttling: This solution aims to limit the rate of requests from the botnet by making it costly to build circuits, either economically (e.g. paying bitcoins) or computationally (e.g. solving a puzzle). This solution fulfills its purpose but also inconveniences normal Tor users.

- Guard node throttling: When a client connects to the Tor network, it first starts a key exchange at a guard node. By limiting the rate that guard nodes accept connection requests, it does not prevent bots from flooding the network but makes it ineffective to run a botnet C&C via Tor. This could be a valid solution if other verifiable services that require a high rate can request permission for a higher connection rate.

- Reuse of failed partial circuits: When a circuit times out, it is destroyed entirely. By reusing partially built circuits a substantial reduction in load for the network can be accomplished if the failure rate of creating a circuit is high enough.

- Hidden service isolation: By isolating the processing of hidden service traffic from ordinary traffic, regular users can be protected from this kind of attack. This could work by introducing new cell types to recognize the type of traffic. It does however intensify the effect for legitimate hidden services.

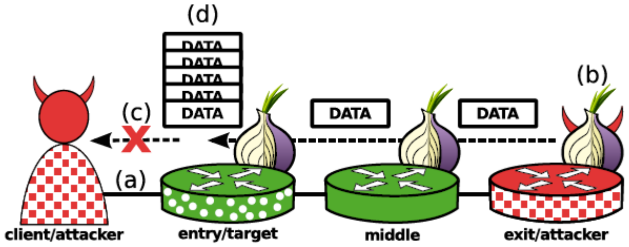

Figure 9: Basic idea of the Sniper Attack: (a) The client creates a circuit with the target as entry. (b) The exit sends data through the circuit, ignoring package window limits. (c) The client stops reading from the TCP stream to the target entry. (d) The target entry buffers the data until termination of the Tor process by the OS. [65]

The Sniper Attack In 2014 a novel and destructive DoS attack against Tor that can be used to anonymously disable arbitrary Tor relays was published [65]. Since the attacker remains hidden while disabling relays in a targeted manner, the attack is called the Sniper Attack. This attack works by utilizing Tor's application level congestion and flow control mechanisms to cause a target relay to buffer a large amount of data in application queues.

To understand the principles of the attack, it is important to know the working of Tor's level congestion and flow control mechanisms. Tor implements an end-to-end sliding window mechanism to control the amount of data directed into the network. The exit relay manages a package window counter for each out flowing stream initiated at 500 and a total package window counter for all streams combined initiated at 1000. For each package the exit relay injects into the circuit, the corresponding stream package window counter, as well as the circuit package window counter, are decremented by one. The exit relay will stop injecting cells from any multiplexed stream whose package window reaches zero, and will stop injecting cells from all streams when the circuit package window becomes zero. The receiving node maintains a sliding window of received packets and once this window is full, the receiver sends a SENDME cell to the exit relay causing the exit relay to increase the package window counters and to restart the transmitting process.

The paper [65] describes a few different versions of the attack, of which the most efficient variant (i.e. the one using the least resources) will be explained below. An adversary starts the attack by creating a circuit that uses the target node as the circuit entry, and initiates the download of two very large files over the circuit. This will result into two streams which can together cause the exit relay to inject up to the 1000 package window limit. By sending SENDME cells to the exit node, the attacker ensures that the exit's package windows does not reach zero and it continues to inject packages into the circuit. However the attacker never reads the cells that arrive at the entry node, the target of the attack. Therefore, cells will continue to flow to and be buffered by the entry node in its application queue, until the entry's Tor process is killed by the OS due to the process consuming too much memory.

CellFlood Attack In contrast to the Sniper attack, that targets relays by downloading large files, the CellFlood attack hinders Tor relays by flooding the relays with difficult to execute circuit setup requests [13]. This enables the attacker to reduce the processing capacity of the targeted relays using little bandwidth. The attacker uses the fact that processing a *create* command takes 4 times longer than generating it. The *create* command is used by a client to extend a circuit. The attacker generates a continuous stream of *create* commands for the targeted relays, which consumes all their computational resources. This results in *create* commands from honest clients that are rejected.

Attacks on hidden services An adversary can also aim to bring down a Hidden Service. A hidden service relies on a collection of introduction points that can be used to introduce a client to the hidden service. The list of introduction points associated with a certain hidden service is stored on Hidden Service Directory Servers (HSDirs). Therefore, HSDir servers are in a position to make a hidden service unreachable by refusing to answer a client's request to receive the list of introduction points [19]. However, an attacker needs to control multiple HSDir servers in order to stop clients from creating connections to the hidden service completely. Another way to impede anyone of creating a connection with the hidden service is by DoSing the introduction points of a particular hidden service [11]. A way to defend against such attacks is proposed by Syverson and Øverlier [96]. This paper describes the strategy of adding a large number of contact points between a client and the introduction points, which prevents an attacker from identifying the introduction points.

Related work In 2014 B. Conrad and F. Shirazi published a paper which analyzes the effectiveness of DoS attacks on Tor [31]. Multiple scenarios are simulated by using different strategies in choosing which OR to attack. The effectiveness is measured by looking at download times for files and the amount of compromised circuits.

DoS attacks that aim to make the Tor network unavailable, might improve by targeting so called *supernodes* [75]. Supernodes are relays that excel in both availability and bandwidth. If those relays are attacked with a DoS attack, other relays may not handle all additional traffic. It might be very difficult to detect a DoS attack on supernodes if the strength of the attack is gradually increased, a so called loop attack [75].

4.6 Supportive Attacks

In this section we describe a number of attacks that do not directly aim to de-anonymize Tor users or disrupt the Tor network but rather are helpful to perform a de-anonymization attack or a disruptive attack at a later point in time.

Influencing Tor's Guard Selection Most attacks on Tor are traffic correlation attacks, where both the entry node and the exit node are required to perform the attack, it can be beneficial for an attacker to force a client to choose a malicious node as guard node, or entry guard. A client only uses guards nodes as Tor entry nodes. This means that if none of the guard nodes are malicious the user can never connect to a malicious entry node [123]. Guard nodes are usually replaced after 30 - 60 days [76]. Replacement is done in a so called guard selection round, where a set of non-selected guard nodes is chosen to be included in the guard list. The chance that a guard node is included in the guard list is bigger for long-running or high bandwidth nodes.

A paper published in 2015 introduces an attack that aims to shorten the the time interval between guard selection rounds [76]. Assumed is that the attacker controls multiple guard nodes. The attacker should also be able to identify and manipulate Tor traffic between the client and an entry node. Identifying Tor traffic can be done by an AS using the method described in Paragraph 4.6.

By analyzing the entry nodes the client connects to, the attacker is able to get the clients guard list. The attacker blocks Tor traffic from the client to all guard nodes except for one. Leaving one guard node reachable makes sure that the client's communications are not disturbed. Since a new guard selection round is performed when less than 2 guards are online [45], blocking all guard nodes except for one results in a new guard selection round. This is an opportunity for one of the malicious guard nodes to be included in the guard list. This process continues until a malicious guard node is included in the guard list. Experiments confirm that in 80% of the cases a malicious guard node is included in the guard list within 20 guard selection rounds. Exper-

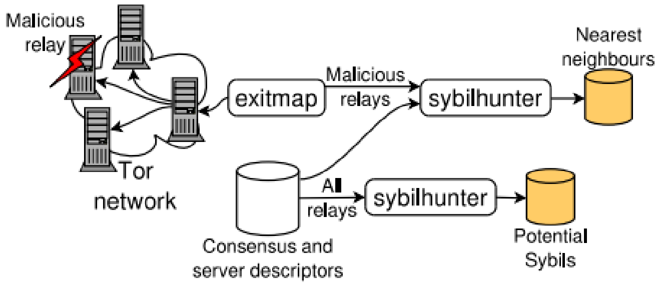

Figure 10: Setup of Sybil attack: two datasets as input to the attacker, consensus and server descriptors; malicious relays together with the exitmap [100]

iments also show that this attack forces an new guard selection round every 1.5 minutes. This would include a malicious guard node into the client's guard list within 30 minutes.

The Sybil attack In June 2010, the number of active Tor relays suddenly increased in a matter of hours. It turned out that somebody set up several hundred Tor relays on PlanetLab machines [100]. This may look harmless, but it can actually be used as an attack on the Tor network called a Sybil attack.

In the Sybil attack, an adversary controls many virtual identities in order to obtain disproportionately large influence in the network. The effectiveness of many attacks on Tor depends on the consensus weight of the attacker, which is the amount of traffic an attacker can observe. As the consensus weight grows, a number of other Tor attacks become easier to execute. Examples of attacks that are easier in combination with a Sybil attack are the fingerprinting and correlation attacks.

Besides simplifying other attacks, the Sybil attack poses risks on the usage of the Tor network and therefore on the anonymity of its users. The effectiveness of Tor depends on the reliability of the Tor relays. Unreliable relays can both degrade the user experience and impair the anonymity guarantees provided by Tor. Certain users will refrain from using the system when encountering issues caused by unreliable Tor relays. Less users means a decrease in the overall anonymity of the network. The remaining users will continue using the network with a lower anonymity, presenting better opportunities for observation. This problem can be exploited by adding malicious relays and strategically affecting the reliability of anonymous communications to increase the odds of an adversary compromising user anonymity [6].

Practical defenses against Sybil attacks are challenging, these attacks will probably always be possible in anonymity networks without a central authority [100]. However, since Sybil relays typically behave and appear similarly, there are some heuristics that can be used to detect a Sybil attack to some extends. Relays that are part of a Sybil attack often join and leave the network simultaneously, they have common configuration parameters, and may frequently change their identity fingerprint to manipulate Tor's distributed hash table.

Packet Size Analysis Attack In 2011 a low-cost technique that distinguishes Tor traffic from non-Tor encrypted (HTTPS) traffic was published [14]. For this attack, it is only needed to intercept traffic and analyze it. Therefore, this attack can be performed by a passive adversary.

After analyzing Tor traffic, it is concluded in [14] that the size of the third packet is about 140 bytes and the size of the fifth packet is about 920 bytes. Using these simple heuristics 98% of actual Tor traffic can be classified as Tor traffic. Furthermore, a large fraction of the packets is just bigger than 512 bytes, the size of a Tor cell. The experiments have been performed in controlled conditions, but these patterns might be visible in the real world Tor network. By analyzing the packet sizes of intercepted traffic, an adversary could distinguish Tor traffic from regular traffic.

Tor Authentication Protocol Attack In 2009, Y. Zhang published a paper for an attack on the Tor Authentication Protocol (TAP) if a user has multiple concurrent sessions [151] of TAP running.

The TAP forms the basis of Tor's security and is used to negotiate session keys between a user and the ORs in a circuit. A vulnerability was discovered if a user runs multiple concurrent sessions of TAP.

This attack works if the attacker has control over one malicious OR A. When a user connects to OR A and negotiates a session key, then negotiates a session key with another OR B, the attacker is able to trickily interleave the messages from both sessions to make different session keys between the user and the non-malicious OR B. While this attack does no direct harm, it violates the original object of the TAP protocol.

4.7 Revealing Hidden Services

It might be interesting for an attacker to reveal a hidden service. This kind of attack is especially interesting for governments that try to pinpoint the location of a hidden service. There are a number of research papers that describe ways to reveal hidden services.

First Node Attack In the first paper that aims to reveal hidden services the attackers' relays try to become the relay in a circuit that is directly connected to the hidden service's server [95]. This would immediately reveal the location of the hidden service to that node.

To become the first node from the hidden service's server in a circuit, the attacker needs a malicious node and a client that connects to a hidden service. The client will connect to the hidden service and send a certain timing pattern in the communication. If the malicious node is on the circuit it will detect this pattern. Since the client knows the IP-addresses of all nodes until the rendezvous point, the malicious node can detect whether it is one of those nodes. Usually, there are three nodes between the Rendezvous point and the Hidden Service. If the malicious node is next to the rendezvous point, this can be detected, since the rendezvous point's IP-address is known by the attacker. If the malicious node is in the circuit, but not next to the rendezvous point or between the rendezvous point and the client, it has to be first or the second node after the hidden service. By using timing analysis the malicious node can detect whether it is the first or the second node. If the malicious node is the first node, the location of the hidden service is revealed. Otherwise, the attack will be run again until the malicious node becomes the first node in a circuit.

The chances that this attack succeeds is reduced by introducing guard nodes. Since the hidden service only creates circuits to guard nodes, if your node is not a guard node of the hidden service, it will never be selected.

Related work More recent, in 2013 an attack that used similar techniques was presented [19]. This attack technique is modified and used by Abbott et al. to pinpoint clients of a web service [1].

Clock Skew Attack Another approach is taken in [88], which is improved in [150]. These attacks are able to pick the hidden services from a list of candidate servers. By creating a lot of requests to a Hidden Service the temperature of the server will rise, resulting in a different clock skew. This clock skew can be derived from the timestamps that are received from the hidden service. By comparing the timestamps from all the candidate servers with their sample they can detect a matching clock skew. This reveals the location of the hidden service.

5 Attack Detection

Many papers describe attacks, but give no way to detect or expose them. A paper by Winter et al. [143] published in 2014 describes how malicious exit relays can be exposed for many common kinds of attacks. Two tools were developed, one to detect active Man in the Middle (MitM) attacks and the other for detecting credential sniffing.

The 'man in the middle' in Tor circuits are the exit nodes. There the final layer of encryption is removed and the traffic is sent to its actual destination. This allows the owner of this exit node to see and even actively modify the traffic. This is often used to exploit vulnerabilities in order to make the connection insecure or inserting malicious code in web pages. The tool 'exitmap' can detect popular MitM attacks. It runs on a single machine and asynchronously creates circuits with a set of exit relays as end points. Most detection methods work by comparing the fingerprint of a certificate fetched through Tor with the expected one that is hard-coded in the client. Examples of attacks that can be detected are HTTPS tempering, sslstrip (rewriting HTTPS to HTTP) and DNS query censoring.

Credential sniffing is a passive MitM attack. Rather than actively modifying traffic it simply looks for credentials in the traffic. The other tool, 'HoneyConnector', can detect credential sniffing. It creates bait connections over Tor using randomly generated unique credentials over FTP and IMAP. By monitoring which accounts were accessed, the malicious exit relay that was sent a unique account could be identified and exposed.

Over a period of several months, all exit relays (~ 950 at the time) were monitored. Using 'exitmap', 40 malicious exit relays were identified. Using modified FTP and IMAP servers, 'HoneyConnector' was deployed on multiple hosting providers. A total of 255 login attempts were made tracing back to 27 sniffing relays, only two of which were also caught by 'exitmap'.

While it is possible for the Tor project to blacklist certain relays as exit relays, since most attackers do not publish contact information or other hints, it is difficult to identify the actual attacker. They might just shut down their malicious exit node and set up another one if they notice that their relay is blacklisted. Therefore, a patch for the TorButton extension of the TorBrowser was developed to detect some attacks automatically for regular users.

6 Attacks and Countermeasures

To defend against the threat posed by de-anonymizing attacks, there have been a number of research efforts by academic and industrial agencies on developing various countermeasures. In general, countermeasures can be deployed from three perspectives: network layer, protocol layer and application layer [148] [117].

Network Layer Since network traffic characteristics can be exploited to de-anonymize users, a basic idea of defense is to remove or falsify the features of traffic correlated with users [148]. These features include packet size distribution, packet order, traffic volume, traffic time, and so on. *Packet padding techniques* can be used to alter packet sizes in order to prevent that features such as packet length and packet order can be inferred [17]. For example, the size of each packet can be fit into the same size with a maximum transmission unit (MTU). The traffic time can be obfuscated by adding delay between each packet to increase the traffic time [12]. Besides, *dummy traffic techniques* can be used to inject dummy packets into original traffic in order to bewilder the traffic volume [12]. Moreover, *traffic morphing techniques* can be applied to alter traffic patterns to look like other traffic patterns. For example, to confuse a web based fingerprinting attack, the web server can select a target page and then imitate the packet size distribution of that web page [148] [67]. Mainly, countermeasures at the network layer are more general and can also be applied in anonymous communication systems other than Tor [12].

Protocol Layer Protocol-level padding and dummy techniques can be applied to obfuscate traffic features and thereby hinder de-anonymizing attacks [43]. This should be done with a random amount of padding in order to improve security. Actually, secure shell, TLS and IPsec apply such protocol-level padding techniques to line plaintext up with block cipher boundaries, causing some obfuscation in the packet size [148]. Tor does not use circuit-level padding techniques because it can significantly decrease the performance of the circuit. But it is certainly possible to design protocol-level padding and dummy techniques in such a way that it reduces the overhead caused by itself.

Application Layer HTTP features and background traffic can be exploited to hide traffic features from user flows. HTTP pipelining and HTTP ranges can be used to modify the packet sizes of incoming and outgoing messages [80]. Additionally, the order of HTTP requests at the client side can be adjusted to alter the traffic pattern. Methods based on background traffic can be applied at the application level by loading a fake web page in the background while a user is browsing the target web page. Generally, countermeasures at the application layer are specific for some applications and cannot be widely applied [78].

From the description of the various countermeasures, we can conclude that there is an increasing need for hybrid techniques that can be deployed at multiple layers simultaneously [148]. This way various attacks of different types can be obstructed effectively. Moreover, the trade-off between security and performance need to be taken into account to provide an overall and secure solution to various attacks [22].

7 Ethical Vulnerabilities of Tor

Tor has been proven to be used for real evil content. We consider this an often neglected vulnerability. According to CloudFlare 94% of the requests that are done across the Tor network are intrinsically malicious [102]. Recently, evidence has been found that al-Qaeda and other terrorist organizations are using the Tor network to propagate their causes [34]. Due to this illegal content, Tor does not have an unquestionable moral high ground.

7.1 Values and Principles of Tor

As explained on the Tor Project website, the Tor Project is based on the values and principles of net neutrality, right to anonymity online, freedom of speech and the right to privacy [129]. This makes Tor a powerful tool for many morally right uses. The Tor network empowers freedom of speech to those living under repressive governments and in countries with restrictions on Internet [121]. Countries like China are known for censoring their citizens' access to the Internet; Tor provides a way around this control [128]. For informers, Tor provides a safe way to leak information to journalists. In fact, Edward Snowden released information on the NSA's PRISM program to news organizations via Tor [93] [109]. However, the values and principles of Tor also introduce some ethical issues that will be looked into more closely next.

7.2 Tor and Criminal Behaviour

Nowadays, freedom is one of the keywords of the internet. How far should this freedom go? Should we allow Tor users to perform illegal activities with a small chance of being convicted? The type of actions we are encouraging by providing anonymization services should be handled very carefully. This makes it necessary to reflect on the impact that Tor's freedom has on its users [32].

Tor lets people with evil intentions provide and use illegal services in a relatively secure way, meaning that it is not easy to trace the source of an illegal service and the location of the users of the service. Illegal activities can be hidden using Tor's hidden service protocol [124]. This makes them only accessible via the Tor network and they do not get indexed like the rest of the Internet. Among the hidden services of Tor are drug marketplaces, weapon marketplaces, contract killers, hackers and child pornography [149]. A recent study shows that about 44% of the websites hosted as a hidden service are of criminal intent [20]. This study also shows that more than 50% of hidden service addresses can be accessed by a port that is used by "SkyNet", a botnet that can be used for DDoS attacks or Bitcoin generation [59]. This suggests that a lot of servers that host hidden services are part of “SkyNet”.

The fact that Tor allows all kinds of communities to grow makes many Tor users uneasy. It may even undermine the network’s user base since the criminal activities on Tor might prevent potential users from using Tor. Merely using Tor can make you an attractive target for the government, even if you only use the service for legal purposes [93]. Because of this, users might worry to be associated with these illegal activities and therefore decide not to use Tor. As one of our team members noted on Tor during a discussion: “I would not use it at home.” Innocent users do not want to be generalized with people that use Tor for illegal purposes.

7.3 The Dark Web: from Snowden to Silk Road

Not all hidden services remain hidden. Silk Road is a well-known example of a hidden service that got exposed and whose accused administrator Ross Ulbricht got arrested [114] [71]. According to the FBI, the Silk Road servers were pinpointed to Iceland after a CAPTCHA field in the login page of the Silk Road website was not configured to be used via the Tor network [131]. However, several security experts claim that this story is not true [55] [33]. According to them, the FBI was able to locate the Silk Road servers by accessing a PHPmyadmin configuration file. Access to the configuration file would have been gained by password sniffing [55] [33]. More recently, in 2015 the FBI was able to seize the servers of “Playpen”, a bulletin board that was used to distribute child pornography [107]. The FBI then ran the service for two weeks after exposure but included identification software and were able to identify 1500 users of “Playpen”. This case started the debate whether government organizations are allowed to hack Tor users in order to identify them [18].

In 2013 former NSA contractor Edward Snowden released thousands of classified NSA documents on their PRISM Program to news organizations via Tor [93] [109]. The PRISM Program is the NSA’s surveillance program to track online communication for which they tapped many Internet users and nine internet firms, including Facebook, Google, Microsoft and Yahoo [81]. The documents that Snowden leaked revealed that Tor users have also been targeted by the NSA for years. In a presentation acquired by Snowden titled “Tor Stinks”, the NSA admits that it will “never be able to deanonymize all Tor users all the time” [70]. The files exposed the organization’s struggles with deciphering emails and encrypted chat logs on Tor, despite its abilities to hack into online communication systems. Snowden was charged with two counts of violating the Espionage Act and theft of US Government property [49]. On June 21, 2013, the U.S. Department of Justice dropped charges against Snowden and made an exception for political offenses. A subject of controversy, Snowden has been called a hero and a traitor by the public [25]. His actions have triggered debates over mass surveillance and the tension between national security and privacy. But that is exactly what Snowden aimed for: “I didn’t want to change society. I wanted to give society a chance to determine if it should change itself. All I wanted was for the public to be able to have a say in how they are governed.” [53] Snowden has always publicly supported Tor. According to him Tor is a critical technology in defense of our publication right. “The design of the Tor system is structured in such a way that even if the US Government wanted to subvert it, it couldn’t because it’s a decentralized authority [99].” Whether government organizations are allowed to use identification software on Tor and the consequences this has for Tor users is discussed in Paragraph 7.5.

7.4 Tor and Informed Consent

There is a major ethical issue going along with the principle of freedom of speech: uninformed consent of the Tor users [121]. In reality most users have no knowledge of what is being downloaded by their connection on the Tor network. It could include the illegal activities of someone else on the network. This is a major problem with the Darknet: the user’s nodes will often be used for the propagation of data that the majority do not approve of [128]. The idea of any depravity or illegality being routed through or being stored on your system makes many users uneasy. The counter side of a liberal view towards freedom of information and speech is that you cannot choose what to approve. There is enough criminal behavior on Tor that most users want no part in proliferating. Should society give up its standards for anonymity?

7.5 Tor and Violations of Privacy

From a technical perspective Tor can provide a high level of privacy. However, privacy might be reduced as Tor users are outlawed by the government. Can the government intrude the Tor network and its users’ computers to retrieve information about hidden services? This would also be an intrusion of Tor user’s privacy. What makes the Silk Road case interesting, is the discussion whether the privacy of Mr. Ulbricht has been violated during the exposure of Silk Road.

After the arrest of Mr. Ulbricht his lawyers accused the FBI of violating the right of privacy by the fourth amendment of their client when the FBI accessed his servers in Iceland [133]. The fourth amendment ensures the privacy of every American citizen [50]. According to the judge the FBI did not violate Mr. Ulbricht's fourth amendment right of privacy by hacking his servers in Iceland [58]. The judge argued that Mr. Ulbricht had not timely shown that the servers belonged to him. However, if Mr. Ulbricht would have shown that the servers were his, this might get him a conviction. This seems like an impossible situation for Mr. Ulbricht. However, according to the judge Mr. Ublricht should have released a statement that the servers where his, without risking that this statement would have been used against him in a court of law.

With hidden electronic data there is always the discussion if it is legal for law enforcement agencies to hack the users in order to identify them and if it is not a violation of the fourth amendment [57]. In the case of Silk Road the government argued that the servers of Mr. Ulbricht were foreign property where evidence of a criminal act could be found, which allows them to hack the servers. A warrant to search electronic data is allowed if the government cannot tell where the data is located [119]. However, Tor works in such a way that the government could have never known where the Silk Road servers where located without compromising the Tor network. Therefore government organizations are seeking a way to adjust the legal checks of the Fourth Amendment in order to be able to legally hack users connected to Tor [58]. The requested change would allow government organizations to acquire a warrant to search electronic data without providing any specific details as long as the target computer location has been hidden through a technical tool like Tor or a virtual private network [119]. This kind of discouragements of Tor use by governments or other third parties, for example by not protecting data sent over Tor by the fourth amendment, are labeled as *lawyer-based attacks.*

7.6 Illegality as a Consequence

Balancing the ethical and moral uses of Tor against the opportunity for misuse by criminals poses the question 'Should users be allowed to be anonymous online?'. It is important to note that criminal activity on Tor is a consequence, not a goal, of the network's commitment to freedom of speech. Just as large, growing cities attract criminals, it is unavoidable that the growth of Tor has made the network appealing for shady activities [93]. Therefore the use of Tor should be regulated in cooperation with law enforcement agencies. Some hidden services are immoral and should be punished, just like they would be in 'the real world'. And if that would not be the case anymore, if Tor would become a platform that does not make any judgments of its use, how do we then judge the acts of a Tor volunteer?

8 Financial Vulnerabilities of Tor

The Tor network is suffering from continuous starvation. While they have a loyal fan-base, they have no business model and no devoted Tor developers to rely on. In the last two years Tor's annual revenue was reported holding steady at about 2.5 million [127]. This is a moderate budget considering the number of Internet users involved in the network and the impact they have. Since 2012 the Tor network continues to grow steadily with an average rate of 18% per year in Tor relays [126]. This year Tor reported to support about 20 contractors and to have a user base that is up to several million people each day [127].

Aside from their donation campaign, Tor's services are made possible by more than 7,000 volunteers running as relay operators and by the huge amount of analysis Tor gets from research groups and individual programmers [127]. This constant peer review has become one of their strengths over the past years. As a result, the network's success and continuity mainly depends on the thousands of financial and non-financial volunteers that contribute to everything from system administration to global outreach and education [125]. Tor survives because of grants from foundations and individuals, but the important contributions are time, research and user commitment, instead of money. Therefore the more significant concerns are around global outreach, the social dynamics within the Tor community and collaborative practices.

The network's security and effectiveness may be harmed by the lack of staff and continuous code integration. In order to prevent this, a pile of the network's features need to be researched, implemented and deployed on a voluntary basis [11]:

- Hidden Service operators need to be made aware of the shortcomings of the Tor architecture in order to solve scaling issues and to improve its security.
- Researchers need to be introduced to various research topics and questions regarding anonymous communication services to stimulate further research in Tor's protocol, cryptography and the mechanisms of countermeasures against deanonymizing attacks.
- Software developers need to be introduced to the pile of coding tasks left to be done and the issues that involve Tor's codebase.

The statements above only touch issues that involve Tor's codebase or its security, but if Tor wants to be truly successful and influential it is also essential to build a dynamic ecosystem around the network. Extensions like privacy-preserving archiving systems, anonymous file sharing, easy-to-use publishing platforms and chat systems would give a boost to the network's growth and its institutional funding [11].

9 Conclusion

In this survey, we analyzed the technical, ethical and financial vulnerabilities of the Tor network.

Technical Vulnerabilities

The technical vulnerabilities of Tor are increasingly exploited by de-anonymizing attacks. We have elaborated on attacks that have been published, including end-to-end and single-end attacks proposed by active or passive adversaries. Based on their method and goal, we categorized the existing attacks on Tor into seven groups: *Correlation attacks*, *Congestion attacks*, *Timing attacks*, *Fingerprinting attacks*, *Denial of Service attacks*, *Supportive attacks* and *Revealing Hidden Services attacks.* Most de-anonymizing techniques that are applied in these categories are based on the concepts of traffic analysis, traffic confirmation and forgery of node identities. With traffic analysis an adversary simply observes inputs and outputs of the network and correlates their timing patterns. The attacker distinguishes data flow patterns from normal traffic caused by Tor's encryption. Similarly, with traffic confirmation an attacker uses the weaknesses of the constructed Tor nodes and other related services to control or observe the relays of both ends of a Tor circuit. Then he analyzes individual network links in order to validate his suspicion. More recent attacks use a totally different approach that is based on the forgery and manipulation of node identities and since these attacks often work in a distributed manner, they can pose a big threat to the Tor network.

Each academic and industrial agency leverages unique methods to attack Tor because of different goals. Therefore different countermeasures need to be taken to mitigate the risks posed by each attack. Generally, the countermeasures can be deployed from three perspectives: *network layer* (to de-anonymize the communication between users), *protocol layer* (to hide traffic features associated with users) and *application layer* (to remove traffic features from user flows). However, in the last thirteen years the attacks on Tor have grown to be more complex and effective. There is an increasing need for hybrid techniques that that can be deployed at multiple layers simultaneously - while taking the trade-off between security and performance into account - to provide an overall and secure solution to various attacks. Additionally, a new method has been developed by [143] to detect most de-anonymizing attacks more easily and to expose their malicious exit relays more quickly. By checking certificates and creating bait accounts on decoy connections, malicious nodes can be exposed and blacklisted as exit nodes in the Tor network. Detection is the first step in the obstruction of attacks so further research on this should be investigated.

The race of developing new attacks to compromise the anonymity of Internet users will continue. Therefore it is important to establish a strong theoretical background in both the academic and industrial world in order to be able to keep studying interactions between attacks and countermeasure mechanisms.

Ethical Vulnerabilities

The Tor Project is based on the principles of net neutrality, right to anonymity online, freedom of speech and the right to privacy. But these values of Tor also introduce a number of ethical issues. The freedom and cover Tor provides attracts criminal behavior. A recent study shows that about 44% of the websites hosted as a hidden service is of criminal intent [20]. Because of this Tor has developed a bad reputation, which may prevent potential users from using Tor since they do not want to be associated with illegal activities. Another ethical issue that goes along with freedom of speech is uninformed consent of users who might not always agree with the content that is being downloaded by their connection on the Tor network. These recent developments have forced law enforcement agencies to intrude the Tor network and its users to retrieve information about hidden services. This has caused lawyer-based attacks and adjustments on the fourth amendment to be a point of discussion. The ethical issues also raise questions about the privacy of Tor. Do these issues justify third parties to hack Tor and pose a threat to its privacy in the name of the public good? Illegality is a consequence, not a goal, of the network's commitment to freedom of speech. Therefore the services provided by Tor need to be regulated and controlled in cooperation with government organizations to guarantee safety and justice in the cyber world.

Financial Vulnerabilities

The Tor project deals with financial insecurities and continuous resource starvation because of lack of a business model. The network's success and continuity mainly depend on the thousands of financial and non-financial volunteers. To ensure the security and effectiveness of Tor in the future, the network's features need to be researched, implemented and deployed on a voluntary basis. Hidden Service operators and developers need to be made aware of the shortcomings of the Tor architecture and researchers need to be introduced to various research questions regarding anonymous communication services. This way further research and development can be stimulated to create a dynamic ecosystem around Tor and to boost the network's growth and institutional funding.

References

[1] Timothy G Abbott et al. "Browser-based attacks on Tor". In: *Privacy Enhancing Technologies*. Springer. 2007, pp. 184–199.

[2] A. Acquisti and P. Syverson. "On the economics of anonymity". In: *Financial Cryptography, 7th International Conference* (2003).

[3] M. Allman, V. Paxson, and E. Blanton. *TCP Congestion Control*. Standard. IETF, 2009.

[4] R. J. Anderson. "The eternity service". In: *In Proceedings of Pragocrypt '96* (1996).

[5] Tom Anderson et al. *UW CSE — Systems Research — Rocketfuel*. 2005. URL: http://research.cs.washington.edu/networking/rocketfuel/.

[6] Prateek Mittal Anupam Das Nikita Borisov and Matthew Caesar. "Re3 : Relay Reliability Reputation for Anonymity Systems". In: (2014).

[7] Arma. *Did the FBI Pay a University to Attack Tor Users?* 2015. URL: https://blog.torproject.org/blog/did-fbi-pay-university-attack-tor-users (visited on 03/21/2016).

[8] Arma. *Tor security advisory: "relay early" traffic confirmation attack*. 2014. URL: https://blog.torproject.org/blog/tor-security-advisory-relay-early-traffic-confirmation-attack/ (visited on 03/21/2016).

[9] Daniel Arp, Fabian Yamaguchi, and Konrad Rieck. "Torben: A Practical Side-Channel Attack for Deanonymizing Tor Communication". In: *ASIACCS*. 2015, pp. 597–602.

[10] Daniel Arp, Fabian Yamaguchi, and Konrad Rieck. *Torben: Deanonymizing Tor communication using web page markers*. Tech. rep. Technical Report IFI-TB-2014-01, University of Göttingen, 2014.

[11] Asn. *Hidden Services need some love*. 2013. URL: https://blog.torproject.org/blog/hidden-services-need-some-love (visited on 03/31/2016).

[12] Adam Back, Ulf Möller, and Anton Stiglic. "Traffic analysis attacks and trade-offs in anonymity providing systems". In: *Information Hiding*. Springer. 2001, pp. 245–257.

[13] Marco Valerio Barbera et al. "CellFlood: Attacking Tor onion routers on the cheap". In: *Computer Security–ESORICS 2013*. Springer, 2013, pp. 664–681.

[14] John Barker, Peter Hannay, and Patryk Szewczyk. "Using traffic analysis to identify the second generation onion router". In: *Embedded and Ubiquitous Computing (EUC), 2011 IFIP 9th International Conference on*. IEEE. 2011, pp. 72–78.

[15] Kevin Bauer et al. "Low-resource routing attacks against tor". In: *Proceedings of the 2007 ACM workshop on Privacy in electronic society*. ACM. 2007, pp. 11–20.

[16] Kevin Bauer et al. "On the optimal path length for Tor". In: *HotPets in conjunction with Tenth International Symposium on Privacy Enhancing Technologies (PETS 2010), Berlin, Germany*. 2010.

[17] Laurent Bernaille et al. "Traffic classification on the fly". In: *ACM SIGCOMM Computer Communication Review* 36.2 (2006), pp. 23–26.

[18] James Billington. *Tor user blames FBI for being caught during massive child pornography website sting*. 2016. URL: http://www.ibtimes.co.uk/man-blames-fbi-being-caught-during-massive-child-pornography-website-sting-1539526 (visited on 03/19/2016).

[19] Alex Biryukov, Ivan Pustogarov, and R Weinmann. "Trawling for tor hidden services: Detection, measurement, deanonymization". In: *Security and Privacy (SP), 2013 IEEE Symposium on*. IEEE. 2013, pp. 80–94.

[20] Alex Biryukov et al. "Content and popularity analysis of Tor hidden services". In: *Distributed Computing Systems Workshops (ICDCSW), 2014 IEEE 34th International Conference on*. IEEE. 2014, pp. 188–193.

[21] Stevens Le Blond et al. "One bad apple spoils the bunch: exploiting P2P applications to trace and profile Tor users". In: *arXiv preprint arXiv:1103.1518* (2011).

[22] Xiang Cai et al. "A systematic approach to developing and evaluating website fingerprinting defenses". In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM. 2014, pp. 227–238.

[23] Xiang Cai et al. "Touching from a distance: Website fingerprinting attacks and defenses". In: *Proceedings of the 2012 ACM conference on Computer and communications security*. ACM. 2012, pp. 605–616.

[24] Abdelberi Chaabane, Pere Manils, and Mohamed Ali Kaafar. “Digging into anonymous traffic: A deep analysis of the tor anonymizing network”. In: *Network and System Security (NSS), 2010 4th International Conference on*. IEEE. 2010, pp. 167–174.

[25] S. Chakrabarti. *Let me be clear – Edward Snowden is a hero*. 2015. URL: `http://www.theguardian.com/commentisfree/2015/jun/14/edward-snowden-hero-government-scare-tactics` (visited on 03/31/2016).

[26] S. Chakravarty, A. Stavrou, and A.D. Keromytis. *LinkWidth: A Method to Measure Link Capacity and Available Bandwidth Using Single-end Probes*. technical. Columbia University, 2008.

[27] S. Chakravarty, A. Stavrou, and A.D. Keromytis. “Traffic analysis against low-latency anonymity networks using available bandwidth estimation”. In: *Computer Security – ESORICS 2010*. 2010.

[28] Sambuddho Chakravarty, Angelos Stavrou, and Angelos D Keromytis. “Identifying proxy nodes in a Tor anonymization circuit”. In: *Signal Image Technology and Internet Based Systems, 2008. SITIS’08. IEEE International Conference on*. IEEE. 2008, pp. 633–639.

[29] Sambuddho Chakravarty et al. “On the effectiveness of traffic analysis against anonymity networks using flow records”. In: *Passive and Active Measurement*. Springer. 2014, pp. 247–257.

[30] Andrew Christensen. “Practical onion hacking: finding the real address of tor clients”. In: *FortConsults advisory http://www. f ortconsult. net/image s/pdf/Practical_Onion_Hacking. pdf* (2009).

[31] Bernd Conrad and Fatemeh Shirazi. “Analyzing the effectiveness of dos attacks on tor”. In: *Proceedings of the 7th International Conference on Security of Information and Networks*. ACM. 2014, p. 355.

[32] Henry Corrigan-Gibbs and Bryan Ford. “Welcome to the world of human rights: please make yourself uncomfortable”. In: *Security and Privacy Workshops (SPW), 2013 IEEE*. IEEE. 2013, pp. 1–4.

[33] Joseph Cox. *How Did the FBI Find the Silk Road Servers, Anyway?* 2014. URL: `http://motherboard.vice.com/read/how-did-the-fbi-find-the-silk-road-servers-anyway` (visited on 04/01/2016).

[34] Joseph Cox. *ISIS Now Has a Propaganda Site on the Dark Web*. 2015. URL: `https://motherboard.vice.com/read/isis-now-has-a-propaganda-site-on-the-dark-web` (visited on 04/01/2016).

[35] George Danezis. “Statistical disclosure attacks”. In: *Security and Privacy in the Age of Uncertainty*. Springer, 2003, pp. 421–426.

[36] George Danezis and Paul Syverson. “Bridging and fingerprinting: Epistemic attacks on route selection”. In: *Privacy Enhancing Technologies*. Springer. 2008, pp. 151–166.

[37] Norman Danner, Danny Krizanc, and Marc Liberatore. “Detecting denial of service attacks in Tor”. In: *Financial Cryptography and Data Security*. Springer, 2009, pp. 273–284.

[38] Sam DeFabbia-Kane. “Analyzing the effectiveness of passive correlation attacks on the tor anonymity network”. PhD thesis. Wesleyan University, 2011.

[39] R. Dingledine, N. Mathewson, and P. Syverson. “Tor: The Second Generation Onion Router”. In: *Proceedings of the 13th USENIX Security Symposium* (2004).

[40] Roger Dingledine and Nick Mathewson. “Tor protocol specification”. In: *URL: https://gitweb. torproject. org/torspec. git* (2008).

[41] Roger Dingledine, Nick Mathewson, and Paul Syverson. *Tor: The second-generation onion router*. Tech. rep. DTIC Document, 2004.

[42] John R Douceur. “The sybil attack”. In: *Peer-to-peer Systems*. Springer, 2002, pp. 251–260.

[43] Kevin P Dyer et al. “Peek-a-boo, i still see you: Why efficient traffic analysis countermeasures fail”. In: *Security and Privacy (SP), 2012 IEEE Symposium on*. IEEE. 2012, pp. 332–346.

[44] M. Edman and P. Syverson. “As-awareness in tor path selection”. In: *n CCS ’09: Proceedings of the 16th ACM conference on Computer and communications security*. ACM. 2009, pp. 380–389.

[45] Tariq Elahi et al. “Changing of the guards: A framework for understanding and improving entry guard selection in Tor”. In: *Proceedings of the 2012 ACM workshop on Privacy in the electronic society*. ACM. 2012, pp. 43–54.

[46] Nathan S Evans, Roger Dingledine, and Christian Grothoff. “A Practical Congestion Attack on Tor Using Long Paths”. In: *USENIX Security Symposium*. 2009, pp. 33–50.

[47] J. Feigenbaum, A. Johnson, and P. Syverson. “Probabilistic analysis of onion routing in a black-box model”. In: *Proceedings of the 2007 ACM workshop on Privacy in electronic society*. ACM. 2009, pp. 1–10.

[48] Joan Feigenbaum, Aaron Johnson, and Paul Syverson. “Preventing active timing attacks in low-latency anonymous communication”. In: *Privacy Enhancing Technologies*. Springer. 2010, pp. 166–183.

[49] P. Finn and S. Horwitz. *U.S. charges Snowden with espionage*. 2013. URL: `https://www.washingtonpost.com/world/national-security/us-charges-snowden-with-espionage/2013/06/21/507497d8-dab1-11e2-a016-92547bf094cc_story.html` (visited on 03/31/2016).

[50] *Fourth Amendment*. n.d. URL: `https://www.law.cornell.edu/constitution/fourth_amendment` (visited on 03/18/2016).

[51] Felix C Freiling, Thorsten Holz, and Georg Wicherski. *Botnet tracking: Exploring a root-cause methodology to prevent distributed denial-of-service attacks*. Springer, 2005.

[52] Xinwen Fu et al. “One cell is enough to break tor’s anonymity”. In: *Proceedings of Black Hat Technical Security Conference*. Citeseer. 2009, pp. 578–589.

[53] B. Gellma. *Edward Snowden, after months of NSA revelations, says his mission’s accomplished*. 2013. URL: `https://www.washingtonpost.com/world/national-security/edward-snowden-after-months-of-nsa-revelations-says-his-missions-accomplished/2013/12/23/49fc36de-6c1c-11e3-a523-fe73f0ff6b8d_story.html` (visited on 03/31/2016).

[54] Yossi Gilad and Amir Herzberg. “Spying in the dark: TCP and tor traffic analysis”. In: *Privacy Enhancing Technologies*. Springer. 2012, pp. 100–119.

[55] Robert Graham. *Reading the Silk Road configuration*. 2014. URL: `http://blog.erratasec.com/2014/10/reading-silk-road-configuration.html#.Vv59_Wh96Uk` (visited on 04/01/2016).

[56] Andy Greenberg. *Carnegie Mellon Denies FBI Paid for Tor-Breaking Research*. 2015. URL: `http://www.wired.com/2015/11/carnegie-mellon-denies-fbi-paid-for-tor-breaking-research/` (visited on 03/21/2016).

[57] Andy Greenberg. *Feds ‘Hacked’ Silk Road Without a Warrant? Perfectly Legal, Prosecutors Argue*. 2014. URL: `http://www.wired.com/2014/10/feds-silk-road-hack-legal/` (visited on 03/18/2016).

[58] Andy Greenberg. *Judge Rejects Defense That FBI Illegally Hacked Silk Road—On a Technicality*. 2014. URL: `http://www.wired.com/2014/10/silk-road-judge-technicality/` (visited on 03/18/2016).

[59] Claudio Guarnieri and Mark Schloesser. *Skynet, a Tor-powered botnet straight from Reddit*. 2012. URL: `https://community.rapid7.com/community/infosec/blog/2012/12/06/skynet-a-tor-powered-botnet-straight-from-reddit` (visited on 03/19/2016).

[60] Krishna P Gummadi, Stefan Saroiu, and Steven D Gribble. “King: Estimating latency between arbitrary internet end hosts”. In: *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*. ACM. 2002, pp. 5–18.

[61] Andrew Hintz. “Fingerprinting websites using traffic analysis”. In: (2003). URL: `http://freehaven.net/anonbib/cache/hintz02.pdf` (visited on 03/16/2016).

[62] Nguyen Phong Hoang, Yasuhito Asano, and Masatoshi Yoshikawa. “Anti-RAPTOR: Anti routing attack on privacy for a securer and scalable Tor”. In: *Advanced Communication Technology (ICACT), 2015 17th International Conference on*. IEEE. 2015, pp. 147–154.

[63] Nicholas Hopper. “Challenges in Protecting Tor Hidden Services from Botnet Abuse”. In: *Financial Cryptography and Data Security (Volume 8437 of the series Lecture Notes in Computer Science)*. 2014, pp. 316–325.

[64] Markus Huber, Martin Mulazzani, and Edgar Weippl. “Tor HTTP usage and information leakage”. In: *Communications and Multimedia Security*. Springer. 2010, pp. 245–255.

[65] Johnson Jansen Tschorsch and Scheuermann. “The Sniper Attack: Anonymously Deanonymizing and Disabling the Tor Network”. In: (2014). URL: `http://www.nrl.navy.mil/itd/chacs/sites/www.nrl.navy.mil.itd.chacs/files/pdfs/13-1231-3743.pdf` (visited on 03/20/2016).

[66] Aaron Johnson et al. “Users get routed: Traffic correlation on Tor by realistic adversaries”. In: *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM. 2013, pp. 337–348.

[67] Marc Juarez et al. “A critical evaluation of website fingerprinting attacks”. In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM. 2014, pp. 263–274.

[68] Sachin Kadloor et al. “Low-cost side channel remote traffic analysis attack in packet networks”. In: *Communications (ICC), 2010 IEEE International Conference on*. IEEE. 2010, pp. 1–5.

[69] Leo Kelion. *Tor attack may have unmasked dark net users*. 2014. URL: `http://www.bbc.com/news/technology-28573625`.

[70] S. M. Kelner. *Snowden Leaks Show NSA Targets Tor*. 2013. URL: `http://www.eweek.com/security/snowden-leaks-show-nsa-targets-tor.html` (visited on 03/31/2016).

[71] Alex Konrad. *Feds Say They’ve Arrested ’Dread Pirate Roberts,’ Shut Down His Black Market ’The Silk Road’*. 2014. URL: `http://www.forbes.com/sites/alexkonrad/2013/10/02/feds-shut-down-silk-road-owner-known-as-dread-pirate-roberts-arrested/#523cb4b34b63` (visited on 03/18/2016).

[72] Albert Kwon et al. “Circuit fingerprinting attacks: passive deanonymization of tor hidden services”. In: *24th USENIX Security Symposium (USENIX Security 15)*. 2015, pp. 287–302.

[73] M. Leech et al. *SOCKS Protocol Version 5*. Standard. IETF, 1996.

[74] Brian Neil Levine, Clay Shields, and N Boris Margolin. “A survey of solutions to the sybil attack”. In: *University of Massachusetts Amherst, Amherst, MA* (2006).

[75] Chenglong Li et al. “Super nodes in Tor: existence and security implication”. In: *Proceedings of the 27th Annual Computer Security Applications Conference*. ACM. 2011, pp. 217–226.

[76] Quangang Li, Peipeng Liu, and Zhiguang Qin. “A Stealthy Attack Against Tor Guard Selection”. In: (2015).

[77] Marc Liberatore and Brian Neil Levine. “Inferring the source of encrypted HTTP connections”. In: *Proceedings of the 13th ACM conference on Computer and communications security*. ACM. 2006, pp. 255–263.

[78] Z. Ling et al. “A New Cell-Counting-Based Attack Against Tor”. In: *IEEE/ACM Transactions on Networking (Volume:20, Issue: 4)*. IEEE. 2012, pp. 1245–1261.

[79] Karsten Loesing. *Analysis of circuit queues in tor*. 2009.

[80] Xiapu Luo et al. “HTTPOS: Sealing Information Leaks with Browser-side Obfuscation of Encrypted Flows.” In: *NDSS*. 2011.

[81] E. Macaskill and G. Dance. *U.S. charges Snowden with espionage*. 2013. URL: `http://www.theguardian.com/world/interactive/2013/nov/01/snowden-nsa-files-surveillance-revelations-decoded` (visited on 03/31/2016).

[82] N Boris Margolin and Brian Neil Levine. “Quantifying resistance to the sybil attack”. In: *Financial Cryptography and Data Security*. Springer, 2008, pp. 1–15.

[83] Nick Mathewson and Roger Dingledine. “Practical traffic analysis: Extending and resisting statistical disclosure”. In: *Privacy Enhancing Technologies*. Springer. 2004, pp. 17–34.

[84] Damon McCoy et al. “Shining light in dark places: Understanding the Tor network”. In: *Privacy Enhancing Technologies*. Springer. 2008, pp. 63–76.

[85] Jon McLachlan and Nicholas Hopper. “On the risks of serving whenever you surf: vulnerabilities in Tor’s blocking resistance design”. In: *Proceedings of the 8th ACM workshop on Privacy in the electronic society*. ACM. 2009, pp. 31–40.

[86] Prateek Mittal et al. “Stealthy traffic analysis of low-latency anonymous communication using throughput fingerprinting”. In: *Proceedings of the 18th ACM conference on Computer and communications security*. ACM. 2011, pp. 215–226.

[87] M. Mulazzani, M. Huber, and E. Weippl. “Anonymity and Monitoring: How to Monitor the Infrastructure of an Anonymity System”. In: *IEEE Transactions on Systems, Man and Cybernetics* (2010).

[88] S. Murdoch. “Hot or Not: Revealing Hidden Services by their Clock Skew”. In: *Proceedings of the 13th ACM Conference on Computer and Communication Security* (2006).

[89] Steven J Murdoch and George Danezis. “Low-cost traffic analysis of Tor”. In: *Security and Privacy, 2005 IEEE Symposium on*. IEEE. 2005, pp. 183–195.

[90] Steven J Murdoch and Piotr Zieliński. “Sampled traffic analysis by internet-exchange-level adversaries”. In: *Privacy Enhancing Technologies*. Springer. 2007, pp. 167–183.

[91] E. Y. Vasserman N. Hopper and E. Chan-Tin. “How much anonymity does network latency leak?” In: *ACM Transactions on Information and System Security,* ACM. 2010, pp. 13–28.

[92] Gabi Nakibly and F Templin. *Routing loop attack using IPv6 automatic tunnels: Problem statement and proposed mitigations*. Tech. rep. RFC 6324, August, 2011.

[93] W. Nicol. *A Beginner's Guide to Tor: How to navigate through the underground Internet*. 2016. URL: `http://www.digitaltrends.com/computing/a-beginners-guide-to-tor-how-to-navigate-through-the-underground-internet/` (visited on 03/18/2016).

[94] G. O'Gorman and S. Blott. "Large scale simulation of tor: modelling a global passive adversary". In: *Proceedings of the 12th Asian computing science conference on Advances in computer science,* Springer-Verlag. 2007, pp. 48–54.

[95] L. Overlier and P. Syverson. "Locating Hidden Servers". In: *IEEE Symposium on Security and Privacy* (2006).

[96] Lasse Overlier and Paul Syverson. "Valet services: Improving hidden servers with a personal touch". In: *Privacy Enhancing Technologies*. Springer. 2006, pp. 223–244.

[97] Lasse Øverlier and Paul Syverson. "Improving efficiency and simplicity of Tor circuit establishment and hidden services". In: *Privacy Enhancing Technologies*. Springer. 2007, pp. 134–152.

[98] Andriy Panchenko et al. "Website fingerprinting in onion routing based anonymization networks". In: *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society*. ACM. 2011, pp. 103–114.

[99] M. Perry. *This is What a Tor Supporter Looks Like: Edward Snowden*. 2015. URL: `https://blog.torproject.org/blog/what-tor-supporter-looks-edward-snowden` (visited on 03/31/2016).

[100] Karsten Loesing Philipp Winter Roya Ensafi and Nick Feamster. "Identifying and characterizing Sybils in the Tor network". In: (2016).

[101] Ryan Pries et al. "A new replay attack against anonymous communication networks". In: *Communications, 2008. ICC'08. IEEE International Conference on*. IEEE. 2008, pp. 1578–1582.

[102] Matthew Prince. *The trouble with Tor*. 2016. URL: `https://blog.cloudflare.com/the-trouble-with-tor/` (visited on 04/01/2016).

[103] Tor Project. *Tor Metrics*. URL: `https://metrics.torproject.org/` (visited on 03/29/2016).

[104] Michael G Reed, Paul F Syverson, and David M Goldschlag. "Anonymous connections and onion routing". In: *Selected Areas in Communications, IEEE Journal on* 16.4 (1998), pp. 482–494.

[105] Yakov Rekhter and Tony Li. "A border gateway protocol 4 (BGP-4)". In: (1995).

[106] Nicky van Rijsbergen and Kevin Valk. "Tor vs the NSA". on website. URL: `http://www.covert.io/research-papers/security/Tor%20vs%20NSA.pdf`.

[107] Mary-Ann Russon. *FBI crack Tor and catch 1,500 visitors to biggest child pornography website on the dark web*. 2016. URL: `http://www.ibtimes.co.uk/fbi-crack-tor-catch-1500-visitors-biggest-child-pornography-website-dark-web-1536417` (visited on 03/19/2016).

[108] Juha Salo. "Recent Attacks On Tor". In: *Aalto University* (2010).

[109] Bruce Schneier. "Attacking Tor: how the NSA targets users' online anonymity". In: *The Guardian* 4 (2013).

[110] Bruce Schneier. *How the NSA Attacks Tor/Firefox Users With QUANTUM and FOXACID*. 2013. URL: `https://www.schneier.com/blog/archives/2013/10/how_the_nsa_att.html` (visited on 03/10/2016).

[111] Andrei Serjantov, Roger Dingledine, and Paul Syverson. "From a trickle to a flood: Active attacks on several mix types". In: *Information Hiding*. Springer. 2002, pp. 36–52.

[112] Micah Sherr, Matt Blaze, and Boon Thau Loo. "Scalable link-based relay selection for anonymous routing". In: *Privacy Enhancing Technologies*. Springer. 2009, pp. 73–93.

[113] Yi Shi and Kanta Matsuura. "Fingerprinting attack on the tor anonymity system". In: *Information and Communications Security*. Springer, 2009, pp. 425–438.

[114] Gerry Smith. *Here's The Seemingly Ordinary Man Who Allegedly Ran The Internet's Biggest Black Market*. 2014. URL: `http://www.huffingtonpost.com/2013/10/02/silk-road-closed_n_4032116.html` (visited on 03/18/2016).

[115] Gerry Smith. *Meet Tor, The Military-Made Privacy Network That Counts Edward Snowden As A Fan*. July 18, 2013. URL: `http://www.huffingtonpost.com/2013/07/18/tor-snowden_n_3610370.html` (visited on 04/01/2016).

[116] R. Snader and N. Borisov. "Improving Security and Performance in the Tor Network through Tunable Path Selection". In: *IEEE Transactions on Dependable and Secure Computing* (2010).

[117] Robin Snader and Nikita Borisov. "A Tune-up for Tor: Improving Security and Performance in the Tor Network." In: *NDSS*. Vol. 8. 2008, p. 127.

[118] Kiran P Somase and Neeru Yadav. "Analysis of a new cell-counting-based attack against connection based Tor". In: *International Journal* 2.1 (2014).

[119] S. Spike. *FBI Seeks To Legally Hack You If You're Connected To TOR Or a VPN*. 2015. URL: https://yro.slashdot.org/story/15/01/20/1540241/fbi-seeks-to-legally-hack-you-if-youre-connected-to-tor-or-a-vpn (visited on 03/21/2016).

[120] Yixin Sun et al. "RAPTOR: routing attacks on privacy in tor". In: *24th USENIX Security Symposium (USENIX Security 15)*. 2015, pp. 271–286.

[121] J. Temperton. *Why the Tor browser and your privacy are under threat*. 2014. URL: http://www.expertreviews.co.uk/software/internet-security/1401061/why-the-tor-browser-and-your-privacy-are-under-threat (visited on 03/21/2016).

[122] Fabrice Thill. "Hidden Service Tracking Detection and Bandwidth Cheating in Tor Anonymity Network". PhD thesis. Master thesis. University of Luxembourg. 2014. URl: https://www. cryptolux. org/images/b/bc/(c it. on p. 96), 2014.

[123] Tor. *Tor FAQ: What are Entry Guards?* n.d. URL: https://www.torproject.org/docs/faq.html.en#EntryGuards (visited on 03/09/2016).

[124] Tor. *Tor: Hidden Service Protocol*. n.d. URL: https://www.torproject.org/docs/hidden-services.html.en (visited on 03/18/2016).

[125] Tor. *Tor Project: Annual Report 2009*. 2009. URL: https://www.torproject.org/about/findoc/2009-TorProject-Annual-Report.pdf (visited on 04/01/2016).

[126] Tor. *Tor Project: Annual Report 2012*. 2012. URL: https://www.torproject.org/about/findoc/2012-TorProject-Annual-Report.pdf (visited on 04/01/2016).

[127] Tor. *Transparency, Openness, and our 2014 Financials*. 2016. URL: https://blog.torproject.org/blog/transparency-openness-and-our-2014-financials (visited on 04/01/2016).

[128] Tor. *Who uses Tor?* 2014. URL: https://www.torproject.org/about/torusers.html.en (visited on 03/18/2016).

[129] Tor. *Why we need Tor*. 2014. URL: https://www.torproject.org/about/overview.html.en (visited on 03/18/2016).

[130] C. Troncoso and G. Danezis. "The bayesian traffic analysis of mix networks". In: *Proceedings of the 2009 ACM Conference on Computer and Communications Security* (2009).

[131] Liam Tung. *Silk Road site's CAPTCHA led FBI to main servers*. 2014. URL: http://www.cso.com.au/article/554443/silk-road-site-captcha-led-fbi-main-servers/ (visited on 03/18/2016).

[132] S. Ioannidis V. Pappas E. Athanasopoulos and E. P. Markatos. "Compromising anonymity using packet spinning". In: *In Proceedings of the 11th Information Security Conference*. ISC. 2008, pp. 287–296.

[133] Kate Vinton. *Alleged Silk Road Creator Ross Ulbricht's Fourth Amendment Rights Were Violated, Lawyers Say*. 2014. URL: http://www.forbes.com/sites/katevinton/2014/08/04/alleged-silk-road-creator-ross-ulbrichts-fourth-amendment-rights-were-violated-lawyers-say/#6449252d11a5 (visited on 03/21/2016).

[134] Tao Wang. "Website Fingerprinting: Attacks and Defenses". In: (2016).

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

[136] Tao Wang et al. "Effective attacks and provable defenses for website fingerprinting". In: *23rd USENIX Security Symposium (USENIX Security 14)*. 2014, pp. 143–157.

[137] Xiaogang Wang et al. "A novel flow multiplication attack against Tor". In: *Computer Supported Cooperative Work in Design, 2009. CSCWD 2009. 13th International Conference on*. IEEE. 2009, pp. 686–691.

[138] Xiaogang Wang et al. "A potential HTTP-based application-level attack against Tor". In: *Future Generation Computer Systems* 27.1 (2011), pp. 67–77.

[139] Xinyuan Wang, Shiping Chen, and Sushil Jajodia. "Network flow watermarking attack on low-latency anonymous communication systems". In: *Security and Privacy, 2007. SP'07. IEEE Symposium on*. IEEE. 2007, pp. 116–130.

[140] Xinyuan Wang and Douglas S Reeves. “Robust correlation of encrypted attack traffic through stepping stones by manipulation of interpacket delays”. In: *Proceedings of the 10th ACM conference on Computer and communications security.* ACM. 2003, pp. 20–29.

[141] Zachary Weinberg et al. “StegoTorus: a camouflage proxy for the Tor anonymity system”. In: *Proceedings of the 2012 ACM conference on Computer and communications security.* ACM. 2012, pp. 109–120.

[142] Rungrat Wiangsripanawan, Willy Susilo, and Rei Safavi-Naini. “Design principles for low latency anonymous network systems secure against timing attacks”. In: *Proceedings of the fifth Australasian symposium on ACSW frontiers-Volume 68.* Australian Computer Society, Inc. 2007, pp. 183–191.

[143] Philipp Winter et al. “Spoiled Onions: Exposing Malicious Tor Exit Relays”. In: *Privacy Enhancing Technologies (Volume 8555 of the series Lecture Notes in Computer Science).* 2014, pp. 304–331.

[144] Matthew K Wright et al. “Passive-logging attacks against anonymous communications systems”. In: *ACM Transactions on Information and System Security (TISSEC)* 11.2 (2008), p. 3.

[145] Matthew K Wright et al. “The predecessor attack: An analysis of a threat to anonymous communications systems”. In: *ACM Transactions on Information and System Security (TISSEC)* 7.4 (2004), pp. 489–522.

[146] Matthew Wright et al. “An Analysis of the Degradation of Anonymous Protocols.” In: *NDSS.* Vol. 2. 2002, pp. 39–50.

[147] Matthew Wright et al. “Defending anonymous communications against passive logging attacks”. In: *Security and Privacy, 2003. Proceedings. 2003 Symposium on.* IEEE. 2003, pp. 28–41.

[148] Ming Yang et al. “De-anonymizing and countermeasures in anonymous communication networks”. In: *Communications Magazine, IEEE* 53.4 (2015), pp. 60–66.

[149] Richard B Yetter. “Darknets, cybercrime & the onion router: Anonymity & security in cyberspace”. PhD thesis. Utica College, 2015.

[150] Sebastian Zander and Steven J Murdoch. “An Improved Clock-skew Measurement Technique for Revealing Hidden Services.” In: *USENIX Security Symposium.* 2008, pp. 211–226.

[151] Yang Zhang. “Effective attacks in the tor authentication protocol”. In: *Network and System Security, 2009. NSS’09. Third International Conference on.* IEEE. 2009, pp. 81–86.

[152] Y. Zhu and R. Betatti. “Anonymity vs. Information Leakage in Anonymity Systems”. In: *IEEE Int’l Conf. Distributed Computing Systems.* 2005.

[153] Y. Zhu et al. “Anonymity analysis of mix networks against flow-correlation attacks”. In: *GLOBECOM - IEEE Global Telecommunications Conference.* 2005.

[154] Y. Zhu et al. “Correlation-based traffic analysis attacks on anonymity networks”. In: *IEEE Transactions on Parallel and Distributed Systems* (2010).