DEPARTMENT OF ELECTRONICS AND COMMUNICATION
B M S COLLEGE OF ENGINEERING
(AUTONOMOUS COLLEGE UNDER VTU, BELGAUM)
BANGALORE – 560019

2021-22

6TH SEMESTER ALTERNATE ASSESSMENT
IN

COMPUTER COMMUNICATION NETWORK
(16EC6DCCCN)

TOR: THE ONION NETWORK

BY

SANATH N UPADHYAYA
1BM19EC139

Course Instructor
Prof. K. SUJATHA
Associate Professor, Dept of ECE

ABSTRACT

Decentralized internet is a people-powered kind of internet that makes the web more democratic as there is no hosting company. In this paper, we review about Tor, a top level protocol network for safely communicating over the internet. This provides a decentralized abstraction layer over the already exsited centralized networks. The powerful algorithm working behind Tor, the Onion routing protocol, will be explained. We will dicuss the various kinds of attacks that have been discovered over the years. Finally, we will look at alternative protocols and their workings.

TABLE OF CONTENTS

1. INTRODUCTION

The Internet is a vast layer of nodes and connections that connect billions of devices together. All of these devices, provide so much knowledge at our hands, that we currently call this time, the ‘ age of information’. Anything that happens around the world, can be known by anyone else at the touch of a finger. With so much data going about, this brings us the concept of internet privacy. Governmens and rogue agencies can steal the data and use it for monetary purposes. This prevents basic human rights for internet freedom.

Hence, we are in need of alternate technologies, that allows us to browse and surf the internet anonymously, benfitting from the interent, but preventing tracking. One such technologies is Tor, short for ‘The Onion Browser’ is a free and open source software for enabling anonymous communication.

It was developed in the 1990s by the US Naval Rsearch Laboratory. It was brainchild of mathematician Paul Syverson, Michel G.Reed and Dvaid Goldschlag. In 2004, the Naval Research Laboratory released the code for Tor under a free liscence. In 2006, Roger Dingledine and Nick Mathewson, and five others founded the Tor Project, a research education organization reponsible for maintaining Tor.

The Tor Project states that Tor users include "normal people" who wish to keep their Internet activities private from websites and advertisers, people concerned about cyber-spying, and users who are evading censorship such as activists, journalists, and military professionals.

Tor works on the real-world Internet, requires no special privileges or kernel modifications, requires little synchronization or coordination between nodes, and provides a reasonable tradeoff between anonymity, usability, and efficiency. Using Tor makes it more difficult to trace a user's Internet activity. Tor's intended use is to protect the personal privacy of its users, as well as their freedom and ability to communicate confidentially through IP address anonymity using Tor exit nodes.

Tor works based on the Onion Routing Protocol (hence the name), which we will discuss later in this paper. It mainly encryptes the data and then passes it across various voluntary nodes in a randomized manner. Tor is not meant to completely solve the issue of anonymity on the web. Tor

is not designed to completely erase tracking but instead to reduce the likelihood for sites to trace actions and data back to the user.

Tor can also provide advanced services, where servers can receive requests only through Tor and can be used to provide Onion Services / Hidden Services. Here, the server only remains in the Tor network and cannnot be accessed by regular users. Considering all the various attacks on Tor in the past few years, the odds of detection and de-anonymization is still low, abou one in two million. Hence, it can be assumed to be a pretty safe network.

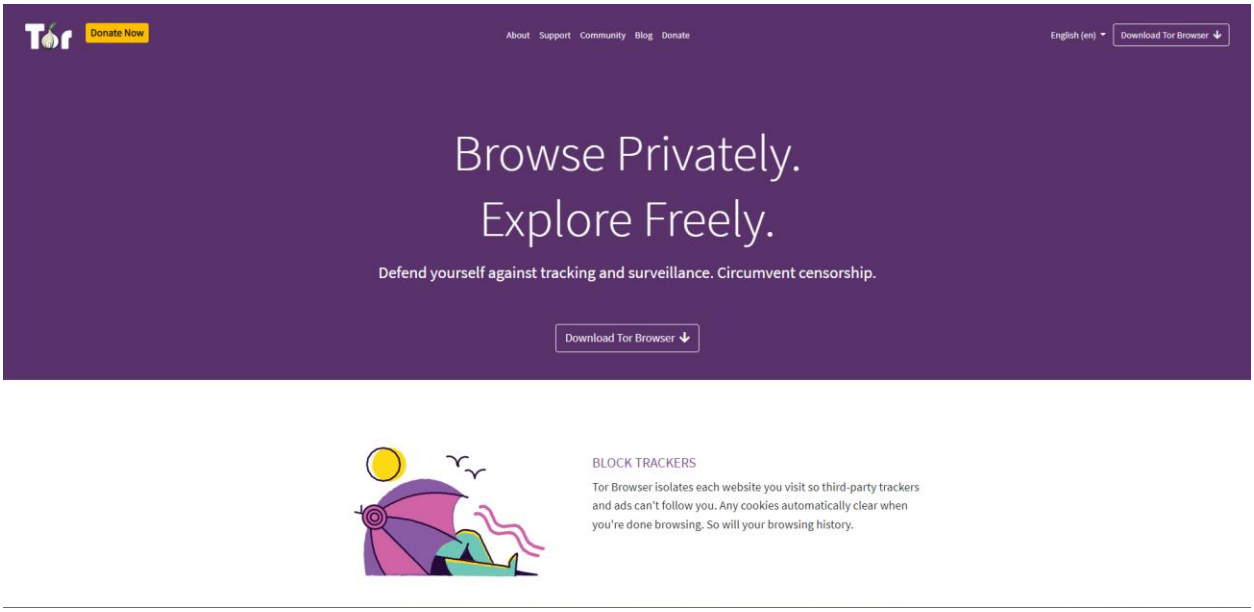

Figure 1 Home Page of Tor

2. LITERATURE SURVEY

Since Tor is an open source tool, there will be discussion of some of the original papers describing tor.

[1] Onion Routing for Anonymous and Private Internet Connections

This paper introduces the technique of onion routing which is the core principle of how Tor works.The authors also explain about the various non linear routing techniques that existed at that time. This paper is considered to be the paper that led to the origin of Tor.

[2] Tor: The Second-Generation Onion Router

This paper actually is the main paper for Tor . The authors introduce the working of Tor, its advantages, it's limitations and the security risks. This is an extention to the previous onion routing protocol and Tor is called second generation onion router. This paper also contains the threat model design of Tor and some of it's problems as highlighted by some of the earliest users.

[3] Anonymity with Tor: A Survey on Tor Attacks

In this paper, ths author go through an extensive number of attacks that have been lauchned agains the Tor network over the years since its conception. All types of attacks, from flash based attacks to browser fingerprinting attacks are discussed here. This gives us a pretty good understanding of the security vulnerabilites of the Tor network.

[4] Browser Based Attacks on Tor

In this paper, there is a discussion of a new browser based Attack on Tor, which uses Javascript and HTML to deanonymize a single user. The authors also discuss about the various other types of deanonymizing attacks such as flash attacks, etc. This attack also explains how a malicious node can modify HTTP traffic and use this to find out the IP of the sender.

[5] RFC 7686

This document defines the Onion Services or Hidden Services that can be offered on the Tor Network. It was defined by IETS in October 2015. This goes into detail of how the Hidden service protocols work and how certain services can be provided only here.

3. OPERATION OF TOR

The Onion Routing protocol is the main protocol behind the Tor Network. It was first proposed in mid 1990s in US. It was further developed and patented in 1998. In onion routing messages are encapsulated in several layers of encryption. The encrypted Data is transmitted through a series of network nodes called onion routers, each of which "peels" away a single layer, uncovering the data's next destination. When the final layer is decrypted, the message arrives at its destination. The sender remains anonymous because each intermediary knows only the location of the immediately preceding and following nodes.

The basic element of the network is a Node or a computer with internet connection. There are mainly 3 types of nodes:

- **Entry Node:** The client enters this network first
- **Relay Node:** The intermediate node, which only knows its previous and next nodes
- **Exit Node:** The final node which relays the data to the destination server

The final component of the Tor network are **Directory Servers.** The main purpose of these is to maintain the updates of the Tor nodes. Directory servers keep monitoring all the activities of the nodes and replace the unhealthy nodes with a healthy node for the stability of the tor network. [10] The following image gives a visual representation of how the network works

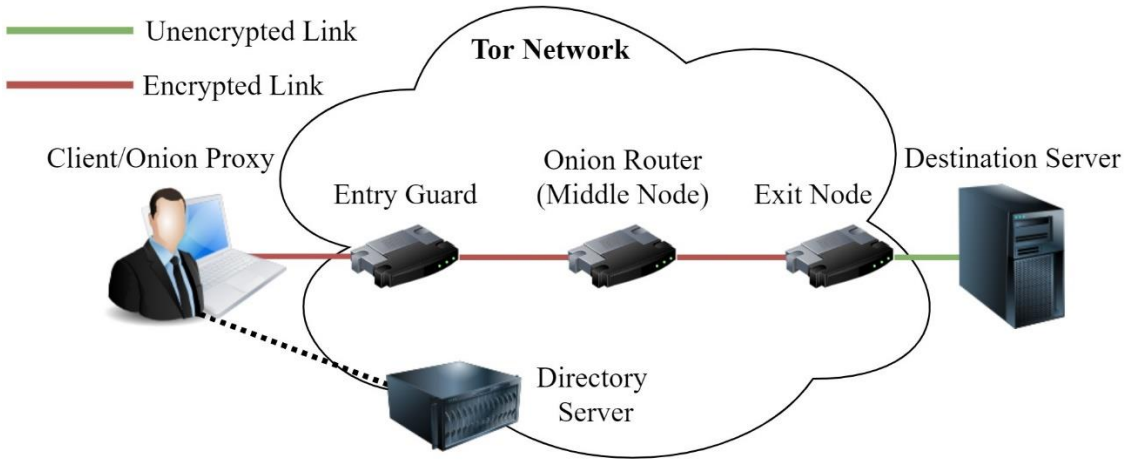

Figure 2

We will do an in-depth analysis on how this communication happens. All of these has been mentioned in [2]

Each normal user runs as a Onion Router (OR) without any external privileges. Each maintains a TLS connection to every other onion router. Each user runs local software called Onion Proxy (OP) to fetch directories, establish circuits across the network, and handle connections from user applications. These onion proxies accept TCP streams and multiplex them across the circuits. The onion router on the other side of the circuit connects to the requested destinations and relays data. Onion routers communicate with one another, and with users' OPs, via TLS connections with ephemeral keys. Using TLS conceals the data on the connection with perfect forward secrecy, and prevents an attacker from modifying data on the wire or impersonating an OR. Traffic is forwarded along connecitons in fixed-size cells. Each cell is 512 bytes long, consisting of header and payload. Each have a circuit identifier (circID) based on which they can be classified into two types: Control cells and Relay cells.

1. *Control Cells:* These carry control cell commands such as Padding, Create/Created and Destroy

Figure 3 The Control Cell

2. *Relay cells:* These cells have an additional header containing a streamID. They can be carrying multiple commands which are as follows: Relay Begin, Relay end, Relay teeardown, Relay connected, Relay extend/extended , Relay Truncate/Truncated, Relay Sendme and Relay Drop.

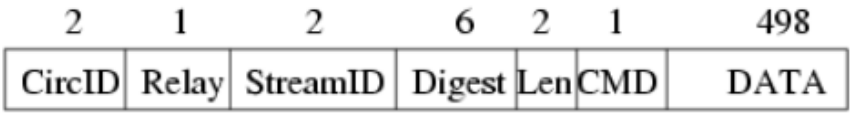

Figure 4 The Relay Cell

Onion Routing originally built one circuit for each TCP stream. In Tor, each circuit can be shared by many TCP streams. To avoid delays, users construct circuits preemptively. The circuits are constructured as follows

1. The client OP, Alice sends a create cell to the first node in her chosen path (Bob) She chooses a new circID CAB not currently used on the connection from her to Bob. The create cell's payload contains the first half of the Diffie-Hellman handshake (g^x), encrypted to the onion key of Bob.
2. Bob responds with a created cell containing gy along with a hash of the negotiated key $K=g^{xy}$. Here, the first connection is established.
3. To extend the circuit further, Alice sends a relay extend cell to Bob, specifying the address of the next OR (Carol) and an encrypted g^{x2} for her.
4. Bob copies the half-handshake into a create cell, and passes it to Carol to extend the circuit. (Bob chooses a new circID CBC not currently used on the connection between him and Carol.
5. When Carol responds with a created cell, Bob wraps the payload into a relay extended cell and passes it back to Alice. Now the circuit is extended to Carol, and Alice and Carol share a common key $K2 = g^{x2\,y2}$.
6. To extend the circuit to a third node or beyond, Alice proceeds as above, always telling the last node in the circuit to extend one hop further.
7. Once Alice has established the circuit (so she shares keys with each OR on the circuit), she can send relay cells. Upon receiving a relay cell, an OR looks up the corresponding circuit, and decrypts the relay header and payload with the session key for that circuit.
8. If the cell is headed away from Alice the OR then checks whether the decrypted cell has a valid digest. If valid, it accepts the relay cell and processes it. Otherwise, the OR looks up the circID and OR for the next step in the circuit, replaces the circID as appropriate, and sends the decrypted relay cell to the next OR.
9. OPs treat incoming relay cells similarly: they iteratively unwrap the relay header and payload with the session keys shared with each OR on the circuit, from the closest to farthest. If at any stage the digest is valid, the cell must have originated at the OR whose encryption has just been removed.

10. To construct a relay cell addressed to a given OR, Alice assigns the digest, and then iteratively encrypts the cell payload (that is, the relay header and payload) with the symmetric key of each hop up to that OR. Because the digest is encrypted to a different value at each step, only at the targeted OR will it have a meaningful value.
11. When an OR later replies to Alice with a relay cell, it encrypts the cell's relay header and payload with the single key it shares with Alice, and sends the cell back toward Alice along the circuit. Subsequent ORs add further layers of encryption as they relay the cell back to Alice.
12. To tear down a circuit, Alice sends a destroy control cell. Each OR in the circuit receives the destroy cell, closes all streams on that circuit, and passes a new destroy cell forward.

So this, is how circuit establishment and data communication occur in the Tor network. Here's a visual representaton of circuit establishment

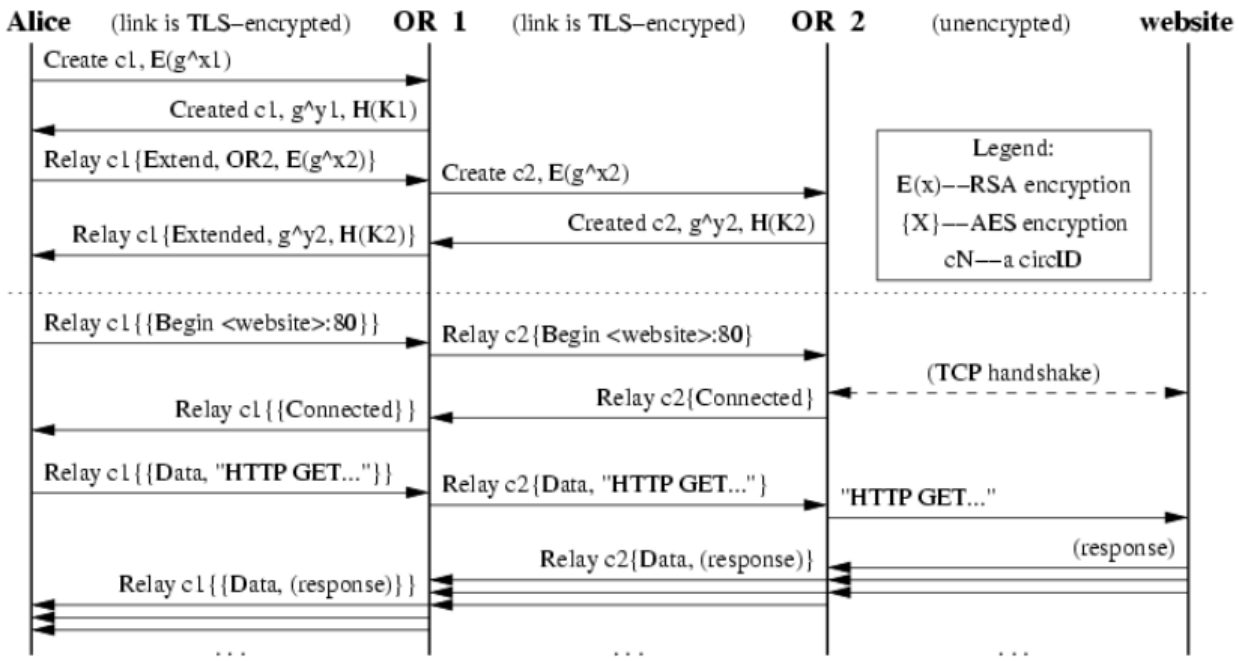

Figure 5

In brief, we first gave an overview of routing works in Tor and then we discussed the actual key exchange mechanism in the protocol. Hence, the working of the basic services of Tor has been successfully demonstrated.

4. ADVANCED TOPICS

Tor is the most famous networks that is used all across the world for various purposes with anonymity. Hence, it is also the victim of various attacks and tests by various governmental and research institutions over the decades. Some of the services provided by Tor has been improved. We give a brief intro to all these advanced topics here.

4.1 HIDDEN SERVICES

Tor can also provide anonymity to websites and other servers . Rather than revealing a server's IP address (and thus its network location), an onion service is accessed through its onion address, usually via the Tor Browser. The Tor network understands these addresses by looking up their corresponding public keys and introduction points from a distributed hash table (DFT) within the network. This type of anonymity protects against distributed DoS attacks: attackers are forced to attack the onion routing network because they do not know hidden service's IP address.

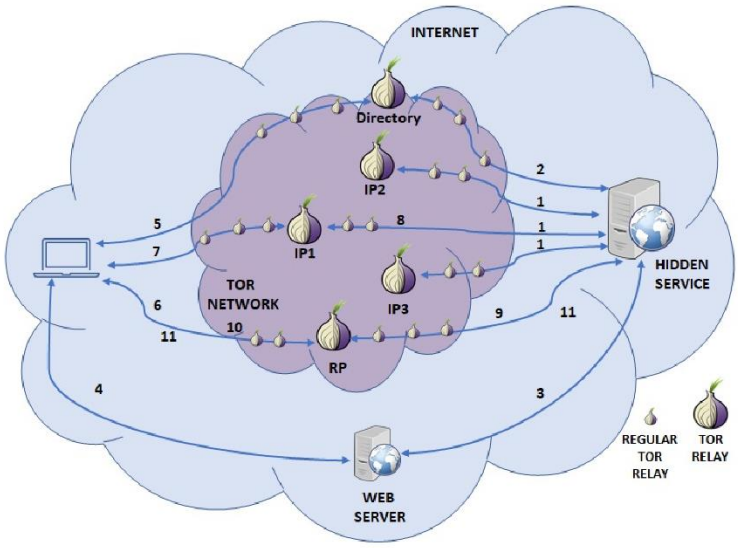

Figure 6 Example Architecture of a Hidden Service

4.2 ATTACKS AGAINST TOR

As mentioned earlier, various parties, including governments and law enforcement agencies, are interested in attacks that assist in de-anonymising the Tor Network, disrupting its operations, and bypassing is censorship circumvention mechanisms. There are many attacks on Tor over the

years, based on different vulnerabilities. Each one of the attacks mentioned above have subvariants which are divided in active and passive attacks.

The Tor Project has patched up most of the errors Some of these categories include De-anonymisation attacks, network disruption attacks, censorship attacks and many other general attacks like traffic confirmation attacks, correlation attacks etc.[4] More information can be gained from that paper.

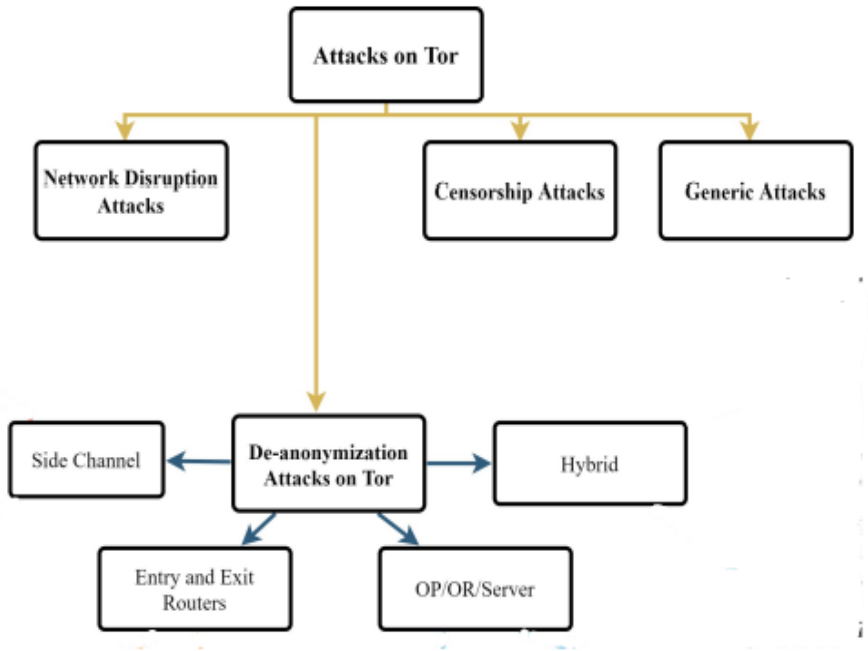

Figure 7 A diagram of attacks on Tor

To patch up all these attacks, the Tor Browser, provides various levels of security depending on individual user needs. It can be classified into:

1. **Standard (default):** At this security level, all browser features are enabled. However, this pertains to the lowest level of security
2. **Safer:** At this security level, many of the services are disabled. JavaScript is disabled on non-HTTPS sites. Mechanisms for displaying math equations are disabled.
3. **Safest:** At this security level, highest levels of security are provided. Here, JavaScript is disabled by default. Math symbols, fonts, icons are all disabled.

4.3 ALTERNATIVE PROTOCOLS

Even before the implementation of announcement of Tor, there were multiple other protocols that could be used for anonymization as mentioned in the original Tor paper.[2]

One of the alternatives is Mix Networks. These are routing protocols that create hard-to-trace communication by using a chain of proxy servers known as mixes which multiplexes multiple data from multiple senders , permutate them and send it out in random order to next destination. It was first mentioned by David Chaum in 1981. Applications that are based on this include anonymous remailers, key-based routing, etc. Mix-nets don't use encryption on layers as onion routing does. Hence, we can say that onion routing is a special implementation of mix-nets.

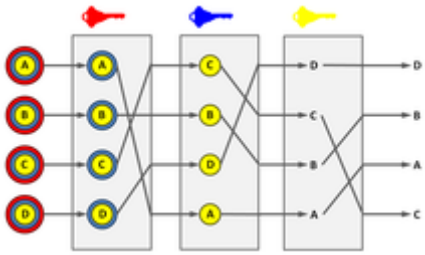

Figure 8 Simple implementation of mix net

Alternative routing protocols include Klein Bottle Routing, which claims to provide services and fill the gap between onion routing and mix network and can be widely employed in anonymous communication.[6].

Various other networks like IPFS, IP2, etc exist that can provide anonymous services, but none of them are as famous as Tor, or have the same amount of security.

5. CONCLUSION

In this paper, we discussed the Tor Network and how it works. The Onion Routing protocol was discussed here. An exact working of the Tor protocol as described in the original paper was summarized. Advanced topics of Tor such as hidden services and security attacks were touched upon which is a huge field of research in the current era.

In all these years, Tor has been a matter of intense controversy, being blamed for the usage of Illegal services in many countries. Tor is also used for illegal activities. These can include privacy protection or censorship circumvention,[30] as well as distribution of child abuse content, drug sales, or malware distribution.

The Tor Project states that Tor users include "normal people" who wish to keep their Internet activities private from websites and advertisers, people concerned about cyber-spying, and users who are evading censorship such as activists, journalists, and military professionals.

Overall, the network has it’s instances of good usage and illegal usage. But here, we only discuss the impressive technology behind it. We can successfully conclude that this tool can be used in the preservation of freedom of information.

6. BIBLIOGRAPHY

[1] Goldschlag, David & Reed, Michael & Syverson, Paul. (1999). Onion Routing for Anonymous and Private Internet Connections. Communications of the ACM. 42. 10.1145/293411.293443.

[2] Dingledine, Roger & Mathewson, Nick & Syverson, Paul. (2004). Tor: The Second-Generation Onion Router. Paul Syverson. 13.

[5] Karunanayake, Ishan & Ahmed, Nadeem & Malaney, Robert & Islam, Md Rafiqul & Jha, Sanjay. (2021). De-anonymisation attacks on Tor: A Survey. IEEE Communications Surveys & Tutorials. PP. 1-1. 10.1109/COMST.2021.3093615.

[4] Abbott, Timothy & Lai, Katherine & Lieberman, Michael & Price, Eric. (2007). Browser-Based Attacks on Tor. 184-199. 10.1007/978-3-540-75551-7_12

[5] https://datatracker.ietf.org/doc/html/rfc7686

[6] Peng, Kun & Nieto, Juan & Desmedt, Yvo & Dawson, Ed. (2006). Klein Bottle Routing: An Alternative to Onion Routing and Mix Network. 4296. 296-309. 10.1007/11927587_25.

[7] https://en.wikipedia.org/wiki/Tor_(network)

[8] https://en.wikipedia.org/wiki/Onion_routing

[9] https://thesecmaster.com/detailed-anatomy-of-the-tor-network-structure-of-the-tor-network

[10] https://hackernoon.com/how-does-tor-really-work-5909b9bd232c

[11] https://en.wikipedia.org/wiki/Mix_network