

Network Traffic Encryption Tools in short

Comparison of tools and use case & study literature review

Michail M

COMPUTER SCIENCE YEAR 3, COMPUTER AND NETWORK SECURITY

SCHOOL OF ARCHITECTURE, COMPUTING & ENGINEERING

BSc in Computer Science

Michail Markou

CN6003 – COMPUTER AND NETWORK SECURITY

UEL NUMBER

2020732

Date

11/11/2021

Network Traffic Encryption Tools in short

Michail Markou

University of East London

Contents

1.	Abstract.....	1
2.	Introduction	1
2.1.	Brief Encryption & Internet History	1
2.2.	Internet Service Providers.....	3
2.3.	Encrypted Network communication	4
2.3.1.	How it works	5
2.3.2.	VPN types and use cases.....	8
2.4.	Anonymous Network communication	13
2.4.1.	The Onion Routing - How it works [47].....	14
2.4.2.	Tor Hidden Services – How it works	17
2.4.3.	Vulnerability issues & Forensics.....	19
2.5.	The caveat law thing	20
3.	Security tools.....	22
3.1.	OpenVPN.....	22
3.1.1.	Brief.....	22
3.1.2.	Pros and Cons.....	22
3.2.	TOR.....	24
3.2.1.	Brief.....	24
3.2.2.	Pros and Cons.....	24
3.3.	Compare Results	25
4.	Real Case Research Analysis Literature Review	29
4.1.	The case Study	29
4.1.1.	Tor options	29
4.1.2.	torrc.....	29
4.1.3.	Relay node operation.....	29
4.1.4.	Hidden Service	31

References	33
Appendix	39
Glossary.....	39
Figure 1 Internet connectivity options from end-user to tier 3/2 ISPs (tiers are ISP's).....	4
Figure 2 https://hackernoon.com/decentralized-vpn-the-evolution-of-tor-hkv3uix	5
Figure 3 VPN Architecture Diagram with Routers/ VPN Concentrators	6
Figure 4 VPN Architecture Diagram with UTM Appliances.....	7
Figure 5 VPN Architecture Diagram using Wireless Controller.....	7
Figure 6 https://www.bitvpn.net/blog/centralized-vs-decentralized-vpn/	8
Figure 7 virtual network device interfaces used by VPN clients to establish virtual instances of physical networking connections.	11
Figure 8 https://tb-manual.torproject.org/about/	15
Figure 9 https://www.youtube.com/watch?v=QRYzre4bf7I	17
Figure 10 Normal Onion Routing and Hidden Services.....	19
Figure 11 World map of encryption laws and policies.....	21
Figure 12 https://www.gp-digital.org/world-map-of-encryption/	22
Figure 13 https://restoreprivacy.com/vpn/openvpn-ipsec-wireguard-l2tp-ikev2-protocols/	24
Figure 14 Tor relay config in torrc.....	30
Figure 15 Tor metrics	31
Figure 16 torrc for hidden services	32
Figure 17 Find your .onion address.....	33

1. Abstract

Traffic Encryption tools for anonymous¹ communications networks were born to protect the privacy, security of our internetwork communications, preventing tracking, censorship and traffic analysis they are by far the most famous in the encryption category that everybody uses for privacy without transparency².

In these articles, we present two industry-standard open-source software solutions Tor anonymization network & OpenVPN with a systematic literature review on the research made on Tor and VPN types presenting the state of the art, history, architecture, protocols and the different research challenges to be addressed. This comprehensive review has been developed from various selection articles and showcase main findings and use cases with each tool's advantages and disadvantages, limitations and Law issues, not all VPNs were created equally and they include many components e.g., encryption algorithms/ciphers, Encapsulation techniques, how they are going to push the traffic through the tunnel and many more strategy patterns. we are going to examine in high view the general idea, but to understand a specific tool first we are going to see how their underlying mechanics of a VPN works and Tor is based upon a VPN architecture.

Keywords: Network Security tools; Tor; VPN; OpenVPN; Traffic Encryption; hidden services; survey; decentralized peer to peer network infrastructure;

2. Introduction

2.1. Brief Encryption & Internet History

Cryptography for the first time appeared in ~1900 BC, it is used either to alter the meaning of a text with inscriptions and not hide it, changing its form for something more dignified (e.g., in Ancient Egypt) or to hide secret messages between government diplomacies, agencies and military for spy purposes and wars.

Fast-forwarding to around 100 BC, Julius Caesar used a form of encryption for messages to his army generals posted on the war front. This substitution cipher, known as Caesar cipher but as with all encryption systems it's some kind of mathematic formula and the way you design the system of this formula mat-

ters. Cesar cipher depends on the secrecy of the system and not on the encryption key. Once the system is known (secret crypto), these encrypted messages can easily be decrypted [1].

The first hardcore ciphers which used encryption keys were Enigma machines used in World War 1 and WWII. Eventually Alan Turing, a Cambridge University mathematician and logician, provided much of the original thinking that led to the design of the cryptanalytical bombe machines that were instrumental in eventually breaking the naval Enigma [2].

Our needs became our actions, Alan Turing was considered the father of

¹ Anonymization differs from encryption in a way that encrypted is something that can be seen as activity from a third party but anonymization instead is when we want to be invisible from any eye, that no activity took place.

² From traffic perspective not actual code implementation because they are open-source most of them.

modern computer science [3] [4] in the late ~1950s because cryptography as a science after him became a heavy thing but there was a catch the information could not move around without a physically humans' effort. Then was where an old idea arose again of connecting computers together creating a network, first steps towards global communication began in the 1960s [5] [6] [7]. Following that because of many wars at that time and government conspiracies a project called ARPANET [6] has been developed which ultimately evolved into what we know as the internet. The TCP/IP model came alive in ~1970s [8] [9] by 2 DARPA scientists and was used from project ARPANET a military network-grade communication level for exchanging successfully large data sizes without corruption but TCP/IP version 1; was meant for pure information locomotion without any encryption in mind at that time IBM in 1970s realized that their customers were demanding some form of encryption, so they formed a "crypto group" headed by Horst-Feistel. They designed a cipher called Lucifer which turned out to be called in our days Data Encryption Standard (DES) [1]. In 1981 TCP/IP version 4 was released [10] and adapted in the project ARPANET which allows multiple separate networks could be joined into a network of networks forming the Internetwork in short internet. The fourth iteration of TCP/IP IPv4³ like other previous versions of it had its drawbacks there was not built-in for encryption by default then where modules/plugins kicked in creating on top of it secure communications. In the

early ~1990s, IP-layer encryption [11] [12] came to life named Internet Protocol Security aka IPsec. IPsec is a security protocol suite that encrypts the packets of data flows to provide secure encrypted communication between two computers over an Internet Protocol network (IPv4), i.e., between a pair of hosts (host-to-host), between a pair of security gateways (network-to-network) or between a security gateway and a host (network-to-host). To achieve this encryption, it uses cryptographic keys like 3DES while establishing mutual authentication between agents at the beginning of a session and negotiation. It supports network-level peer authentication, data origin authentication, data integrity, data confidentiality (encryption), and replay protection (CIA triangle). In the Later 1990s got adopted the first VPN in history by Microsoft Peer-to-Peer Tunneling Protocol (PPTP) [13], VPN came from the idea of virtual circuit, the basic structure of the virtual circuit is to create a logical path from the source port to the destination port. This path may incorporate many hops between routers for the formation of the circuit. The final, logical path or virtual circuit aka VPN tunnel acts in the same way as a direct connection between the two ports. In this way, two applications could communicate over a shared network securely with privately bypassing all Internet traffic through the tunnel. This happens either on a computer interface client pc (remote access VPN) or Router interface (network-to-network). Thus, keeping the user data secure and private. At that point in time, the core prin-

³ Unlike IPv6 which has encryption built-in mechanisms.

ciple of Tor, Onion routing, was developed in the mid-1990s by Naval Research Laboratory employees, mathematician Paul Syverson, and computer

scientists Michael G. Reed and David Goldschlag, to protect intelligence communications online [14].

To conclude, history teaches us:

1. Privacy and secrecy were always an issue.
2. Technology is an echo of our lives.
3. The secrecy of your message should always depend on the secrecy of the key, and not on the secrecy of the encryption system. (This is known as Kerckhoffs's principle.)
4. Related to the above, always use ciphers that have been publicly reviewed and have been established as a standard. Using "secret crypto" is bad, because just like the Caesar cipher, once the system is known, all messages can be decrypted. For example, if your key is compromised, an attacker could access your messages; however, if the attacker can compromise the crypto system itself, they can obtain the plain text of every message (not just for a single person) encrypted by that system.

2.2. Internet Service Providers

In the beginning, the Internet was a bunch of universities, government and research facilities servers/hosts (research networks). As years go and the internet became the new on demand standard from futuristic and technological evolvement view/perspective soon business markets started to implement a solution for global communication services and joined the backbone⁴ of the Internet [15] (commercial networks). There were many ISPs with a different type of Internet connection from the last-mile of ISP perspective connection types (e.g., xDSL "synchronization") but the 3-layer hierarchical model⁵ of the main network consisted of VPN technologies and leased lines for business solutions the rest end-users

were either plain old telephone system (POTS/PSTN) or dial-up as analogue-digital circuits and in many cases ISDN for either of them which still was a public switched telephone network access [16] [17] [18].

The client consumer was using the internet above a public line meaning anyone could reach him and it wasn't encrypted definitely not anonymized [19].

The most common way for business clients to connect computers between multiple offices was by using a Private/Leased line either with point-to-point [20] implementation or Point-to-Multipoint [21] and an ISDN for public access which had more than 1 telephone

⁴ Backbone is the main road leading you to whole system/core network in Cisco hierarchical topology like a high road you can take any exit to main big cities or villages it's not a leaf/dead-end

⁵ A proper network consists of 3 Main Layers/tiers according to CISCO Front-mid-backhaul [77] (->) Access Network -> Aggregation/Distribution -> Mobile/fixed et al Core Layer in a Data Center [75] [76] [16].

incoming line phone calls ability⁶. Point-to-Point created their privacy/security circuit between two branches but if they wanted also public access, they should use a different connection like ISDN in most cases. Point-to-multipoint had the ability to connect more than 2 branches as long as a public one e.g., ISDN. From the ISP perspective, the leased line is based on MPLS VPN technology [18] [22] mainly with hub and spoke business network design [23] [24] due to it must pass through ISP from all branches and older days as frame-relays (hub n spoke). Leased lines, are private network connections meaning nobody can reach you from the outer world, that a telecommunications company could lease to its customers. Leased lines provided a company with a way to expand its private network beyond its

immediate geographic area. These connections form a single wide-area network (WAN) for the business. Though leased lines are reliable and secure, the leases are expensive, with costs rising as the distance between offices increases [25] [26]. Again, Privacy concerns despite being leased because the ISP is the main core where all traffic passes and you can't really tell if it is entering encrypted or unencrypted in their network because it happens in the edge Router of their network though this could be easily bypassed from them any time. It didn't take too long that VPN technology upscaled with solutions and end-user got different needs [27] than a large organization but Encryption and Anonymization were always different things.

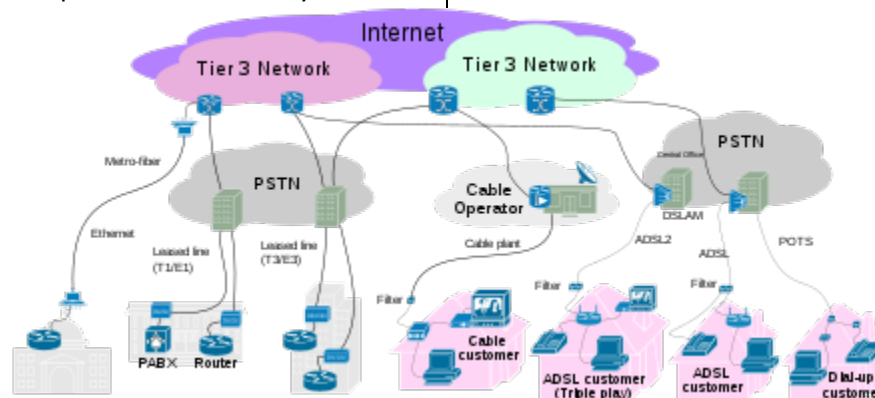


Figure 1 [Internet connectivity options from end-user to tier 3/2 ISPs \(tiers are ISP's\)](#)

2.3. Encrypted Network communication

VPN technology got a boost with a plethora of implementation techniques in the market started with many VPN's but all of them had the same approach of security at the implementation level but not In Business Policy level this is where they differ, this is the true privacy target e.g., log sharing with third parties.

⁶ ISDN was useful due to because its public (most of the times 1 active data line + BRI/PRI voice channels) [78] meaning an incoming call could be made but

point-to-point had no call ability due to had no public access as line directly.

A VPN system could typically use tunnels, firewalls and proxy servers and its architecture are either centralized or the new generation of VPN's WEB 3.0 based decentralized.

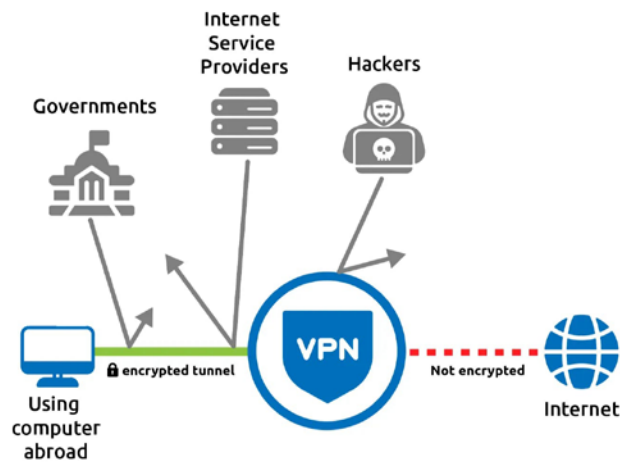


Figure 2 <https://hackernoon.com/decentralized-vpn-the-evolution-of-tor-hkv3uix>

2.3.1. How it works

Organizations in order to use the Internet as a private WAN, they need to overcome two main challenges. First, networks often communicate with a variety of protocols and not IP e.g., IPX so there is this issue of compatibility using cross-platform, but the Internet is designed to carry and process IP traffic. So, VPNs must implement a way to pass this traffic of non-IP protocols from one network to another. Second main thing is that data packets are crossing the internet in wide open without any encryption and/or privacy “in clear text” you may say.

Consequently, anyone who can see Internet traffic can also read the data contained in the packets. This is the main concern of privacy and security especially from a business perspective when confidential business information is crossing the

Internet. VPN's can eliminate these obstacles by using a strategy called tunnelling which instead of packets crossing the internet out in the open, data packets are first encrypted for security, and then encapsulated in an IP package by the VPN and tunnelled through the internet.

“To illustrate the concept, let's say we're running NetWare on one network, and a client on that network wants to connect to a remote NetWare server. The primary protocol used with traditional NetWare is IPX. So, to use a generic layer-2 VPN model, IPX packets bound for the remote network reach a tunnel initiating device - perhaps a remote access device, a router, or even a desktop PC, in the case of remote-client-to-server connections - which prepares them for transmission over the

Internet. The VPN tunnel initiator on the source network communicates with a VPN tunnel terminator on the destination network. The two agree upon an encryption scheme, and the tunnel initiator encrypts the packet for security. Finally, the VPN initiator encapsulates the entire encrypted package in an IP packet. Now, regardless of the type of protocol originally being transmitted, it can travel the IP-only Internet. And,

because the packet is encrypted, no one can read the original data. On the destination end, the VPN tunnel terminator receives the packet and removes the IP information. It then decrypts the packet according to the agreed-upon encryption scheme and sends the resulting packet to the remote access server or local router, which passes the hidden IPX packet to the network for delivery to the appropriate destination [28]."

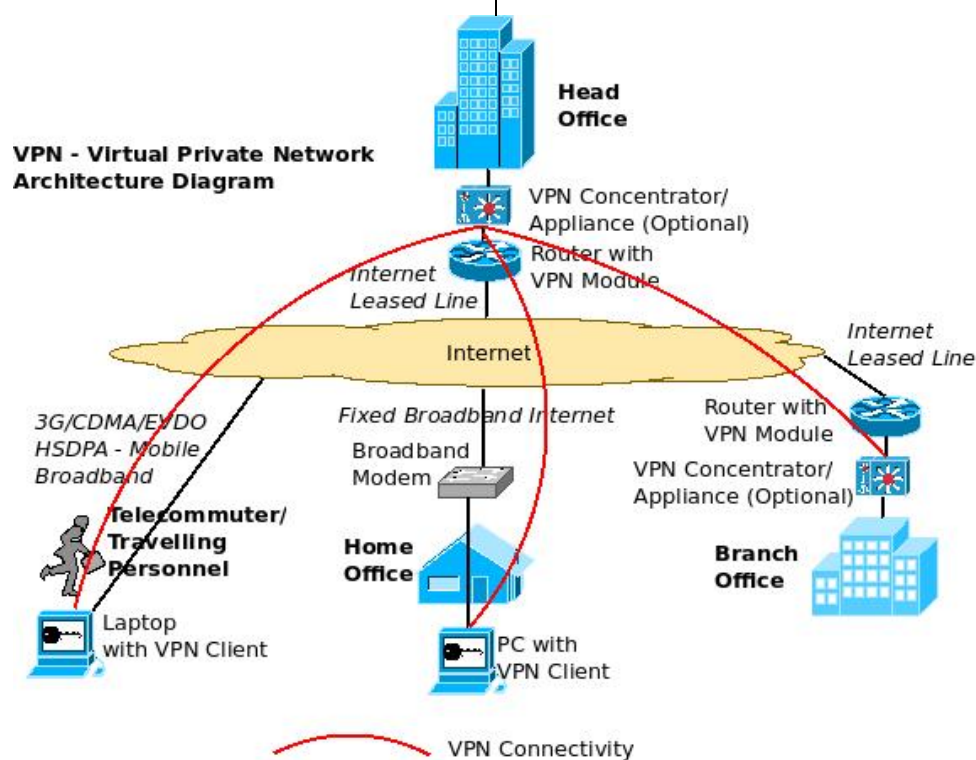


Figure 3 [VPN Architecture Diagram with Routers/ VPN Concentrators](#)

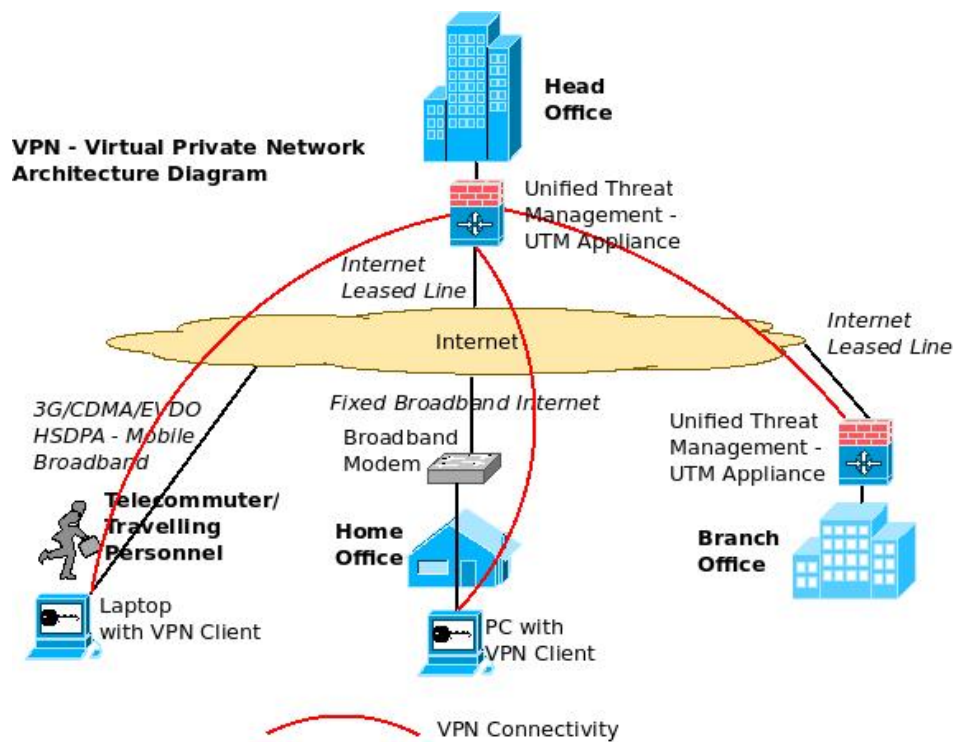


Figure 4 [VPN Architecture Diagram with UTM Appliances](#)

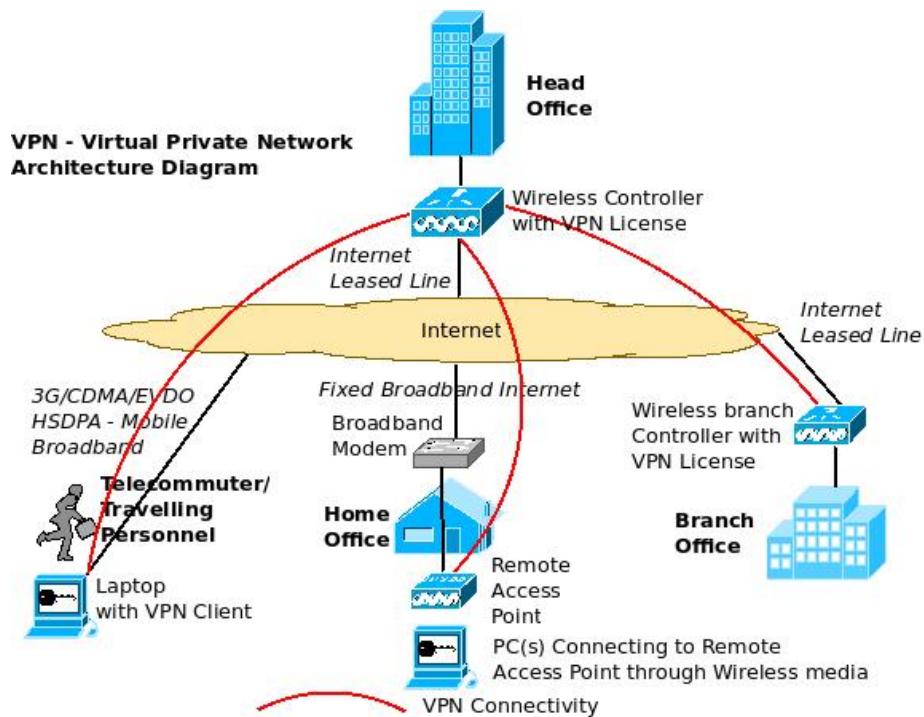


Figure 5 [VPN Architecture Diagram using Wireless Controller](#)

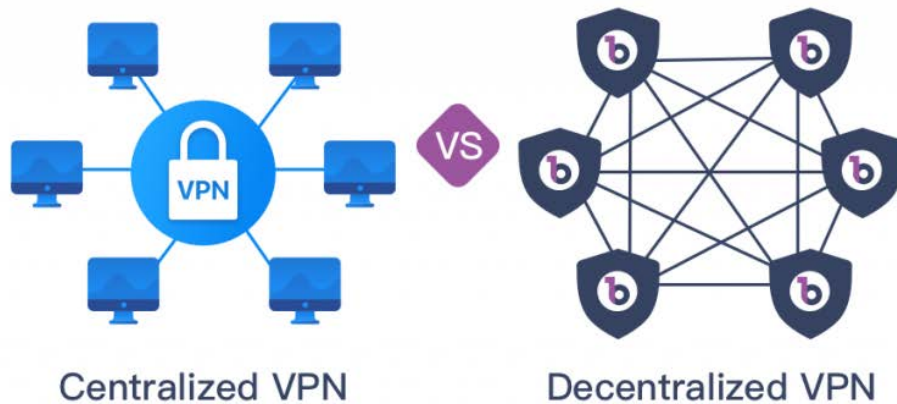


Figure 6 <https://www.bitvpn.net/blog/centralized-vs-decentralized-vpn/>

2.3.2. VPN types and use cases

2.3.2.1. *Features & Advantages* [29] [30] [31] [32] [33] [34]

- Centralized VPN (Classic)
- Decentralized VPN (blockchain based support)
- Encryption before actual data transmission
- Client-based or client-less (with specific software or just browser)⁷
- Data encryption of outbound interface (not just the browsers traffic)
- Auditing/Log information
- Automatic Fail-Over backup line
- SSL VPN's are better for disaster recovery/ business continuity as it allows for anywhere anytime access to the corporate networks for authorized users.
- VPN is scalable and cost-saving meaning can be extended to any branch/ individual user who has access to the Internet via Internet Leased Lines/ Fixed Broadband/ Mobile Broadband etc. Usually preferred over the leased line since leases are expensive, and as the distance between offices increases, the cost of leased line increase
- Saves time and expense for employees who commute from virtual workplaces
- VPN helps to centralize all IT resources and allows for centralized administration of critical IT resources at Data Center's in the head office
- Provide extended connections across multiple geographic locations without using a leased line.
- Improving security mechanism of data using encryption techniques.

⁷ Usually, software client is downloaded by redirecting automatically when connecting to a "VPN" portal.

- Provides flexibility for remote/travelling offices and employees to use the business corporate resources in the intranet over an existing Internet connection as if they're directly connected to the network with a VPN client software access (with SSL VPN technologies, even the client software may not be required, just a standard browser).
- VPN's allow for selective access to vendors and partners via an external portal in order to increase their efficiency and get work done faster (extranet).
- Stop unwanted Data collection (e.g., tracking, ads).
- Unblock/bypass region restrictions for digital content (e.g., due to country policy)
- VPN provider could send your traffic through Tor by your demand even without you using any special Tor browser
- Possible End-to-end-encryption (E2EE)⁸ e.g., plain text to ciphered text

2.3.2.2. *Disadvantages*

- VPN connection is slow.
- No Anonymization except if its dVPN
- VPN connection stability is mainly in control of the internet scalability, factors outside an organization control.
- Differing VPN technology may not work together due to immature standards.
- Because the connection travels over public lines, a strong understanding of network security issues and proper precautions before VPN deployment are necessary.

2.3.2.3. *Types*

- Remote Access VPN (for end-users e.g., host-to-server/network/site)
 - Layer 2 VPN
 - Layer 3 VPN
 - Layer 4 VPN
- Site-to-Site VPN (for businesses, router-to-router/network-to-network)
 - L2 VPN
 - L3 VPN

2.3.2.4. *Usability & Terminology*

- **Tunnel:** Tunneling or encapsulation is a technique of packaging one network packet insider another. The encapsulated packet is called the tunnelled packet and the outer is called the transport packet. An outer IP header is added to the original header and between the two of these headers is the security information specific to the tunnel. The outer header includes the source and the destination nodes/end-points of the tunnel while the inner header identifies the original sender and the recipient of the packet [35].

⁸ End-to-end-encryption depends if the host that connects to a remote network or a host and this network or host (remote) implements any encryption at all to the received traffic or the target-service outside of exit node VPN is using encryption, so the user must be aware about technology implementation at Layer 3 (e.g., IPsec) and/or Layer 7 (e.g., https)

- **Client-to-site/Remote access/user:** VPN allows a user to connect to a private network and access its services and resources remotely. The connection between the user and the private network happens through the Internet and the connection is secure and private. It is useful for business users as well as home users.

Home users, or private users of VPN, primarily use VPN services to bypass regional restrictions on the Internet and access blocked websites. Users' consciousness of Internet security also uses VPN services to enhance their Internet security and privacy.

For example, a corporate employee, while travelling, uses a VPN to connect to his/her company's private network and remotely access files and resources on the private network [27].

- **Site-to-Site:** VPN is also called as Router-to-Router VPN and is mostly used in corporates. Companies, with offices in different geographical locations above a Wide Area Network (WAN) link and they, use Site-to-site VPN to connect the network of one office location to the network at another office location. When multiple offices of the same company are connected using a Site-to-Site VPN type, it is called as Intranet-based VPN. When companies use Site-to-site VPN type to connect to the office of another company, it is called as Extranet based VPN. Basically, Site-to-site VPN create a virtual bridge between the networks at geographically distant offices and connect them through the Internet and maintain secure and private communication between the networks. Since Site-to-site VPN is based on Router-to-Router communication, in this VPN type one router acts as a VPN Client and another router as a VPN Server. The communication between the two routers starts only after authentication is validated between the two [27] [36].
- **Layer 4 VPN (network independent):** Suitable for roaming with mobile devices (we are not referring to ISP's mobile network perspective). A Layer 4 VPN is network-independent as it sits above the network layer 3. The session stays open regardless of whether or not the client switches between networks, maintaining a continuous connection for both the user and the application. Since there are no specific requirements for network infrastructure equipment, a Layer 4 VPN is thus more reliable and robust [37].
- **Layer 3 VPN (TUN)⁹:** General idea is an interface that belongs to at least L3 Device-ability does Routing with the remotely connected broadcast domain with a secure tunnel between them you get different IP range addresses in a different subnet. This has lower traffic overhead since the broadcast packet will not pass at least by default (it depends on how the technology of Layer 3 VPN is implemented on code) [38] [39]. (Figure 7)
- **Layer 2 VPN (TAP):** General idea is an interface that belongs to at least L2 device has L2 abilities and does bridging segments like switches. This carries whole Layer

⁹ It is worth noting that TUN/TAP devices are only used by certain VPN protocols (such as OpenVPN and others) but not from some like IKEv2/IPsec but again you can implement the idea of TUN/TAP with either L2 IPsec or L3 IPsec e.g., L2TP/IPsec or IKEv2/IPsec

2 mechanics on the VPN tunnel like broadcast, overhead traffic to pass but may have scaling issues again depends on how it is implemented in code to work how many features of an actual Layer 2 network will bring [38] [39]. (Figure 7)

- **Centralized VPN:** Classic/Traditional/Direct VPN works like a proxy, there is one main Provider which stores and log your online activity (or not). The drawback is that you depend on Provider's judgement policy for sharing your data or if it gets an attack the centralized approach will be a big problem for your privacy. Another concert of this option is the closed source code of this nature of an VPN provider you really can't tell what's going on behind the scenes. This idea of a centralized nature is what called WEB2 architecture where a business centralized entity can have all the power of your data [33] [40] [41]. (Figure 6)
- **Decentralized VPN (dVPN):** The nature of this peer-to-peer mesh network structure prevents any type of logging from being possible it works ~like TOR network each node hope add a layer of encryption without anyone knowing the initial source, only knows who was previous or next node as packet hop/cross from one node to another. It is an open-source code which means transparency of its actions and support Blockchain evolutionary technology which is implemented in WEB3 architecture but at the same time every node in contrast with Tor its paid you pay per minute every node your traffic crosses usually with GAS on Ethereum blockchain. Everyone plays role and can be the server by owning something, in a blockchain environment you are not controlled by a global main business entity's action so there is transparency to the network (because you can read the open-source code) but not its transactions, this brings anonymization to any middle node level attack except the first hop node or last exit node. As with most P2P infrastructures, the more participants which join the network, the stronger and more robust it becomes. [33] [40] [41] [42] [43]. (Figure 6)

TUN and TAP in the network stack

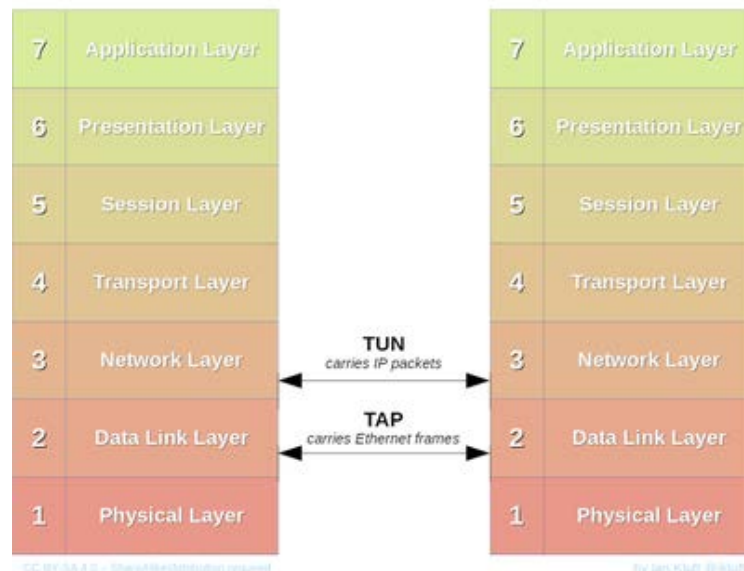


Figure 7 [virtual network device interfaces used by VPN clients to establish virtual instances of physical networking connections.](#)

2.3.2.5. *Famous Types of VPN protocols [27] [29] [44] [45] [39] [32]*

The above two VPN types are based on different VPN security protocols that “sit” on top of VPN terminology. Each of these VPN protocols offer different features and levels of security, and are explained below:

- IPsec
 - Layer 3 OSI
 - Network firewalls’ policy possible to deny it
 - 2 operation modes: Transport or tunnelling mode
 - Not a protocol but a framework/suite of protocols
 - CIA and anti-replay
 - Encryption algorithms (e.g., 3DES, Diffie-hellman)
 - Site-to-site VPN tunnel
 - Client-to-site (remote user) VPN tunnel
 - Site-to-site access (requires host-based clients)
 - Between two servers to authenticate and/or encrypt traffic
- L2TP
 - Extension of PPTP
 - Tunnel Layer 2 Traffic over layer three connections
 - “Bridge” Remote Lan’s – same subnet
 - No encryption by-default
 - Can use IPsec
- PPTP
 - obsolete
- SSL VPN & TLS
 - App-specific encryption (e.g., SMTP, FTP, XMPP, NNTP, HTTP/s)
 - Application Layer 7
 - TLS new version of SSL
 - No need for a special software client for most cases (browser-based/clientless with an optional thin client) - remote access
 - Portal through a web browser to access application
 - Used everywhere because HTTPS traffic is most allowed (wi-fi public hotspots)
- OpenVPN (more on this later)
 - Open-Source
 - Point-to-point
 - Site-to-site
 - Custom security based on SSL/TLS
- SSH
 - Any network service encryption
 - Client-to-site/server
 - Used in File Transfer Protocols mechanisms
- VPLS

- L2 bridging is used in ISP's or big corporate disjoint networks. (Old method was L2TP. A use case could be e.g., from ADSL end-user to Broadband Remote Access Server (BRAS) [46]).
- MPLS L3 VPN
 - Communication between business branches with encryption and traffic forwarding between them on ISP's side is used above any first-mile from business/customer perspective link typically leased line.

Et al.

Features	SSL	IPSEC
<i>Identity authentication</i>	One-way authentication Mutual authentication Digital certificate	Mutual authentication Digital certificate
<i>Encryption</i>	strong	Very strong
<i>Encryption type</i>	Key length 40 bits to 128 bits	Key length 56 bit to 256 bits
<i>Full security</i>	E2E	Network edge to the client, encryption only between the VPN gateway
<i>Access</i>	Easy Selection at any time, anywhere, No Firewall blocking policy	Access restrictions to the defined controlled user access. Firewall policy may deny it
<i>Installation</i>	Easy	Complex
<i>Application</i>	Web, File sharing, Email	All protocols based on IP service
<i>User</i>	Customers, partners, suppliers, users, remote users more	Internal users
<i>network</i>	Operates at layer 4-7	Operates at layer 3
<i>Gateway location</i>	Usually deployed behind the firewall	Usually implemented on the firewall
<i>Scalable</i>	Easy configuration and expansion	Easy expanding at the server end but difficult for the client
<i>Cost</i>	Low	High

2.4. Anonymous Network communication

Tor consists of a network of relay servers that are run by volunteers all over the world. When people hear about The Onion Router (Tor) which is an implementation of onion routing (and it is called this way because onions have layers and this networking protocol also has layers) the first thing that comes in mind is the Dark-web aka Tor hidden services, Deep-web, Services-web and is a different part from how Tor works in a conceptual term of encryption and traffic forwarding. Tor differs from confidentiality which is usually associated with encryption (e.g., encrypting messages) and even if some people see and get these messages can't read what it is but

sometimes, we don't want people to see at all in the first place that we sent messages. Below in summary everything is explained from previous statements. [47]

2.4.1. The Onion Routing - How it works [47]

Tor is a circuit-based low-latency anonymous communication service [48] basically an overlay network of virtual tunnels over a public network and not inherently a peer-to-peer network. That allows you to improve your privacy and security on the Internet (anonymity) on TCP-based applications like web browsing, secure shell and instant messaging et al. Tor works (by bouncing connections to different routers so they're hard to track and provides anonymity) sending your traffic through three random servers (also known as *relays*) in the Tor network from through the Browser to host adapters interface encapsulated from application Layer in several layers of encryption, analogous to encapsulation in the OSI layer 7 model [49]. The resulting 'onion' is a fully encapsulated message which been then routed by onion routers (partitioned in equally 512B cell messages in onion network in order to create indistinguishable traffic movement for making tracking almost impossible at least inside the Tor network), a series of nodes in a network with each node peeling away a layer of the 'onion'

and therefore uncovering the data's next destination. The last relay in the circuit (the "exit relay") then sends the traffic out onto the public by stripping the last security layer of encryption 'onion' (so you might be getting a plaintext of the original message) to the Internet while keeping the original author anonymous because each node in the networks is only aware of the preceding and following nodes in the path (except the first node that knows who actually the sender is, but doesn't know the final destination) without any extra added encryption except if the original socket was made upon an encrypted-based-service (e.g., HTTPS), so the last exit-node is basically to your behalf is acting as a proxy.

This has led to 'attacks' on which the NSA runs servers in order to attempt to be the first and last nodes in the network. If the NSA server is the first node, it knows where the message is from. If the NSA server is the last node, it knows the final destination and what the message says [50].

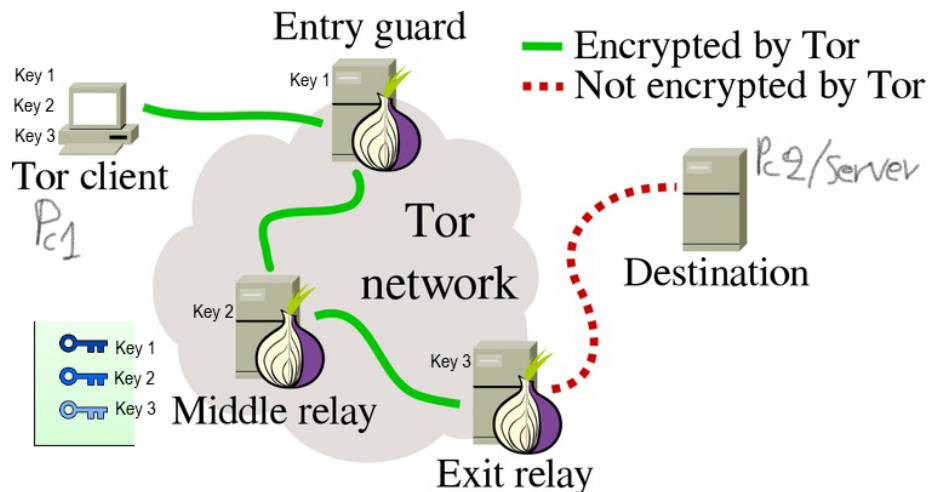


Figure 8 <https://tb-manual.torproject.org/about/>

Relay 1 (Entry-guard) knows who you are, but not where you're going.

Relay 2 (middle node) only knows where to forward it in the network.

Relay 3 (exit-node) knows where you're going, but not who you are.

2.4.1.1. Example

The image above (Figure 8) illustrates a user browsing to different websites over Tor. The middle computers represent relays in the Tor network, while the three keys represent the layers of encryption (multiple layers in additive form) between the user and each relay/server¹⁰ so it confuses the spy what's going on. To expand in further detail no one in this network knows anything about the whole

connection, they just know what's before them and what's after them. Let's assume PC1 (top left) wants to send traffic to PC2 (webserver Right) the first thing that happens is, it gets a list of guard nodes to choose from and establishing shared symmetric keys (advanced encryption standard aka AES keys) encryption schema with these 3 nodes (Key1, Key2, Key3) by creating a circuit (a pre-

¹⁰ Anyone can be a part of relay network and the client machine is not only isolated in Tor Network it can do other things as well if it is capable of. With a software setup on client machine or laptop you can be either middle relay node or exit-node or both at the same time but not entry-guard and exit-node together now being the Guard Entry node is a more specific issue and must be trusted because this is where literally you could be spied from third parties and to alert them that you are using Tor also to point out an Entry Guard node can't be at the same time

exit-node because is like this concept. If you use a proxy or VPN, you have some anonymity, but also a single point of failure. If someone compromises the machine doing the relaying, they know your IP and the IP of where you're going (you have been deanonymized) so basically sensitive security mechanism can't be played at the same time from the same server. [79]. Your internet connection must be good too.

built path with at least 3 hops)¹¹
¹² using key exchange (Diffie-Hellman) the first Tor node is the Entry guard node that takes the traffic from PC1 in an encrypted format 3 times (Key1, Key2, Key3). Then the entry-guard-node decrypts/deciphers/takes off the first encryption layer with Key1 and he is the only one that knows the IP Address of the original source but can't read the message because it is already encrypted another 2 times later the guard node forwards that message to middle-relay (Key2). That middle-relay node does the same and forwards to exit-node. The exit-relay then uses his own private Key 3 to strip the last layer of security and unveils the original message then forwards traffic to the server here is where you lose your actual anonymity because the Service you are going to access knows that you had interaction with it and a log with a timestamp but does not know your actual location. Then the reverse process happens in the backward pass. Web server (PC2) send to exit-node as it only knows that this was the Source IP address. The exit-node get the data packet from PC2 it encrypts them with Key 3 (first layer added back in) and sees that the

new destination IP is the middle node and sends it there the process continues until all layers of security have been added back together like onion hence the name until reaches Guard node where he has the original location IP address of the original source that gets the Packet (PC1) and decrypts/decipher all layers of security with 3 Keys. The only caveat here is it only works with 3 Tor nodes adding more in between won't provide better security but it will definitely enhance the delay time and make it slower.

So, to sum up exit-node (output) knows who the server (destination) is because sees the packet raw as it was sent it from PC1 but does not know who about PC1 only that someone in this Tor network wants to access this website. Middle relay doesn't know anything at all except where to forward the message and another interesting thing is middle-relay does not know either who the entry-guard-node is because the Tor the way it works is when you decrypt a layer of security (in the process of forward pass before web server gets the message) you don't not know have many layers there are left despite we discussed about 3 intermediate

¹¹ Below 2 hops the encryption schema won't work you won't be anonymous.

¹² First the connection and shared key is established in PC1 with guard node (or exit-node depends on

implementation schema) then guard node extends/adds to the circuit by instructing middle relay node to establish shared key with PC1 and every time they do that the previous messages are already encrypted so nobody knows what's going on.

nodes concept we know that but the way it works on computers they don't (it could be more than 3 though). Now the guard node (input) knows that the

Client is PC1 with that IP address and it is talking to Tor without knowing the message because of 2 other layers of security in that moment that can't decrypt.

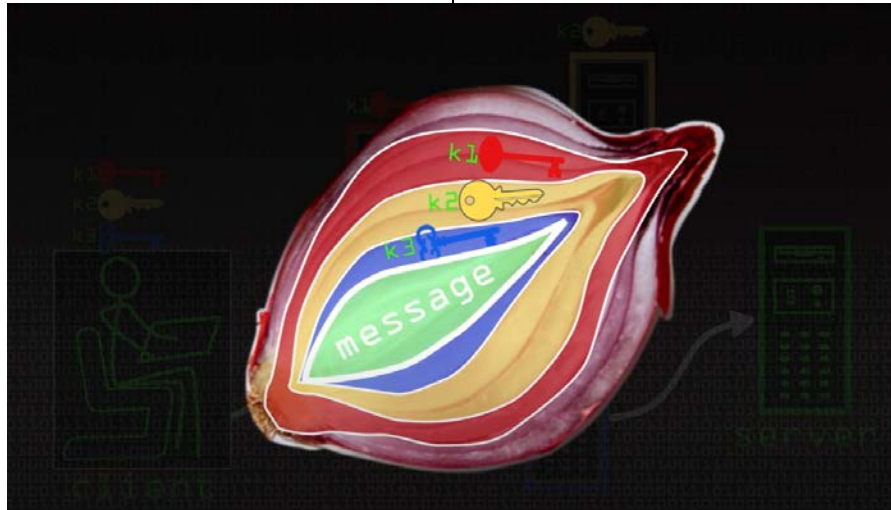


Figure 9 <https://www.youtube.com/watch?v=QRYzre4bf7I>

2.4.2. Tor Hidden Services – How it works

Tor Hidden Services (THS)(HS) aka Dark web tries to hide IP location of a server inside Tor Network without any exit node to reveal any detail [51]. A normal usage of Tor with just anonymization and can be depicted in picture Figure 8 and Figure 10 whereas THS is Figure 10.

These services can be identified because their addresses are finished in “.onion” and any of them requires that any citizen executes the rendezvous protocols to contact the service in an anonymous way. That is one of the reasons that THS is linked with various illegal activity but “Tor is critical technology not just in terms of privacy protection but in defence of our publication right” [52]. However, it can be used by journalists and to provide freedom

of speech are being offered by human rights and whistle-blowing organizations [53].

For a user to communicate with an HS mainly 3 nodes are required chosen by HS and 3 by the user, but the bigger the network the more robust it can be. So, in contrast with a normal anonymous communication to a site outside the Tor network (surface | deep web) more nodes take place. One of those 3 chosen by the user becomes the Rendezvous point (RP) that in simple words relays the client's encrypted messages to the services and vice versa.

“With a little more detail, a user who wants to connect or visit a hidden service must know his onion address, a string of 16

alphanumeric characters. Then, the client searches for information in a directory node (HSDir) that is basically a hash table (DHT) distributed (among all nodes in the network) that contains the descriptors of each HS and finds an introduction point (IP) that the hidden service has previously defined. Simultaneously, the client also chooses a node that will function as RP and builds a circuit to it. Once that client connects to one of these nodes, the IP transmits a notification to the hidden service, that contains the address of the rendezvous point. When the Hidden Service recognizes the RP, it creates a new circuit to it, and sends a new message to the client through the IP, in order to this close the initial connection with the IP, and both of them keep the circuit built towards the RP, the communication is finally established. Both the IPs and RP have three intermediate jumps between them and the user, and between them and the server. Therefore, none knows the identity of each part. In addition, each Tor message is a fixed 512B which creates another “layer” of indistinguishability of Tor’s traffic analysis movement only inside the network. Finally, the communication of the HS and user travel through 6 hops and use the RP as an intermediary, so they do not identify each other. This as far as it relates to Client-server anonymity now a server maybe does not want a “hide” protection but only wants the user’s to be protected so in that case 3 IPs will

be excluded from the part process and will directly will go from the server to RP.

As a final note Hidden services use another different WEB protocol for interfacing with services on top of them meaning its just another language of the web that normal WEB services (Clearnet/surface/deep-web) like google it’s not implemented to talk to this Dark web protocol there is no API and cannot index cannot talk to Dark web. If someone with knowledge how Dark web works can easily create an API that extracts Hidden services by indexing them and posting them in Clearnet (normal web) on a public forum (also to note that Dark web can be indexed because is a normal web just different from the usual we know e.g., torch dark web search engine or Hidden wiki).

To sum up HS hide server physical location IP through use of onion protocol by choosing randomly some relays and build circuits to them and ask them to act as IP by telling them a public key then the onion service assembles an onion services descriptor which contains its public key (of server) and a summary of each IP and signs this descriptor with his private key. Then uploads this descriptor to a distributed hash table like torrent are tracked these days and this descriptor is the onion address e.g., dshjad786ndds.onion (alphanumeric characters) if its V2 is a 16-character or 56-character (for V3) name that is derived from

services public key. These addresses do not use DNS and there is not one in order to take them down as central authority like what happens in Clearnet. So, to resolve this Tor uses a Consensus of directory authorities' nodes which they are special relays in Tor network scattered and clustered while they are responsible to update and distribute a master list of all Tor relays by making a vote and thus reaching the consensus (common

vote) of that list (v3 only addresses are being hold in this directory)¹³. Basically, they are gate keepers tracking status of Tor relays as valid or not, online/offline, bandwidth status of HS relays et al.

More details on Tor protocol, directory server, rendezvous protocol can be found in Tor's specifications [54] and in its design document [48]." [53]

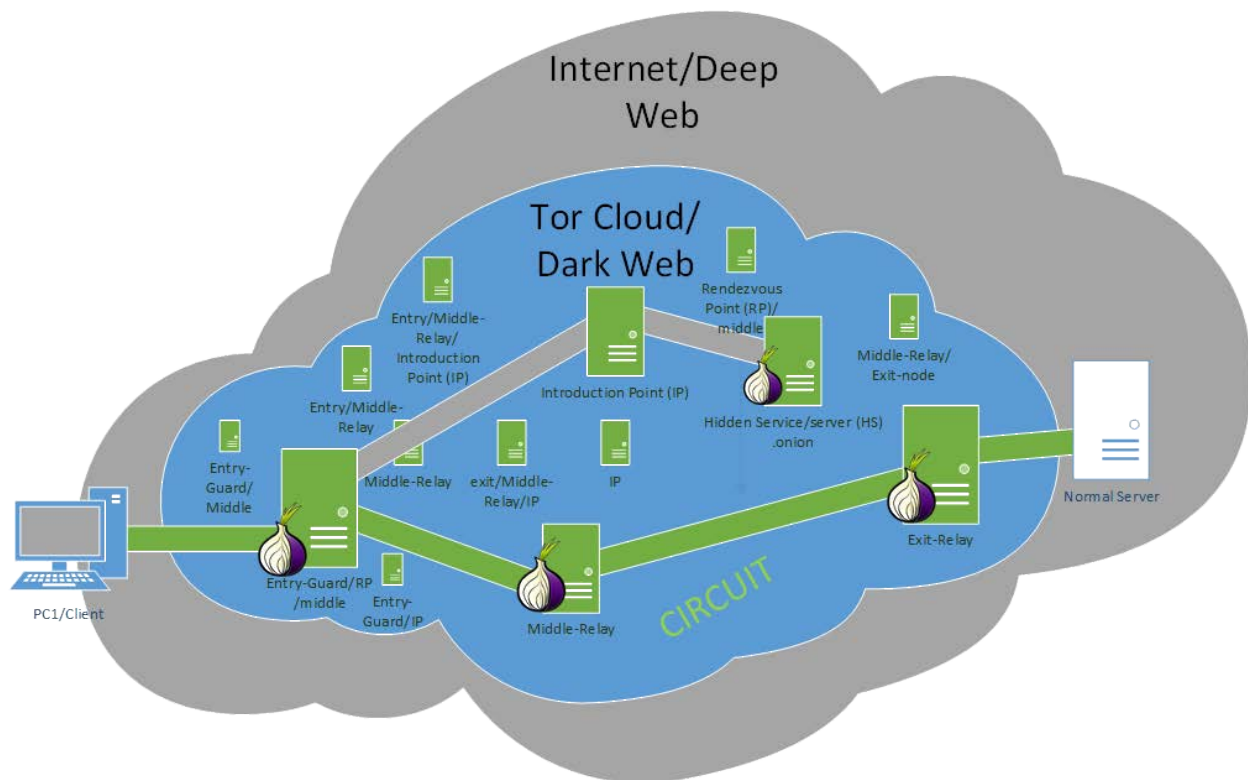


Figure 10 Normal Onion Routing and Hidden Services

2.4.3. Vulnerability issues & Forensics

The actual vulnerability is that your traffic is weak at the Entry Guard node and Exit-node¹⁴ if someone promiscuous/monitors those two nodes¹⁵ and carefully synchronizes events/patterns that happened in different stages then you are in trouble (meaning that in

¹³ Version v2,3 represents security level of encryption algorithm HS not many people want to access a V2 (risk of deanonymizing)

¹⁴ Exit Relays are public known

¹⁵ except if use VPN on exit node mode for outgoing traffic;

entry and exit more than 1 connection/user could be alive and track down the pattern). But as with all things if somebody can have the resources to run a big part of the network their own servers either for blockchain or Tor for malignant reasons and convert the Web3 approach to Web2 with a centralized entity to control everything then the anonymity and privacy loss of protection is always a possible undergo threat [55].

2.5. The caveat law thing

Each Law enforcer in a country has its own rules when comes to restrictions. In the WEB 2.0 era where centralization is the main thing and dependency, where everything is controlled by a single entity e.g., government, corporate then you have to play by their own rules. According to Meta (former Facebook), E2EE is delayed up to 2023 due to concerned parties to ensure bad actors do not abuse the system [56]. Sure, such power can lead to many crimes but encryption and privacy are not secure if it can be examined by these parties under their demand (Results). Also, in below (Figure 11) we can see where encryption is permitted or almost banned. Again, this is a very controversial thing because there is not a clear answer/solution for what is right or wrong.

Web 3.0 comes to help even more with this problem where privacy becomes a

bigger thing for the anxious end-user [40] [41] which introduces blockchain technologies with a peer-to-peer infrastructure network and everything gets a decentralized approach, yet as everything, this is going to be exploited at some time in the future by parties. But let's face it data collection and mining even spying was always a thing even before technology arrived so it is in our will to be heading there and if you want a positive thought out of this, we are just actions and branching out of options we made (maybe something more); consider the fact that someone can gather these a new lifeform could be designed like a realistic Artificial intelligence in the future which will bring breakthroughs to all science/STEM job fields and change completely our lives, a system that could recognize all patterns and reduce even crime rates.

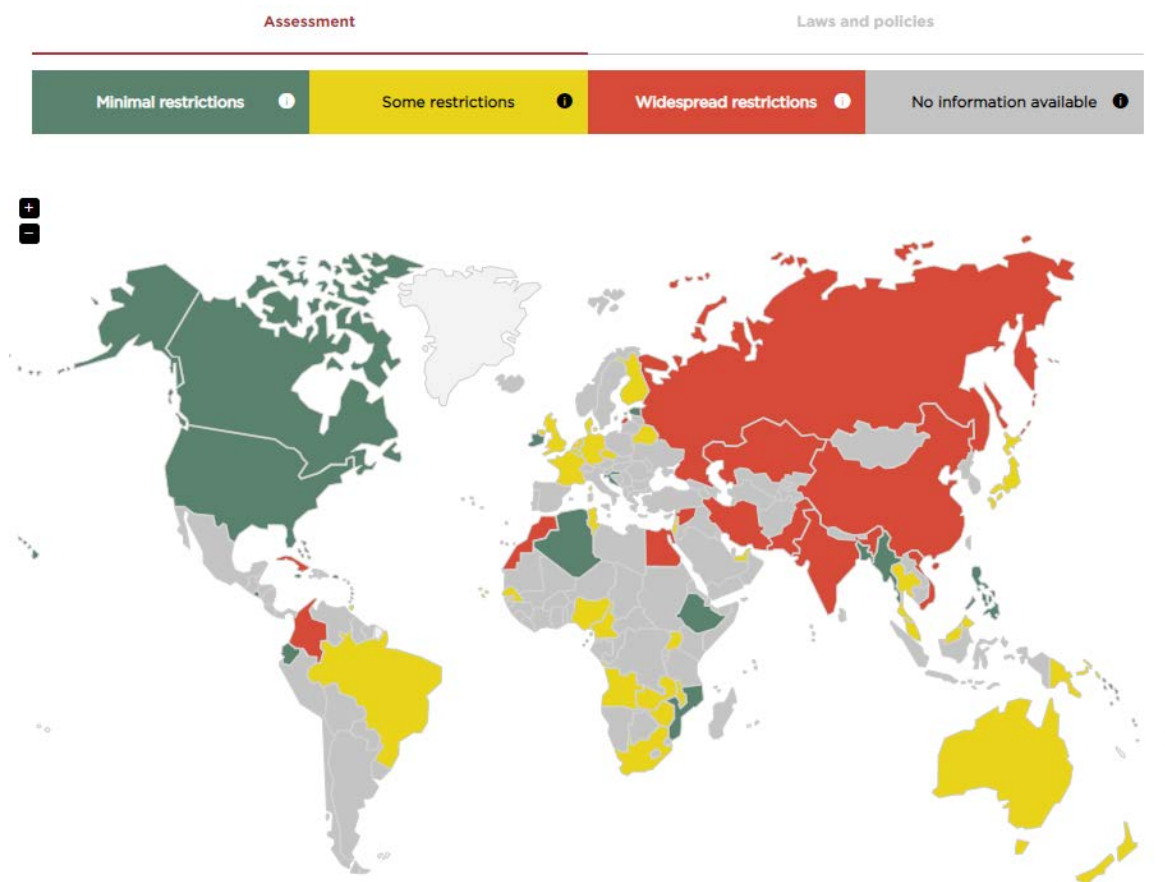


Figure 11 [World map of encryption laws and policies](#)

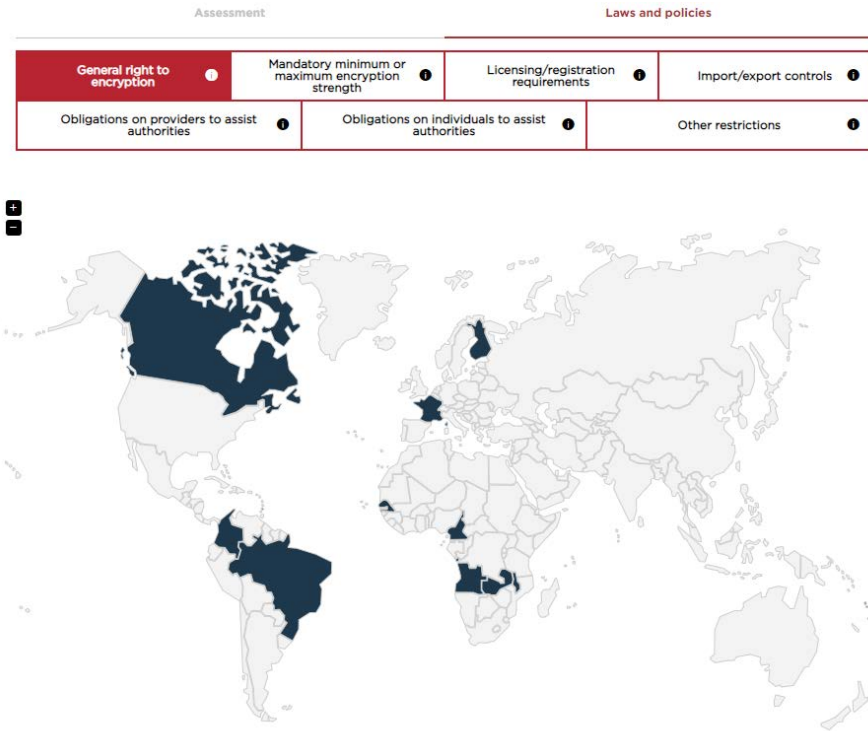


Figure 12 <https://www.gp-digital.org/world-map-of-encryption/>

3. Security tools

3.1. OpenVPN

3.1.1. Brief

An open-source VPN system which means transparency to the network (you can read its code) and implements secure point-to-point, site-to-site connection with both client and server applications¹⁶ or any network your physical or cloud ecosystem needs, whether AWS, CGP, Azure et. al in routed or bridged configurations and remote access facilities. It's first released in 2001, written in C and is available on a variety of platform systems. Some Networking gear support it by default or it can even be installed in an Open/DD-Wrt Linux based router firmware system and is under GNU license (GPLv2) [57].

3.1.2. Pros and Cons

3.1.2.1. Pros [58] [59] [60]

- Open-Source
- Site-to-site
- Client-to-site

¹⁶ You install the server application and the client by configuring them in your own rules.

- E2EE¹⁷
- Data encryption on outbound interface/forward direction even without the browser's traffic
- Interfaces (L2/L3 - TAP/TUN)
- Security (high-end ciphers with high-bit encryption keys)
- Cross-platform
- Good firewall compatibility (any port of TCP/UDP frameworks e.g., as HTTPS masked)
- Network Address Translation (NAT) friendly (traversal)
- Perfect Forward Secrecy (secret key changes per session client/server)¹⁸
- No Cost/pricing for manual configuration

3.1.2.2. Cons [58]

- Centralized;
- Requires additional software client
- Complex manual configuration
- No free-roaming for mobile devices in the matter if IP L3 changes connection/session will be lost as from mobile carries perspective it has no issue because there is no L3 address change between radio access network (RAN) change.
- Costs for ready GUI solution (license)
- Limit in the number of servers and connections (free edition)
- Bandwidth or/and latency due to many overheads and encryptions (user space is more prominent to cause problems here, not always as a cons)
- Proxy Support (if not compatible proxy found)

¹⁷ traffic outside the tunnel is encrypted only if protocols used are encryption-based (i.e., SSH, pop3 etc.) and if its remote-access (host-to-host) then possible there is not E2EE except if the target host service/socket connection that the traffic is forwarded is using encryption (e.g., if website you visit is HTTP the final exit VPN node will not have encryption as referred to host-to-host-to-target-website).

¹⁸ Makes almost impossible for a Man in the Middle attack from decrypting all data packets gathered among different period of times/sessions because it needs to find every time the new crypto key to decipher.

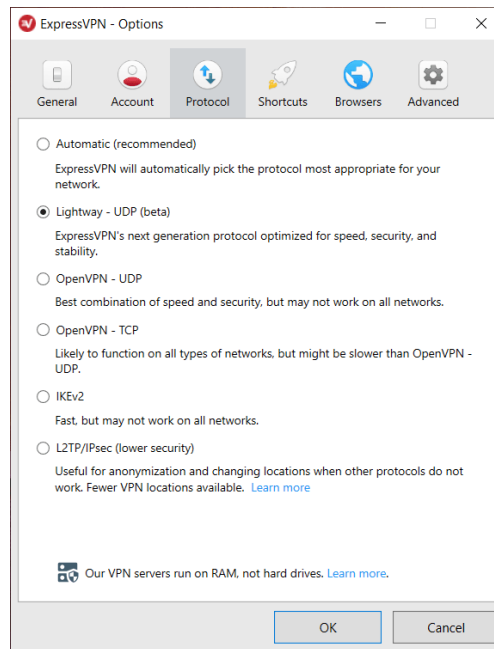


Figure 13 <https://restoreprivacy.com/vpn/openvpn-ipsec-wireguard-l2tp-ikev2-protocols/>

3.2. TOR

3.2.1. Brief

Tor, short for The Onion Router, is free and open-source software for enabling anonymous communication. It directs Internet traffic through a free, worldwide, volunteer overlay network, for concealing a user's location and usage from anyone conducting network surveillance or traffic analysis. Using Tor makes it more difficult to trace the Internet activity to the user. Tor's intended use is to protect the personal privacy of its users, as well as their freedom and ability to conduct confidential communication by keeping their Internet activities unmonitored [14]. It is written in C, Python, Rust and is available for many platforms.

3.2.2. Pros and Cons

3.2.2.1. Pros [34] [61] [62]

- Open-Source
- Anonymization
- Free
- Decentralized without blockchain for payments
- No auditing
- Secure Browsing (HTTPS everywhere, NoScript and encrypted data)
- Democratic activities (you can post anything by hiding yourself)
- Deep web access (Deep web and blocked sites from regular search engines)
- .onion websites (only accessible via through the Tor's browser and they are related to deep/dark web)
- Added encryption Layers does not slow down Tor. At [47] video timestamp 12:50 – 14:00

3.2.2.2. Cons

- Slow Connection compared to regular browsers
- Exit Nodes don't encrypt traffic
- Illegal use
- Startup time of browser is slower while seeks the available guard node servers to connect.
- Some scripts are blocked, disable certain online features
- Data encryption only through the browser's traffic and inside the Tor network¹⁹
- No E2EE²⁰

3.3. Compare Results

OpenVPN vs Tor

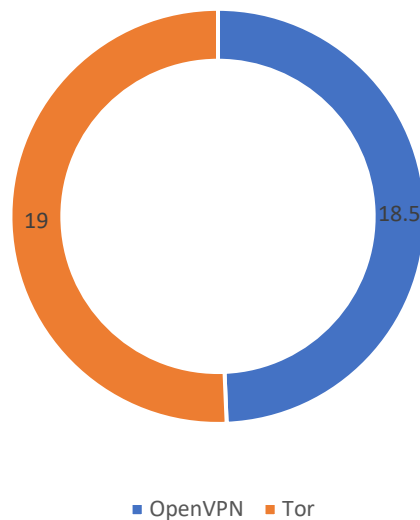
Criteria	dVPN (possible OpenVPN in future?)	OpenVPN	Tor
Open-Source	+	+	+
Fast Speed	+, -	+, -	-
Easy to Setup	+, -	-	+
Easy to Learn	+, -	-	+
Network Interfaces	+	+	-
Scalability	+	+	+, -
Anonymization	+	-	+
Encrypted connections	+	+	+
Packet encapsulation Encryption	+	+	+
Data payload Encryption	+ (any service is behaving like "using" "HTTPS" due to blockchain nature) [63] [64]	yes, if the target service is on site	-
Protects all online connections	+, - (e.g., site-to-site or host-to-host)	+, -	-
Ensures privacy	+	+, -	+
End-to-end encryption (E2EE)	+	Only for site-to-site and target service to be inside on target site	-

¹⁹ The actual data payload for Application Layer on TCP/IP model is not encrypted by Tor so the exit-node will remove the final layer of encrypted packet of Tor network but the Application Layer will be on its own from now and on depending what service the user initiated to make contact/socket with.

²⁰ Again, User must be aware of final target service that want to reach (http, https) the outbound Interface of user will encrypt Traffic because is going through browser to the interface as inbound (top-to-down encapsulation OSI model) and also the Traffic will be getting extra layers of encryption inside Tor Network but on final exit-node the actual Application Data tier layer will be on its own.

<i>Decentralized peer-to-peer nodes</i>	+	-	+
<i>Flexibility on Business Network Ecosystem</i>	+	+	-
<i>Business Support (if can be applicable)</i>	+	+	+, -
<i>End-user support</i>	+	+	+
<i>Hard to trace</i>	+	+, -	+
<i>User-friendly</i>	+, -	+, -	+
<i>Unlock streaming content</i>	+	+	Yes, but not usable due to its speed
<i>Deep web</i>	+	+	+
<i>Dark web (hidden services)</i>	+, -	-	++
<i>Roaming (not from mobile RAN ISP perspective)</i>	? (Depends on implementation)	-	-
<i>Easy to choose exit node</i>	+	+	-
<i>Industry Standard</i>	+	+	+
<i>No Audit/Log Centralized</i>	+	-	+
<i>Product & Community Support</i>	+, -	+	+
<i>Cost/Pricing</i>	Pay as you go	Free or subscription	Free

Compare Results



3.3.1.1. Notes to take & Chosen Criteria explanation

When comparing products, we have to keep in mind some core things.

If it's long enough out there so we can rely upon it as a business and adapt it under the umbrella term "industry standard" from its features, business strategy adoption and adaptation, product support and product availability in many forms.

- **Open-Source:** To learn & develop & understand. Share == care
- **Fast Speed:** Internet speed metered connection based on latency depending on the crypto algorithm.
- **Easy to Setup:** Usability out of the box or how complex can it be with manual config.
- **Easy to Learn:** Abstraction of the complex underlying system + more customers approach it.
- **Network Interfaces:** Easy configuration and addition of any virtual or physical kernel driver network communication device.
- **Scalability:** Easily can be expanded on demand or/and cross-platforming accessibility.
- **Anonymization:** Hide completely activity of individuals
- **Encrypted connections:** Setup channel and data flow encryption
- **Packet encapsulation Encryption:** Encryption at Layer 3
- **Data payload Encryption:** Payload layer 5-7 encryption
- **Protects all online connections:** Automatically encrypt all inbound traffic for outgoing encrypted tunnelled destinations.
- **Ensures privacy:** Encryption of the content of the activity.
- **End-to-end encryption (E2EE):** From source to service target application destination encryption schema (like simple SSL or SSL over complex blockchain).
- **Decentralized peer-to-peer nodes:** Not controlled by a single entity.
- **Flexibility on Business Network Ecosystem:** Business support + scalability on infrastructure or Software as a Service solution (SaaS).
- **Business Support (if can be applicable):** Network Level Corporate adaptation solutions architecture.
- **Hard to trace:** Implementation of actual specification on architecture to hide information.
- **User-friendly:** Intuitive and intriguing (G)UI design for Human-computer interaction (HCI) accessibility.
- **Unlock streaming content:** Bypass Local geographical restrictions
- **Deep web:** Ability to access completely the surface web
- **Dark web (hidden services):** A web replica of freedom and anonymity
- **Roaming (not from mobile RAN ISP perspective):**
- **Easy to choose exit node:** Easier access to distant geographical local physical services.
- **Industry Standard:** Just it works for everything so adaptation for everyone is welcome. The ecosystem expands.
- **No Audit/Log Centralized:** No history and stored data of any transaction or interaction.
- **Product & Community Support:** Improving and improving (software) and customer support with solutions and add-ons/tools for a robust ecosystem.
- **Cost/Pricing:** Easy access for educational or corporate usage

3.3.1.2. Results

The giveaway to take is that there is no perfect solution everything can satisfy someone and dissatisfy someone else. Every tool has its usages in the right spot and under a provisioned network designed strategically. Is about what a client or/and stakeholder wants in a project. They could work as a complementary solution to each other to enhance security even more.

OpenVPN is suitable for a simple user that wants privacy without anonymity and any cost or a business that implements intranet/extranet VPN access (remote access) to its employees or entire site-to-site connection bridging and routing for merging the gap of disjoint form or policy network and making it like a layer 2 part by sharing the segment or layer 3 i.e., running adjacency routers with neighborships through the tunnel. The main actual drawback is it does not use E2EE so the user must be aware of the target service tier and the remote VPN server is Centralized and could possibly log you (storing your data) for third parties sharing i.e., Data mining, government spy, hackers (an attack).

Tor on the other hand is very robust for anonymity something OpenVPN does not provide meaning that for a user that wants to be invisible at least from spies in the middle²¹. But is quite non-suitable for enterprise networks that want privacy it's more user-driven not enterprise-driven but it could be done.

A combination of both solutions (Tor + OpenVPN) comes in two flavors (approaches) [43]:

- VPN on Tor (source host (bind to VPN) -> Tor -> ISP -> VPN (exit) -> Target service)
 - Connect Tor then use a VPN: Requires complex manual configuration. Your VPN server acts as the final exit node and the Tor's exit node will carry the packet still encrypted without compromising your activity (due to VPN).
 - ISP can see you that you are using Tor but it wouldn't be able to trace you while you keep your IP address hidden from the VPN service from further auditing/logging.
 - If there is an attack on Tor's exit-node your packet will remain encrypted because the node would not be able to tell your activity due to VPN.
- Tor over VPN (source host -> VPN -> ISP -> Tor (exit) -> Target service)
 - Connect to your VPN then use Tor: This way you encrypt the traffic before it enters the Tor Network, and also hides your IP address in Tor's exit node
 - ISP can't tell that you are using Tor
 - VPN provider will trace Tor activity and be able to log. A Workaround is to use a decentralized VPN, which cannot keep user logs.

While regular VPN offer protection to their users (for a price), the real fight against surveillance and censorship is a shared one because the current providers do nothing to

²¹ An attack to guard-node and/or exit-node could be proven fatal for user's anonymization.

address the infrastructural flaws of the internet so there is no immunity to corporate or government control. [43]

There is not a single fit-all-size solution or a true security mechanism that protects you from a data breach, every method is just an additive layer of security that could be stripped from an attack.

Anecdotally the more complex layer of the OSI model is Layer 8 user/political layer because technology is human-driven or an echo of our lives and security is totally something that we want but always in our advantages, not someone's else that carries everywhere from "our" to a single human or organization (everything must be secure but not secure in the same time) [65].

4. Real Case Research Analysis Literature Review

4.1. The case Study

4.1.1. Tor options

Tor has many options on how to operate we are going to examine how to setup a Tor Relay node and a Hidden Service Server. Both procedures require *torrc* file to be adjusted accordingly it is located under: (it may be hidden)

- Linux: `/etc/tor/torrc`
- Windows: `"path_to_tor"\Tor Browser\Browser\TorBrowser\Data\Tor\torcc`
- macOS: `~/Library/Application Support/TorBrowser-Data/Tor/torrc`

4.1.2. torrc

torrc is a configuration file to edit how Tor will behave on various services such as Hidden service and the Relay node. Misconfiguration or following any unusual practice of that file can lead to the server compromising its security and being vulnerable to attacks. Any modification to the file itself should be done while you have closed the Tor Browser because it may erase your modification because of its run-time [66].

4.1.3. Relay node operation

According to documentation there are plenty of Tor relay options to choose from and you can run multiple modes of the relay at the same time but not all of them. For example, you cannot be an entry guard and exit node at the same time [67].

Also, before choosing relay mode also, we should check any legal complaint and attraction of law enforcement agencies [68].

In order to configure it as middle-relay or exit-node we follow these rules and generally speaking we must have some bandwidth capabilities as well more on this here [69] [70]:

- Use a Physical machine or/and virtual or could solution (e.g., DigitalOcean)
- Install Linux ubuntu or kali linux and update it, install any repos for python and install python if not already installed.
- Add repos if can't find Tor in `sudo apt install`

- Make sure you have a Firewall policy for ports and Port forwarding on Network Device/Router (NAT/PAT) to allow the daemon socket you are going to open for Tor e.g., port 4000 or 9001.
- `sudo nano /etc/tor/torrc`
- uncommented and add the following lines (remove #) as depicted in Figure 14
- Also, you can configure bandwidth any many more options you can find here [67]
- `sudo tor` or `(systemctl enable tor and systemctl start tor)`
- You will appear in public registry in couple of days after building trust level cause of traffic you can find moer in this Figure 15.
- Contratz you just contribute to this amazing project with adding another node relay!

```

kali@kali: /etc/tor
File Actions Edit View Help
kali@kali: /etc/tor x root@kali: /var/lib/tor/hidden_service x
GNU nano 5.4 torrc *

## Once you have configured a hidden service, you can look at the
## contents of the file ".../hidden_service/hostname" for the address
## to tell people.
##
## HiddenServicePort x y:z says to redirect requests on port x to the
## address y:z.

HiddenServiceDir /var/lib/tor/hidden_service/
HiddenServicePort 80 127.0.0.1:8080

#HiddenServiceDir /var/lib/tor/other_hidden_service/
#HiddenServicePort 80 127.0.0.1:80
#HiddenServicePort 22 127.0.0.1:22

##### This section is just for relays #####
#
## See https://www.torproject.org/docs/tor-doc-relay for details.

## Required: what port to advertise for incoming Tor connections.
ORPort 9001
ExitPolicy reject *: * #depends on accept reject your ISP will be able to see your activity
Nickname TorTestWarning #These will be known in the network
ContactInfo youremail@domain.toplevel #if anything goes wrong Tor will email you
## If you want to listen on a port other than the one advertised in
## ORPort (e.g. to advertise 443 but bind to 9090), you can do it as
## follows. You'll need to do ipchains or other port forwarding
## yourself to make this work.
#ORPort 443 NoListen

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line  M-E Redo
  
```

Figure 14 Tor relay config in torrc

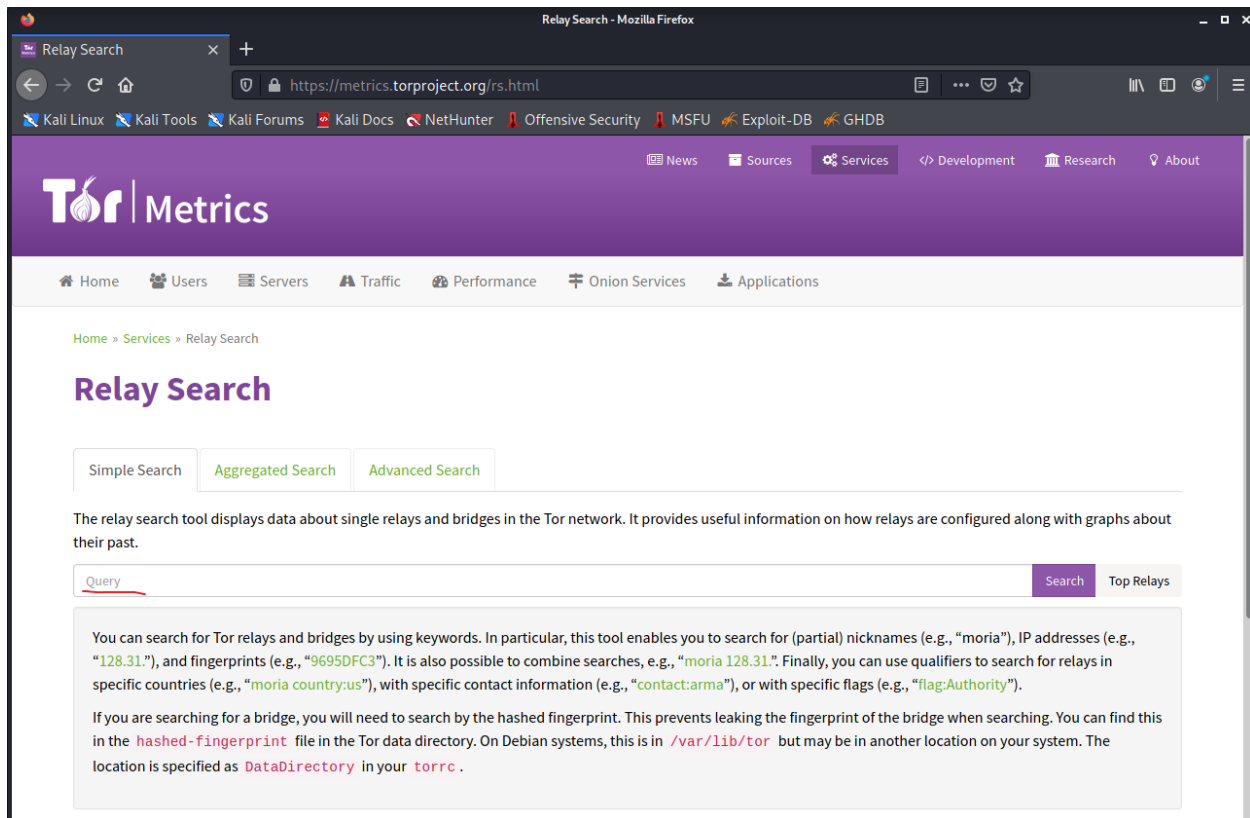


Figure 15 Tor metrics

4.1.4. Hidden Service

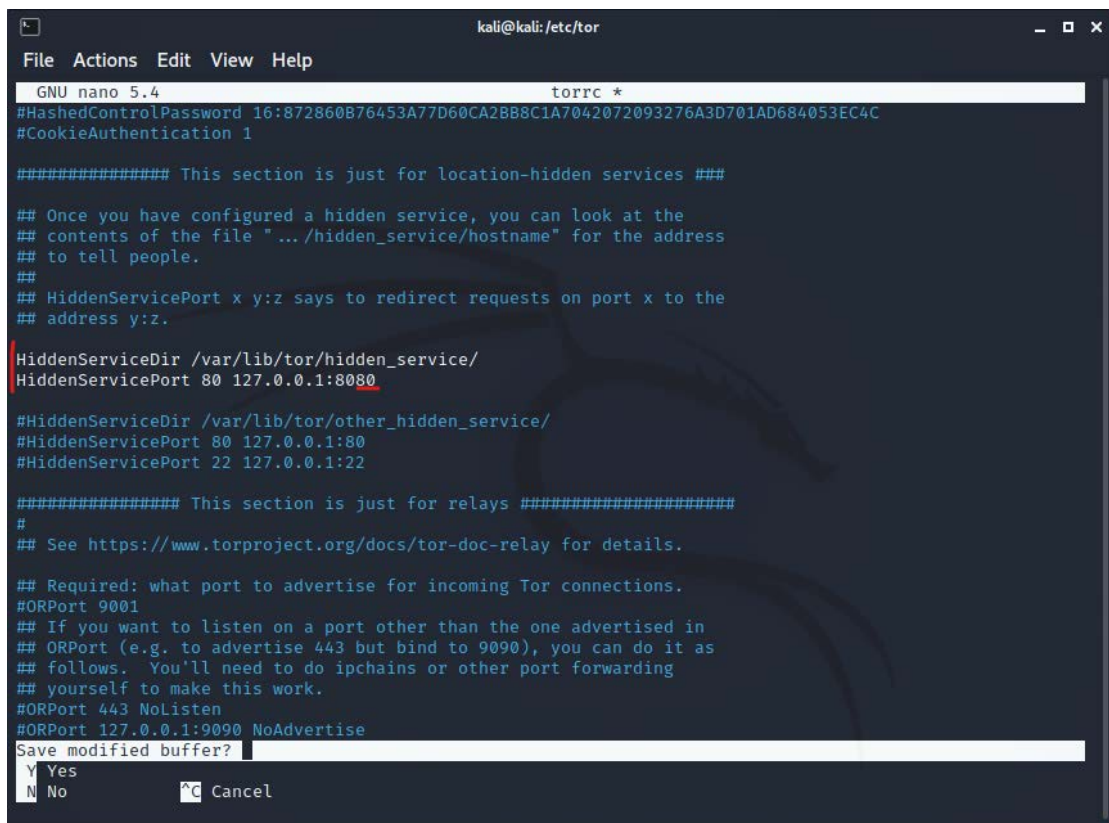
According to the documentation, you must first make a directory for the server to live in: [71]

- Either to Physical machine, virtual machine or cloud solution (e.g., DigitalOcean)
- Install Linux Ubuntu or kali Linux and update it, install any repos for python and install python if not already installed.
- Add repos if can't find Tor in sudo apt install
- mkdir tor_service (anywhere)
- cd tor_service
- python3-m http.server --bind 127.0.0.1 8080
 - This IP is recommended by Tor in order to avoid get discovered by some services
- touch index.html
- nano index.html
- <html><body> Test Tor Hidden Service App</body></html>
- Save it
- Open a Web Browser and type url to port "localhost:8080". The Web page runs on Python in local machine now we want to host it in Tor
- Install Tor and Tor Browser
- sudo apt update

- `sudo apt install -y tor torbrowser-launcher`
- `whereis tor`
- `cd /etc/tor`
- `sudo nano torrc`
- uncommented the following lines (remove the # in both of them and add 8080 as port) as depicted in Figure 16
- save it
- `sudo tor`
- The hidden service is running in order to see it in Tor browser you need the “.onion” address you can get it in `cd /var/lib/tor/hidden_service` as a `sudo su` user before go there as depicted in Figure 17.
- Copy the address paste it inside Tor Browser and voila the Python web server is hosted inside Tor Network as a Hidden service and everybody with this address from now on can access your Hidden Service!

Is recommended to have your own dedicated server like apache, tomcat or any other HTTP software because you are going to run long-term and get a lot of traffic probably so python server maybe not suitable for the job.

Nothing is 100% safe so be warned.



```

kali@kali: /etc/tor
File Actions Edit View Help
GNU nano 5.4 torrc *
#HashedControlPassword 16:872860B76453A77D60CA2BB8C1A7042072093276A3D701AD684053EC4C
#CookieAuthentication 1

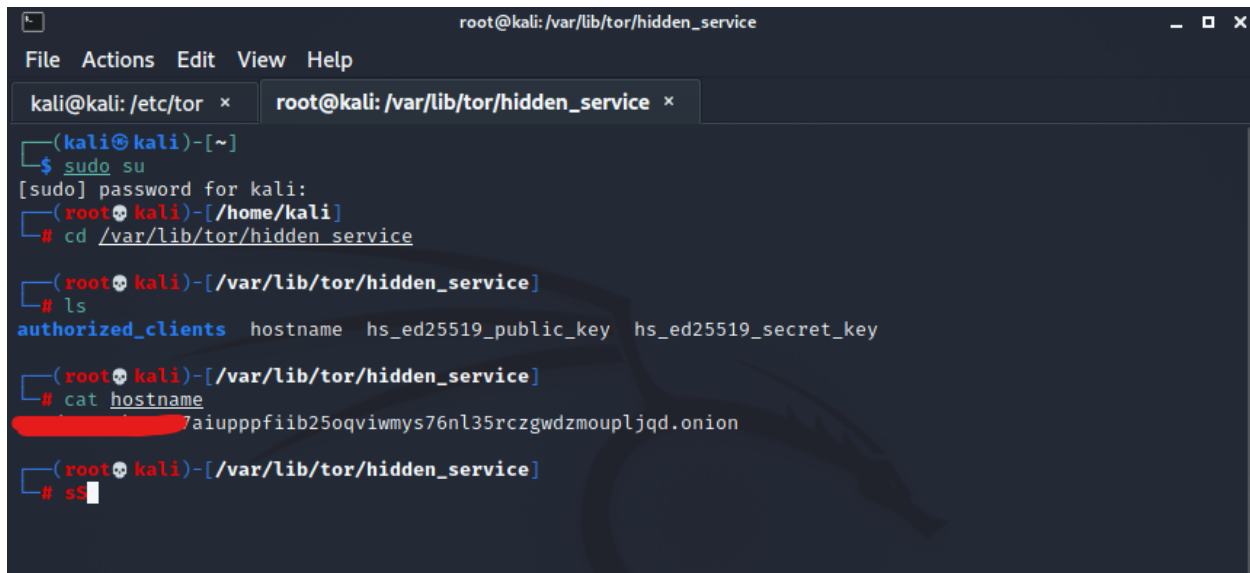
##### This section is just for location-hidden services #####

## Once you have configured a hidden service, you can look at the
## contents of the file ".../hidden_service/hostname" for the address
## to tell people.
##
## HiddenServicePort x y:z says to redirect requests on port x to the
## address y:z.
HiddenServiceDir /var/lib/tor/hidden_service/
HiddenServicePort 80 127.0.0.1:8080

#HiddenServiceDir /var/lib/tor/other_hidden_service/
#HiddenServicePort 80 127.0.0.1:80
#HiddenServicePort 22 127.0.0.1:22

##### This section is just for relays #####
#
## See https://www.torproject.org/docs/tor-doc-relay for details.
## Required: what port to advertise for incoming Tor connections.
#ORPort 9001
## If you want to listen on a port other than the one advertised in
## ORPort (e.g. to advertise 443 but bind to 9090), you can do it as
## follows. You'll need to do ipchains or other port forwarding
## yourself to make this work.
#ORPort 443 NoListen
#ORPort 127.0.0.1:9090 NoAdvertise
Save modified buffer? [Y] Yes
N No ^C Cancel
  
```

Figure 16 torrc for hidden services



```

root@kali: /var/lib/tor/hidden_service
File Actions Edit View Help
kali@kali: /etc/tor x root@kali: /var/lib/tor/hidden_service x
(kali@kali)-[~]
$ sudo su
[sudo] password for kali:
(root@kali)-[/home/kali]
# cd /var/lib/tor/hidden_service

(root@kali)-[/var/lib/tor/hidden_service]
# ls
authorized_clients  hostname  hs_ed25519_public_key  hs_ed25519_secret_key

(root@kali)-[/var/lib/tor/hidden_service]
# cat hostname
7aiupppfiib25oqviwmys76nl35rczgdzmoupljqd.onion

(root@kali)-[/var/lib/tor/hidden_service]
# sS

```

Figure 17 Find your .onion address

References

- [1] H. S. published, "A Brief History of Cryptography," redhat, 14 08 2013. [Online]. Available: <https://access.redhat.com/blogs/766093/posts/1976023>. [Accessed 11 11 2021].
- [2] "Enigma machine - Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Enigma_machine. [Accessed 11 11 2021].
- [3] "Alan Turing - Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Alan_Turing. [Accessed 11 11 2021].
- [4] "Alan Turing," New Scientist, [Online]. Available: https://www.newscientist.com/people/alan-turing/?utm_source=rakuten&utm_medium=affiliate&utm_campaign=3690980:Linkbux&utm_content=10&ranMID=47192&ranEAID=wizKxmN8no4&ranSiteID=wizKxmN8no4-edVE._ke9k8oXwf4sksltw. [Accessed 11 11 2021].
- [5] "A Brief History of the Internet," Usg.edu, [Online]. Available: https://www.usg.edu/galileo/skills/unit07/internet07_02.phtml. [Accessed 11 11 2021].
- [6] "ARPANET - Wikipedia," [Online]. Available: <https://en.wikipedia.org/wiki/ARPANET>. [Accessed 11 11 2021].
- [7] "History of the Internet - Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/History_of_the_Internet. [Accessed 11 11 2021].

- [8] "Internet protocol suite - Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Internet_protocol_suite. [Accessed 11 11 2021].
- [9] "History of TCP/IP," scos, [Online]. Available: <https://scos.training/history-of-tcp-ip/>. [Accessed 11 11 2021].
- [10] "IPv4 - Wikipedia," [Online]. Available: <https://en.wikipedia.org/wiki/IPv4>. [Accessed 11 11 2021].
- [11] "IPsec - Wikipedia," [Online]. Available: <https://en.wikipedia.org/wiki/IPsec>. [Accessed 11 11 2021].
- [12] "Introduction to Cisco IPsec Technology," Cisco, [Online]. Available: https://www.cisco.com/c/en/us/td/docs/net_mgmt/vpn_solutions_center/2-0/ip_security/provisioning/guide/IPsecPG1.html. [Accessed 11 11 2021].
- [13] L. Center, "VPN History & The Future of VPN Technology," CactusVPN, [Online]. Available: <https://www.cactusvpn.com/beginners-guide-to-vpn/vpn-history/>. [Accessed 11 11 2021].
- [14] "Tor (network) - Wikipedia," [Online]. Available: [https://en.wikipedia.org/wiki/Tor_\(network\)](https://en.wikipedia.org/wiki/Tor_(network)). [Accessed 11 11 2021].
- [15] "Internet service provider - Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Internet_service_provider. [Accessed 11 11 2021].
- [16] C. Press, " Hierarchical Network Design Overview (1.1) > Cisco Networking Academy Connecting Networks Companion Guide: Hierarchical Network Design," Cisco, [Online]. Available: <https://www.ciscopress.com/articles/article.asp?p=2202410&seqNum=4>. [Accessed 11 11 2021].
- [17] "Understanding Using MPLS-Based Layer 2 and Layer 3 VPNs on EX Series Switches - TechLibrary," Juniper Networks, [Online]. Available: https://www.juniper.net/documentation/en_US/junos/topics/concept/mpls-ex-series-vpn-layer2-layer3.html. [Accessed 11 11 2021].
- [18] L. Cittadini, G. D. Battista and M. Patrignani, "MPLS Virtual Private Networks," [Online]. Available: http://sigcomm.org/education/ebook/SIGCOMMeBook2013v1_chapter6.pdf. [Accessed 11 11 2021].
- [19] "Public switched telephone network - Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Public_switched_telephone_network. [Accessed 11 11 2021].
- [20] "PPPoE on Ethernet," Cisco, [Online]. Available: https://www.cisco.com/c/en/us/td/docs/ios/bbds/ios/bbds/configuration/guide/bba_ppoe_enet.html. [Accessed 11 11 2021].
- [21] "MPLS Point-to-Multipoint Traffic Engineering," Cisco, [Online]. Available: https://www.cisco.com/c/en/us/td/docs/ios/mpls/configuration/guide/12_2sr/mp_12_2sr_book/mp_te_p2mp.html. [Accessed 11 11 2021].

- [22] "Introduction to Cisco MPLS VPN Technology," Cisco, [Online]. Available: https://www.cisco.com/c/en/us/td/docs/net_mgmt/vpn_solutions_center/2-0/mpls/provisioning/guide/PGmpls1.html. [Accessed 11 11 2021].
- [23] "Configuring Scalable Hub-and-Spoke MPLS VPNs," Cisco, [Online]. Available: https://www.cisco.com/c/en/us/td/docs/ios/mpls/configuration/guide/convert/mp_l3_vpns_book/mp_cfg_hub_spoke.html. [Accessed 11 11 2021].
- [24] "MPLS/VPN Hub-and-spoke Topology," [Online]. Available: <http://etutorials.org/Networking/MPLS+VPN+Architectures/Part+2+MPLS-based+Virtual+Private+Networks/Chapter+11.+Advanced+MPLS+VPN+Topologies/MPLS+VPN+Hub-and-spoke+Topology/>. [Accessed 11 11 2021].
- [25] "Leased Line Definition, Explanation, and Example," study-ccna, [Online]. Available: <https://study-ccna.com/leased-line/>. [Accessed 11 11 2021].
- [26] "Leased line - Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Leased_line. [Accessed 11 11 2021].
- [27] "Types of VPN and types of VPN Protocols," vpnoneclick, [Online]. Available: <https://www.vpnoneclick.com/types-of-vpn-and-types-of-vpn-protocols/>. [Accessed 11 11 2021].
- [28] Y. Fujimoto, T. (JP) and T. Ohsawa, VIRTUAL PRIVATE NETWORK, U.S, 2004.
- [29] "Introduction to VPNs," networklessons, [Online]. Available: <https://networklessons.com/cisco/ccna-routing-switching-icnd2-200-105/introduction-to-vpns>. [Accessed 11 11 2021].
- [30] "The Many Use-Cases of VPN: How a VPN Can be Useful," webhostingsecretrevealed, [Online]. Available: <https://www.webhostingsecretrevealed.net/blog/security/how-a-vpn-can-be-useful/>. [Accessed 11 11 2021].
- [31] "An Overview of Enterprise VPN – Virtual Private Network," [Online]. Available: <https://excitingip.com/780/an-introduction-for-enterprise-vpn-virtual-private-network/>. [Accessed 11 11 2021].
- [32] "Cisco Business VPN Overview and Best Practices," Cisco, [Online]. Available: <https://www.cisco.com/c/en/us/support/docs/smb/routers/cisco-rv-series-small-business-routers/1399-tz-best-practices-vpn.html>. [Accessed 11 11 2021].
- [33] "Centralized vs Decentralized VPN, or VPN vs dVPN, Which Is the Better Protection for My Online Security and Privacy?," bitvpn, [Online]. Available: <https://www.bitvpn.net/blog/centralized-vs-decentralized-vpn/>. [Accessed 11 11 2021].

- [34] "The Pros And Cons Of Using Tor Browser," [Online]. Available: <https://www.noobslab.com/2020/01/the-pros-and-cons-of-using-tor-browser.html>. [Accessed 11 11 2021].
- [35] "Virtual private network - Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Virtual_private_network. [Accessed 11 11 2021].
- [36] "Cisco Site-to-Site VPN Technologies Comparison At-A-Glance," Cisco, [Online]. Available: https://www.cisco.com/c/dam/en/us/products/collateral/ios-nx-os-software/enterprise-class-teleworker-ect-solution/prod_brochure0900aecd80582078.pdf. [Accessed 11 11 2021].
- [37] S. Communications, "Finding the right VPN solution," [Online]. Available: <https://sectraprodstorage01.blob.core.windows.net/communication-uploads/sites/4/2020/09/finding-the-right-vpn-solution.pdf#:~:text=As%20a%20rule%2C%20a%20traditional,the%20client%20and%20the%20server..> [Accessed 11 11 2021].
- [38] "OpenVPN Tap vs Tun Mode," stackexchange, [Online]. Available: <https://security.stackexchange.com/questions/46442/openvpn-tap-vs-tun-mode>. [Accessed 11 11 2021].
- [39] "Layer 3 VPNs User Guide for Routing Devices," Juniper, [Online]. Available: <https://www.juniper.net/documentation/us/en/software/junos/vpn-l3/topics/topic-map/l3-vpns-overview.html>. [Accessed 11 11 2021].
- [40] "Web 1.0, Web 2.0 and Web 3.0 with their difference," geeksforgeeks, [Online]. Available: <https://www.geeksforgeeks.org/web-1-0-web-2-0-and-web-3-0-with-their-difference/>. [Accessed 11 11 2021].
- [41] "Morgan Stanley's Top Stock Picks for the \$8 Trillion Metaverse Opportunity (Ep. 450), timestamp 3,41 - 14,44," YouTube, [Online]. Available: https://www.youtube.com/watch?v=g6BzocTEBig&list=PLiTVVRdEvpm6UuNLky1I58fy_eavP5t92&index=20. [Accessed 11 11 2021].
- [42] "THE BLOCKCHAIN BANDWIDTH INFRASTRUCTURE for web 3.0 (dVPN)," sentinel, [Online]. Available: <https://sentinel.co/>. [Accessed 11 11 2021].
- [43] "Decentralized VPN: The Evolution of Tor?," hackernoon, [Online]. Available: <https://hackernoon.com/decentralized-vpn-the-evolution-of-tor-hkv3uix>. [Accessed 11 11 2021].
- [44] "VPLS Overview," Juniper, [Online]. Available: https://www.juniper.net/documentation/en_US/junos/topics/concept/vpls-security-overview.html. [Accessed 11 11 2021].

- [45] "MPLS VPN Overview," Juniper, [Online]. Available: https://www.juniper.net/documentation/en_US/junos/topics/concept/mpls-security-vpn-overview.html. [Accessed 11 11 2021].
- [46] "Layer 2 Tunneling Protocol - Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Layer_2_Tunneling_Protocol. [Accessed 11 11 2021].
- [47] "How TOR Works- Computerphile," YouTube, [Online]. Available: <https://www.youtube.com/watch?v=QRYzre4bf7I>. [Accessed 11 11 2021].
- [48] R. Dingledine, N. Mathewson and P. Syverson, "Tor: The Second-Generation Onion Router," [Online]. Available: <https://fermatslibrary.com/s/tor-the-second-generation-onion-router>. [Accessed 11 11 2021].
- [49] H. Zimmermann, "OSI Reference Model - The ISO Model of Architecture for Open Systems Interconnection," in *IEEE Transactions on Communications*, vol. 28, no. 4, pp. 425-432, April 1980.
- [50] B. Schneier, "Attacking Tor: how the NSA," the guardian, [Online]. Available: https://cyber-peace.org/wp-content/uploads/2013/06/Attacking-Tor_-how-the-NSA-targets-users-online-anonymity_-World-news_-theguardian.pdf. [Accessed 11 11 2021].
- [51] "TOR Hidden Services - Computerphile," YouTube, [Online]. Available: https://www.youtube.com/watch?v=IVcbq_a5N9I. [Accessed 11 11 2021].
- [52] M. Perry, "This is What a Tor Supporter Looks Like: Edward Snowden | Tor Blog," 2015.
- [53] L. Diana, T. Huete and R.-M. Antonio, "Tor Hidden Services: a systematic literature review," *MDPI*, vol. 1, no. 3, pp. 496-518, 2021.
- [54] "Tor Project. Tor Design Documents," 2021.
- [55] "Someone Is Running Hundreds of Malicious Servers on the Tor Network and Might Be De-Anonymizing Users," gizmodo, [Online]. Available: <https://gizmodo.com/someone-is-running-hundreds-of-malicious-servers-on-the-1848156630>. [Accessed 05 12 2021].
- [56] "Meta Delays Offering End-to-End Encryption by Default Until 2023: Here's Why," [Online]. Available: https://www.makeuseof.com/meta-delays-end-to-end-encryption-until-2023/?utm_source=MUO-FB-P&utm_medium=Social-Distribution&utm_campaign=MUO-FB-P&fbclid=IwAR3oGMRuVTXgFJvcx_AliVC53D2Gvxwp5Tj5RhGB83Q1mkr7esLuVcBNLc. [Accessed 11 11 2021].
- [57] "OpenVPN - Wikipedia," [Online]. Available: <https://en.wikipedia.org/wiki/OpenVPN>. [Accessed 11 11 2021].
- [58] "OpenVPN – Pros and Cons," [Online]. Available: <https://bobcares.com/blog/openvpn-pros-and-cons/>. [Accessed 11 11 2021].

- [59] "Is OpenVPN end to end encryption?," [Online]. Available: <https://theknowledgeburrow.com/is-openvpn-end-to-end-encryption/>. [Accessed 11 11 2021].
- [60] "6 Advantages and Disadvantages of OpenVPN | Limitations & Benefits of OpenVPN," [Online]. Available: <https://www.hitechwhizz.com/2020/08/6-advantages-and-disadvantages-drawbacks-benefits-of-openvpn.html>. [Accessed 11 11 2021].
- [61] "Advantages and disadvantages of tor browser," [Online]. Available: <https://www.itrelease.com/2021/05/advantages-and-disadvantages-of-tor-browser/>. [Accessed 11 11 2021].
- [62] "The pros and cons of the Tor browser," [Online]. Available: <https://broadbanddeals.co.uk/guides/the-pros-and-cons-of-the-tor-browser/>. [Accessed 11 11 2021].
- [63] "Blockchain-Enabled End-to-End Encryption for Instant Messaging Applications," Cornell University, [Online]. Available: <https://arxiv.org/abs/2104.08494>. [Accessed 11 11 2021].
- [64] "Using Blockchain to Facilitate End-to-End Encryption (E2EE)," hackernoon, [Online]. Available: <https://hackernoon.com/using-blockchain-to-facilitate-end-to-end-encryption-e2ee-b1r221q>. [Accessed 11 11 2021].
- [65] "Layer 8," computerhope, [Online]. Available: <https://www.computerhope.com/jargon/l/layer8.htm>. [Accessed 11 11 2021].
- [66] "I'm supposed to 'edit my torrc'. What does that mean?," torproject, [Online]. Available: <https://support.torproject.org/tbb/tbb-editing-torrc/>. [Accessed 11 11 2021].
- [67] "Tor Project | Technical Setup," torproject, [Online]. Available: <https://community.torproject.org/relay/setup/>. [Accessed 11 11 2021].
- [68] "What is a Tor Relay?," eff, [Online]. Available: <https://www.eff.org/pages/what-tor-relay>. [Accessed 11 11 2021].
- [69] "Relay Operations," torproject, [Online]. Available: <https://community.torproject.org/relay/>. [Accessed 11 11 2021].
- [70] "TorRelayGuide," torproject, [Online]. Available: <https://gitlab.torproject.org/legacy/trac/-/wikis/TorRelayGuide>. [Accessed 11 11 2021].
- [71] "Tor Project | Onion Services," torproject, [Online]. Available: <https://community.torproject.org/onion-services/>. [Accessed 11 11 2021].
- [72] "Sample Configuration Using the ip nat outside source static Command," Cisco, [Online]. Available: <https://www.cisco.com/c/en/us/support/docs/ip/network-address-translation-nat/13773-2.html>. [Accessed 11 11 2021].

- [73] "Network Address Translation (NAT) FAQ," Cisco, [Online]. Available: <https://www.cisco.com/c/en/us/support/docs/ip/network-address-translation-nat/26704-nat-faq-00.html>. [Accessed 11 11 2021].
- [74] "Cryptanalysis Software Attack," OWASP Foundation, [Online]. Available: <https://owasp.org/www-community/attacks/Cryptanalysis>. [Accessed 11 11 2021].
- [75] "Cisco three-layer hierarchical model," Study CCNA, [Online]. Available: <https://study-ccna.com/cisco-three-layer-hierarchical-model/>. [Accessed 11 11 2021].
- [76] "Hierarchical internetworking model - Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Hierarchical_internetworking_model. [Accessed 11 11 2021].
- [77] "Backhaul (telecommunications) - Wikipedia," [Online]. Available: [https://en.wikipedia.org/wiki/Backhaul_\(telecommunications\)](https://en.wikipedia.org/wiki/Backhaul_(telecommunications)). [Accessed 11 11 2021].
- [78] "Difference Between ISDN BRI and PRI," [Online]. Available: <http://www.differencebetween.net/technology/communication-technology/difference-between-isdn-bri-and-pri/>. [Accessed 11 11 2021].
- [79] "Why can't a Tor node simultaneously be a guard and an exit node?," security stackexchange, [Online]. Available: <https://security.stackexchange.com/questions/200568/why-cant-a-tor-node-simultaneously-be-a-guard-and-an-exit-node>. [Accessed 11 11 2021].
- [80] R. Dingleline, N. Mathewson and P. Syverson, "Tor: The Second-Generation Onion Router," *Naval Research Lab*, 2004.

Appendix

Glossary

Term	Definition
Agnostic	Not Depended on the content e.g., no hardcoded
Abstraction	A high-Level view of things from the final consumer perspective without knowing too much about its underlying mechanics but still able to use it.
LAN	A Private Local network usually small range in logic (overlay) not in physical necessary.
VLAN	Virtual multiple Lan/s on Same Switch. Creates a broadcast domain. Segregation of LAN area groups.
DMVPN	Cisco protocol for dynamic multi-VPN setup

Socket Secure (SOCKS/Proxy)

is a technology created in the early 1990s that uses proxy servers to relay traffic between networks or systems (e.g., you configure for sending and receiving traffic in a pc a proxy IP:Port + credentials if any this can apply to Application level meaning specific application requires proxy in order to communicate or Operating System level that you configure a proxy/socks and your traffic is going through the proxy as relay instead of sending it from your pc directly to Target Service)

TUN/TAP	Kernel Drivers/Virtual network adapters/interfaces that simulates physical connections e.g., ethernet for data transferring, that they are in user space application created virtually and providing packet reception and transmission. VPN software clients can interact with those as P2P or ethernet devices in order to pass or get traffic from user space program (VPN client) in the physical network to forward. L3/L2 used by VPN.
Peer-to-peer	A mesh network with any node connected with one another.
Transparency	Not Clear what's happening in background.
Internet	Connecting computers together around the globe forming a graph/network.
Web	Software layer (A type of software) that runs on the physical internet in order to share information. Other types of software are Email exchange, FTP
Deep web	Content behind passwords that has not been indexed by a search engine (e.g., a private Facebook profile). Is the normal Web part without index.
Dark web	A copy of the Web but with a different language built-in to do the same things as web hence you need proper browser to know how to talk (ask information to get from these pages) to these pages. Talks secret code.
FTP	File transport protocol is a service/app/software more than a protocol (a software that implements a protocol/s) (generally this relates to all Layer 7 services) that you send files to other computers rather than download webpages from them using HTTP

HTTP	A text with links (hypertext - redirect). A way to look at web pages in a domain (IP address)/URL other ways include FTP.
Symmetric NAT	All requests from the same internal IP and port to a specific destination IP and port (target service) are mapped to the same unique external source IP address. If the same internal host sends a packet even with the same source address and port but to a different destination, a different mapping is used. Only an external host that receives a packet from an internal host can send a packet back. [72] [73]
Cipher	A cipher is an algorithm used for encryption or decryption.
Cryptanalysis	Cryptanalysis is a process of finding weaknesses in cryptographic algorithms and using these weaknesses to decipher the ciphertext without knowing the secret key [74]
Internetwork	Network of networks aka The Internet
CIA	Confidentiality, integrity, availability
Key exchange (e.g., Diffie-hellman)	Diffie-Hellman is a way of <i>generating</i> a shared secret between two people in such a way that the secret can't be seen by observing the communication. You're not sharing information during the key-exchange, you're creating a key together. In short is an asymmetric encryption method that send the new shared secret key (symmetric encrypted) which is more lightweight for traffic overhead compared to asymmetric keys data encryption for secure sharing across the open public (internet).
Blockchain	Web 3.0 era where everything is decentralized in a mesh p2p network and every action has a transaction in place either transferring something or creating a web app and putting in in the blockchain. This technology is transparent from code standpoint as open-source but not-transparent for its transactions. It uses digital signatures based on asymmetric key cryptography and its integrity depends upon the crypto system with the key. Very handy for authentication purposes and defend against security attacks i.e., can't be penetrated just compromising one machine. [64]
Asymmetric encryption	More computationally intense than shared symmetrical key. Public private key cipher. The public key can encrypt/sign/authenticate a

	<p>message the private can decipher it/decrypt it/verify. It also used in Digital signature/fingerprint for authentication of the sender with or without certificate for public known authenticity stored in a dedicated server.</p>
Private (signing key) and public (verification) key	<p>They are mathematically linked. The public is visible by others and private is intended to be only on the owner/s possession. The private decrypts the public's data content.</p>
Certificate	<p>A trust level meter of the signature. A certification of a fingerprint meaning its has been approved by other individuals or trusted dedicated website. A signature (sign) gets "Trust" level by certificate it is not needed though for any actual decryption only for identification/authenticity of the source for his level of trust approved by others or an authority.</p>
Sign/Digital Signature	<p>Anyone can encrypt a message using public key and send it somewhere, the decryption takes place only to those who have access to private key that generated that public key. If the message is just encrypted without any signature, then you can still read the message (with private key decryption) but you can't really tell who actually sent this message despite the fact that you could get it in your inbox email from a known address that doesn't mean it's the actual person that is sending it.</p> <p>A sign is a different approach from encryption "de-Sign" does not need decryption key to be known.</p> <p>When you sign a message (which happens with your private key), you generate a key with numbers strange looking (text encrypted basically only to the signature part not the actual message you want to transfer) but it does not say anywhere your name just numbers, now anyone with your public key (That private key who derived that public key can encrypt/decrypt/sign/authenticate) without any password needed, it can "decrypt/de-sign" the signature and reveal your name. If both takes place e.g., Encryption with signature then in order to read it you need the private key and also the "who" signed it (who is the actual sender) is being revealed as well.</p> <p>For a quick example you want to send a secret message to Person A you are Person B both have</p>

asymmetric keys but you don't have each other's private key only public. You encrypt the Message with Person's A public key so he/she can only be the one to decrypt it and alongside you sign it with your private key (Person B). Then the person A needs to have his private key and your public key so both the encrypted message refereeing to him can be revealed as long side with your name revealed by "decrypting" your signature with your public key (Person's B).

signing(message, KEY_{private}) → signature

Signing is a function that takes a message and a private key and producing a digital signature The message could be blank thought but that is pointless.

Anyone can use the public key to verify a digital signature:

verify(m, signature, KEY_{public}) → trueorfalse
