University of Wroclaw

Institute of Informatics

Igor Tryhub:

"Onion routing - how it works? How to use it? Is it secure? TOR - who stands behind it? Why is it not forbidden?"

Wroclaw 2015

Introduction

Onion routing is a technique for anonymous communication over a computer network. In an onion network, messages are encapsulated in multiple layers of encryption in the application layer of a communication protocol stack, analogous to layers of an onion. The encrypted data along with the destination IP address 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 innermost layer is decrypted, the message arrives at its destination without revealing the source IP address. The sender remains anonymous because each intermediary knows only the location of the immediately preceding and following nodes.[1]

TOR is one of the most well-known onion-routing-based software for enabling anonymous communication. The Parliamentary Office of Science and Technology deemed it, with approximately 2.5 mln users daily "by far the most popular anonymous internet communication system."[2] The name is an acronym derived from the original software project name The Onion Router.[3] TOR directs Internet traffic through a free, worldwide, volunteer network consisting of more than six thousand relays [4] to conceal a user's location and usage from anyone conducting network surveillance or traffic analysis.

TOR's use is intended to protect the personal privacy of users, as well as their freedom and ability to conduct confidential communication by keeping their Internet activities from being monitored. It was intended to be a platform for anonymous visiting web-sites, instant messaging and whistle-blowing.

TOR allows not only an opportunity for its users to become more secure and untraceable, but also offers different kinds of service providers to place their services within a .onion special-use top level domain. Such services can be directly accessible only through the TOR anonymity network and it is practically impossible to trace their IP-address, hence the location. Many of those are considered to be Dark web services and are often being argued about by anonymity adversaries:

- darknet markets (illegal drugs, weapons, software exploits);
- hacking services;
- fraud services (trafficking credit card and personal information, counterfeiting services);
- hitmen (contract killers, assassinations for hire);
- illegal and ethically disputed pornography;
- snuff films (murder, suicide);
- terrorism-related activities.

Because the deep web is secret, most of its content is not indexed by any search engine. Furthermore, the domain names of its services are not convenient for humans to memorize, as it is a case in a shallow web browsing. Moreover, the domain names of some services are being changed periodically for security reasons. Therefore, in order to retrieve any information, a user ought to either possess the actual version of the domain name or try to look it up somewhere. The most famous and attended resources can often be found on TORCH (TOR search engine) or Hidden Wiki. The latter one provides a current list of categorized resources along with their short description.[5]

How it works?

Now suppose we are a user who wants to securely send his or her message through the network without being traced back. You as a user download a list of relays inside the TOR network and randomly pick which relays (nodes) you are going to use to rule your traffic through. Then your computer needs to set up a connection with the chosen nodes and negotiate a common symmetrical key. This can be done using Diffie-Hellman Key Exchange algorithm. It relies on modular arithmetic and complexity of backward calculating the one-way functions used by negotiating parties by a man in the middle.

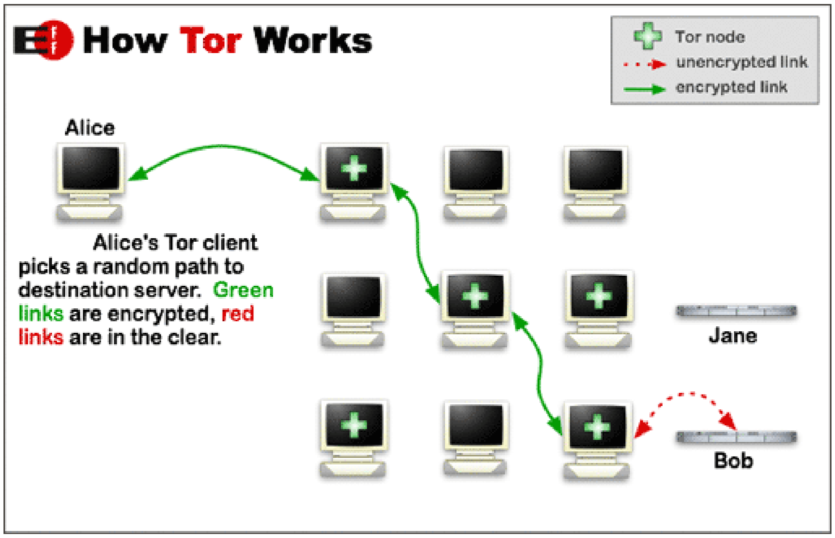
How Tor Works
Tor node
unencrypted link
encrypted link
Alice
Alice's Tor client picks a random path to destination server. Green links are encrypted, red links are in the clear.
Jane
Bob

Image source: https://www.torproject.org/about/overview.html.en

Putting it simple, we can explain it using an example of color mixing. Firstly, Alice and Bob agree upon their common paint and send it over the network. Then, each of them comes up with their secret color, known only to themselves. Afterwards, they mix up the agreed upon colors with their secret colors and send them over the network. Finally, Alice and Bob mix obtained mix from their counterpart with their secret color to get the common secret, which in a magic way appears to be the same for them both. Eventually, they have agreed upon a common secret without giving away they secret colors. And despite the man in the middle could eavesdrop their communication, he will not be able to easily come up with the common secret.

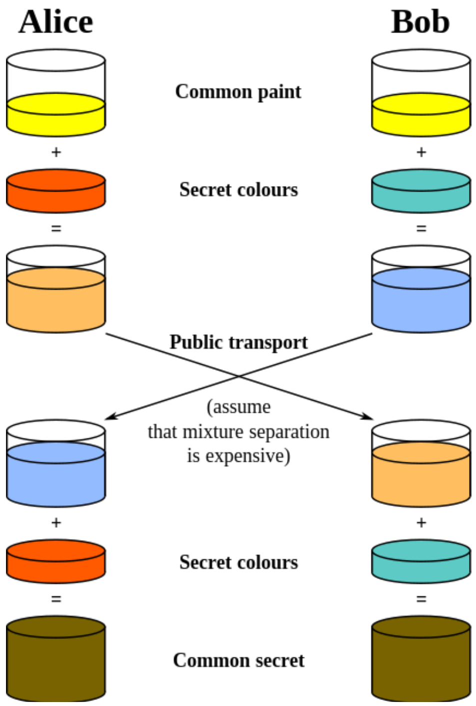

Image source: https://upload.wikimedia.org/wikipedia/commons/thumb/4/46/Diffie-Hellman_Key_Exchange.svg/427px-Diffie-Hellman_Key_Exchange.svg.png

After common keys have been negotiated for each of the chosen nodes, we can choose the order in which those nodes will be passing the information through a secure tunnel. The first node is going to be called the Guard, the middle ones (can be several) are going to be called Relays, and the last one - the Exit. The traffic is encrypted multiple times with different keys, starting from the Exit's key and ending with the Guard's key.

The message is ready for dispatching. At each node it is decrypted in order to read information about the node to which the message should be sent next. The process repeats until our data reaches the Exit node, which decrypts the last layer of encryption and transmit it to the destination. In this way none of the nodes doesn't know who is the actual sender and who is the actual receiver (except for the last one, which by the way can also see transmitted information in a plain text!). The only thing the nodes can see is who they got the packet of data from and who they need to send it to. In such a way we assure that each of the nodes know nothing more than is necessary for communication. In order to increase security, TOR relay routes periodically change (about every 10 min), so that to decrease the probability the compromised nodes take substantial part in communication being sent.

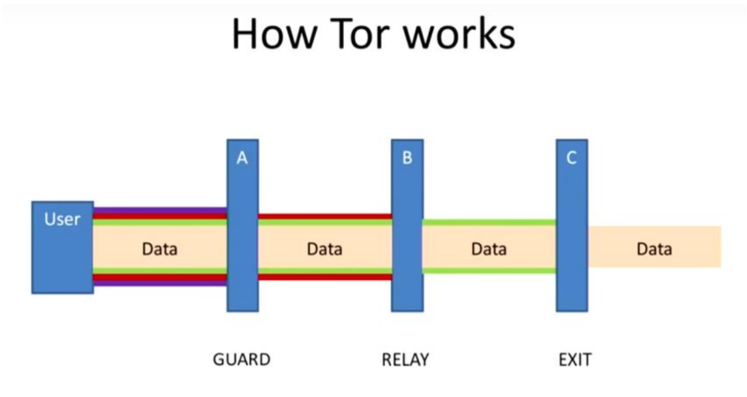

Image source: https://www.youtube.com/watch?v=-oTEoLB-ses&feature=youtu.be&t=1998

Is it secure?

TOR is not meant to completely solve the issue of anonymity on the web. Instead, it simply focuses on protecting the transportation of data so that certain sites cannot trace back the data to a given location. It is still possible for sites to backtrack to a location. TOR is not designed to erase a user's tracks but to simply make it less likely for sites to trace back to them.[6]

TOR, due to its design and internal mechanics, has several weaknesses many of which have been pointed out by scientists, volunteers and even governmental security agency whistle-blowers. Some of them concern the well known facts, such as that downloaded files can contain the pieces of executable code, which can send a request from the user's computer to a designated remote server and in such a way to reveal the IP-address and other sensitive data about the user. Or, as another example, that running an exit relay is potentially risky as it sends unencrypted data with unhidden IP-address. But others appear to be very valuable for the project developers.

For instance, one of the studies conveyed by the University of Luxemburg in 2013 have analyzed the security properties of TOR hidden services and shown that attacks to deanonymize hidden services at a large scale are practically possible with only a moderate amount of resources. They have demonstrated that collecting the descriptors of all TOR hidden services is possible in approximately 2 days by spending less than USD 100 in Amazon EC2 resources. Running one or more guard nodes then allows an attacker to correlate hidden services to IP addresses using a primitive traffic analysis attack. Furthermore, they have shown that attackers can impact the availability and sample the popularity of arbitrary hidden services not under their control by selectively becoming their hidden service directories.

To address these vulnerabilities they have also proposed countermeasures. However, they believe that the problems they have shown are grave enough to warrant a careful redesign of TOR’s hidden services.[7] To support this viewpoint, another article shows that TOR's structure makes it a perfect protocol for botnets. A "Skynet" botnet creator of possible German origin named “throwaway236236” came up with a TOR-powered trojan with DDoS, Bitcoin mining and Banking capabilities. The malware was quite well disguised and its creator would not probably have problems providing he didn't decide to boast on social media about his achievements.[9] Allegedly, the man earned quite a money selling banking information on the darknet. Besides, the bitcoin digging activity only (in 2012 he estimated it to be executed 30% globally by botnets) brought him 40$ per day.

Other scientists described their experiences with an international network of more than 30 nodes. They have come up with a list of open problems in anonymous communication. Let us now highlight some of the problems identified along with proposed solutions:

- **Passive attacks**
    - *Observing user traffic patterns.* Observing a user’s connection will not reveal her destination or data, but it will reveal traffic patterns (both sent and received).
    - *Observing user content.* While content at the user end is encrypted, connections to responders may not be (indeed, the responding website itself may be hostile). While filtering content is not a primary goal of Onion Routing, TOR can directly use Privoxy and related filtering services to anonymize application data streams.

  - *End-to-end timing correlation.* TOR only minimally hides such correlations. An attacker watching patterns of traffic at the initiator and the responder will be able to confirm the correspondence with high probability. The greatest protection currently available against such confirmation is to hide the connection between the onion proxy and the first TOR node, by running the Onion Proxy (OP) on the TOR node or behind a firewall. This approach requires an observer to separate traffic originating at the onion router from traffic passing through it: a global observer can do this, but it might be beyond a limited observer's capabilities.
  - *End-to-end size correlation.* Simple packet counting will also be effective in confirming endpoints of a stream. However, even without padding, we may have some limited protection: the leaky pipe topology means different numbers of packets may enter one end of a circuit than exit at the other.
  - *Website fingerprinting.* All the effective passive attacks above are traffic confirmation attacks, which puts them outside our design goals. There is also a passive traffic analysis attack that is potentially effective. Rather than searching exit connections for timing and volume correlations, the adversary may build up a database of "fingerprints" containing file sizes and access patterns for targeted websites. He can later confirm a user's connection to a given site simply by consulting the database. Defenses could include larger cell sizes, padding schemes to group websites into large sets, and link padding or long-range dummies.
- **Active attacks**
  - *Compromise keys.* An attacker who learns the TLS session key can see control cells and encrypted relay cells on every circuit on that connection; learning a circuit session key lets him unwrap one layer of the encryption. An attacker who learns an Onion Router's (OR) TLS private key can impersonate that OR for the TLS key's lifetime, but he must also learn the onion key to decrypt create cells (and because of perfect forward secrecy, he cannot hijack already established circuits without also compromising their session keys). Periodic key rotation limits the window of opportunity for these attacks. On the other hand, an attacker who learns a node's identity key can replace that node indefinitely by sending new forged descriptors to the directory servers.
  - *Run a recipient.* An adversary running a webserver trivially learns the timing patterns of users connecting to it, and can introduce arbitrary patterns in its responses. End-to-end attacks become easier: if the adversary can induce users to connect to his webserver (perhaps by advertising content targeted to those users), he now holds one end of their connection. There is also a danger that application protocols and associated programs can be induced to reveal information about the initiator. TOR depends on Privoxy and similar protocol cleaners to solve this latter problem.
  - *Run an onion proxy.* It is expected that end users will nearly always run their own local onion proxy. However, in some settings, it may be necessary for the proxy to run remotely— typically, in institutions that want to monitor the activity of those connecting to the proxy. Compromising an onion proxy compromises all future connections through it.
  - *DoS non-observed nodes.* An observer who can only watch some of the TOR network can increase the value of this traffic by attacking non-observed nodes to

shut them down, reduce their reliability, or persuade users that they are not trustworthy. The best defense here is robustness.

- *Run a hostile OR.* In addition to being a local observer, an isolated hostile node can create circuits through itself, or alter traffic patterns to affect traffic at other nodes. Nonetheless, a hostile node must be immediately adjacent to both endpoints to compromise the anonymity of a circuit. If an adversary can run multiple ORs, and can persuade the directory servers that those ORs are trustworthy and independent, then occasionally some user will choose one of those ORs for the start and another as the end of a circuit. If an adversary controls m > 1 of N nodes, he can correlate at most (m/N)^2 of the traffic— although an adversary could still attract a disproportionately large amount of traffic by running an OR with a permissive exit policy, or by degrading the reliability of other routers.
- *Distribute hostile code.* An attacker could trick users into running subverted TOR software that did not, in fact, anonymize their connections—or worse, could trick ORs into running weakened software that provided users with less anonymity. We address this problem (but do not solve it completely) by signing all TOR releases with an official public key, and including an entry in the directory that lists which versions are currently believed to be secure. To prevent an attacker from subverting the official release itself (through threats, bribery, or insider attacks), we provide all releases in source code form, encourage source audits, and frequently warn our users never to trust any software (even from us) that comes without source.[8]

Why is it not forbidden?

Many national security agencies having hard times tracing and fighting the criminals using TOR and similar anonymizing services to commit crimes worldwide. Indeed, if we take a look at the structure of the content of the hidden services hosted in TOR's deep web, we can see that substantial part of those are illegal activities. We would be even more surprised if we looked at the statistics of the structure of data being requested most frequently.

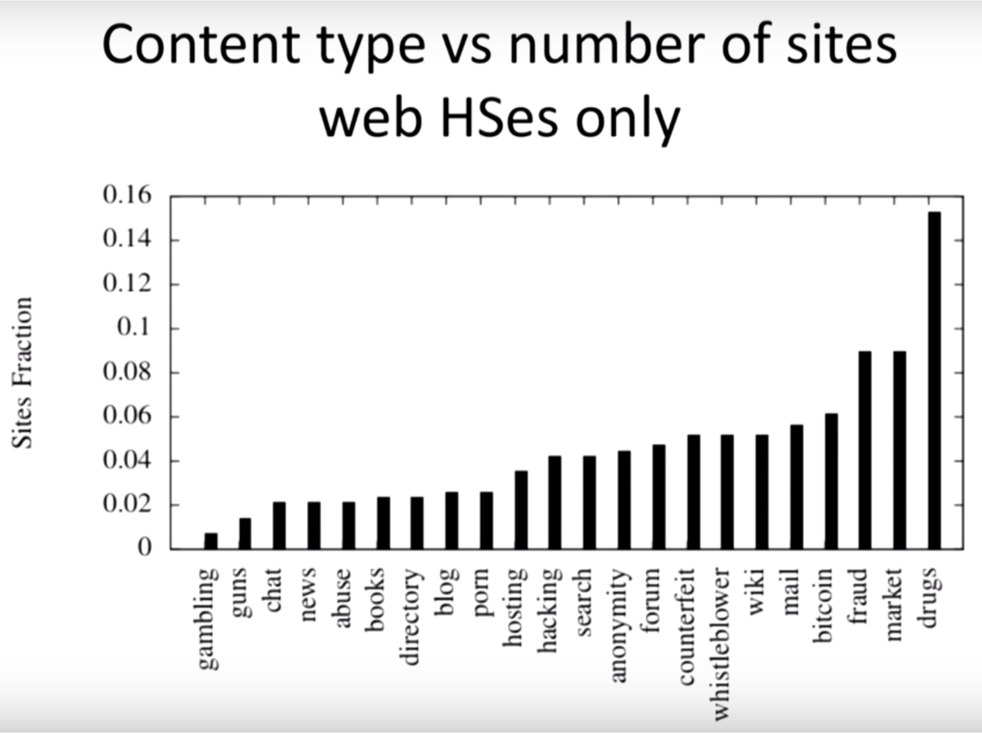

Image source: https://www.youtube.com/watch?v=-oTEoLB-ses&feature=youtu.be&t=1998

But on the other hand, the technology is supported by many activists who advocate freedom of speech. They argue that if someone wants to commit a crime, there are many other ways to do it, and onion routing is by far not the safest option. But for those people who do nothing illegal and just care about their privacy and prosecution by suppressive political systems, this technology can be the only option to share information.

Moreover, it has been shown in the "Is it secure?" section, that up to the date, onion routing has many flaws and therefore faces many threats by potential attackers. It can be assumed, that taking into consideration the budgets and know-how of national security agencies, they would have not much problems pursuing criminals. Otherwise, how can we explain that the most notorious cyber-criminals such as creators of Silk Road and Liberty Reserve were traced and arrested were in such a short period of time?

Possible Improvements and Alternatives

As we have found out, onion routing is a way to protect oneself on the web, but it is not the ultimate all-in-one tool. There are some other good technologies, which can either be used as a substitute to onion routing, or as another layer of protection used concurrently.

One of them is Virtual Private Network (VPN), which can be combined with TOR on different stages of the internet access.

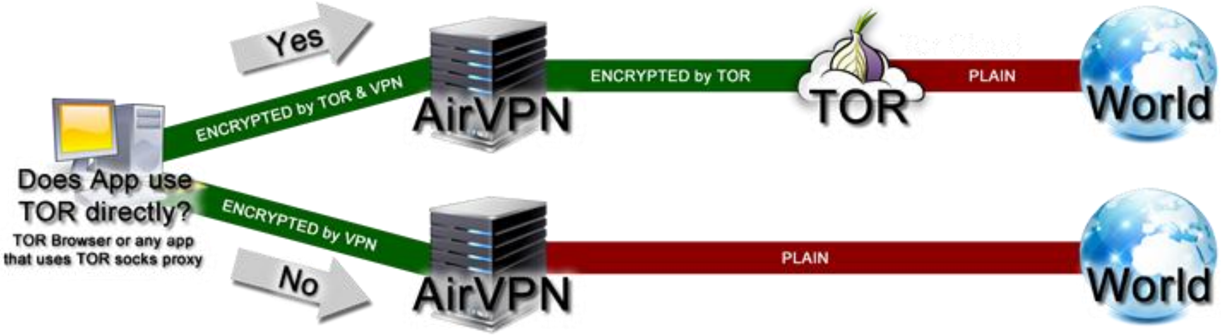

Image source: https://airvpn.org/static/pages/tor/tor_over_airvpn.png

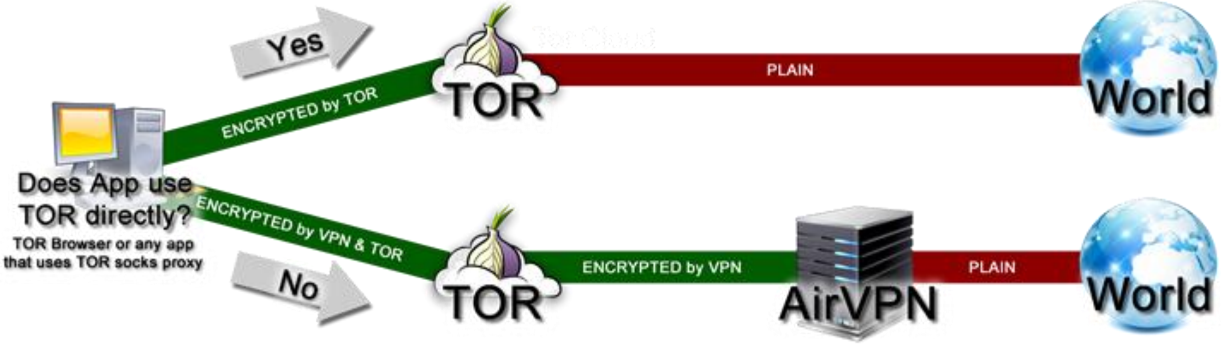

Image source: https://airvpn.org/static/pages/tor/airvpn_with_tor.png

Another one is called I2P (sometimes called garlic-routing). After a preliminary research it appears to be less known, less-usable, but eventually more secure way to perform the same things on the Internet the onion routing allows users to do. The two primary differences between TOR / Onion-Routing and I2P are again related to differences in the threat model and the out-proxy design (though TOR supports hidden services as well). In addition, TOR takes the directory-based approach - providing a centralized point to manage the overall 'view' of the network, as well as gather and report statistics, as opposed to I2P's distributed network database and peer selection.

Benefits of TOR over I2P:

- Much bigger user base; much more visibility in the academic and hacker communities; benefits from formal studies of anonymity, resistance, and performance; has a non-anonymous, visible, university-based leader
- Has already solved some scaling issues I2P has yet to address
- Has significant funding
- Has more developers, including several that are funded
- Big enough that it has had to adapt to blocking and DOS attempts
- Designed and optimized for exit traffic, with a large number of exit nodes
- Better documentation, has formal papers and specifications, better website, many more translations
- More efficient with memory usage
- TOR client nodes have very low bandwidth overhead
- Centralized control reduces the complexity at each node and can efficiently address Sybil attacks

Benefits of I2P over TOR:

- Designed and optimized for hidden services, which are much faster than in TOR
- Fully distributed and self organizing
- Peers are selected by continuously profiling and ranking performance, rather than trusting claimed capacity
- Floodfill peers ("directory servers") are varying and untrusted, rather than hardcoded
- Small enough that it hasn't been blocked or DOSed much, or at all
- Peer-to-peer friendly
- Packet switched instead of circuit switched
  - implicit transparent load balancing of messages across multiple peers, rather than a single path
  - resilience vs. failures by running multiple tunnels in parallel, plus rotating tunnels
  - scale each client's connections at O(1) instead of O(N) (Alice has e.g. 2 inbound tunnels that are used by all of the peers Alice is talking with, rather than a circuit for each)
- Unidirectional tunnels instead of bidirectional circuits, doubling the number of nodes a peer has to compromise to get the same information.
- Protection against detecting client activity, even when an attacker is participating in the tunnel, as tunnels are used for more than simply passing end to end messages (e.g. netDb, tunnel management, tunnel testing)
- Tunnels in I2P are short lived, decreasing the number of samples that an attacker can use to mount an active attack with, unlike circuits in TOR, which are typically long lived.[10]

Bibliography

[1]. http://www.onion-router.net/Publications/CACM-1999.pdf

[2]. http://www.dailydot.com/politics/uk-briefing-tor-child-abuse-minor-role/

[3]. Li, Bingdong; Erdin, Esra; Güneş, Mehmet Hadi; Bebis, George; Shipley, Todd (14 June 2011). "An Analysis of Anonymity Usage". In Domingo-Pascual, Jordi; Shavitt, Yuval; Uhlig, Steve. *Traffic Monitoring and Analysis: Third International Workshop, TMA 2011, Vienna, Austria, April 27, 2011, Proceedings*. Berlin: Springer-Verlag. pp. 113–116.ISBN 978-3-642-20304-6. Retrieved 6 August 2012.

[4]. http://torstatus.blutmagie.de/

[5]. http://thehiddenwiki.org/

[6]. https://www.torproject.org/about/overview.html.en

[7]. http://www.ieee-security.org/TC/SP2013/papers/4977a080.pdf

[8]. http://www.onion-router.net/Publications/tor-design.pdf

[9]. https://community.rapid7.com/community/infosec/blog/2012/12/06/skynet-a-tor-powered-botnet-straight-from-reddit

[10]. https://geti2p.net/en/comparison/tor