

Tor

Anonymous Networks

Devin Vitone

April 30, 2008

Abstract

Tor, a second-generation implementation of onion routing, provides anonymous Internet communications. The Tor project originated as U.S. Navy sponsored research but has since been taken over by the EFF. By routing traffic through a series of volunteer nodes and finally to an exit point Tor can defend against traffic analysis independent of the application layer.

As encryption can be used to hide data contents from a third party, anonymous networks allow the identity of the source and the recipient to remain private. Determining who is talking to who, or who is visiting what website can pose just as large of a security risk as unencrypted data in some cases. An example of this might be a field agent connecting into their departments network and in doing so revealing their affiliation and location to an adversary monitoring connections. In this last case encryption does not help. Another situation warranting use might be when privacy is an issue, encrypting connections may not help if general information is known about a website (perhaps a whistle blower connecting to a watchdog organization report website). Tor networks can also be used to protect against DoS attacks by masking the servers true location and identity. Although Tor has many legitimate uses, many have used it to facilitate illegal activity.

Contents

Abstract	i
Contents	ii
Figures	iv
Abbreviations	v
1 Introduction	1
2 Anonymity Network Applications	1
2.1 Law Enforcement	1
2.2 Military & Government	1
2.3 Businesses & Corporate Networks	2
2.4 Journalists and Political Dissidents	2
2.5 Individual	2
2.6 Criminals	2
3 How Anonymous Networks Work	3
3.1 Anonymous Proxies	3
3.2 Chaum's Mix	4
3.3 Mixed Cascade	5
3.4 Onion Routing	5
3.5 Tor	6
4 Tor Implementation	8
4.1 Connections	8
4.2 Cells & Circuits	9
4.3 Hidden Services	10
5 Tor Strengths	10
6 Tor Weaknesses	12
6.1 DNS Leaks	12

6.2	Browser Leaks	12
6.3	Traffic Analysis	12
6.4	The China Problem	12
7	Conclusion	13

List of Figures

1	Anonymizing Proxy [6]	3
2	Chaum's Mix [5]	4
3	Cascaded Mixed Network[5]	5
4	Onion Layers[1]	6
5	Tor Circuit[2]	7
6	Tor Circuit[2]	7
7	Tor Hidden Service[2]	11
8	Tor Connected Hidden Service[2]	11

Abbreviations

DDoS	Distributed Denial of Service
DoS	Denial of Service
DNS	Domain Name Service
EFF	Electronic Freedom Foundation
NAT	Network Address Translation
TCP	Transmission Control Protocol
UDP	Universal Datagram Protocol
VPN	Virtual Private Network

1 Introduction

Anonymity can be defined as the state of being unidentifiable within a set up of subjects, and determine who is talking to who. It is important to note that anonymity cannot be achieved by one's self. The goal of anonymity is to provide a way to disassociate identities from actions. In the past pen names could be used to conceal one's identity, in the realm of the Internet an anonymity network such as Tor can be used to achieve this goal.

2 Anonymity Network Applications

Many users and organizations make use of an anonymous network for various reasons. Although their motivations and goals may differ, it is the diversity and size of the network that makes the allow these needs to be met.

2.1 Law Enforcement

Police often have the use for covert operations a situation may arise when an officer needs to relay information back to colleagues while not revealing their identity. Anonymous tip lines are also useful to police. An informant may wish to send information to the police but not reveal their identity as it may create a risk of harm from other individuals or even prosecution from police if revealing this information would expose their involvement in crimes.

2.2 Military & Government

To military and governments timing of communications as well as connections between users and known servers can reveal as much information as the data itself. An agent in a foreign country may reveal their identity and location simultaneously by simply connecting to a known server from their respective government, this applies to covert military units as well. Analysis of connections may also correlate with known actions, perhaps a flood of traffic from a certain location before the start of a conflict, major air strike, ground movements, etc. Tor may also be used to harden servers against DoS or DDoS attacks against their servers.

2.3 Businesses & Corporate Networks

Traffic patterns may reveal internal business activities. A competitor may configure their servers to show different content to the known IP addresses of their competitors, this may be done as a means of misdirection. To combat this Tor may be used so the competitor can no longer discriminate against them. Another scenario could be a company wishing to test a new service but not reveal that they are the ones providing it. A sudden spike in traffic from IP addresses associated with an engineering department to a job search site may reveal company instability to outsiders prematurely, using an anonymity network could help prevent this leak of information.

2.4 Journalists and Political Dissidents

Embedded reporters may go under cover in a hostile environment, anonymity is very important in this case. Likewise the unobstructed publishing of information may be important, NATed computers with not public IP addresses can still act as hosts as the connections are tunneled through the NAT. In certain countries, such as China, political speech may be limited by the government, Tor can often circumvent these blocking policies. Whistle blowers can use anonymous networks to expose activities without compromising their jobs, safety or security.

2.5 Individual

Private citizens may wish to preserve their identity whether it is from governments monitoring their traffic, advertisers and companies mining user data. Another case would be a person who contracts a disease but does not want to share this information with others as it may provide embarrassment.

2.6 Criminals

Criminal hackers already have anonymity — Tor networks provide little to no benefit to them — simply hacking other machines and using those to launch attacks provides sufficient anonymity. Nonetheless illegal activity does occur on Tor networks this may range from benign activity considered illegal by governments of the chinese and iranians to activities illegal in countries such as the U.S like pedophilia (very blatantly spoken of and posted on the primary Tor forums).

3 How Anonymous Networks Work

IP packets contain source and destination fields, anyone who can read the packet can see this information. In order to provide anonymity this information must be hidden somehow. That is the task of anonymous network protocols.

Various methods have been employed to create anonymous networks, overtime these implementations built upon previous designs becoming more advanced. The most common method to anonymize traffic is to mix traffic from one host with traffic from another host.

3.1 Anonymous Proxies

Possibly the simplest example of a network model that provides anonymity is the use of a proxy server as in figure 1. A proxy server receives a request from a client to connect to a server, most often a web server, the proxy now makes the request to the destination server as if it were the client. This conceals the identity of a particular client, this can allow traffic to appear to originate from a different location that may not have certain restrictions. Anonymous proxies are unfortunately a single point of failure vulnerable to subpoenas, hackers, etc. This is a very basic example that has many limitations not found in other more advanced configurations.

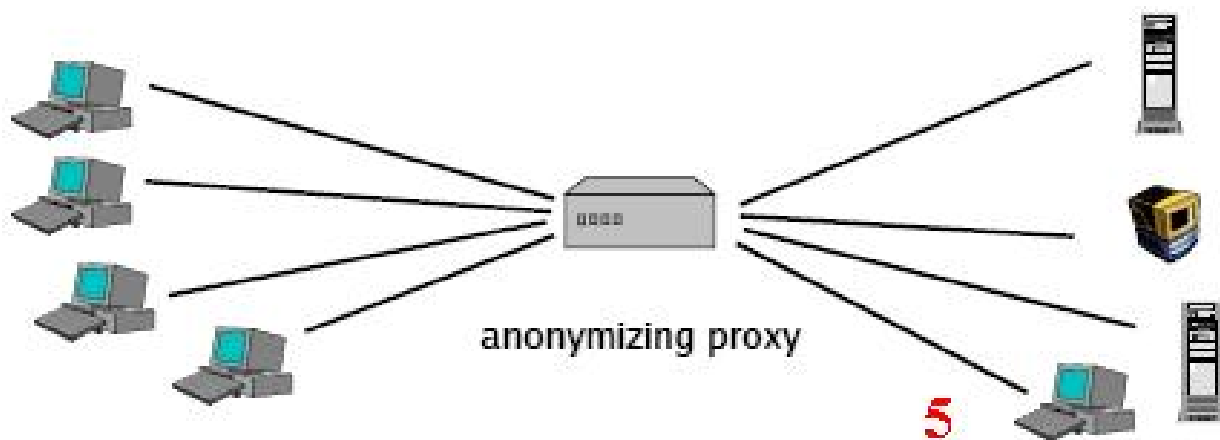


Figure 1: Anonymizing Proxy [6]

3.2 Chaum's Mix

A Chaum's mix is one of the earliest true anonymous networks. Proposed by David Chaum in 1981, special servers called mixes would be used to provide anonymous email. This is very similar to what a proxy server does except traffic is padded and encrypted using a public key – private key method. Chaum's mix network protocol is a Store and Forward Protocol. Figure 2 is a chaum mixed network.

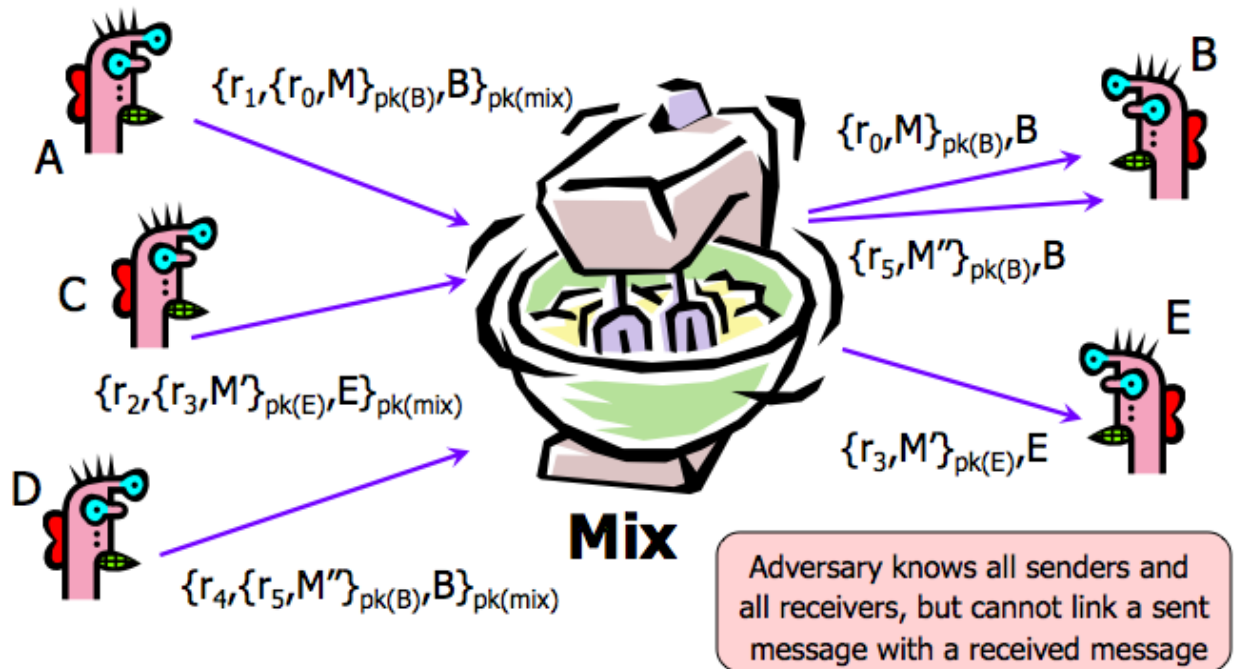


Figure 2: Chaum's Mix [5]

Client A wishes to send message M to B, A encrypts M and random number r_0 with B's private key, A then encrypts with $pk(mix)$ the part previously encrypted concatenated with a $pk(b)$ concatenated with random number r_1 and address B. This message is sent to Mix which will decrypt this outer layer and retrieve address B. The inner part is now sent to B and the message is decrypted revealing the message M.

$$A \rightarrow \text{Mix}: (r_1, (r_0, M)_{pk(B), B})_{pk(mix)}$$

$$\text{Mix} \rightarrow B: (r_0, M)_{pk(B)}$$

Using a Chaum mixed network an adversary can not tell who is sending messages to who. Unfortunately if an adversary has compromised the Mix server they can also determine who is talking to who. Another solution had to be developed to counteract this.

3.3 Mixed Cascade

Building on the simple design of a chaum mixed network, a series of mixed networks can be strung together. So long as one mix server is friendly anonymity is preserved. See figure 3 Mixed networks effectively provide anonymity at the cost of high latencies and high volumes of network traffic — used to pad network traffic — this is suitable for email, but protocols such as http required lower latencies. These high latencies were caused not only by the traffic padding but also by the costly asymmetric encryption and decryption routines necessary at each stage.



Figure 3: Cascaded Mixed Network[5]

3.4 Onion Routing

Borrowing many concepts from Chaum’s mixed networks Onion Routing attempted to overcome some of the high latency limitations of Chaum mixes. The client starts by choosing its route to the server and encrypting that route with each node’s public key in a nested fashion, the very inner layer will contain the clear text message. See figure 4. As in the Chaum mixes several nodes may be compromised but as long as there is at least one honest node anonymity can be preserved. Although this method provides for lower latencies than in Chaum mixes, the process of using asymmetric keys is still costly. Each node only learns the identity of the next node. Onion networks can be set up to establish a reverse route to allow messages to be sent back to the client. This is called a reply onion. The reply onion is sent along with the message to the final recipient, since it is encrypted in the same manner that the initial onion route is it will not expose the identity of the original sender unless the public-key encryption is broken or all the routers along the return path have been compromised by the attacker. Due to the fact that onion routing is not a Store and Forward architecture as chaum mixes are onion routed communications are more susceptible to timing attacks on low traffic networks. Onion routing is limited by the fact it is not a two way connection and has high overhead of the asymmetric key encryption and decryption routines, this makes it impractical for most Internet

traffic.[5][1]

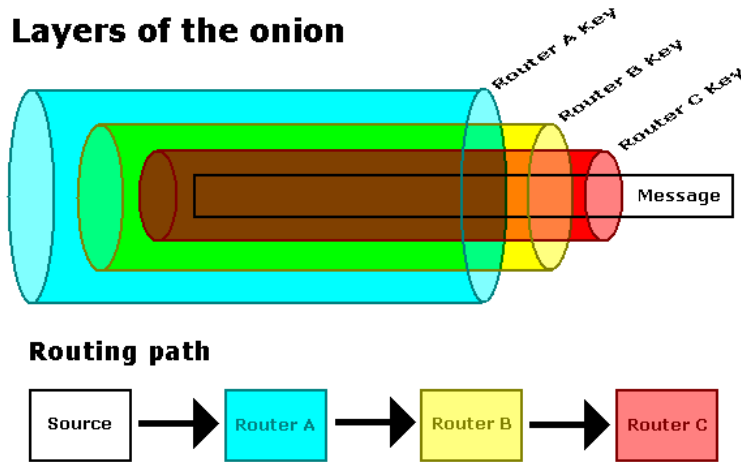


Figure 4: Onion Layers[1]

3.5 Tor

To overcome the limitations of the classical onion networks a second generation of onion routing was created. Rather than using a nested public-key structure to send messages Tor uses private keys. Tor networks are far more suited to transport TCP connections and provide for much lower latencies than classical onion routing do to this architecture.

Tor works by setting up a circuit between the client and destination. Circuits are established when the client requests a list of Tor nodes then selects a path. Each node only knows which two links it is connected to but never the whole path or even how deep into the route they are. While setting up the circuit the path is established one node at a time. A symmetric key is negotiated for each link and all traffic is tunneled through to the next node. Nodes can see only the nodes they receive from and send to. Once established the circuit can be used to route TCP streams, any application using or supporting socks can use this connection. Figure 6 illustrates an established connection. The process of establishing a circuit is repeated approximately every 10 minutes to prevent association of traffic to a particular user. Figure 6 illustrates this process[5, 6, 2]

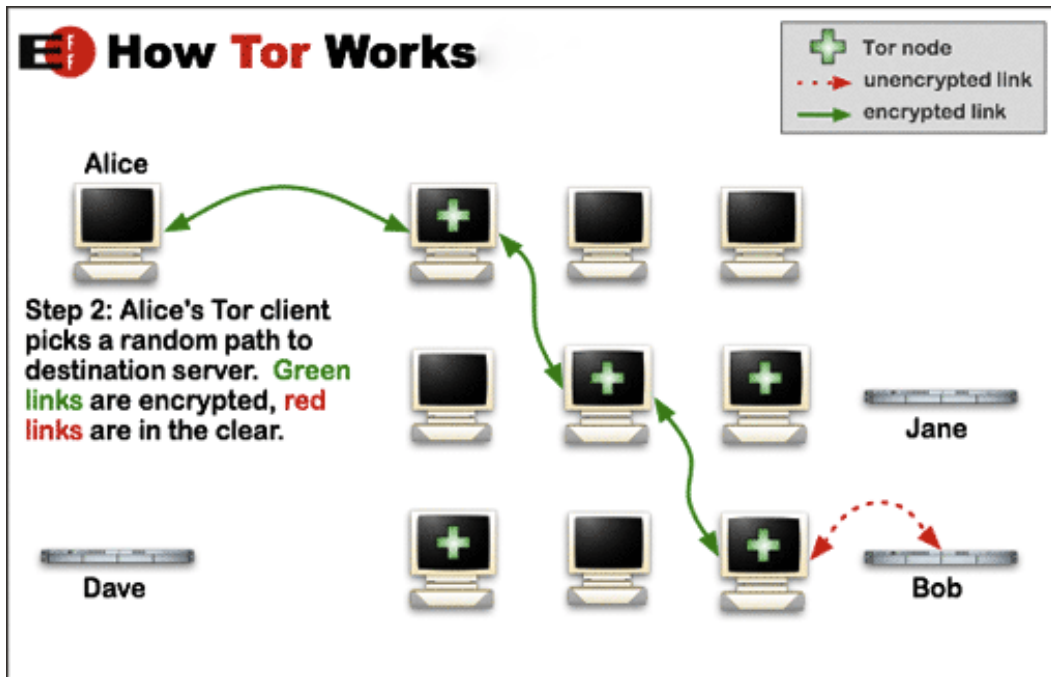


Figure 5: Tor Circuit[2]

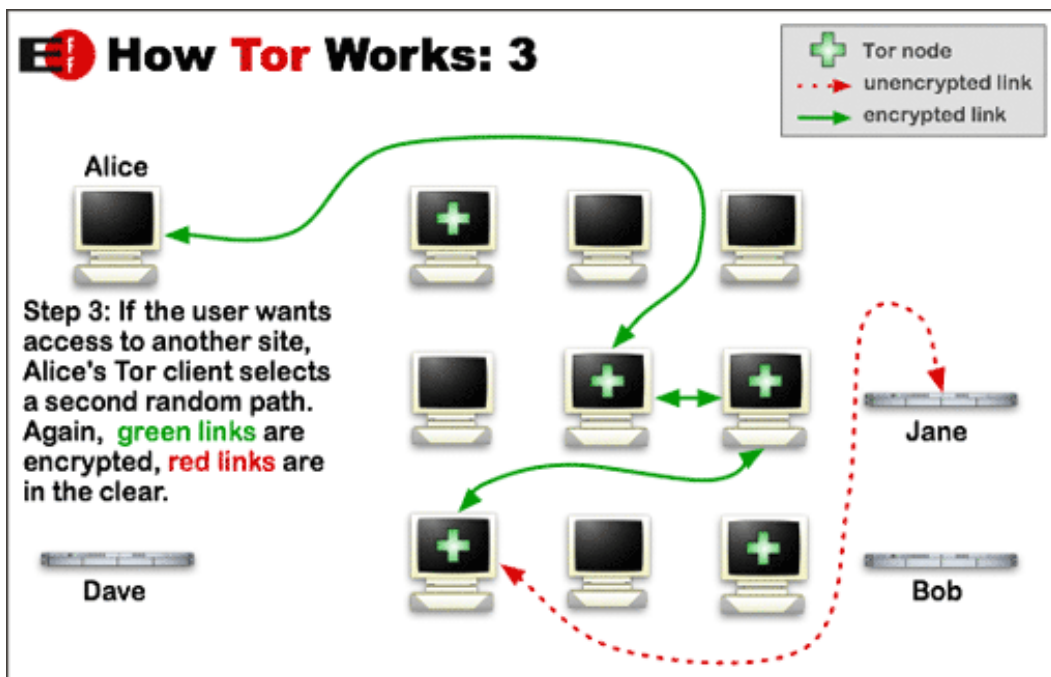


Figure 6: Tor Circuit[2]

4 Tor Implementation

4.1 Connections

As a form of steganography Tor traffic masks it self as other protocols, because of this reason Tor is not vulnerable to protocol filtering. All directory requests use http, and all Tor connections use https (TLS/SSLv3). This includes connections between client and nodes, and node to node. While negotiating the TLS handshake 3 methods are available for use. Older versions used a method of sharing certificates first. One of these is a short term (up to one day) X.509 certificate used to negotiate sessions and the other is a longer term (at least one week) self-signed X.509 certificate, which is used to to verify the identity of the node. After the certificates have been swapped the TLS negotiation continues, each of the parties swaps a list of supported ciphers chosen from and only from the following list:

```
TLS_DHE_RSA_WITH_AES_256_CBC_SHA  
TLS_DHE_RSA_WITH_AES_128_CBC_SHA  
SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA  
SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA
```

This is known as a “Certificate up-front” handshake.

The second type of TLS handshake is the “renegotiation” handshake. In this method the initiator begins the process, the responder sends a one time certificate, then the handshake continues. Once the handshake has completed a Hello is performed, again exchanging a list of supported ciphersuits, only this time in addition to the previous list from the “certificate up-fron” method only this time at least one suite not found in that list much be sent. This is used to differentiate it from the first version of the protocol.

A third method, now required in all new implementations, called “backwards-compatible renegotiations” is designed to allow connections to clients supporting both of the previously mentioned methods. To determine which method to continue as the list of ciphersuites is checked, if there are extras it is a renegotiator, if not it rolls back to support the certification up-front method.

4.2 Cells & Circuits

Fixed length packets called “cells” are used to transport data within the established TLS connection, in the version one protocol there are 3 fields in the cell: CircID, Command, and Payload. CircID is 2 bytes long and specifies which circuit a cell belongs to. Command which is 1 byte long specifies the type of cell: padding, create, created, relay, destroy, create_fast, created fast, versions, and netinfo. The third field is the payload. In version two an additional 2 byte field is used to specify the length of the cell.

When creating a circuit Diffie-Helmen is used to set up a circuit key. Directly from the tor spec the following procedure is used to create a circuit.

When creating a circuit through the network, the circuit creator (OP) performs the following steps:

1. Choose an onion router as an exit node (R_N), such that the onion router’s exit policy includes at least one pending stream that needs a circuit (if there are any).
2. Choose a chain of $(N-1)$ onion routers ($R_1 \dots R_{N-1}$) to constitute the path, such that no router appears in the path twice.
3. If not already connected to the first router in the chain, open a new connection to that router.
4. Choose a circID not already in use on the connection with the first router in the chain; send a CREATE cell along the connection, to be received by the first onion router.
5. Wait until a CREATED cell is received; finish the handshake and extract the forward key Kf_1 and the backward key Kb_1 .

6. For each subsequent onion router R (R_2 through R_N), extend the circuit to R .

To extend the circuit by a single onion router R_M , the OP performs these steps:

1. Create an onion skin, encrypted to R_M ’s public onion key.
2. Send the onion skin in a relay EXTEND cell along the circuit (see section 5).
3. When a relay EXTENDED cell is received, verify KH, and calculate the shared keys.

The circuit is now extended.

If an unrecoverable error is occurs or there are no more stream over that circuit, the circuits are torn down. This can occur hop-by-hop or the entire circuit at once depending upon future need. To open a new data

stream a circuit that can possibly connect to the destination is selected, an unused stream_id is selected and a relay_cell with the following format is used “ADDRESS | ‘:’ | PORT | [00]”. Based upon whether the end relay can connect or not the exit point either responds with a RELAY_END or RELAY_CONNECTED response. If a RELAY_CONNECTED is received by the client data may be passed through the circuit now.[4]

Tor routes are always of length 3, plus 1 if a hidden service is being connected to. Three is said to effectively provide anonymity while keeping the overhead at a minimum.

4.3 Hidden Services

Hidden services are servers reachable only by Tor, in short they allow others to connect to them without revealing the location of the servers. The reasons for this may range from preventing a DoS attack from being effective or simply to prevent others from knowing who is hosting content. To create a hidden service a server chooses a number of relays to act as Introduction Points, these are the machines others will connect through to access the server. The server starts by creating a circuit to the intro points. Figure 7. The service is then advertised with the directory servers.

When a client wishes to connect to a Hidden service it first queries the directory service for the intro points identities. The directory server responds, at this point the client sets up another Tor circuit to a relay to act as a rendezvous point. The client then communicates the identity of the rendezvous point to the server via the intro point. The server now creates a circuit to the rendezvous point, now the connection is complete. Figure 8.

5 Tor Strengths

Tor is one of the more effective and efficient anonymity networks. Combining many of the lower latency features of proxy based solutions with the strength of distributed trust of mixed nets, Tor is both usable and effectively anonymous. With a sizable number of users and easy to use client software Tor can be a practical solution that doesn't require much from the user.

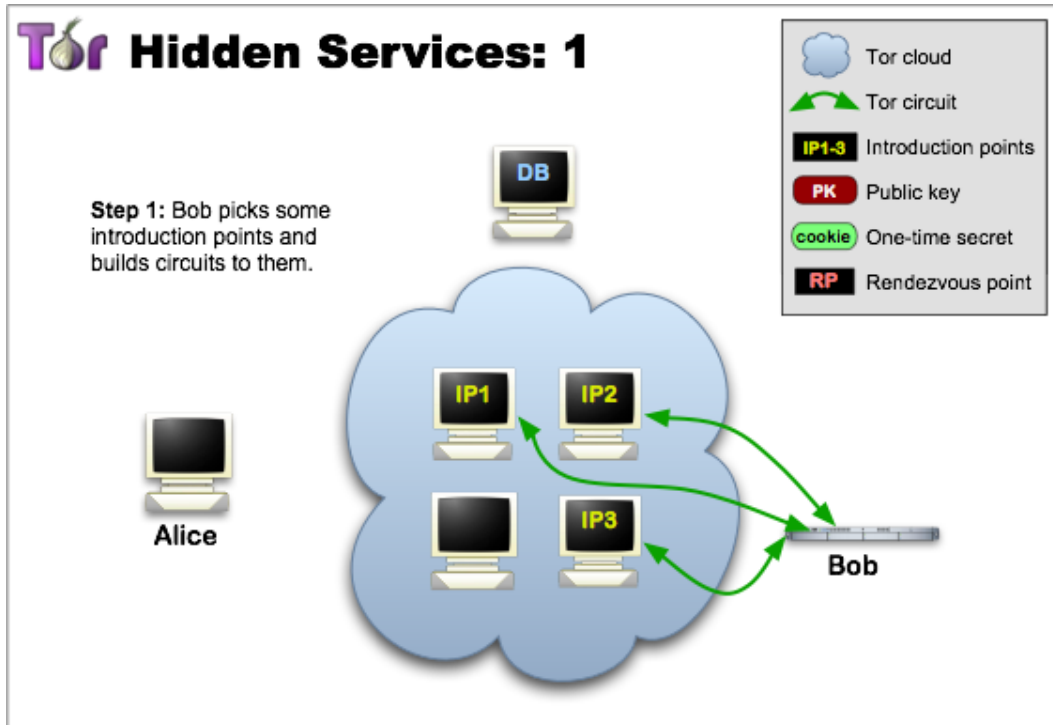


Figure 7: Tor Hidden Service[2]

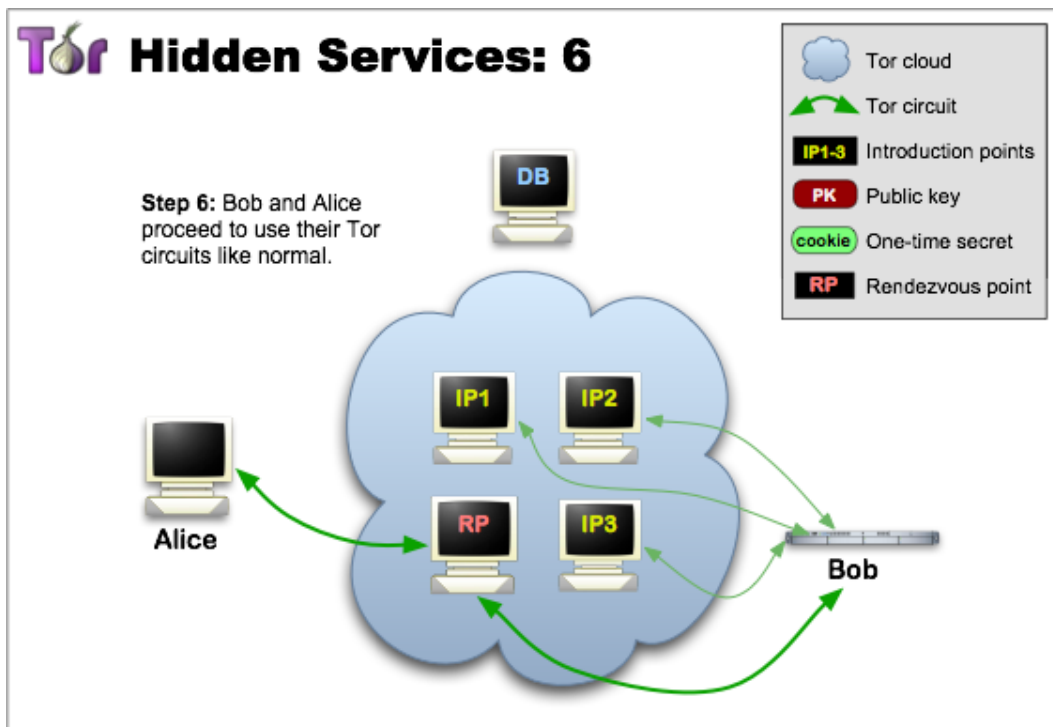


Figure 8: Tor Connected Hidden Service[2]

6 Tor Weaknesses

As with any system there are always flaws, or trade offs that must be made. By managing where the risks are many design decisions were made that still leave vulnerabilities at the cost of efficiency or probability of the risk.

6.1 DNS Leaks

Since DNS is a UDP protocol client machines will often attempt to query DNS servers outside of the Tor network (which is exclusively TCP streams). By making these requests anyone monitoring the clients connection can determine the servers the client is attempting to access. These protocols need to be Torized so that they can connect over a TCP socket.

6.2 Browser Leaks

Java, Javascript, Flash, Silverlight, Quicktime, etc can all be used to collect network information from users, for this reason it is recommended that users disable plugins while using Tor. Tor can provide an infrastructure for anonymity but a user can still give out information since the content of the streams is not monitored by the protocol. Many also make the mistake of assuming encryption is not needed, if it is something that the user does not want others to read it should be encrypted (there is added exposure to other users reading their traffic at exit points).

6.3 Traffic Analysis

Since traffic is not, and cannot be padded in Tor traffic analysis may still be possible. Correlating timing of traffic at end points is the most effective means of attacking Tor. If traffic timing and packet size match up it is possible to determine that a user may be communicating with someone. This requires an attacker to be either extremely lucky to find this or have a global view of the network. [3]

6.4 The China Problem

Having a large number of resources and national control of a network can prove to pose a risk to anonymity. If enough malicious Tor nodes are created by someone with enough resources anonymity can be reduced greatly. If the attacker can see the two ends of the connection they may be able to compromise the anonymity. Additionally if they have the ability to block or cut off nodes they can stop users from connecting or deter-

mine who they are by watching traffic patterns when they cut off links (this is especially true if padding was used). China is a nation in a position to do this, and has been known to attack anonymity networks in the past, for this reason it is sometimes referred to as the china problem. [2]

7 Conclusion

Although many tradeoffs have been made Tor has proven itself effective and usable, with an ever growing network of users the anonymity increases as users have a larger pool to hid in. Nonetheless Tor has limited itself to TCP, which does make it less bloated, and protects against IP finger printing but limits the abilities. From personal research (exploring popular Tor hidden service forums) Tor does appear unfortunately to be used primarily for illegal activity (trading pedophilia). For all its shortcomings and usage by as the Tor Project refers to them 'jerks,' Tor is able to provide legitimate users in totalitarian regimes a means of access to the internet, protection to businesses, and the free flow of information.

[5]

References

- [1] Onion routing. http://en.wikipedia.org/wiki/Onion_routing.
- [2] The Tor Project Inc. The tor project inc. <http://www.torproject.org/>, 2007.
- [3] Steven J. Murdoch and George Danezis. Low-cost traffic analysis of tor. <http://www.cl.cam.ac.uk/~sjm217/papers/oakland05torta.pdf>.
- [4] Nick Mathewson Roger Dingledine. Tor protocol specification. <https://www.torproject.org/svn/trunk/doc/spec/tor-spec.txt>, 2007.
- [5] Vitaly Shmatikoc. Anonymity networks. http://www.cs.utexas.edu/~shmat/courses/cs378_fall107/26anon.ppt, 2007.
- [6] F.W.J. van Geelkerken. Tor:the onion router. <http://www.iusmentis.com/society/privacy/remailers/onionrouting/>.