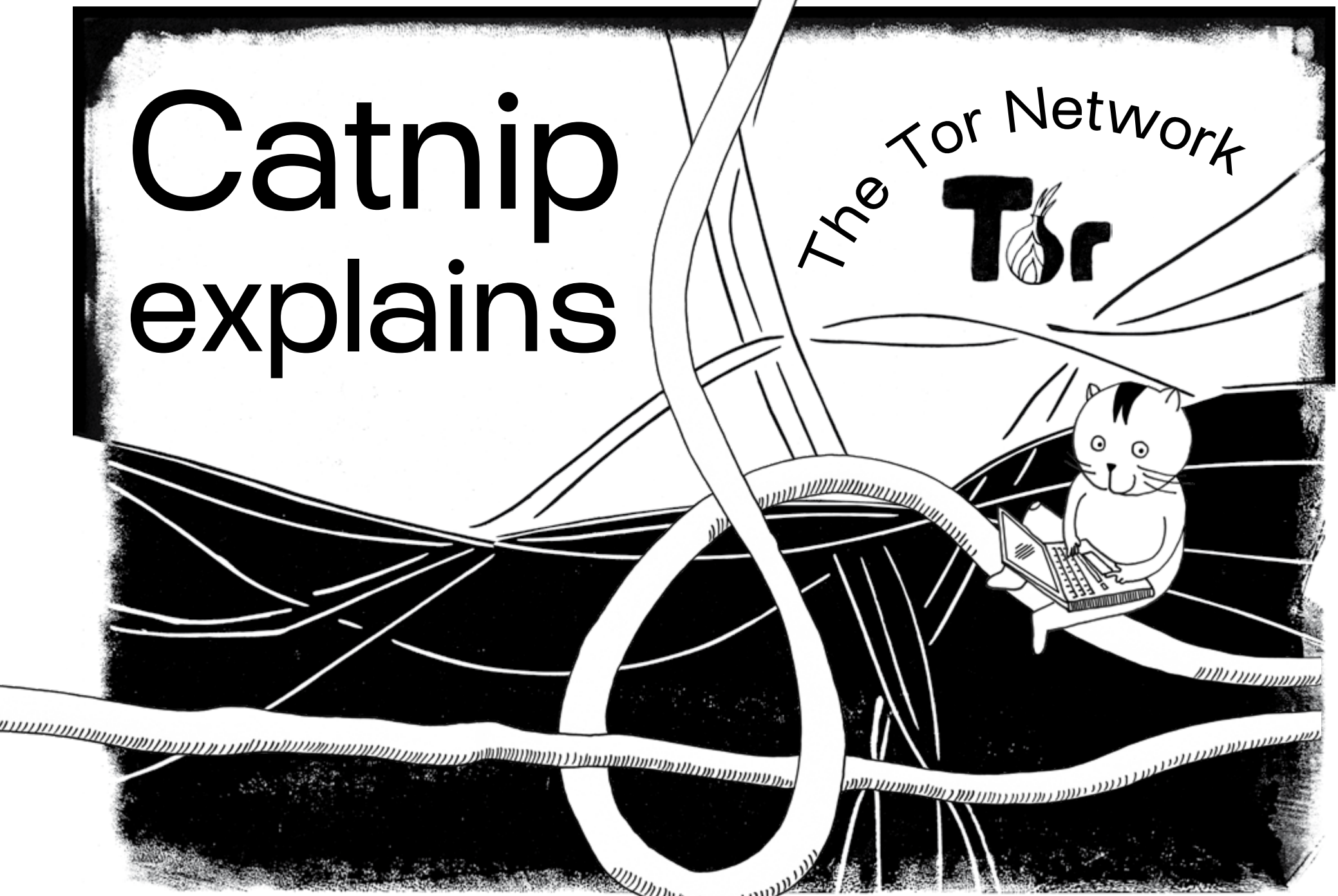
Catnip explains

The Tor Network

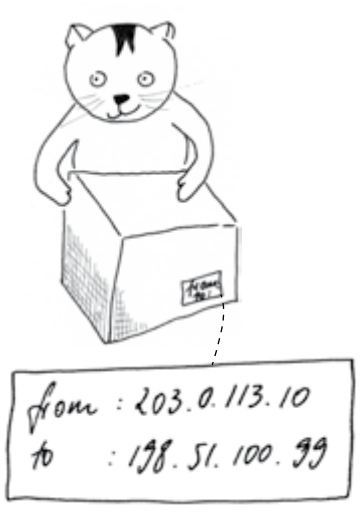


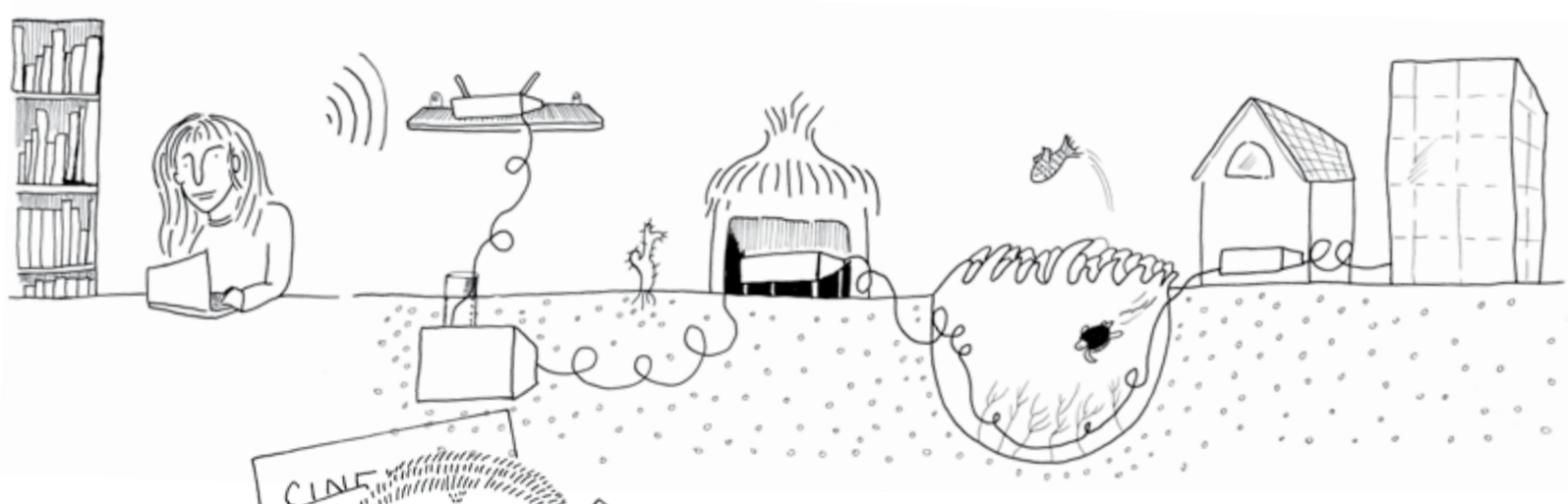
Data travels in packets. And each packet has a tag, or packet header, indicating its source and destination address.

There are no direct connections on the Internet, packets travel through intermediary networks and routers, all of which read the packet tag in order to route the packets to their destination.

All of them know where they received the packet from and the source and destination information contained in the packet header.

All of these intermediaries can copy, store or even alter packets.





This is a problem when states, institutions, employers, parents, network or server operators want to stop us accessing certain content on the Internet through filtering, blocking or throttling Internet traffic.

Attempts to censor content can happen at any stage: at the source, at intermediate routers, or at the destination. These interventions are made by parties operating the routers, the servers or the network equipment that data packets travel through.

To circumvent blocking and filtering, the Tor network (The Onion Routing) can be used to anonymize Internet traffic by hiding source and destination addresses.

Tor is a global network of computers running the Tor software. As of 2021, there are around 6800 “Tor nodes” - machines that make up the Tor network.

The

Any computer can run the Tor software and become a node.

Think of Tor nodes like data checkpoints which take in, treat, and ship out packets traveling over the Tor network.

A Tor circuit

Using Tor, data packets are sent over the Internet's infrastructure like any other packet.

But packets traveling through the Tor network are routed randomly through three nodes (also called relays or hops) before reaching their final destination.

This route, called Tor circuit, is changed every ten minutes to make it harder to observe for potential eavesdroppers.

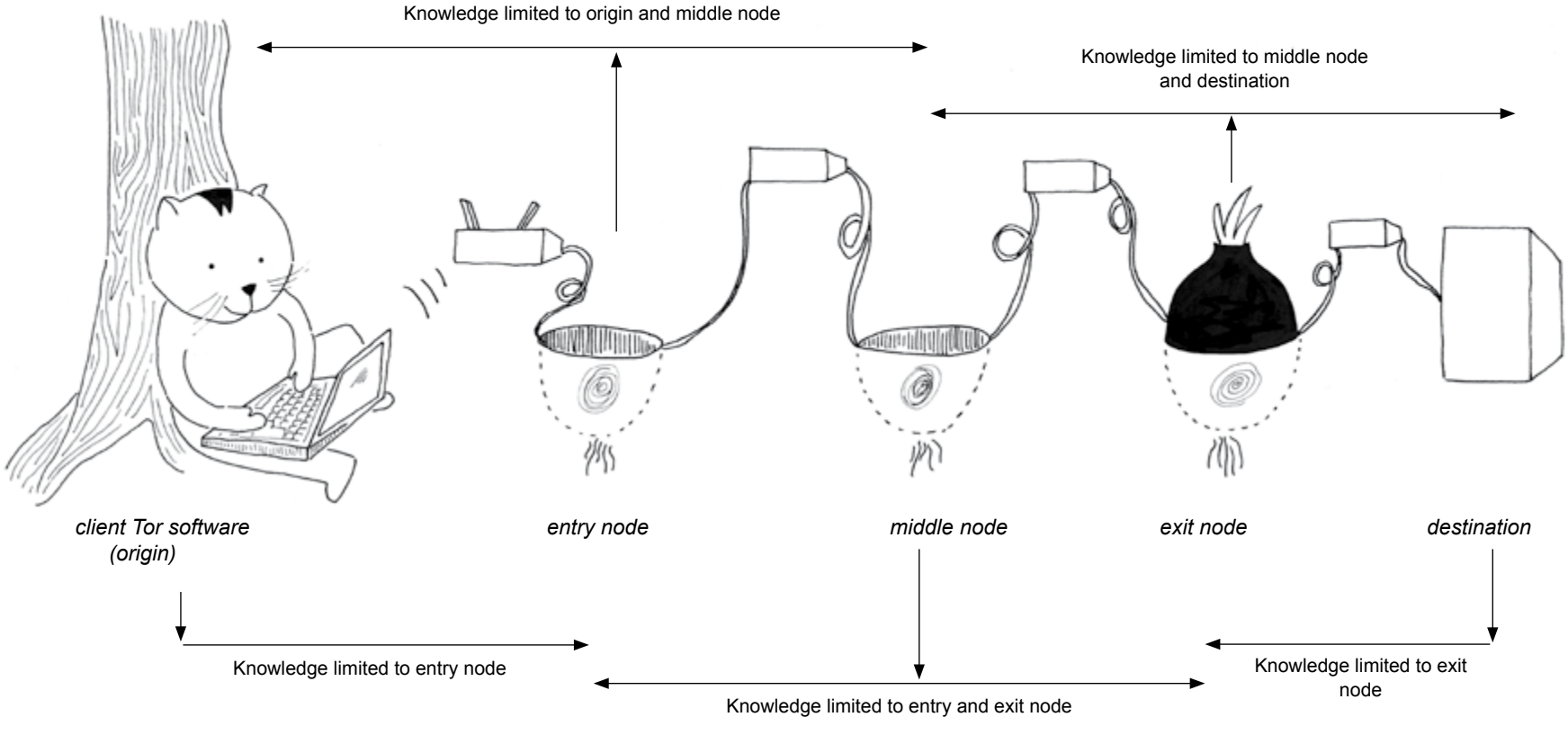
To route the packets down a random path, they are packed like an onion: each is wrapped into three encrypted layers containing a dedicated packet tag.

But only a partial route is encoded on these tags, so that none of the relays know the entire path a packet takes.

At each relay a single layer is peeled off and the packet is then sent to the next relay written on the tag of the underlying layer. This unwrapping and preparing to relay the packets is managed by the Tor software of the node. All that is visible from the outside is that obscure cargo is sent from place to place.

Because of this, the Tor network is often pejoratively called the "dark net".

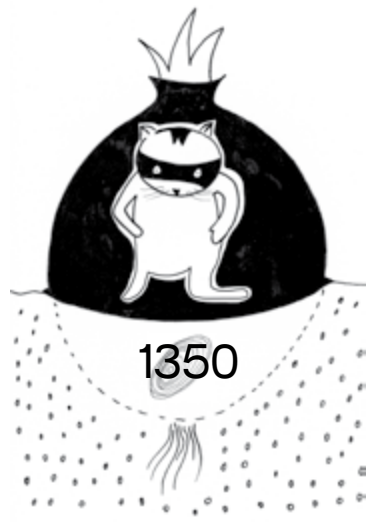
When we say that Tor packet layers are encrypted, it means that only the relay with the correct private encryption key is able to unwrap the corresponding layer.



Tor network

Using Tor, the sender of packets remains anonymous because each intermediary only knows the location of the immediately preceding and following nodes.

Packets which are destined for a website or email server on the clearnet will leave the Tor network after the last hop. These packets appear to originate from the exit node, which is a special Tor node.



In 2021, there are about 1350 exit nodes on the Tor network.

It is important to note that if the initial request that was made didn't use transport or end-to-end encryption, the package contents are visible to the exit node.

Limitations

Because Tor is a privacy and anonymity enhancing tool, it's used for many different things, but it is important to have clear expectations about what Tor is unable to do or in which cases it does not protect your identity or your data.

First, remember that everybody running the Tor software on their server can become a node. This means that there might be eavesdroppers or other malicious entities running Tor nodes. If the same entity could control more than one relay for each Tor circuit, that entity could know everything about you and your traffic.

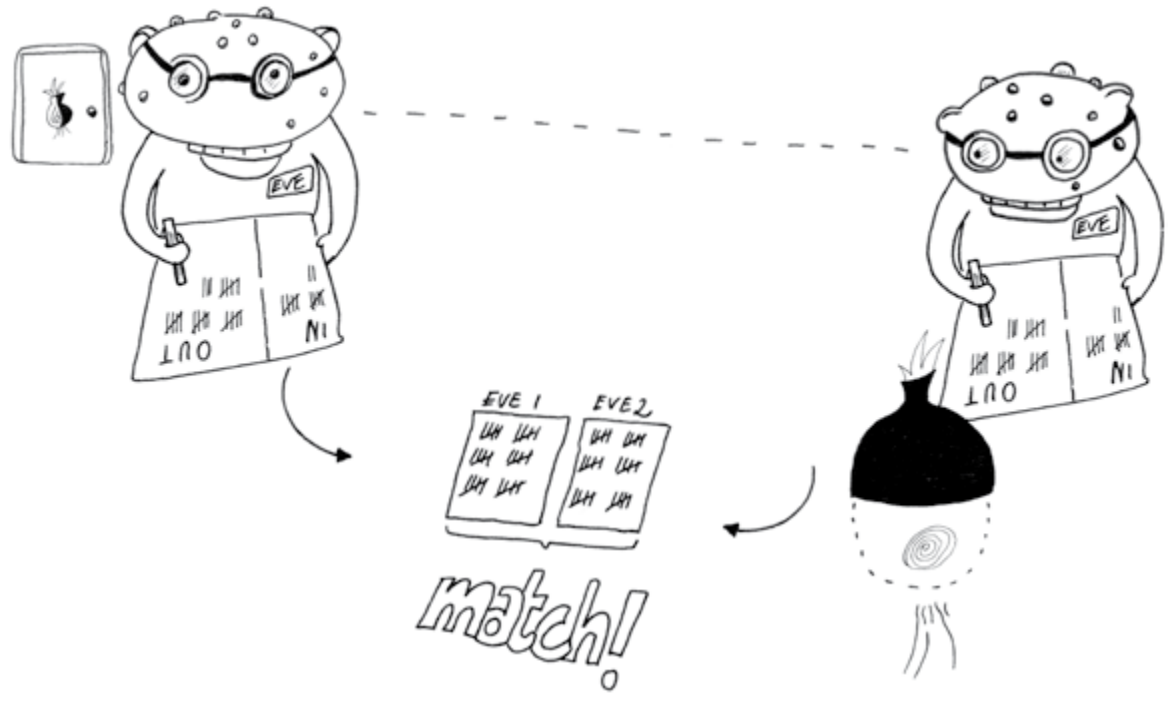
Just having access to an exit node can sometimes be enough for them to know who you are, if you login to a website without using encryption, for example. So, secondly, it's critical to always use encryption on top of using Tor.

Adversaries can even infer on the contents of encrypted packets by closely monitoring the size, timing, and quantity of ingoing and outgoing packets at both origin and destination, performing a targeted traffic analysis. Tor cannot protect from such attackers.

Blocking Tor

In some countries and on some networks known entrances to the Tor network are censored. In this case bridges have to be used as entry points. Bridges are Tor relays that aren't listed in the public main Tor directory. Using bridges, it becomes harder for ISPs and censors to block access to the Tor network. However, when bridge addresses become known to censors, they can equally get blocked.

The same is true for exit nodes. As their addresses are publicly known, it is easy to block packets that originate from an exit node from accessing a website or a server on the clearnet.

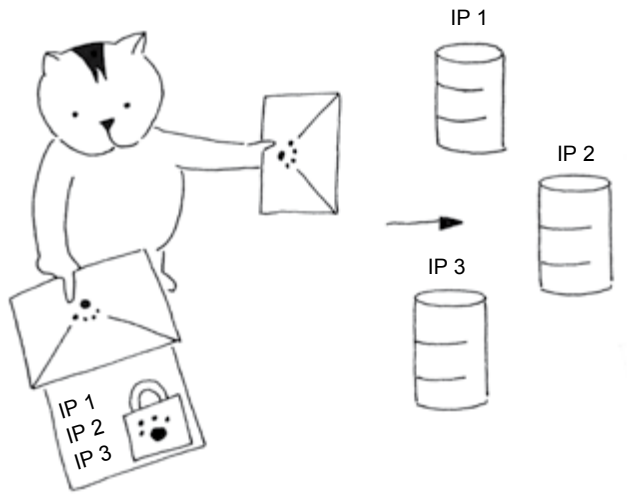


Onion services

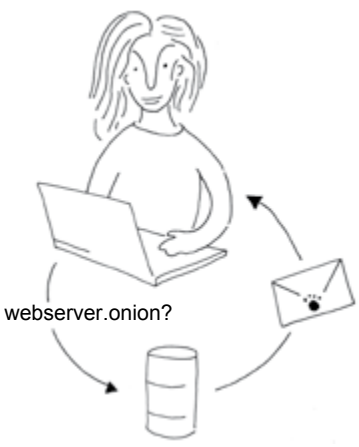
The Tor network provides services that keep traffic end-to-end encrypted. Like Tor end users, an **onion service** is hidden within the network, obscuring the server's location. These services use the top-level domain **.onion**. Let's walk through an example of how a **.onion** service works.

Say Catnip proposes an onion service for a web server, *webserver.onion*.

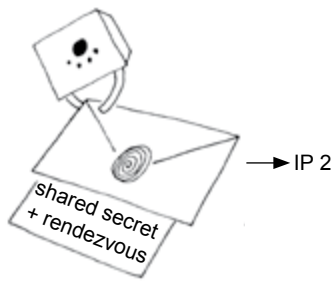
Catnip's Tor software randomly picks three nodes, builds a Tor circuit out of them, and makes them **introduction points**.



Catnip's Tor software then prepares a **descriptor message** that contains the names of the introduction points and the public key belonging to *webserver .onion*. It signs the descriptor message and sends it to a (distributed) database within the Tor network.

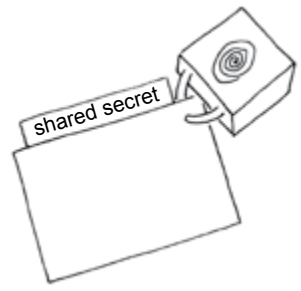


Tor user Alice has read about webserver.onion and wants to visit it. Her Tor software sets up a **rendezvous point** within the network and then asks the database for a signed message from *webserver .onion*.



When she receives this message, her software creates an **introduction message** which contains a secret message encrypted with *webserver.onion's* public key. She sends the introduction message to an introduction point, which then relays the message to Catnip.

The message arrives to Catnip through a Tor circuit of three nodes, none of which knows the entire path, keeping both Catnip and Alice anonymous. Catnip's onion service then decrypts Alice's introduction message and finds the address of the rendezvous point and the secret message.



Catnip can now connect to the rendezvous point, including Alice's secret message in their first rendezvous message. The rendezvous point informs Alice's Tor client when this happens.

Now Alice and Catnip can communicate by using their own three-node Tor circuit to the rendezvous point, which relays their end-to-end encrypted messages. The entry guard is the first node in the circuit.

Using the Tor network

Tor can be configured at the network level, manually, by modifying the way a device or router sends and receives all of its traffic. Or, there are software applications that make using the Tor network easy:

Tor Browser to avoid tracking as you browse the web.

OnionShare to share files anonymously.

Tails, an out-of-the box operating system that makes all network traffic automatically go through the Tor network.

Because Tor currently uses TCP (Transmission Control Protocol) as its packet transport protocol, applications based on UDP (User Datagram Protocol) such as video calls or torrenting don't work over Tor.

Imprint

ARTICLE 19, 2018. Second revised edition 2021. License CC-BY-SA 4.0.
Ulrike Uhlig (research and illustrations). Daniel Kahn Gillmor, Gustavo Gus, intrigeri (technical expertise).
Layout: Ulrike Uhlig and Aline Girard.

ARTICLE 19

Check out our book "How the Internet REALLY Works"
December 2020. No Starch Press, San Francisco, USA.
https://catnip.article19.org ISBN 9781718500297.