Background

- Designed & implemented for US Navy
- Created to protect gov’t communication
- 80% of Tor’s ~$2M funding paid for by US gov’t
Design Goals

- Anonymous
- Low-latency
- Usable
- Flexible
- Simple
Implementation

• TCP
• Create circuit via 3 tor nodes
  – Circuit change every ~10min (configurable)
  – No single point of failure
• Data chunked in 512 byte “cells”
  – Inefficient for bandwidth w/ small data transfers
    • E.g. IRC
  – Used to make it harder to guess what type of content is being transferred by packet size
How Tor Works

Step 1: Alice’s Tor client obtains a list of Tor nodes from a directory server.
How Tor Works

Step 2: Alice’s Tor client picks a random path to destination server. Green links are encrypted, red links are in the clear.
How Tor Works

Step 3: If at a later time, the user visits another site, Alice’s tor client selects a second random path. Again, green links are encrypted, red links are in the clear.
Inside a Circuit
Limitations

• Doesn’t protect:
  – Computer configuration → use Privoxy
  – End-to-end timing attacks
    • Analysis of traffic + timestamp of your client and the destination can pinpoint traffic to you
  – Plugins like Flash can query your local IP

• Also:
  – First server could see who you are
  – 3rd server could see your traffic
Limitations

• Tor IPs are public
• Filtering based on the fingerprint of the Tor TLS handshake
  – Several countries have blocked Tor
    • China, Iran, Japan, Russia
    • Intercept connection between client and 1st relay
    • Solution: bridge relays!

  – Application developers can block Tor
    • Even in US: Craigslist
    • Application server detects 3rd relay’s IP
How to ‘Tor Websites’ work?

• Tor Hidden Services
• Need to connect a client and server s.t.
  – Client info protected from server, AND
  – Server info protected from client
• How?
  – .onion address
  – 2 circuits
• E.g. Silk Road ➔ http://6zyze2mkwyla7jwe.onion
Hidden Service Protocol

Step 1: Bob picks some introduction points and builds circuits to them.
Hidden Service Protocol

**Step 2:** Bob advertises his hidden service — XYZ.onion — to the database
Hidden Service Protocol

Step 3: Alice hears that XYZ.onion exists, and she requests more info from the database. She also sets up a rendezvous point, though she could have done this before.
Hidden Service Protocol

**Step 4:** Alice writes a message to Bob (encrypted with his PK) listing the RP and a one-time secret, and asks a introduction point to deliver it to Bob.
Hidden Service Protocol

**Step 5:** Bob connects to Alice’s rendezvous point and provides her the one-time secret.
Hidden Service Protocol

Step 6: Bob and Alice proceed to use their Tor circuits like normal.
How widespread is it?

• ~4,000 relays
• ~2,000 bridges (non-public relays)
• ~1 GB/s
Usage Graph – last 3 months

Directly connecting users from all countries

The Tor Project - https://metrics.torproject.org/
Usage Graph – last 3 years

Directly connecting users from all countries

The Tor Project - https://metrics.torproject.org/
It’s not sustainable!

• Tons of clients, very few relays
• How do you incentivize people to be Tor relays?
  – More relays leads to:
    → Faster bandwidth, more throughput
    → Less chance of endpoint hijacking (if the new relays aren’t “traitors”)
Possible Incentives

• Relays get “priority”
• Pay for priority service with bitcoins
  – More incentives to ‘cheat’
  – Behavioral economics: people less likely to ‘volunteer’

• Won’t be implemented anytime soon
Open Qs

• Should the circuit/path length be extended to improve security?
• Should Tor un-publicize relay IPs so they don’t get blocked at the application layer?
• Should Tor make every node a relay?