Anonymity

- So far: just talked about encryption
- We leak:
  - Our IP
  - Timing of our messages
  - Sizes of our message
- Idea: use a relay / VPN
  - A big fan in and fan out
  - Idea is to make it impossible to figure out which input corresponds to which output
  - Need to create a TLS connection between server and client that is encrypted to ensure relay
- Problems:
  - A compromised relay gets access to everything
  - Performance
  - Timing analysis still works

Onion Routing

- We use a path of relays
- Each relay only knows what sent it to them and where it is going to send it
- Cannot just make a single encrypted connection between each client / relay and relay / relay because then the relay is seeing unencrypted data
- Let's say we have a path of three relays:
  - C - R1 - R2 - R3 - S
  - C - R1 sets up a TLS connection
  - Using this TLS connection, we set up another TLS connection between C and R2 that goes through R1
    - Nested within the C - R1 connection
    - We continue nesting connections within each other
- TOR typically uses a circuit of three relays

Directory

- We want all clients to use the same directory / database of relays
  - An attack could be to give different clients different sets of relays and use that as an attack
- To mitigate this, TOR uses a bunch of directory servers that each sign the directory

TOR Browser

- Tor browser does more than just use the network
- Trying to limit the amount of unique information leaked
  - Chrome exopses a lot of information like screen size / your graphics card which can build a unique fingerprint of your machine
  - TOR browser tries to do stuff to make your browser look like everyone else's

Bridges

- Bridge = hidden relays
- Used to combat firewalls that are used for censorship
  - You ask a BridgeDB for a bridge relay
  - Rate limited
  - Competent firewall operators can try to block this
    - But if you host the BridgeDB on some CDN like Cloudflare then the operator might have to block all of Cloudflare

    - Current version of TCP still leaks the website that you are connecting to so good firewall operators (i.e. China) can still block it

Anonymous Servers

- So far we have talked about clients being anonymous
- Instead servers want to be anonymous
    - I.e. whistleblowing or illegal websites
    - Also DOS resistant
- Servers choose some relay point to be a "introduction point" / "mailbox"
    - Builds a circuit to that intro point
- Client chooses some relay as the rendezvous point
    - They then build another circuit to the introduction point and sent the name of the rendezvouz point to the server through it
        - It can also send information like a username to try to convince the server to accept the connection
    - The server, if it chooses to accept the connection, will connect to the Rendezvout point
- Clever idea:
    - The name of the onion link will just be the public key
    - This is important because:
        - We don't need a common directory to store this because the server can just sign messages with its private key and the client will see that this is legit
        - It also allows you to encrypt messages to the server with the public key because no TLS CA is going to issue a certificate to the server

Guard Nodes

- If we have a path of three nodes but the ends are both compromised (R1 and R3)
    - Can do sophisticated timing attacks by comparing the times
    - Can bypass an honest node
    - As more and more circuits are built by you, the probability increases to 1 that there will be a compromised circuit
- Guard note idea:
    - Choose and keep first node
    - Only change the second and third node each time
        - You have a chance of getting screwed the first time but your probability won't approach 1 over your lifetime