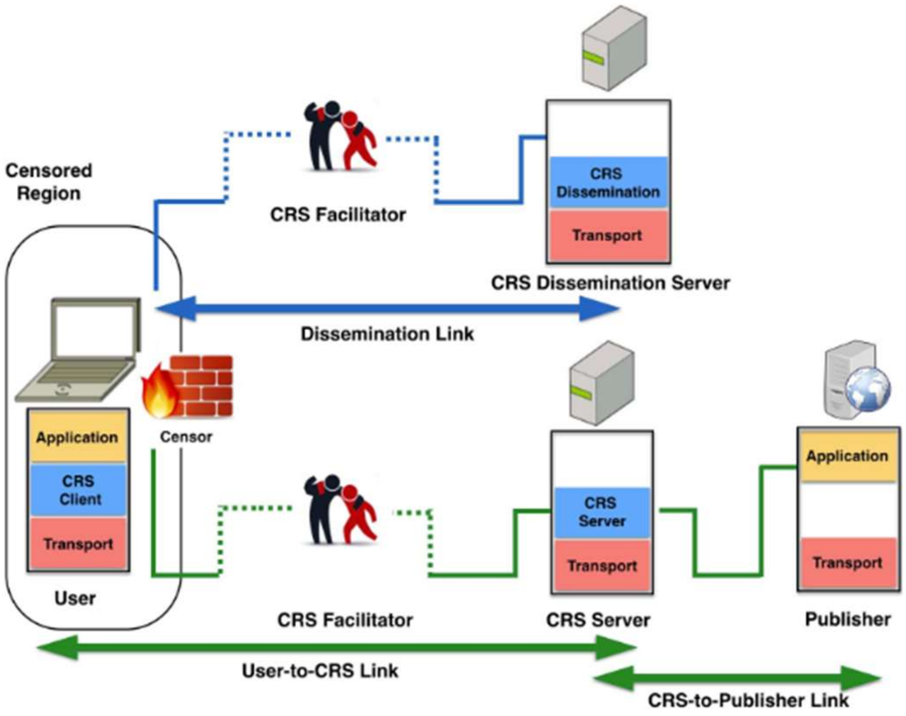
Censorship Resistance Systems

Privacy and anonymity

- Privacy: control over your information
- Anonymity: no information about you is leaked
- Censorship resistance: On top of anonymity, being able to access censored information on the Internet
  - Ongoing cat-and-mouse game with political implications
  - Moral or not?

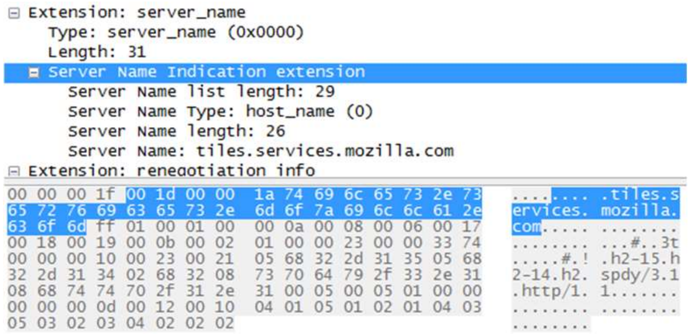
Censorship Resistance Systems



Khattak et al. (2016) “SoK: Making Sense of Censorship Resistance Systems”

TLS

- Starting point: achieves **CIA** principles, but not anonymity
- Higher layer protocols (IP, TCP) reveal source and destination
- TLS Client Hello contains Server Name Indicator
  - Partly solved with encrypted SNI, but not common
- As a result, TLS itself is easily blocked
- VPNs are easily blocked as well

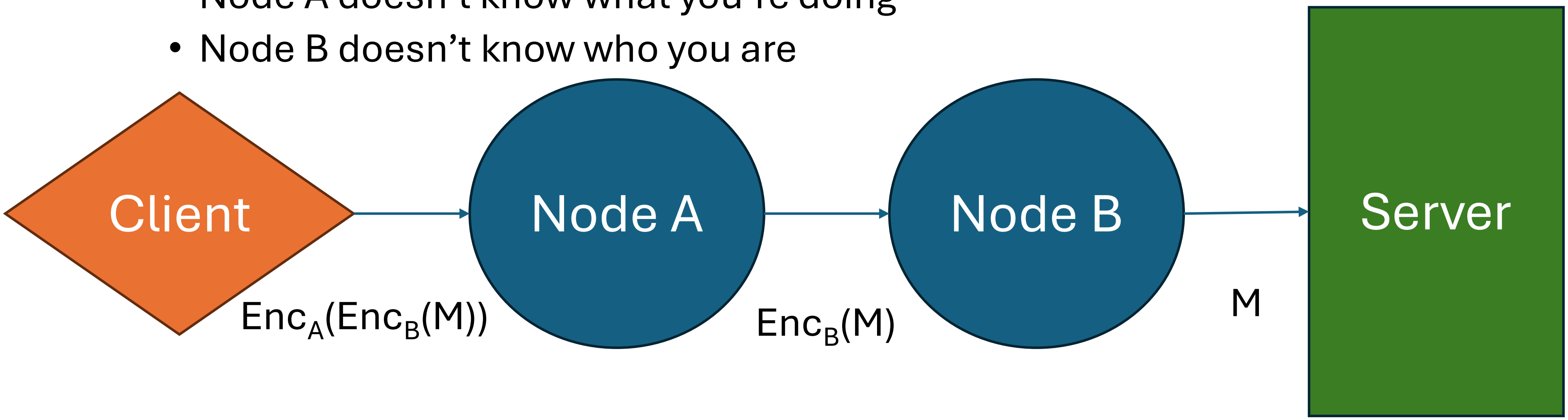


```
⊟ Extension: server_name
    Type: server_name (0x0000)
    Length: 31
  ⊟ Server Name Indication extension
      Server Name list length: 29
      Server Name Type: host_name (0)
      Server Name length: 26
      Server Name: tiles.services.mozilla.com
⊟ Extension: renegotiation info

00 00 00 1f 00 1d 00 00  1a 74 69 6c 65 73 2e 73   ........ .tiles.s
65 72 76 69 63 65 73 2e  6d 6f 7a 69 6c 6c 61 2e   ervices. mozilla.
63 6f 6d ff 01 00 01 00  00 0a 00 08 00 06 00 17   com..... ........
00 18 00 19 00 0b 00 02  01 00 00 23 00 00 33 74   ........ ...#..3t
00 00 00 10 00 23 00 21  05 68 32 2d 31 35 05 68   .....#.! .h2-15.h
32 2d 31 34 02 68 32 08  73 70 64 79 2f 33 2e 31   2-14.h2. spdy/3.1
08 68 74 74 70 2f 31 2e  31 00 05 00 05 01 00 00   .http/1. 1.......
00 00 00 0d 00 12 00 10  04 01 05 01 02 01 04 03   ........ ........
05 03 02 03 04 02 02 02                            ........
```

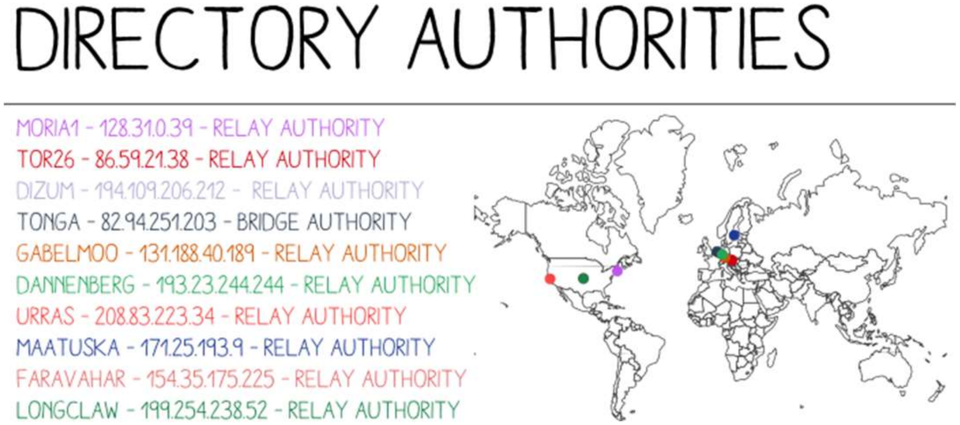
Onion routing (Tor)

- One of the most popular anonymity technologies
- Uses layered (onion) encryption across multiple volunteer nodes
- No assumption of trust in third party
  - Node A doesn't know what you're doing
  - Node B doesn't know who you are



Onion routing (Tor)

- Information about nodes is stored in a public directory
  - Multiple directories co-sign the information to prevent integrity attacks
- Data is wrapped in fixed-length Tor cells, then wrapped in TLS records between Tor relays and client
- Low-latency designed for (relatively) easy browsing



DIRECTORY AUTHORITIES

MORIA1 - 128.31.0.39 - RELAY AUTHORITY
TOR26 - 86.59.21.38 - RELAY AUTHORITY
DIZUM - 194.109.206.212 - RELAY AUTHORITY
TONGA - 82.94.251.203 - BRIDGE AUTHORITY
GABELMOO - 131.188.40.189 - RELAY AUTHORITY
DANNENBERG - 193.23.244.244 - RELAY AUTHORITY
URRAS - 208.83.223.34 - RELAY AUTHORITY
MAATUSKA - 171.25.193.9 - RELAY AUTHORITY
FARAVAHAR - 154.35.175.225 - RELAY AUTHORITY
LONGCLAW - 199.254.238.52 - RELAY AUTHORITY

Jordan Wright

Tor by itself is not censorship-resistant

- Directory information is public, so a list of all nodes is readily available – and blocked
- Tor has non-public nodes known as **bridges**, but access to them is limited and unreliable
- Tor cells are fingerprintable
- Tor proxies are also not probe-resistant

Probing resistance

- Active probing attack: Censor actively connects to potential proxies and tries to get them to respond under their protocol
  - Performed by China’s censor
- Probe resistance: Use secret key (delivered out of band), proxy only responds
  - Tor’s obfs4 protocol (for bridges)
  - Shadowsocks
- Frolov et al. 2020: Many of these can still be cleverly detected
  - Example attack: If the protocol header is expected to be N bytes, the server may choose to wait if it receives N-1 bytes; N is fingerprintable

Probing resistance

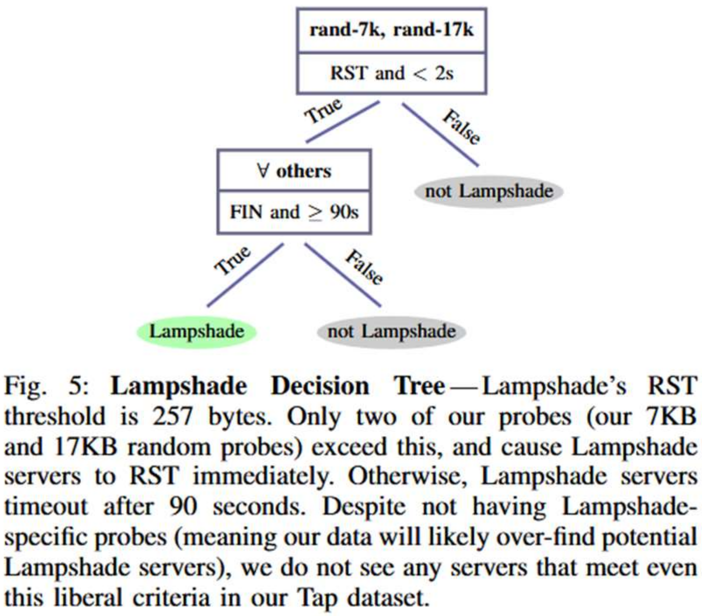


Fig. 5: **Lampshade Decision Tree**—Lampshade's RST threshold is 257 bytes. Only two of our probes (our 7KB and 17KB random probes) exceed this, and cause Lampshade servers to RST immediately. Otherwise, Lampshade servers timeout after 90 seconds. Despite not having Lampshade-specific probes (meaning our data will likely over-find potential Lampshade servers), we do not see any servers that meet even this liberal criteria in our Tap dataset.

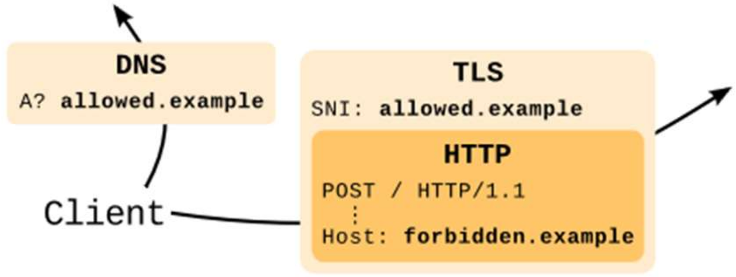
Frolov et al. (2020) "Detecting Probe-resistant Proxies"

Anti-fingerprinting

- Any protocol may have noticeable traffic analysis features from regular traffic
    - e.g. direction, size, timing, order of packets
- CRS needs to mimic another protocol
- Mimicry is extraordinarily difficult:
    - Houmansadr et al. (2013) “The parrot is dead” – all investigated mimicry techniques had implementation flaws
    - Frolov and Wustrow (2019) – CRS systems were fingerprintable, e.g. using old browser versions, using uncommon TLS options

Domain fronting

- Sending a packet that looks like it's going to (uncensored) A, but it will actually go to (censored) B
- This works only if the web server ignores SNI and uses the Host header
- Examples: Tor's meek pluggable transport, Signal
  - Used domain fronting for Google and Amazon, but they blocked the front
  - Tor's meek now uses Azure



Fifield et al. (2015) "Blocking-resistant communication through domain fronting"

Refraction networking

- Cooperating ISP can disguise traffic destination
- Pretend to send packets to uncensored location; once ISP receives it, they will redeliver it to the true destination
  - ISP is outside of censorship range
- Telex:
  - Add a tag that contains the true destination (encrypted)
  - Looks like a random nonce in TLS ClientHello
- Cirripede:
  - True destination is encoded in TCP initial sequence numbers

Refraction networking

- TapDance:
  - Observation: previous schemes require inline blocking
    - Inline blocking blocks further packets between the client and uncensored server
    - Otherwise return packets will confuse the client
  - TapDance instead allows these packets by using ciphertext malleability in CBC to send packets that are valid for the uncensored server but contain a secret request for the censored server
- Deployment study in 2020; up to about 30,000 users
  - Deployed in Psiphon