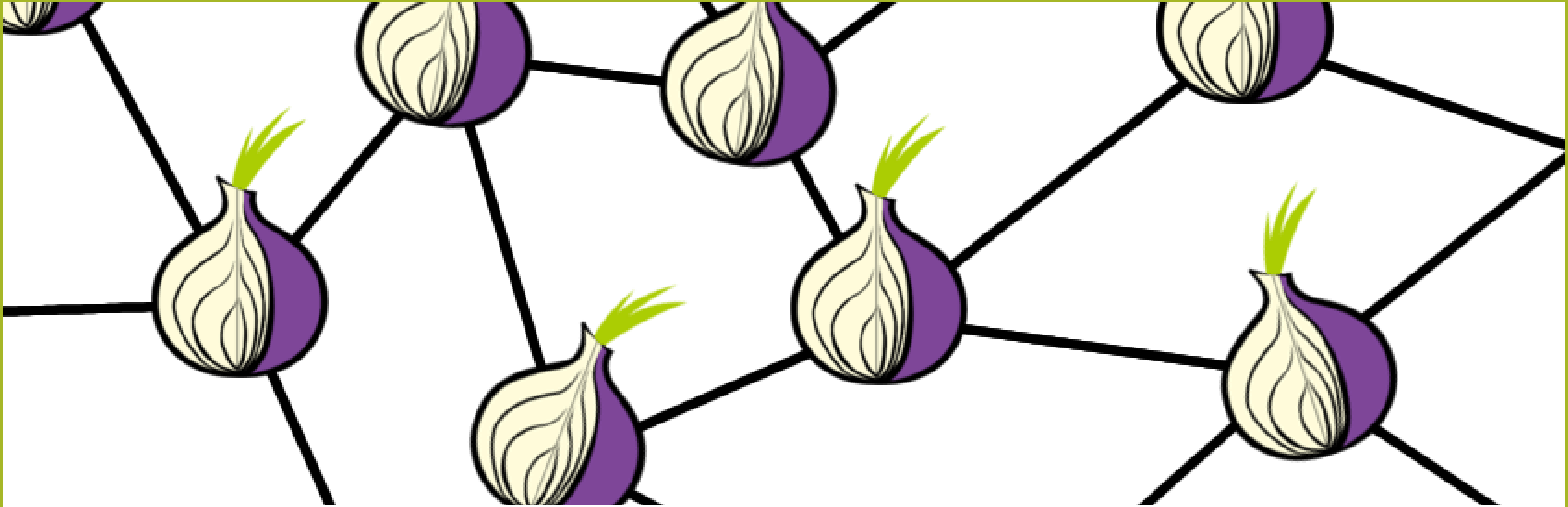


TOR

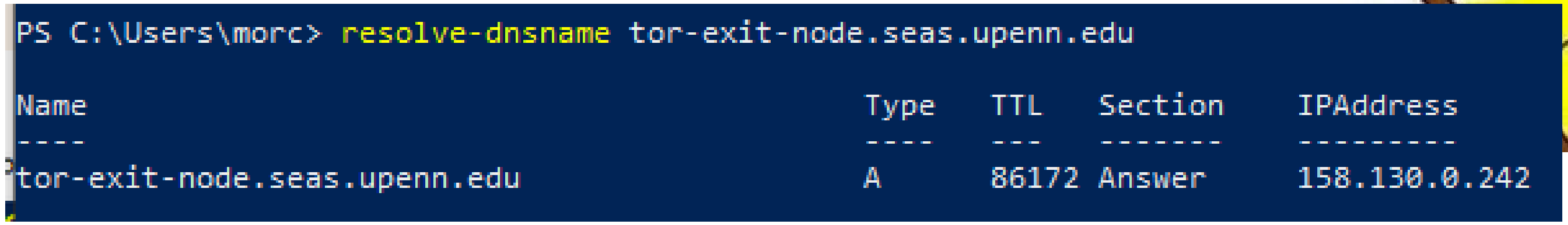
The 2nd Generation Onion Router

1. Tor: **T**he **O**nion **R**outing Project

- 1990s: Onion Routing developed, primary by Navy / DARPA
- 2004: Tor paper comes out
- 2004: EFF (Electronic Frontier Foundation) starts funding Tor
- 2008: Tor Browser developed
- 2015: Tor Messenger released (later discontinued)

- Average users: 2.5 million
  - Spiked in late 2013 … maybe thanks to NSA / Edward Snowden?

First Impressions?


```
PS C:\Users\morc> resolve-dnsname tor-exit-node.seas.upenn.edu

Name                                       Type   TTL   Section    IPAddress
----                                       ----   ---   -------    ---------
tor-exit-node.seas.upenn.edu               A      86172 Answer     158.130.0.242
```

2. Onion Routing

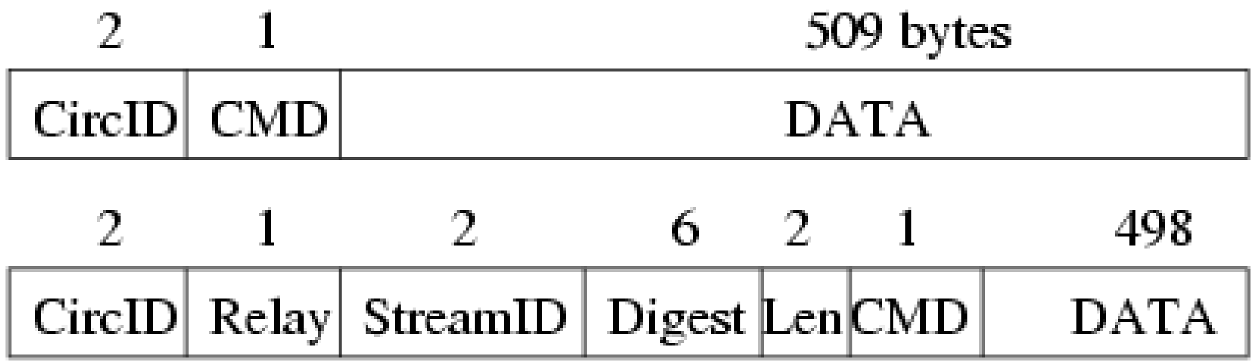
- How is Tor similar/different from what we've previously covered?
  - Mix Network:
    - Similar encryption layers, but using asymmetric vs. symmetric keys
    - Mix-specific keys vs. session-based keys
    - Reuse of circuits is similar to cascades
  - Crowds:
    - Not peer-to-peer
    - Supports more than HTTP
    - Entry/exit nodes only can see content of reqs
- Tor is a low-latency – what are the pros/cons, vs. high-latency designs?
  - High-latency: resists global adversaries but adds… latency! Which affects usability…
  - Low-latency: aims for a "reasonable tradeoff" between anonymity & usability

3. Goals & Assumptions

- Tor's Goals (after anonymity!) – have they accomplished them?
  - Deployability, Usability, Flexibility, Simple Design
- What is Tor choosing *not* to address or implement? Why not?
  - End-to-end attacks:
  - Protocol normalization
  - Full decentralization (it's not peer-to-peer). Does this relate to Crowds, Sybil Attacks?
  - Obfuscating participating users
- What assumptions does the paper make about an attacker? Why is this important?
  - Threat Model: attacker sees some portion of network traffic
  - Defense against traffic analysis, but not traffic confirmation attacks

4. Tor Design: An Overlay Network

- Overlay Network
  - TLS connections between ORs (Onion Routers)
  - User's OP (Onion Proxy) manages connections, accepts TCP streams
- Data exchanged via Cells
  - Fixed size
  - CircID connects cell to circuit
  - Commands tell OR/OP what to do:
    - Control cells: interpreted by rec'ing node
    - Relay cells
      - carry end-to-end data
      - contain streamID
- Any obvious flaws?

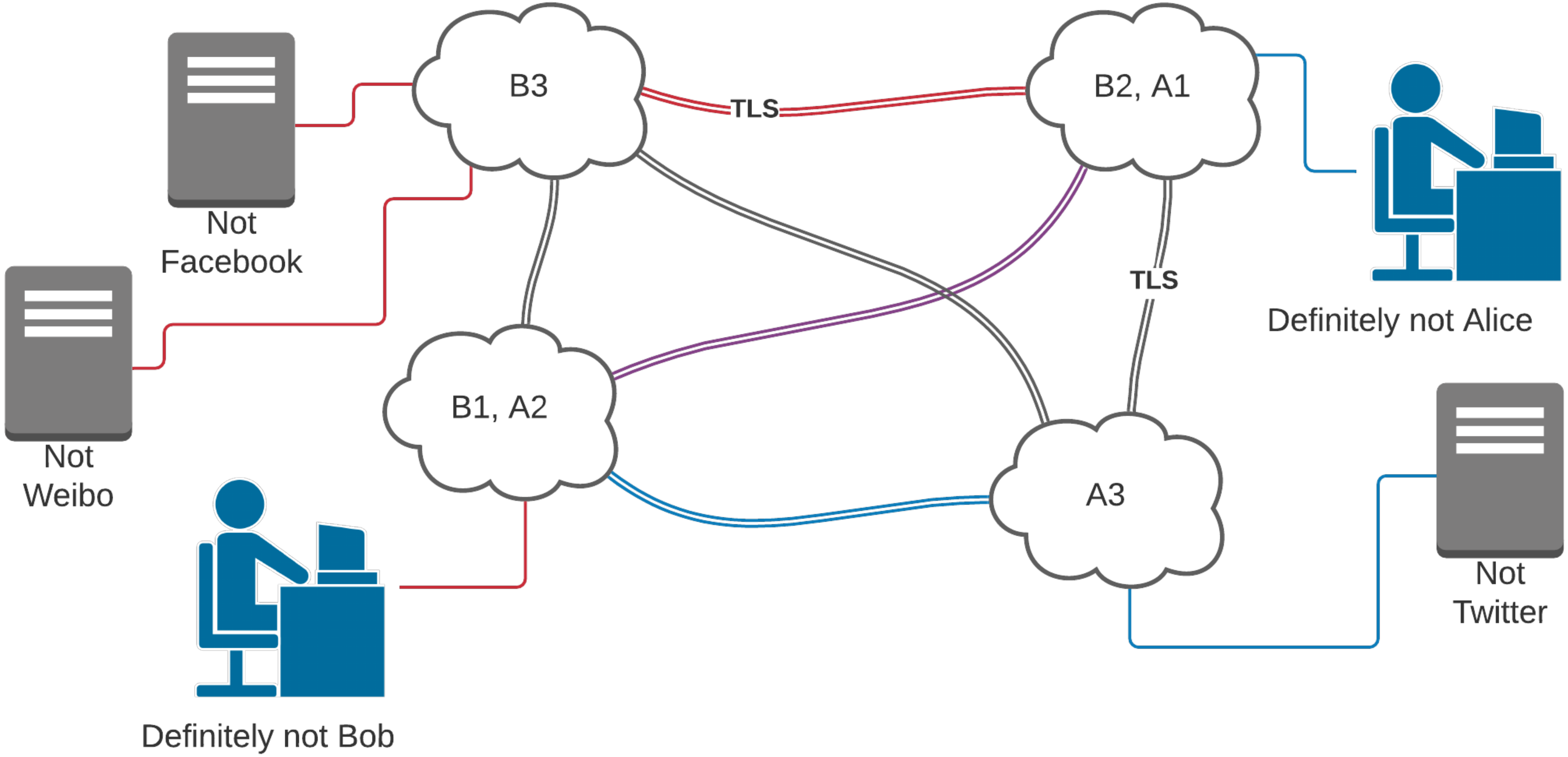


2	1	509 bytes
CircID	CMD	DATA

2	1	2	6	2	1	498
CircID	Relay	StreamID	Digest	Len	CMD	DATA

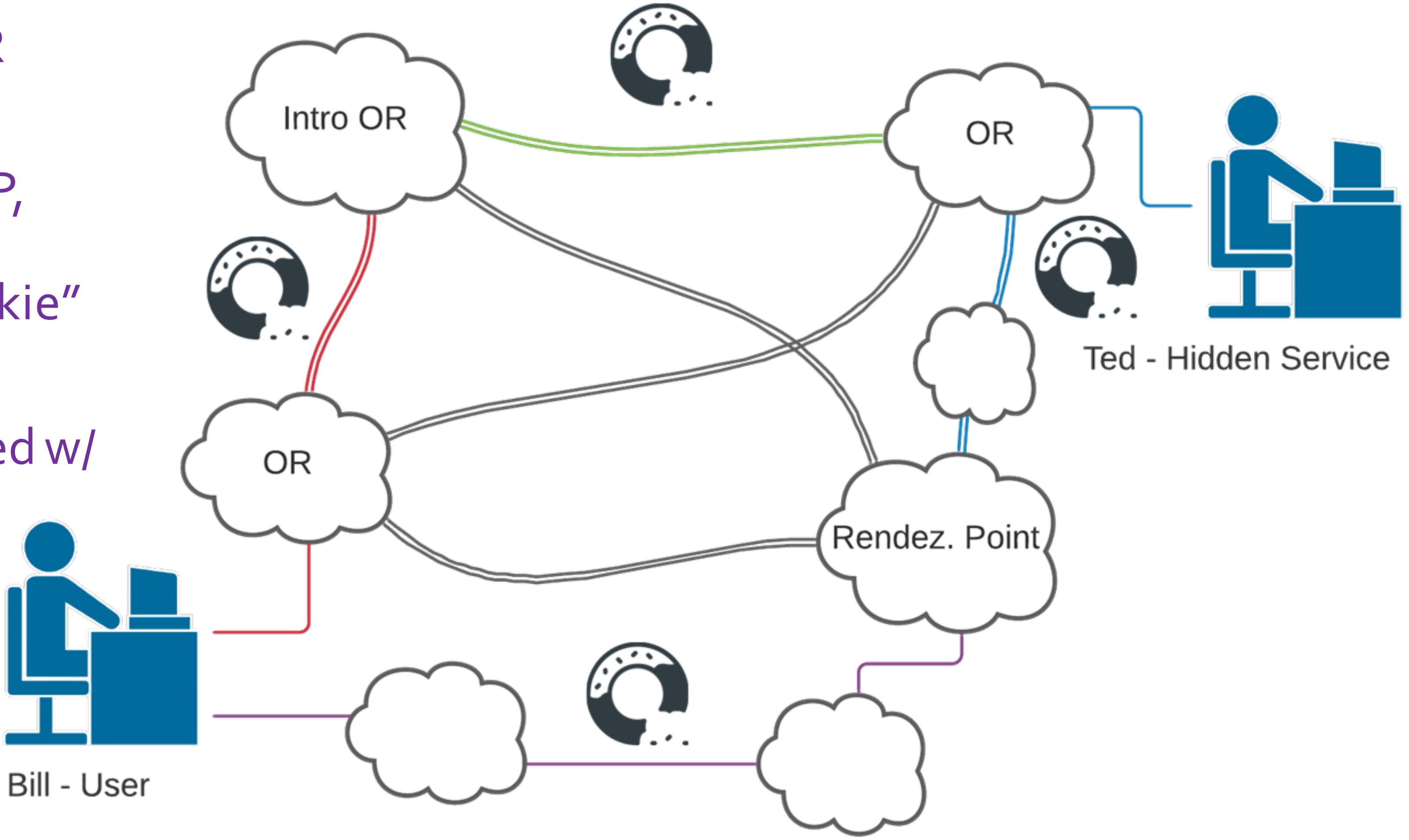
4. Tor Design: Circuits & Streams

- What makes Tor's circuit-building process & usage more secure?
  - "Telescoping", OR doesn't know who originator is, just prev node
  - Fresh key, forward secrecy
  - Circuits can be shared by streams
  - Circuits rebuilt periodically
- Does Tor's design prevent messages from being manipulated/modified?
  - TLS-enforced integrity checks on streams prevent *outsider* manipulation
  - End-to-end encryption of hashes across circuit
- How does Tor handle rate limiting & congestion?
  - Enforces average byte rates, allows for bursts – this mitigates user bandwidth issues
  - Throttling at circuit and stream levels – does this address D/DoS attacks?


B3
TLS
B2, A1
Not Facebook
Not Weibo
Definitely not Alice
TLS
B1, A2
A3
Not Twitter
Definitely not Bob

5. Rendezvous Points & Hidden Services

- Only advertised Intro OR knows hidden service
- Circuit built w/ central RP, connection id'd by user's chosen "rendezvous cookie" (or donut, per diagram)
- Hidden Service advertised w/ .onion TLD, public key
- How secure is this?



6. Other Design Decisions

- Has Tor mitigated Denial of Service attacks? Are the proposed methods feasible?
  - “Not yet implemented any defenses”
  - Back to computational puzzles! Do they really work?
  - Rate limiting to ensure the flow of traffic alongside new handshakes
- Are there any better solutions out there to prevent [D]DoS attacks?
  - Still an issue being worked out in 2021:
    “During the rendezvous protocol, an evil client can send a small message to the service while the service has to do lots of expensive work to react to it.”
  - Newest proposals: 1) use anon tokens, via CAPTCHAs or trusted user tokens, or 2) more puzzles, aka “Proof of Work”, but with updated, dynamic parameter tuning: https://blog.torproject.org/stop-the-onion-denial
- As an exit node, how do you balance being open vs. protecting against abuse? Would you choose stricter or looser exit policies? Why or why not?

6. Other Design Decisions

- How does Tor handle identifying & publishing node info to the network?
  - Clients fetch network info from Directory Servers, which check identity keys of ORs
- What are the implications if a Directory Server is compromised? Does Tor mitigate this?
  - Uses redundancy and synchronization, signs complying directories
  - Consensus health is published! https://consensus-health.torproject.org/
- How does this compare to the "logically centralized, trusted authority" discussed in the Sybil Attack paper?

7. Attacks and Defenses

- Are any of the listed Passive Attacks addressed by Tor?
  - Traffic confirmation attacks are outside the design goals ☹
  - Multiplexing within circuits may mitigate website fingerprinting attacks
- What are some notable Active Attacks?
  - Key compromise: mitigated by key rotation, session keys
  - Iterated compromise: mitigated by session keys, “jurisdiction arbitrage”
  - Onion Proxy compromise
  - Replay Attack: poss. eliminated due to session keys
  - Distribute hostile code: mitigated by signing releases, publishing “known good”
  - …and many more

7. Attacks and Defenses

- Are Directory Servers vulnerable to attack? What are the implications?
  - Subversion of DS's can influence final directory, eg include compromised ORs
  - Tor does not address "directory server dissent"
  - Tor "assumes DS operators will filter out hostile ORs"
- Does Tor prevent attacks against Hidden Services / Rendezvous Points?
  - Many attacks are essentially [D]DoS attacks
  - Restricting request volume, rotating intro points, & intro point testing can mitigate

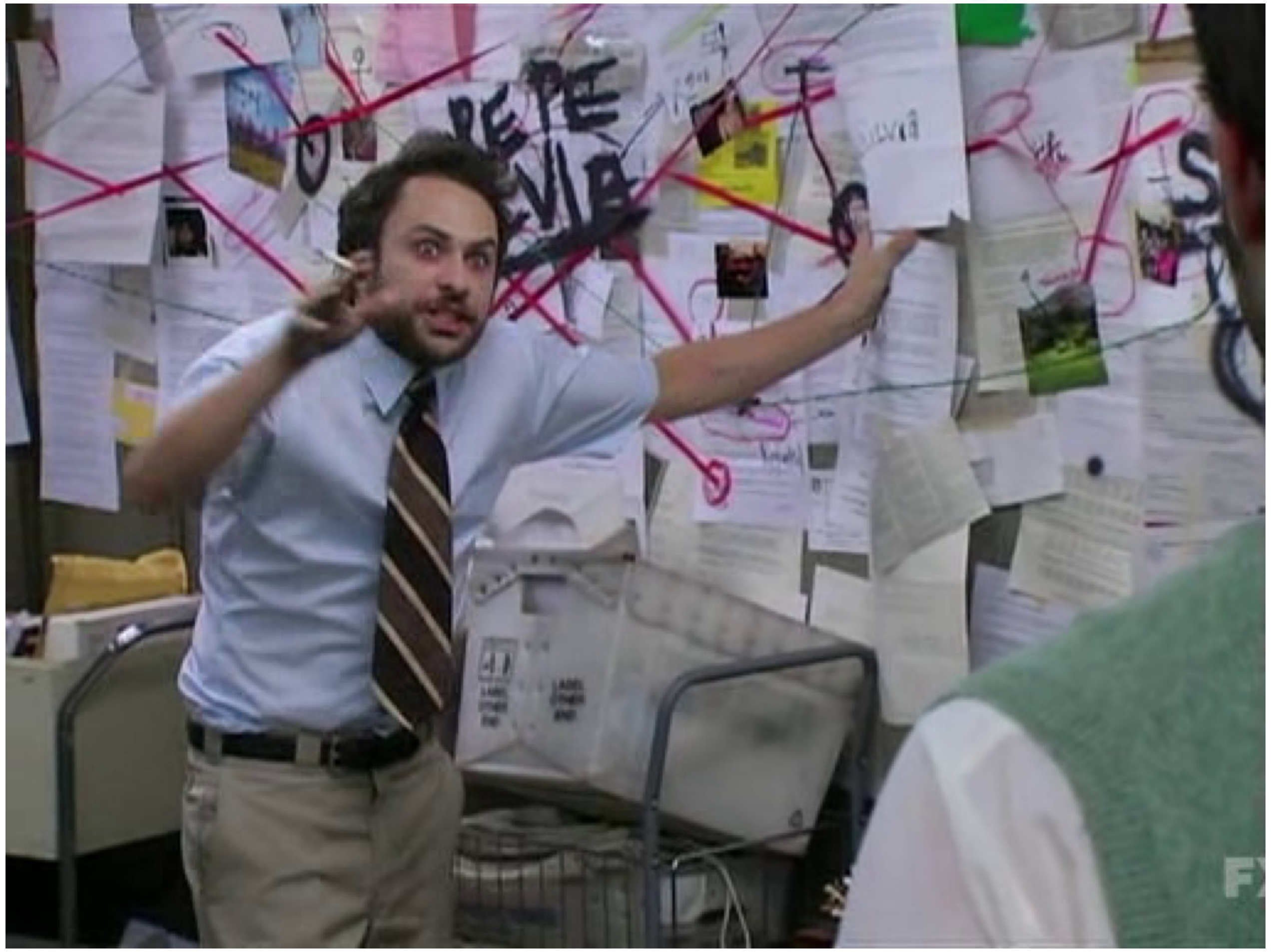
Bonus: 2014 Sybil/Traffic Attack!

- **Sybil + Traffic Confirmation Attack on Tor**: February-July 2014
- **Sybil** Attack: Malicious relays joined Tor at end of January
- **Traffic Confirmation**: Headers modified to exchange signals between malicious nodes

"the relay on one end injects a signal into the Tor protocol headers, and then the relay on the other end reads the signal. These attacking relays were stable enough to get the HSDir ("suitable for hidden service directory") and Guard ("suitable for being an entry guard") consensus flags. Then they injected the signal whenever they were used as a hidden service directory, and looked for an injected signal whenever they were used as an entry guard."

- End Goal: De-anonymize users who operated or used hidden services.
- Were they successful? It's unclear, but Tor "found no evidence that the attackers operated any exit relays". **What are the implications of that finding?**
- OF NOTE: "preventing traffic confirmation in general remains an open research problem"
- More info: https://blog.torproject.org/tor-security-advisory-relay-early-traffic-confirmation-attack

Thwart attackers!



8. Early Experiences & 9. Open Questions

- Tor "in the wild" had no abuse issues in 2004. Is that still true?
- How is performance/latency?
- Tor calls out circuit rotation, path length, and of course traffic confirmation attacks as open questions. Do you think much has changed in the 15+ years since?
- Are any of the other topologies mentioned (cascade, hydra) better?
- Do any of the other open questions remind you of other papers we've read?

10. Future Directions

- Tor: The Next Generation - to boldly go where no Tor has gone before?
- Did many of the ideas/next steps listed happen?
  - Tor now supports millions of users
  - Many nodes and Directory Servers are volunteer-run, but abuse does happen
  - Cover traffic not in use (may not be as useful), & as is expensive. Many issues still open.