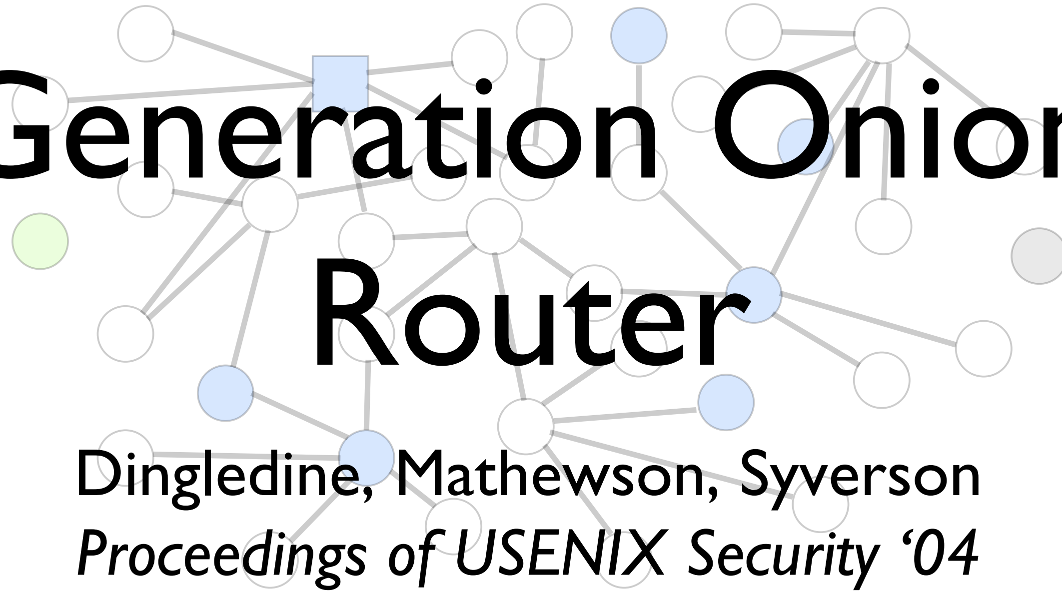
Tor Anonymity Network & Traffic Analysis

Presented by Peter Likarish

This is NOT the presenter's original work. This talk reviews:

Tor: The Second Generation Onion Router

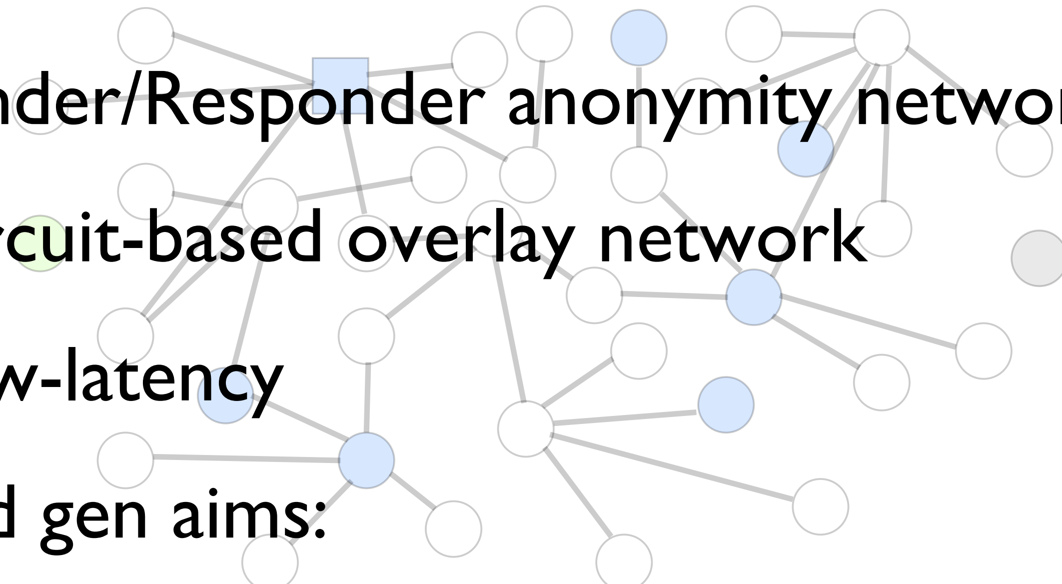


Dingledine, Mathewson, Syverson

Proceedings of USENIX Security '04

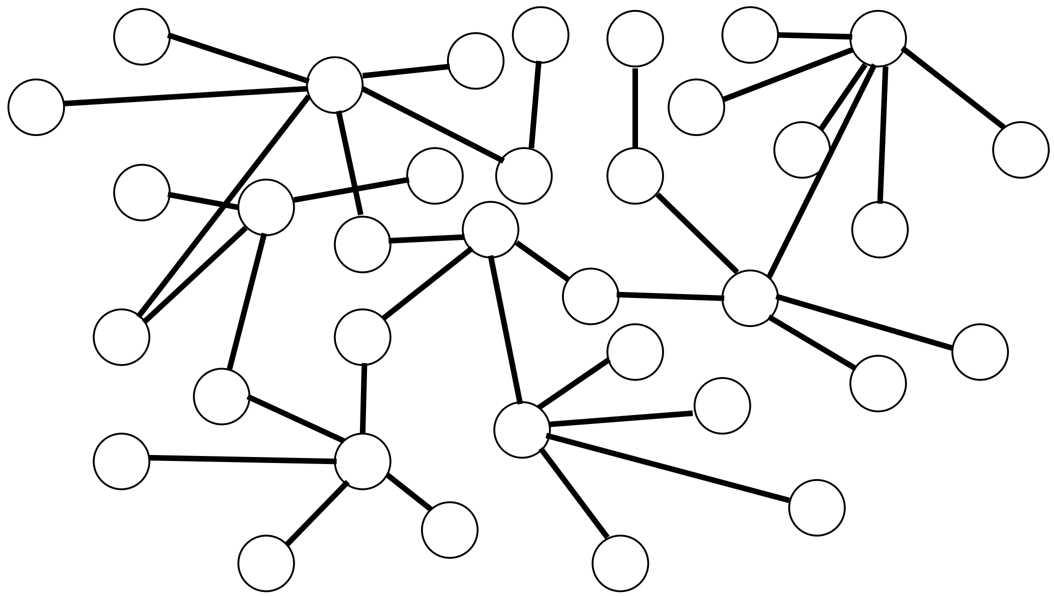
Available at: http://www.usenix.org/publications/library/proceedings/sec04/tech/full_papers/dingledine/dingledine.pdf

What is Tor?

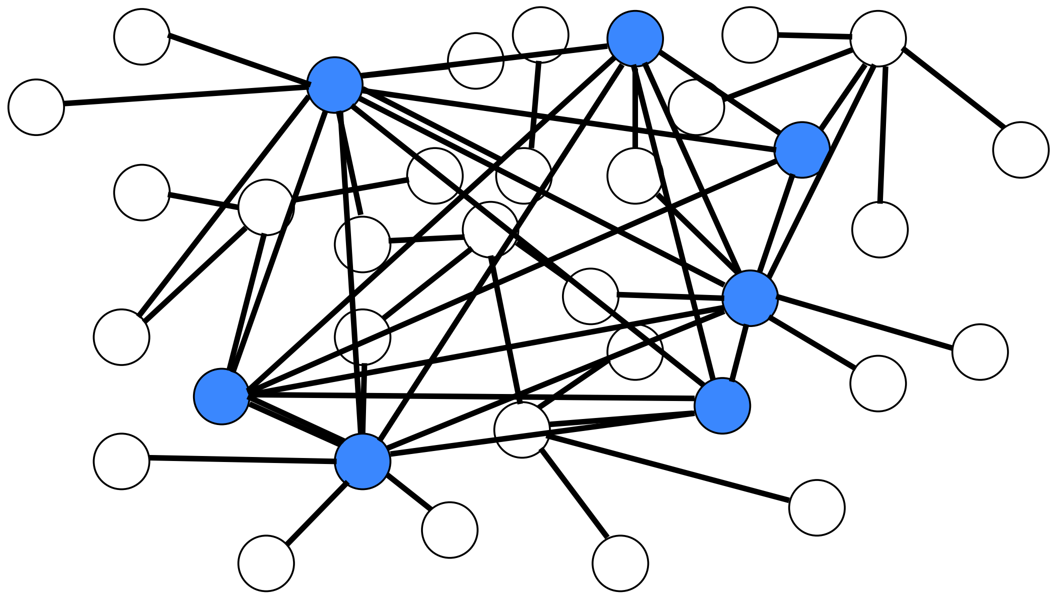


- Sender/Responder anonymity network
- Circuit-based overlay network
- Low-latency
- 2nd gen aims:
  - Perfect forward secrecy, congestion control, directory servers, integrity checking, location hidden servers...

Overlay Networks



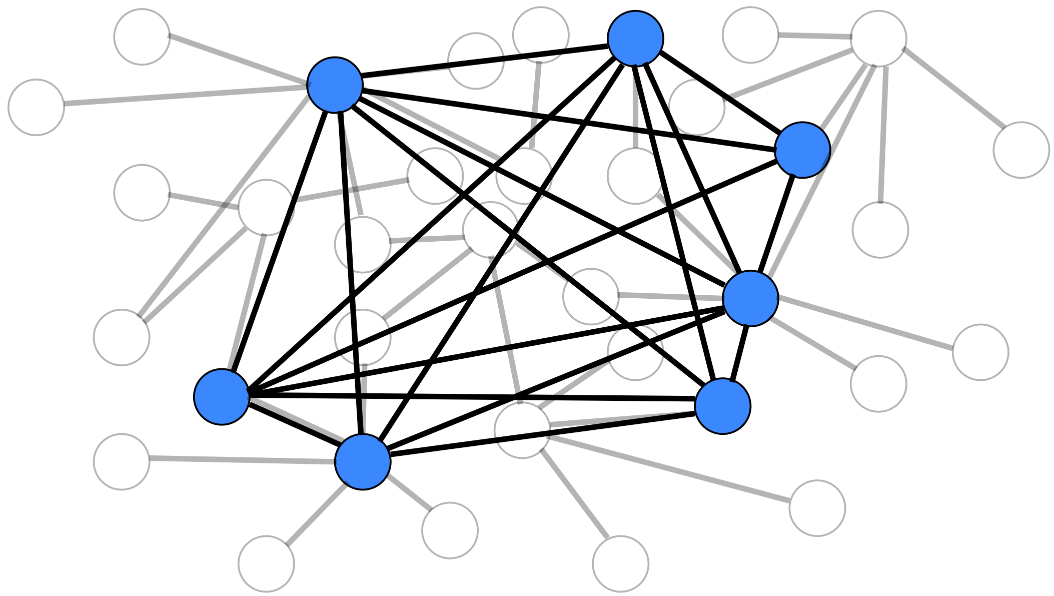
Overlay Networks

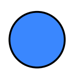


= Overlay (Tor) nodes

= link

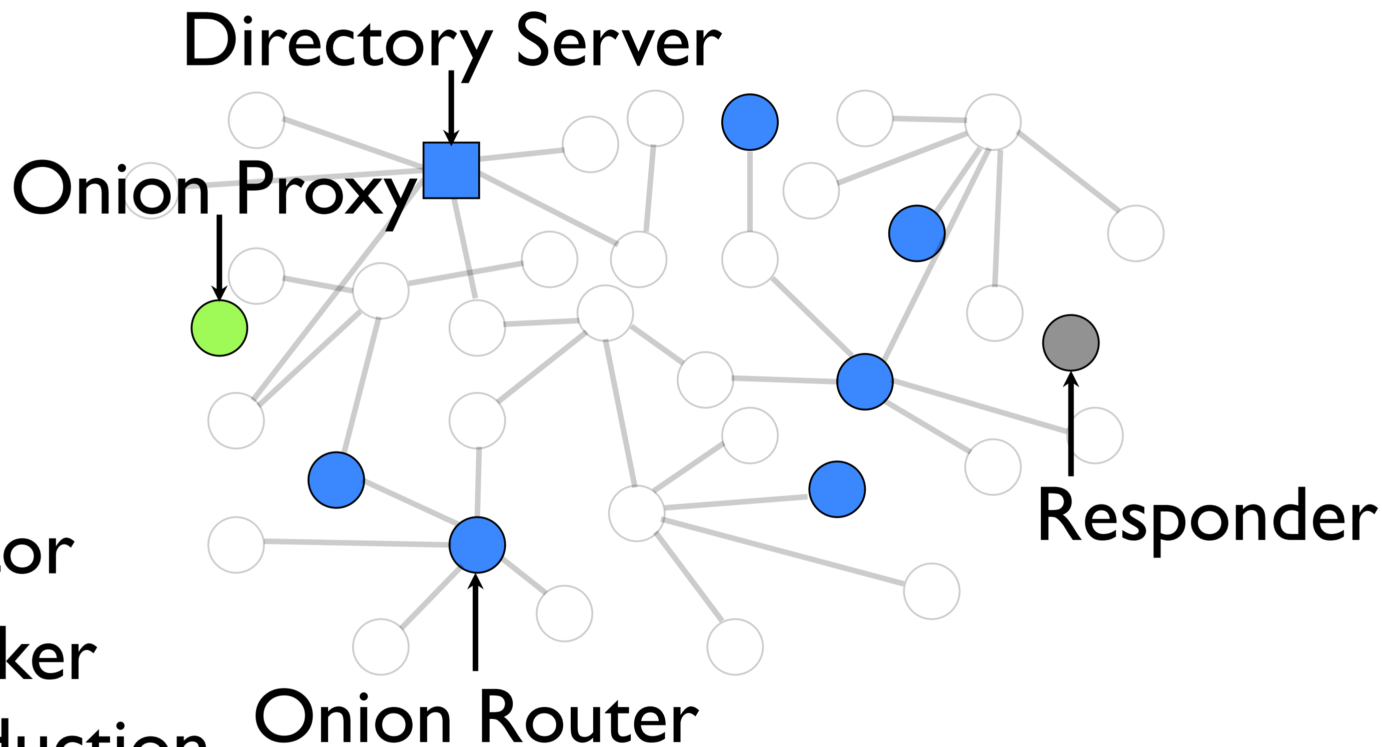
Overlay Networks

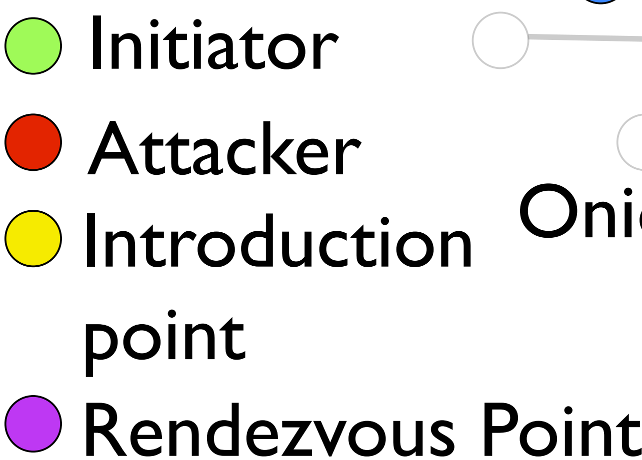


 = Tor node

— = secure link

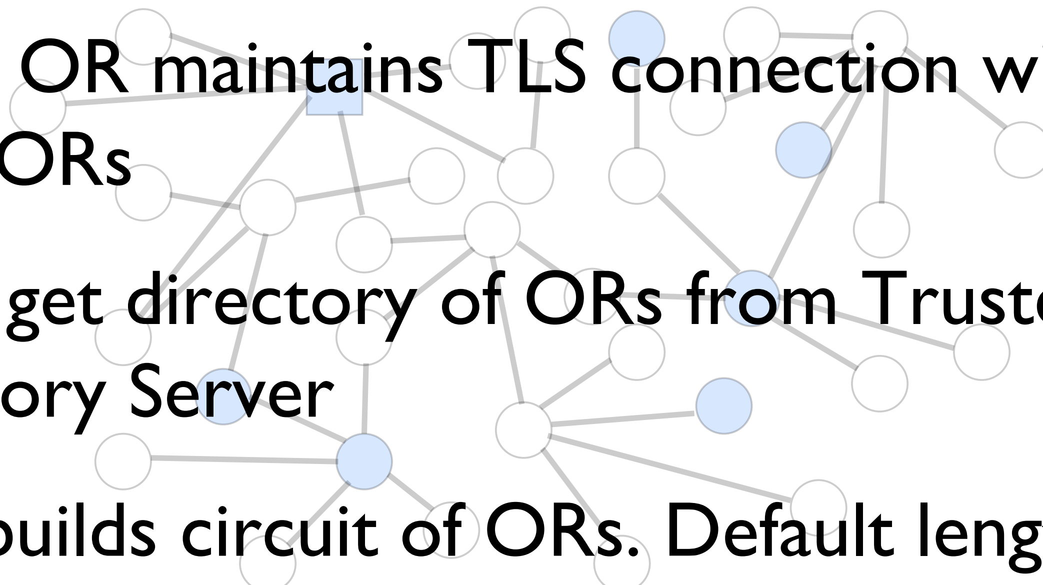
Tor Terminology





Basic Tor ideas

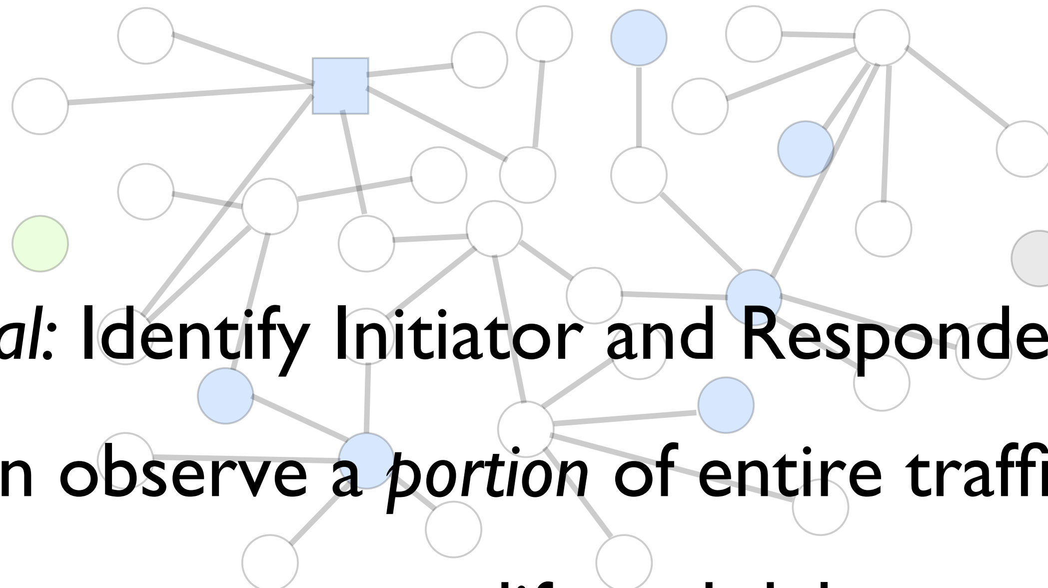
- Each OR maintains TLS connection with the other ORs
- OPs get directory of ORs from Trusted Directory Server
- OP builds circuit of ORs. Default length: 3 ORs.



Tor Threat Model

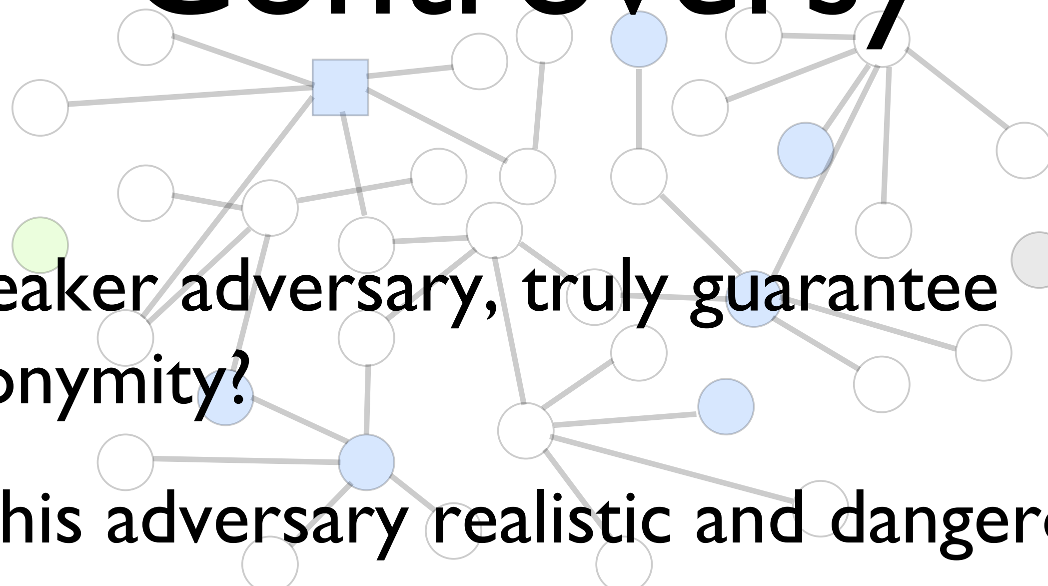
- What type of adversary does Tor attempt to protect users against?
- Typical threat:
  - Global Passive Adversary
- Tor's threat:
  - Partial-view passive adversary

Partial-View Adversary



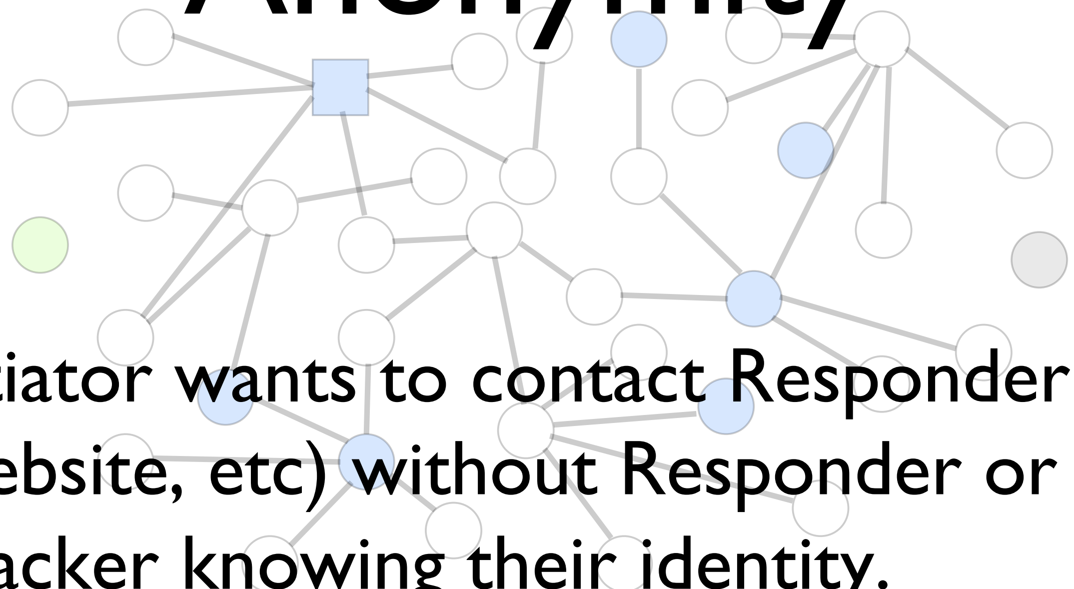
- *Goal:* Identify Initiator and Responder
- Can observe a *portion* of entire traffic
- Can generate, modify and delete traffic
- Can operate Onion routers (ORs) or compromise a % of ORs

Threat Model Controversy



- Weaker adversary, truly guarantee anonymity?
- Is this adversary realistic and dangerous?
- Does it matter?

1st Goal: Initiator Anonymity



- Initiator wants to contact Responder (website, etc) without Responder or any attacker knowing their identity.

Building a Circuit

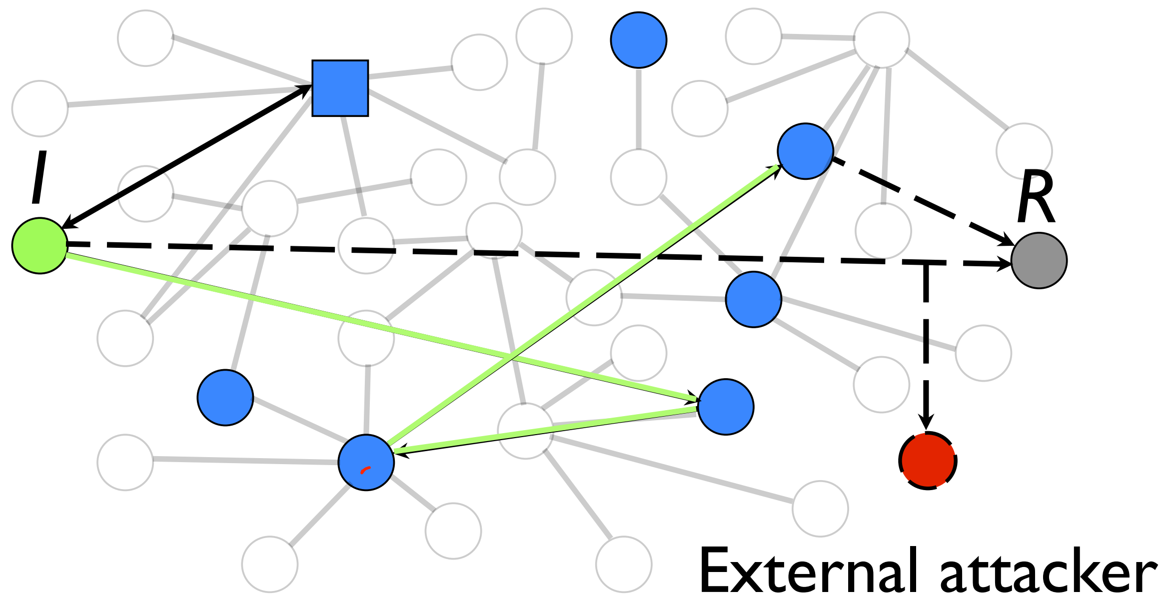
- 1. *I* Gets list of ORs from Directory Server

2. *I* Randomly selects an OR (entry point)

3. *I* Randomly selects an OR, extends circuit

4. *I* Randomly selects a final OR, (exit point)

5. *I* Contacts *R*



Circuit Details

- Tor uses SOCKS proxy
- Creating & extending circuit requires Public Key Crypto
- Communicating over circuit = Diffie-Helman (symmetric crypto)
- Can multiplex TCP connections over circuit, amortize cost of Public Key Crypto
- Rotate circuit to prevent linkability

Circuit Details Cont'd

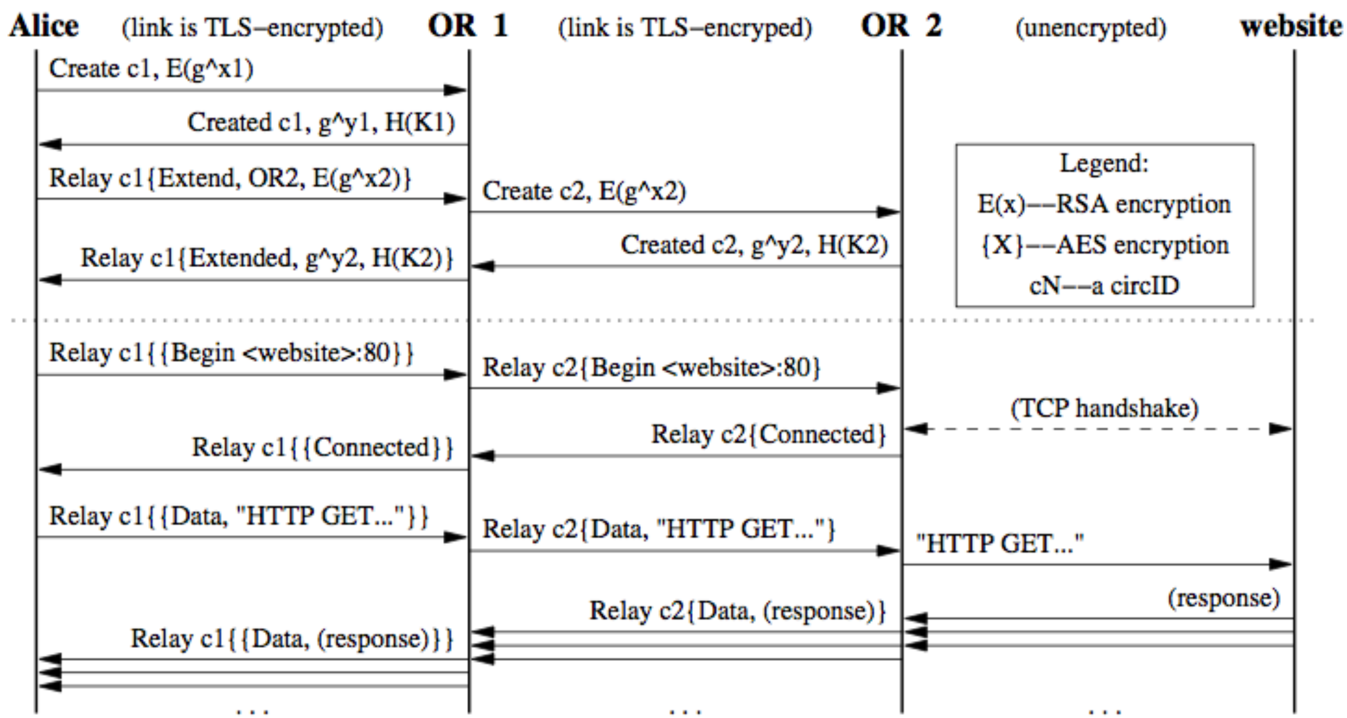
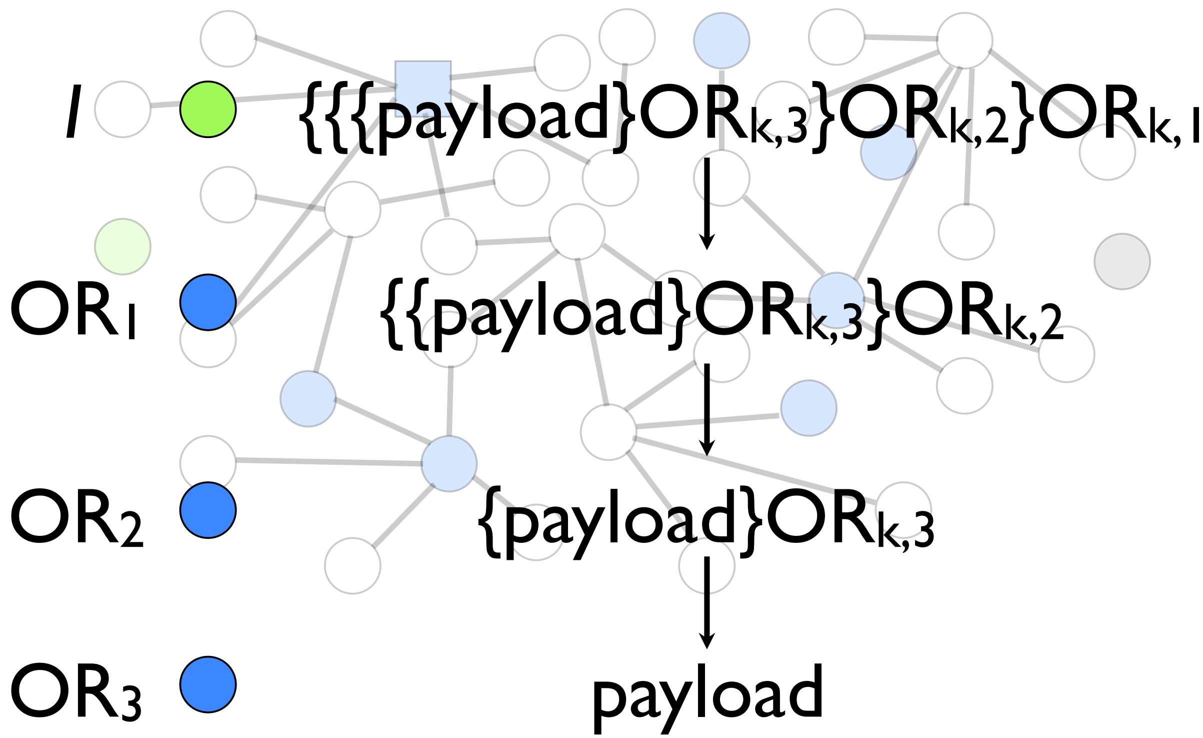


Figure 1 from Dingledine et al.

Cells: Transport over Circuits

- 512 bytes
- Header:
- Circuit ID
- Command
  - Create, extend, destroy circuit
  - relay data, relay begin, relay teardown
- Payload: encrypted payload

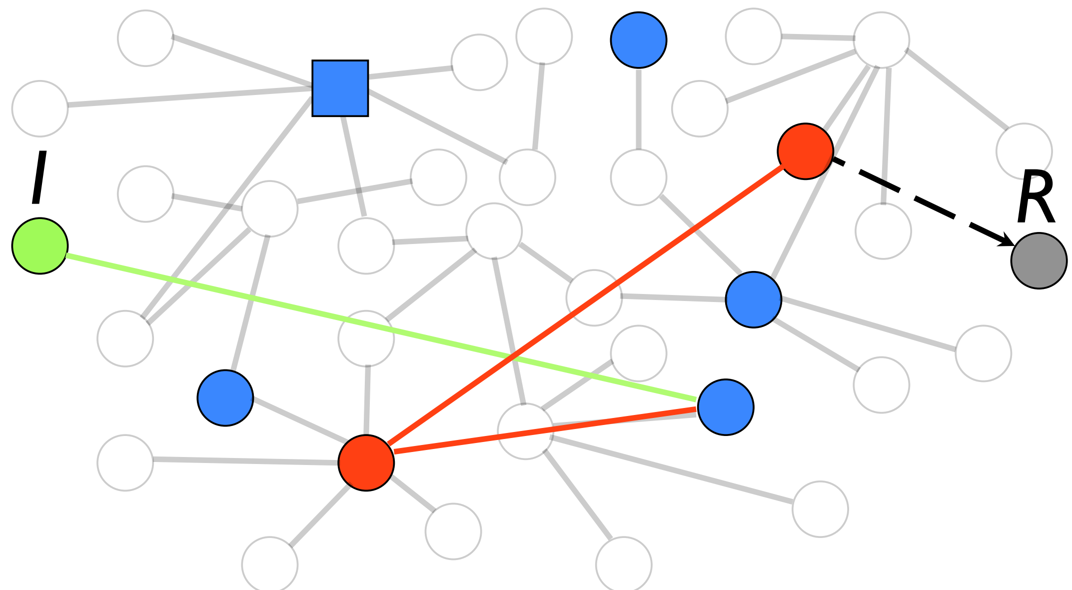
Onion Routing



$OR_{k,i}$ = Ephemeral DH key for circuit

Malicious Onion Routers

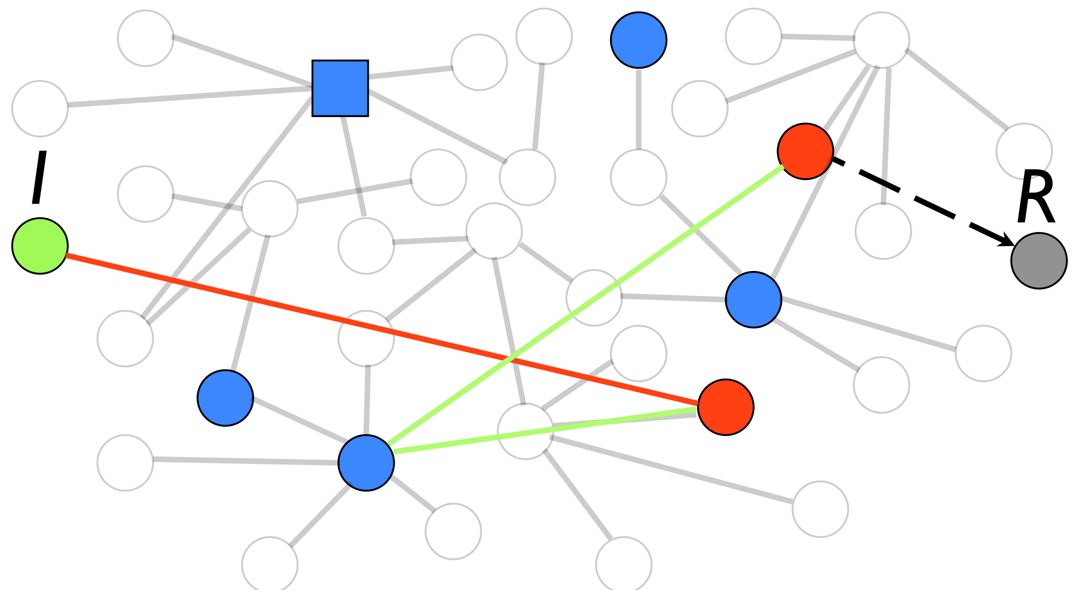
In general, circuits are secure if there is one non-malicious OR in the circuit



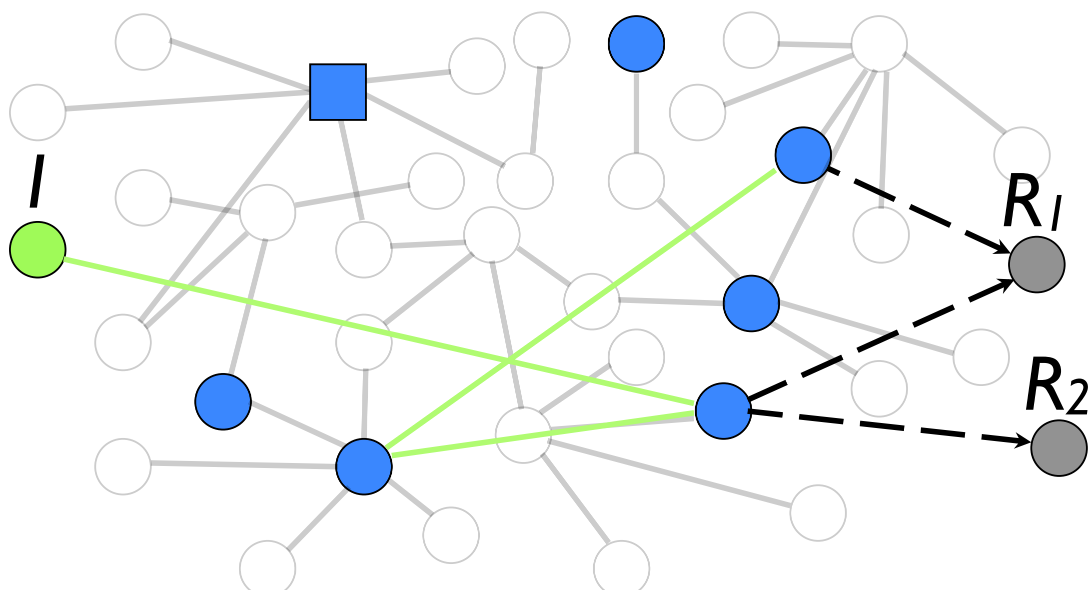
Malicious Entry/Exit Points

If entry/exit points collude, they know that *I* and *R* are using Tor. Can conduct *timing analysis* to try and link *I*/*R*

A colluding clique of size m can observe $(m/N)^2$ of the traffic



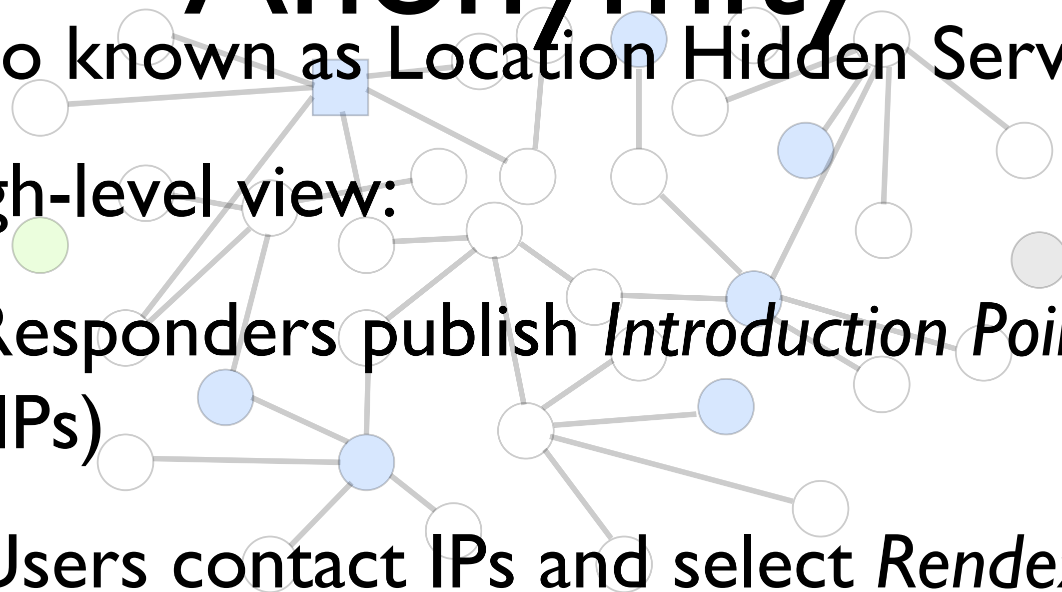
"Leaky-pipe" Circuits



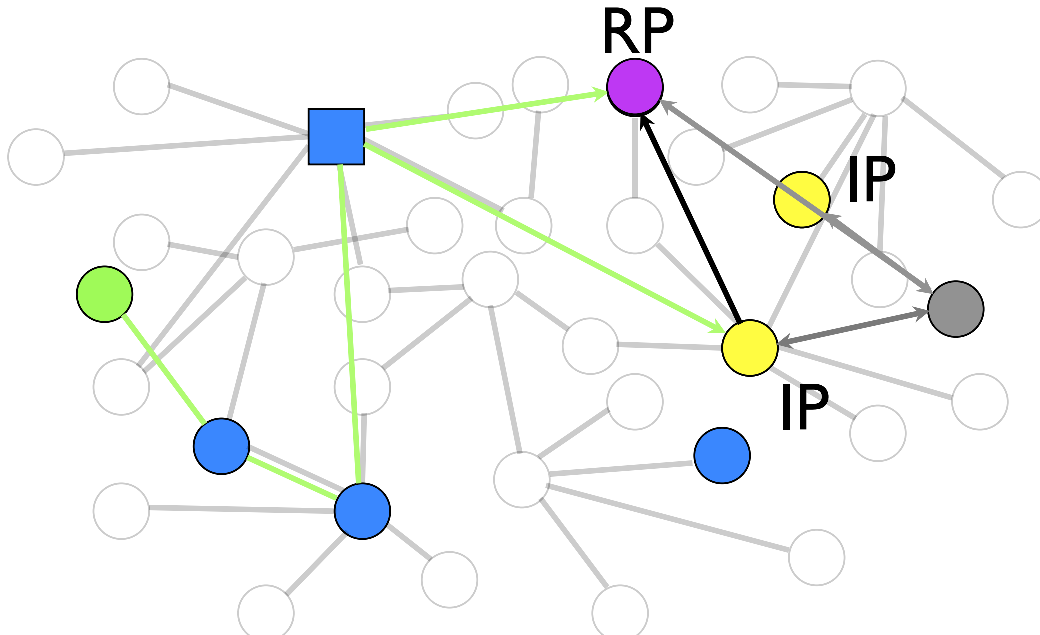
Multiple possible exit points from circuit

2nd Goal: Responder Anonymity

- Also known as Location Hidden Servers
- High-level view:
  - Responders publish *Introduction Points* (IPs)
  - Users contact IPs and select *Rendezvous Point* (RP)
  - User and Responder establish circuit through RP



2nd Goal: Responder Anonymity



What Tor is/does

- Stream integrity checking (TLS)
- Forward Secrecy
  - after circuit demolished, traffic unreadable
- Rate limiting/fairness
- Application transparent

What Tor isn't/doesn't

- Steganographic
  - Does not conceal who is connected
- Prevent end-to-end timing attacks
- Do protocol normalization. No app-level anonymization (cookies/http info)

This is NOT the presenter's original work. This talk reviews:

Low-Cost Traffic Analysis of Tor

Murdoch and Danezis

IEEE Symp. on Security and Privacy '05

Available at: http://ieeexplore.ieee.org.proxy.lib.uiowa.edu/xpls/abs_all.jsp?arnumber=1425067

Goal

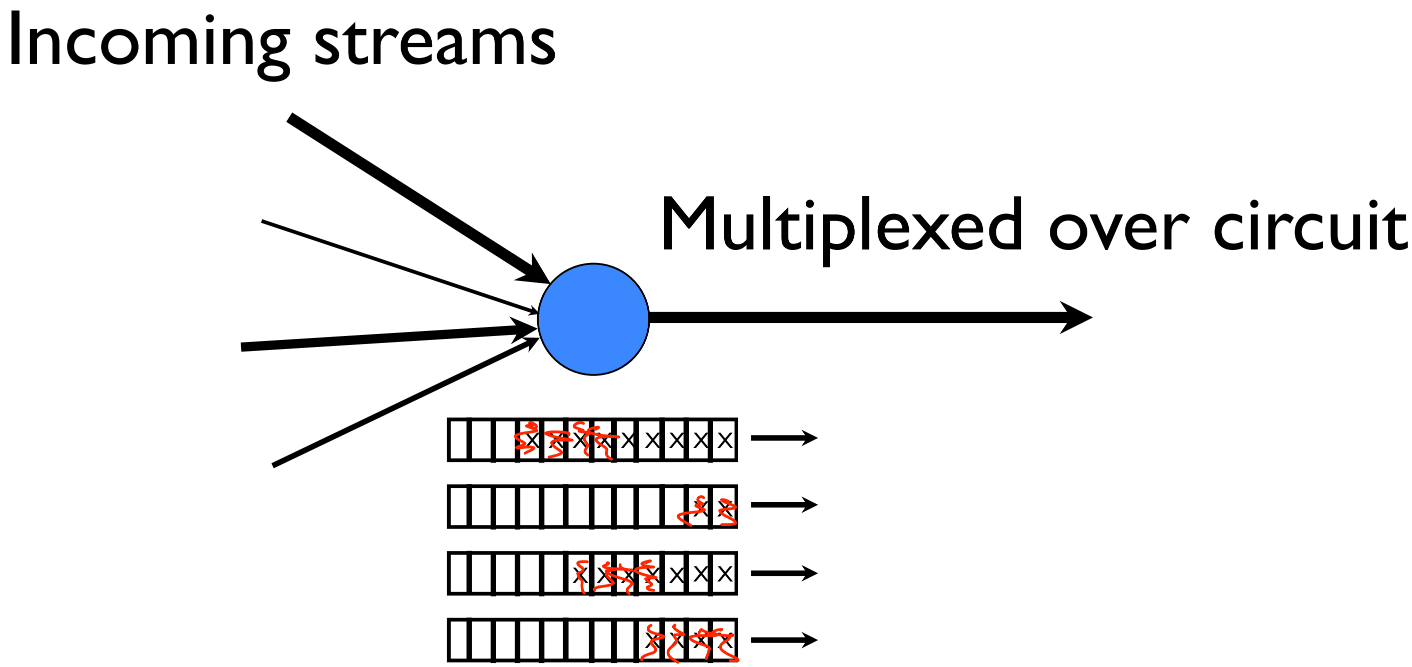
- Show that even within Tor’s limited threat model, traffic analysis/timing attacks are possible.
- Intuition: Use the anonymity network as an oracle to infer network load.
- Assume encrypted tunnels effectively hide bit patterns.

How: Covert Side Channels

- Covert side-channels
- Extra sources of information, does not “break” security used in algorithm.
- In this case, timing attack

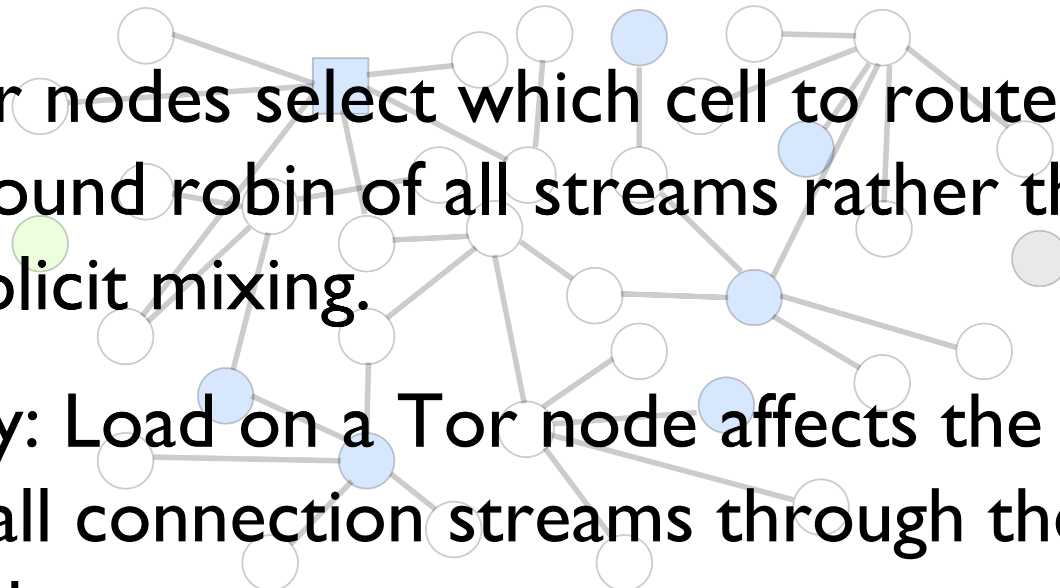
Idea behind attack

- Use the timing signature of an anonymous stream to track the stream through Tor.
- Because Tor is low-latency, it does not engage in traffic-shaping or "mixing" (re-ordering packets from different streams).
- Streams pass through Tor more or less unaltered.



Incoming streams
Multiplexed over circuit

Why it works



- Tor nodes select which cell to route using a round robin of all streams rather than explicit mixing.
- Key: Load on a Tor node affects the latency of all connection streams through the node.
- Compare change in latencies to known traffic patterns

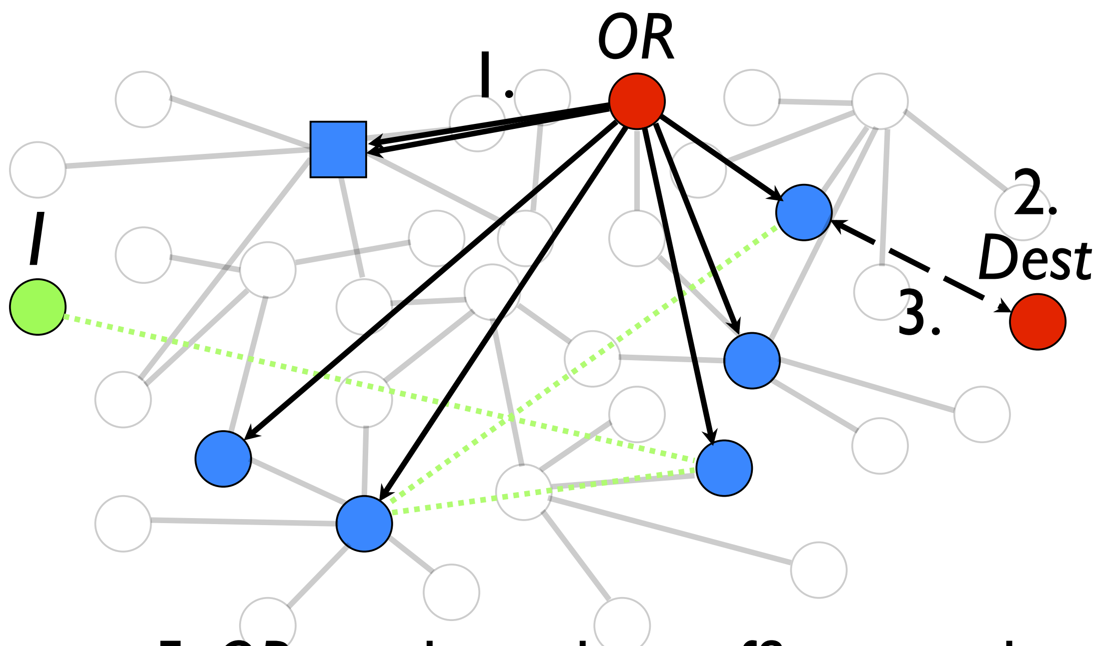
Attack Set-up

1. Malicious OR joins Tor network

We want to observe who I is talking to (dotted = corrupt circuit)

3. User I establishes link with hidden server

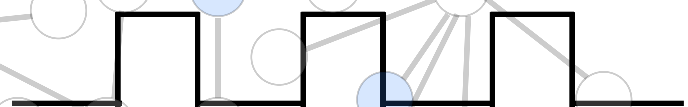
4. *Dest* returns traffic to *I* according to selected pattern



2. Attacker controls/ corrupts a server that Tor users talk to

5. *OR* sends probe traffic to each legitimate OR, if latency is correlated with signal, *I* is using that router

Details

- Signal = bursty 
  - Corrupt server transmits for 10-25 sec
  - Corrupt server is quiet for 30-75
- Corrupt *OR* measures latency of probe traffic. If it is monitoring an OR through which stream passes, latency should increase in correlation with victim signal.

Measuring Correlation

$$c = \frac{\sum S(t) \times L'(t)}{\sum S(t)}$$

- *S(t)* = Indicator variable.
  - 1 if corrupt server is submitting, 0 otherwise.
- *L'(t)* = normalized latency at time *t*
  - Normalized by median latency

Experimental evaluation

- Tested 13 Tor nodes (out of 50 available)
- 11 of 13 cases: correctly identified case in which node was carrying victim traffic compared to stream flowing through other nodes
- Suggest increasing time of test to improve results.
- Also tested for FPs: no ‘echoes’ of stream at other nodes

Good correlation

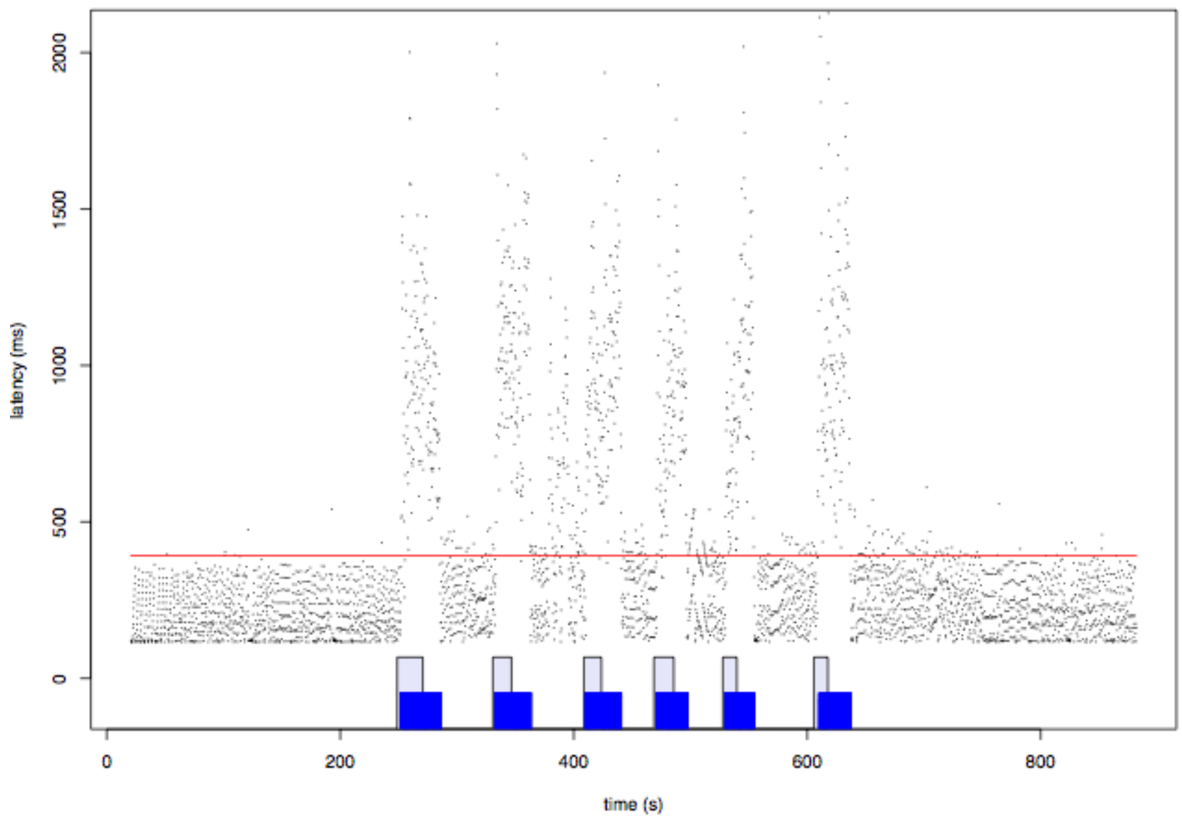
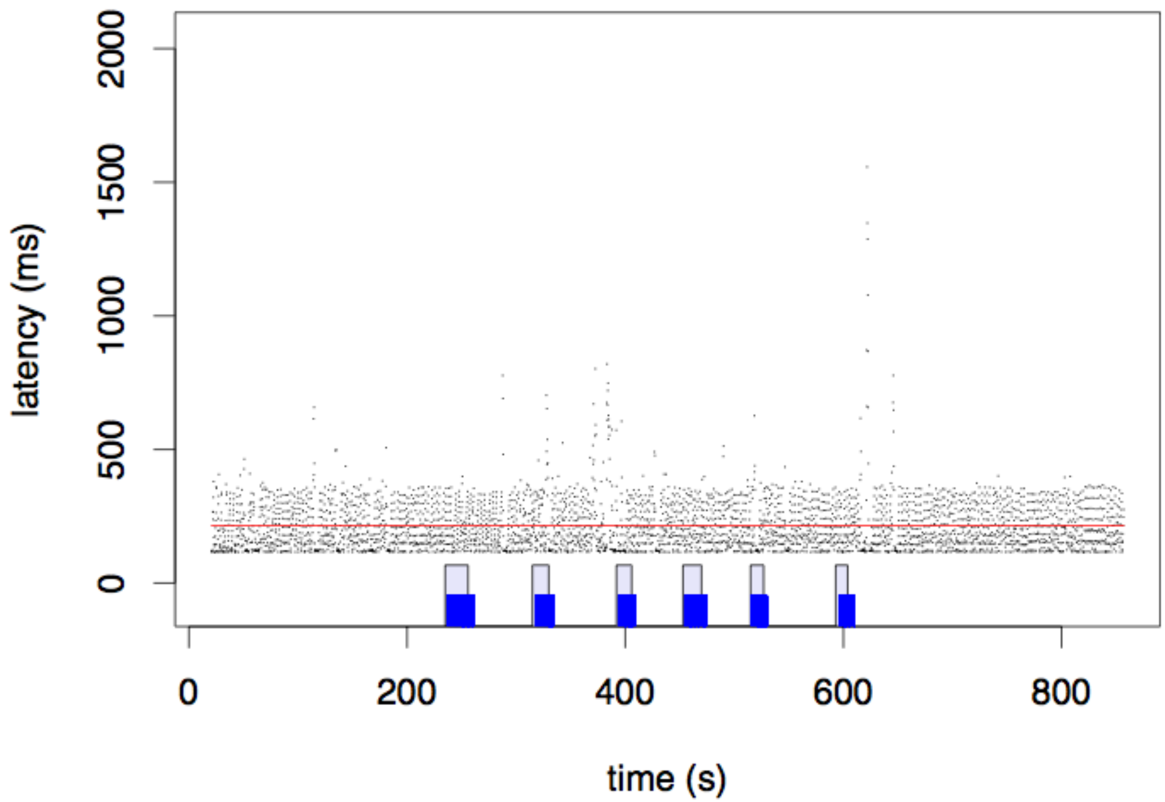


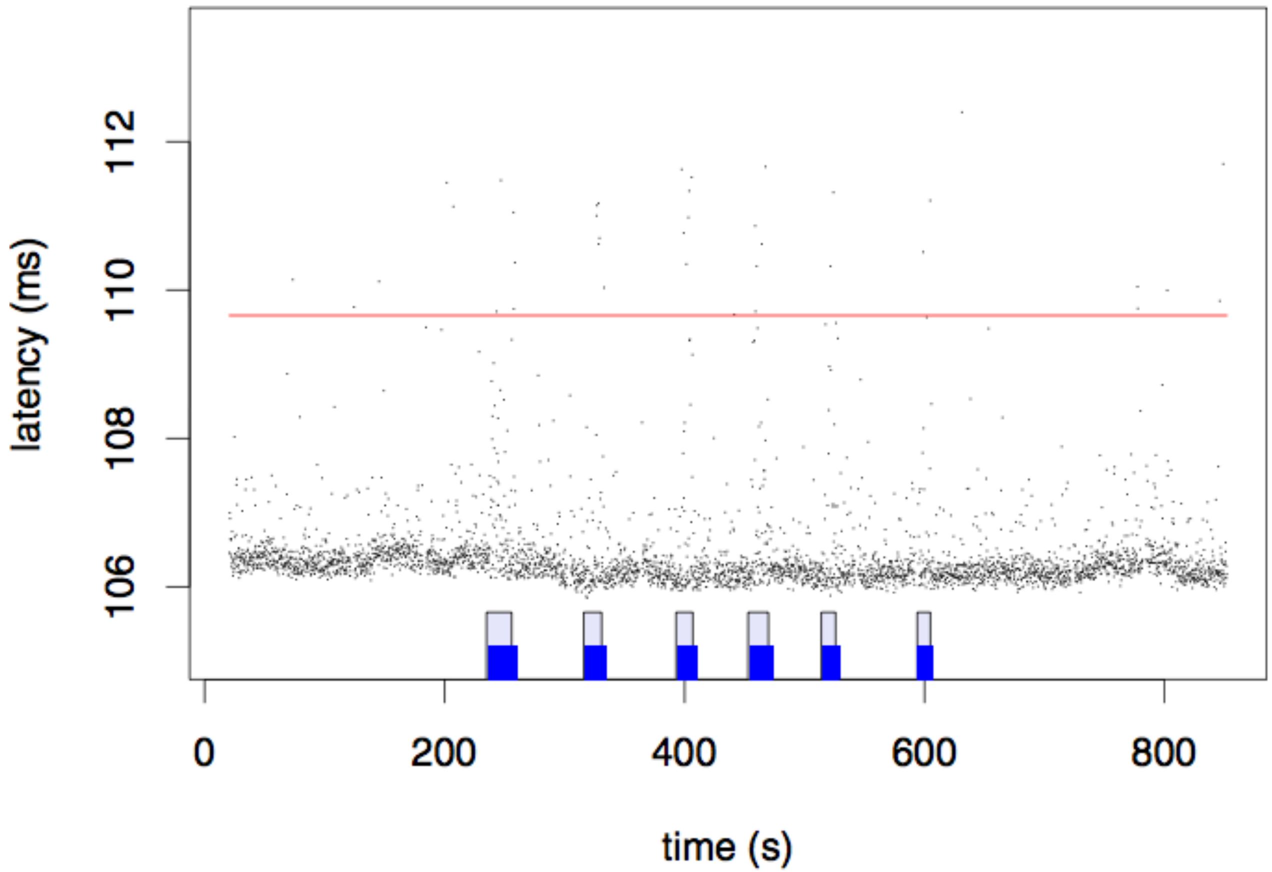
Figure 2. Probe results showing good correlation (Node K)

No echoes



(a) Probe results without traffic pattern (Node K)

Bad Correlation



(b) False negative (Node E)

Results for 13 nodes

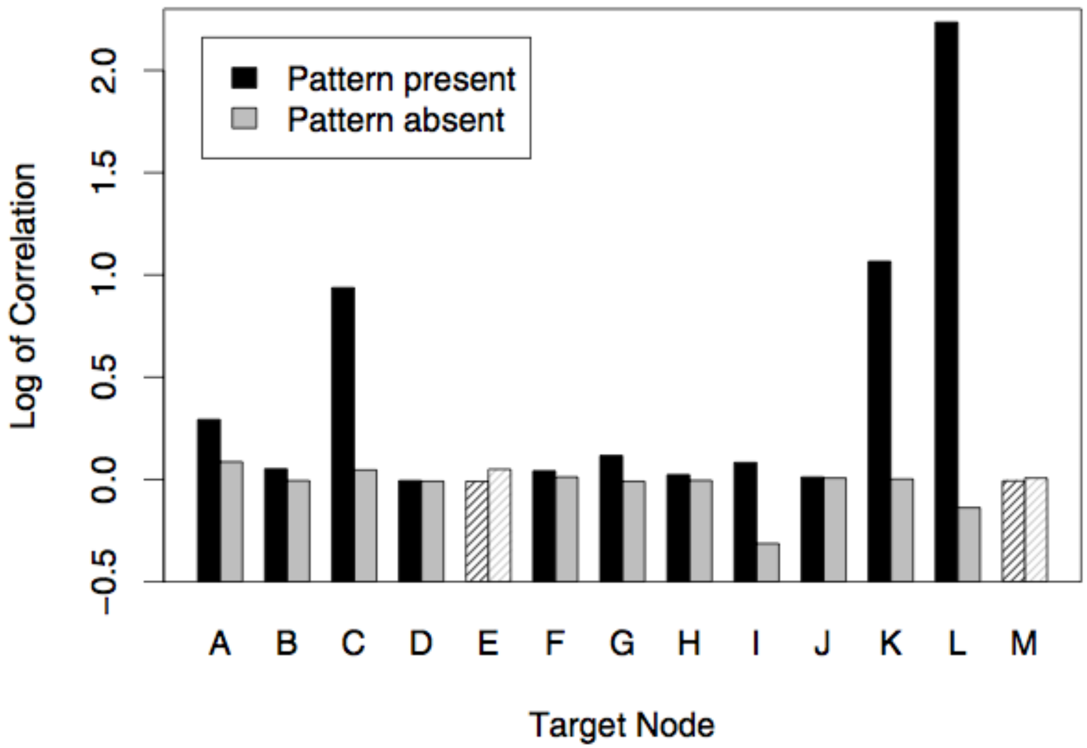


Figure 4. Summary of correlation

Analysis of Attack

- What is the actual reduction in security?
- Is it doable?
- Are there countermeasures?