TorScan: Tracing Long-lived Connections and Differential Scanning Attacks

A. Biryukov, I. Pustogarov, R.P. Weinmann

University of Luxembourg

ivan.pustogarov@uni.lu

September 5, 2012

Overview

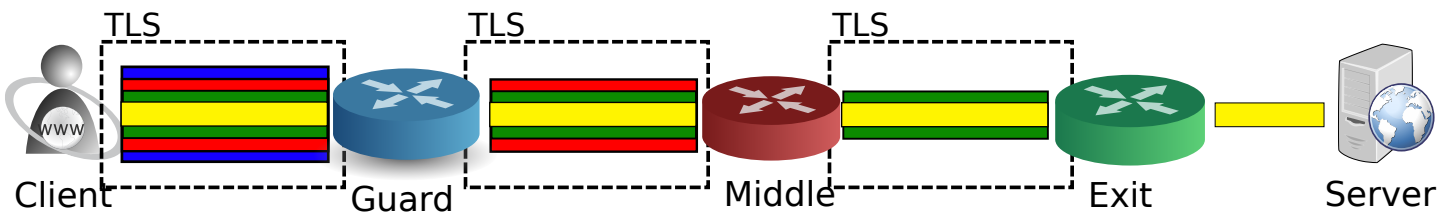
- What is Tor and How it works
- A classification of published attacks on Tor
- Revealing Tor topology information
- Topology-based attacks
- Evaluation of the attacks

Tor anonymity network

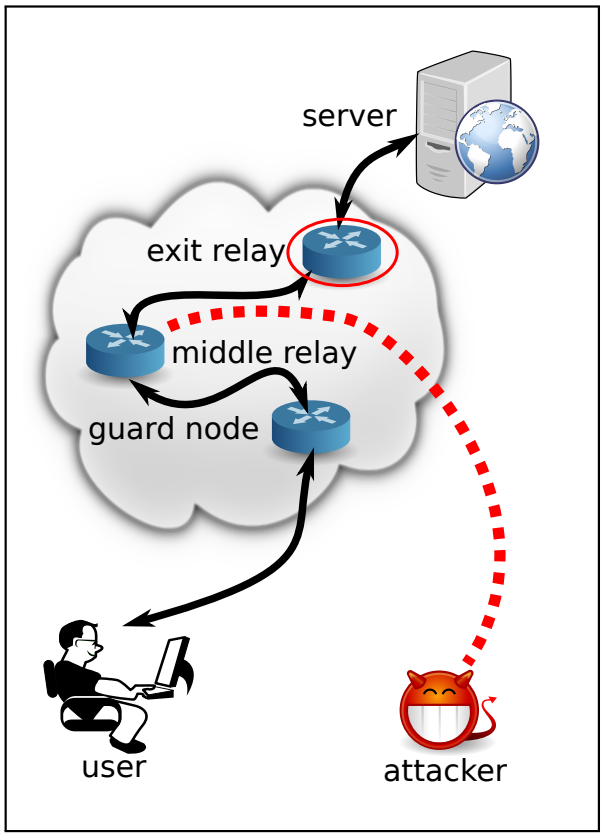
- Tor relays – Internet servers running Tor software
- Both relays and clients run the same software
- Authorities – 9 most trusted relays which maintain and distribute the list of Tor relays (the Consensus)
- Clients route their traffic through a chain of Tor relays
- Tor relays do not delay traffic nor use padding
- Is the most popular anonymity network (>3000 Tor relays; 400,000 users/day).
- It is fast.

Tor anonymity network

- A user chooses three Tor relays – guard,middle,exit – and builds a circuit: exchanging symmetric keys and updating the relays' Tor routing tables
- Telescoping is used to exchange symmetric keys between the client and the relays
- Traffic flows down the circuit in fixed-size cells, which are unwrapped by a symmetric key at each node (like the layers of an onion)
- TLS connections between relays are used as the transport for Tor cells
- **One TLS connection carries many circuits**

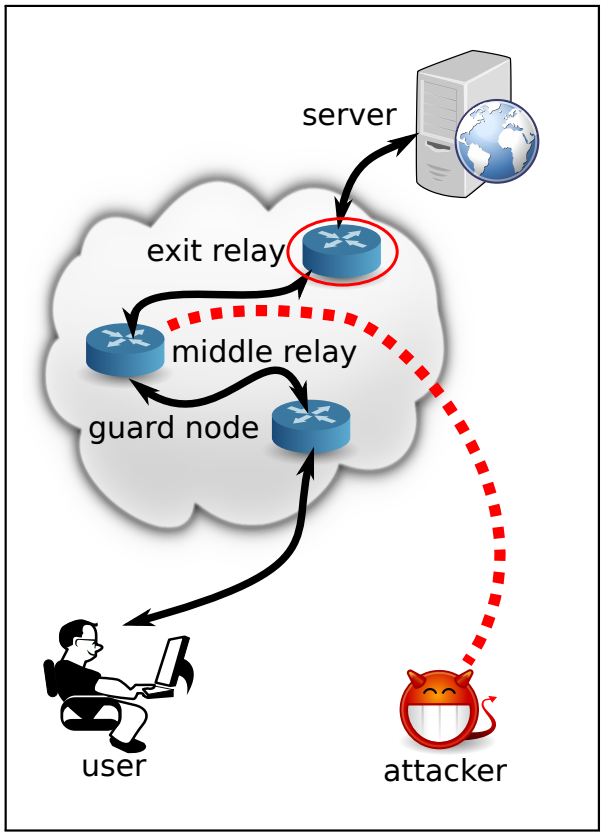


Classification of published attacks



- Passive traffic analysis attacks.
  - Cell counting, time correlation, Website fingerprinting
- Active traffic analysis attacks
  - Watermarking, node clogging, etc.
- Attacks based on information leakage from specific applications.
  - Bittorrent leaking IP addresses, etc.

Intersection attack and Guard nodes



- Controlling the entry node and sniffing the server $\equiv$ locating the client
- If the entry node is chosen randomly, as the client makes many different circuits over time, then the probability that the attacker will see a sample of the traffic goes to 1
- It does not happen if the entry nodes are fixed. **Each client has a pool of three guard nodes which are actual for 1 month**

The attack goal

- We do not reveal the actual IP address of a client
- We do reveal the guard nodes of a client
- Guard nodes are the next point to attack.
- Guard nodes are specific to the user and can be considered as his signature

The key to our attack

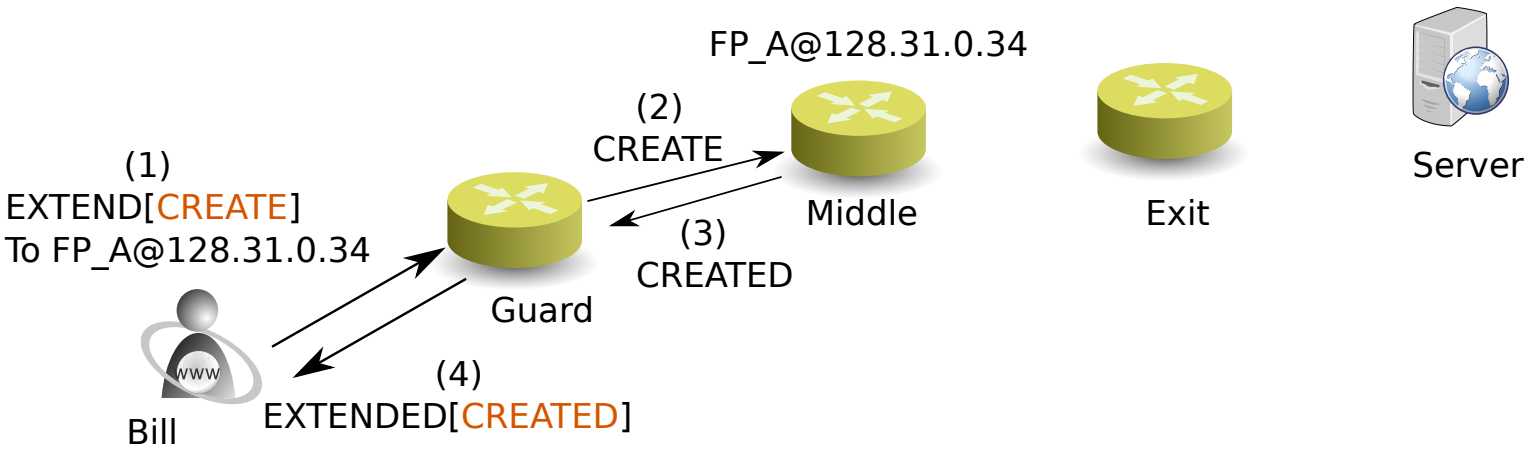
- Before the Tor network was considered as a fully connected graph
- We have found ways to probe the connectivity of a Tor relay.
- We found how topology leakage can be used to trace back a user from an exit relay to a small set of potential entry nodes.

Revealing Tor connectivity (1/5)

- Each node is uniquely identified by its RSA public key (fingerprint)
- Use CREATE/CREATED cells to exchange a Diffie-Hellman key.
- Use EXTEND cell to relay a CREATE cell
- EXTEND cell contains the fingerprint and IP address of the next relay
- EXTENDED cell indicates the success; DESTROY cell indicates a failure

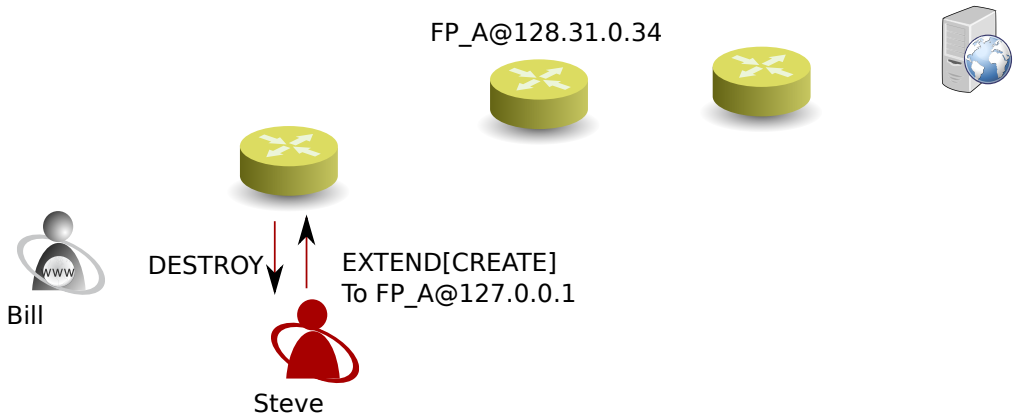
Revealing Tor connectivity (2/5)

- Each node is uniquely identified by its RSA public key (fingerprint)
- Use CREATE/CREATED cells to exchange keys.
- Use EXTEND cell to relay a CREATE cell
- EXTEND cell contains the fingerprint and IP address of the next relay
- EXTENDED cell indicates the success; DESTROY cell indicates a failure



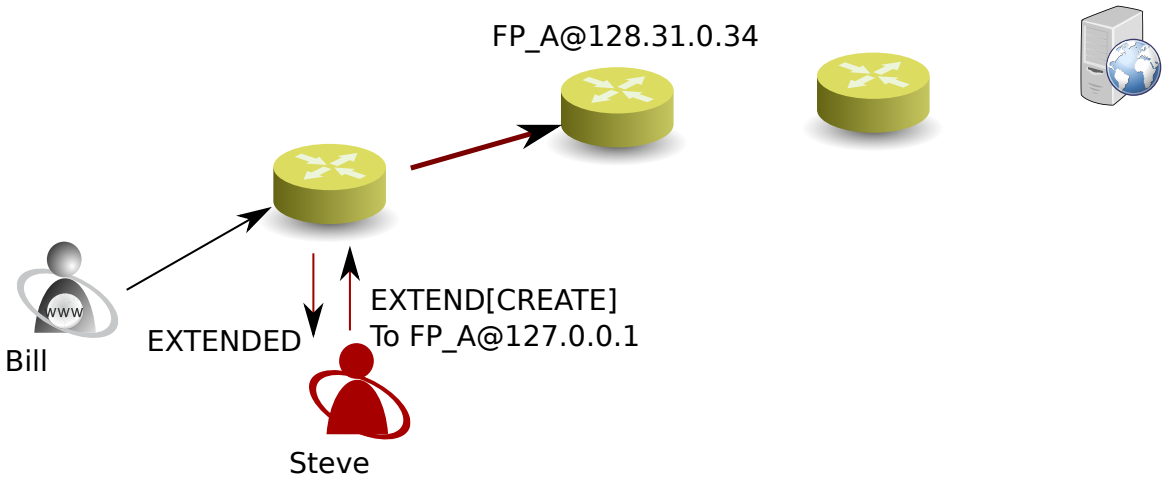
Reavealing Tor connectivity (4/5)

- Each pair of Tor relays tries to keep just one TLS connection
- A connections to Relay A is *Canonical* if IP_A and $fingerprint_A$ of relay A are from the Consensus document
- The canonical connection will be used for all subsequent circuit extension requests to the relay with $fingerprint_A$. **IP address is ignored**
- What happens if I indicate a fingerprint from the consensus and 127.0.0.1 as IP



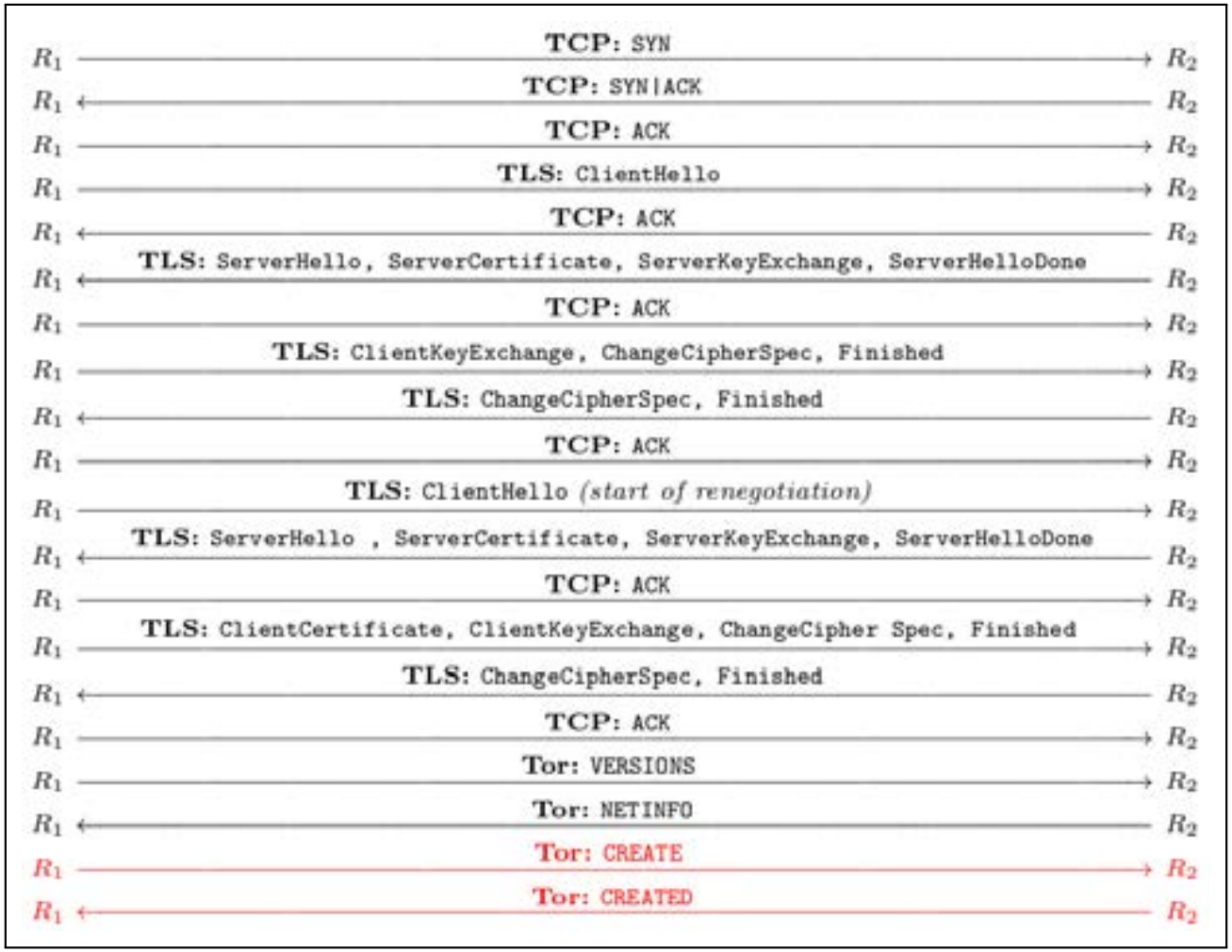
Reavealing Tor connectivity (3/5)

- Each pair of Tor relays tries to keep just one TLS connection
- A connections to Relay A is *Canonical* if IP_A and $fingerprint_A$ of relay A are from the Consensus document
- The canonical connection will be used for all subsequent circuit extension requests to the relay with $fingerprint_A$. **IP address is ignored**
- What happens if I indicate a fingerprint from the consensus and 127.0.0.1 as IP



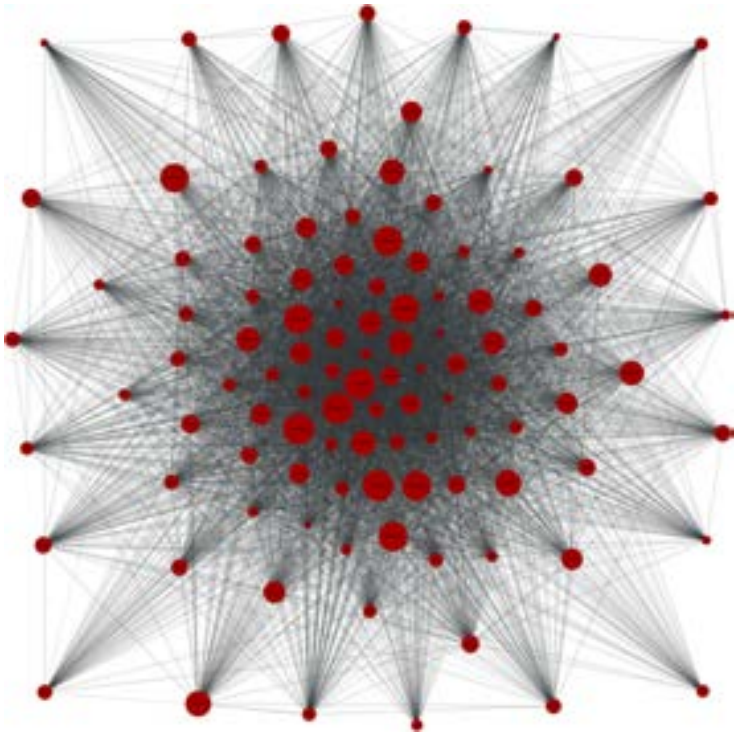
Reavealing Tor connectivity (5/5)

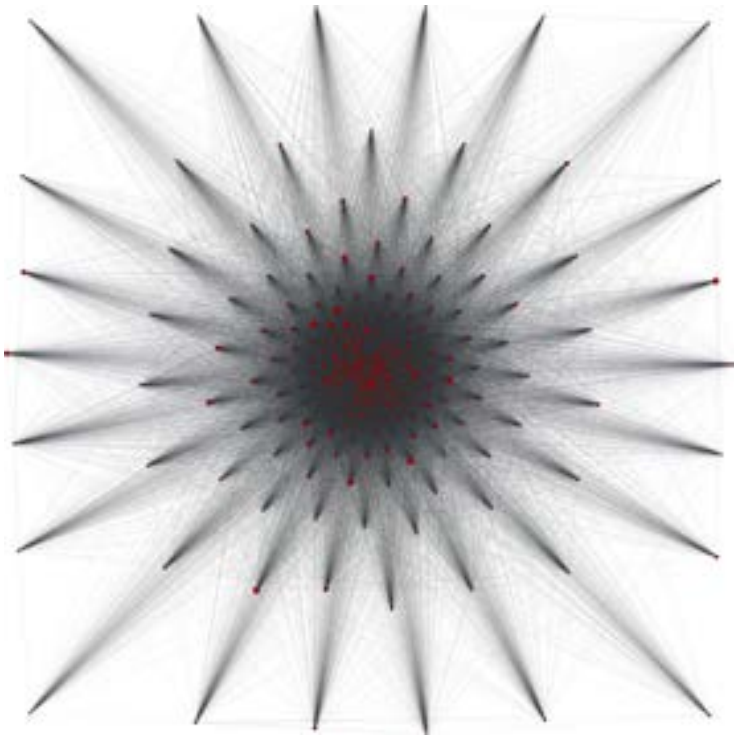
- It takes less time to extend a circuit over already existing connection
- 18 additional steps $\Rightarrow$ at least 18 RTT time slower



Show me!

- One scan takes 30 seconds. Can be easily parallelized.
- 89 USD for scanning the whole network for a day (4 minutes between scans).

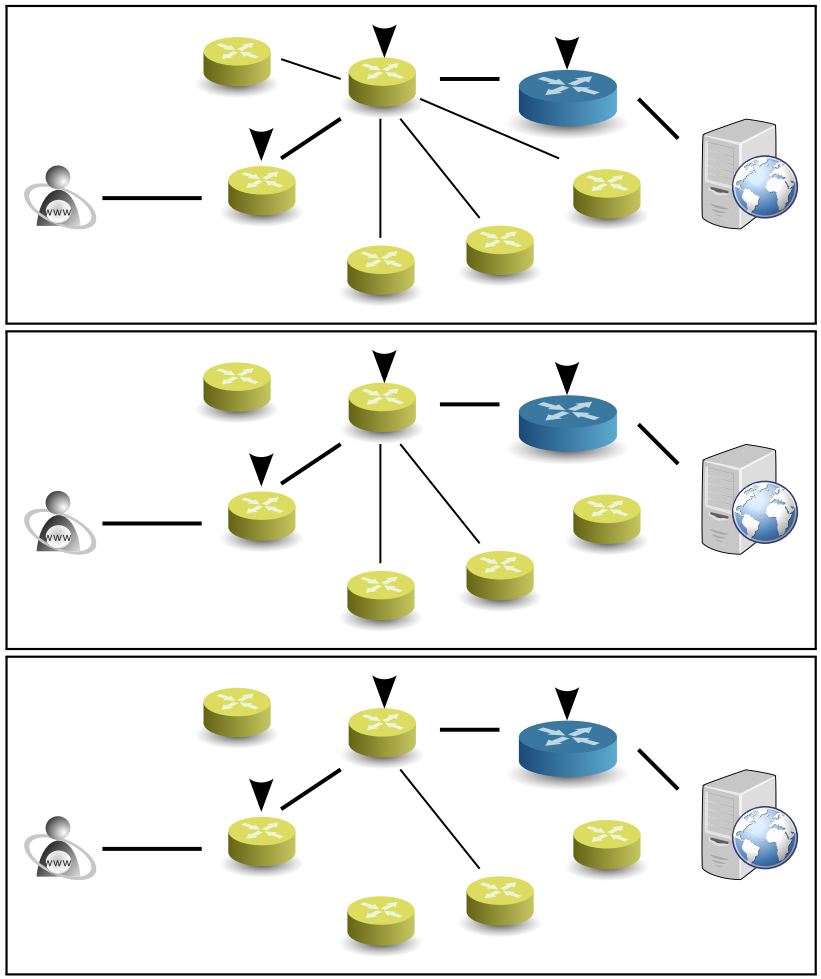




The attack scenarios

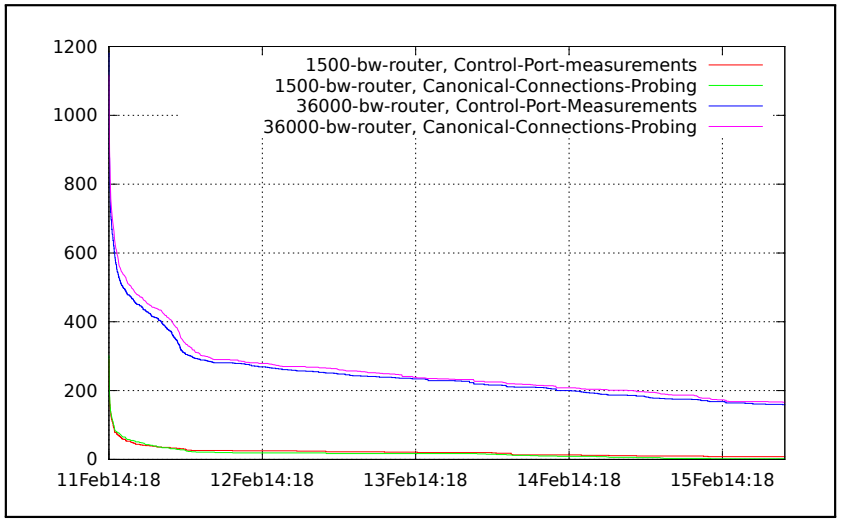
1. We will trace Long-lived connections: long-lived SSH sessions, very large files downloads, file-sharing applications and communications over instant messaging networks.
2. We will trace recurrent connections: Gmail establishes new connections every 2 minutes; web sites with auto-refresh contents. Pseudonymous user is identified by a cookie or a login credential.

Tracing Long-lived connections.



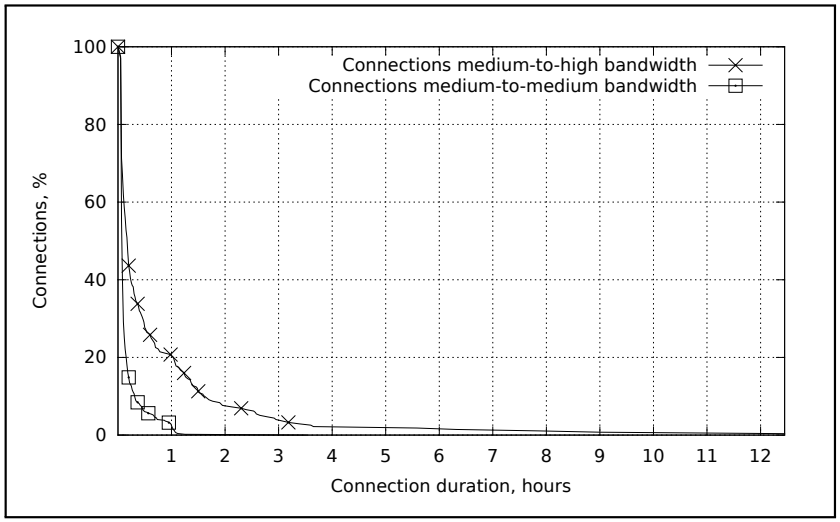
Tracing Long-lived connections. Evaluation

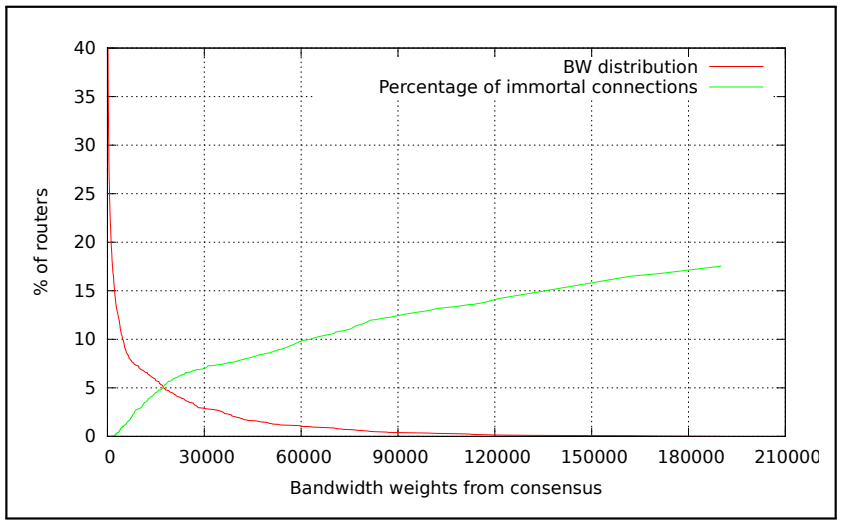
- How long should we wait for other connections to disappear?
- What is the asymptotic behaviour of the decay curve?



Tracing Long-lived connections. Evaluation

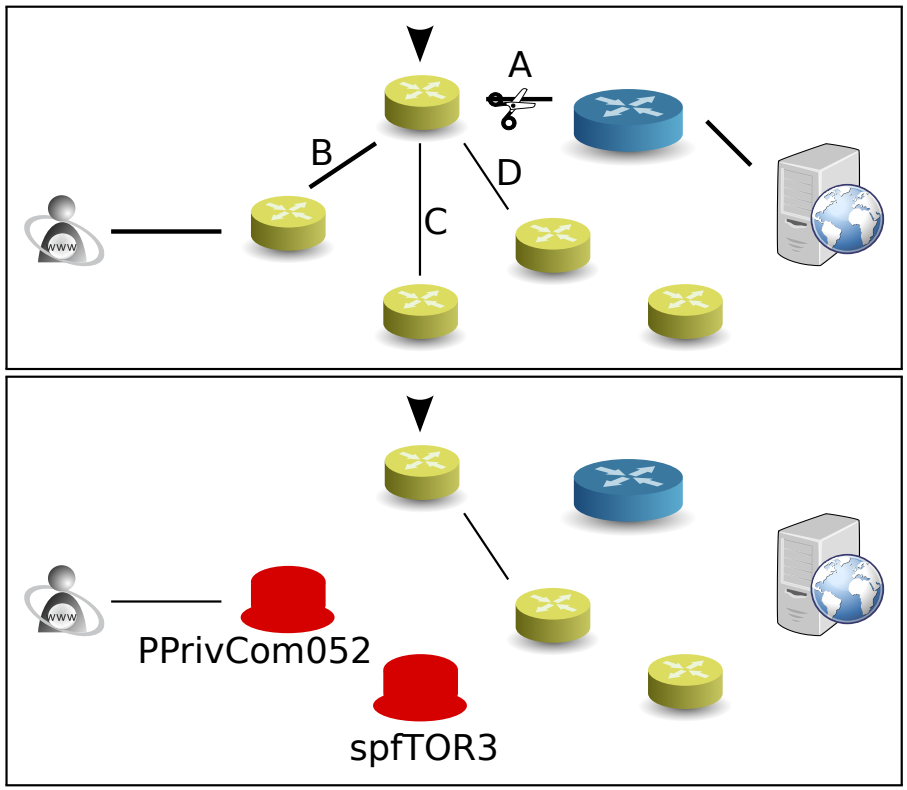
- How long should we wait for other connections to disappear?
  - What is connections duration distribution?
  - Are there "immortal" connections? Incoming circuit rate for this connections is very high
- What is the asymptotic behaviour of the decay curve?





Differential scanning

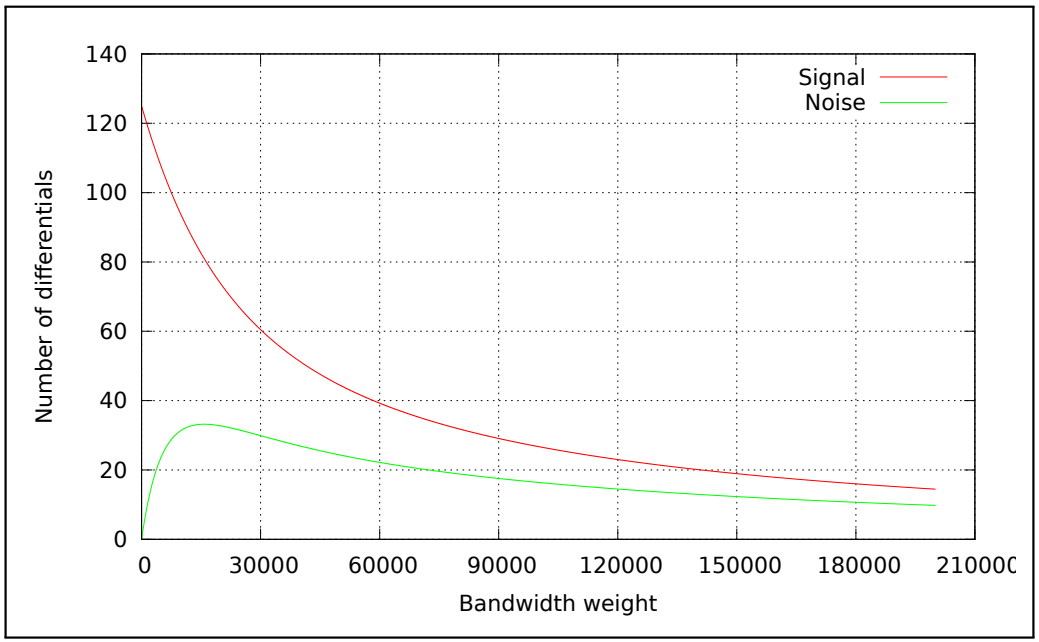
1. Tor Specification: a circuit (route) lives 10 minutes
2. Assume the client establishes recurrent connection to the server
3. The client is identified by his cookie or pseudonym
4. If we kill connection A, there is a probability that connection B will drop as well



Differential scanning. Evaluation

1. What is the probability that connections B drops (Signal)?
2. What is the probability that other connections drops (Noise)?
3. How many tries should we make?

→ `C37B234FAD013453B90375EB55864FEBC876104A`: **58** (PPrivCom052) **bw=36500** ←
`CA1CF70F4E6AF9172E6E743AC5F1E918FFE2B476`: 35 (spfTOR3) bw=29800



Some remarks

- For many existing attacks: exhaustive probing of each link in the Tor network is required
- Existing attacks can become practical again since the amount of links to be probed is significantly reduced

Conclusion

- All prior research on Tor assumed opacity of the Tor network topology meaning that the attacker had to assume a fully connected graph.
- In practice, the real degree of a node in this graph is substantially smaller than its maximum at any given point in time.
- For the first time, we have shown methods to determine the real connectivity of relays in the Tor network and the dynamics of the topology of the whole Tor network.
- Based on this, we described several novel attacks that use this information to deanonymize the entry points of the users into the Tor network.

Thank you.