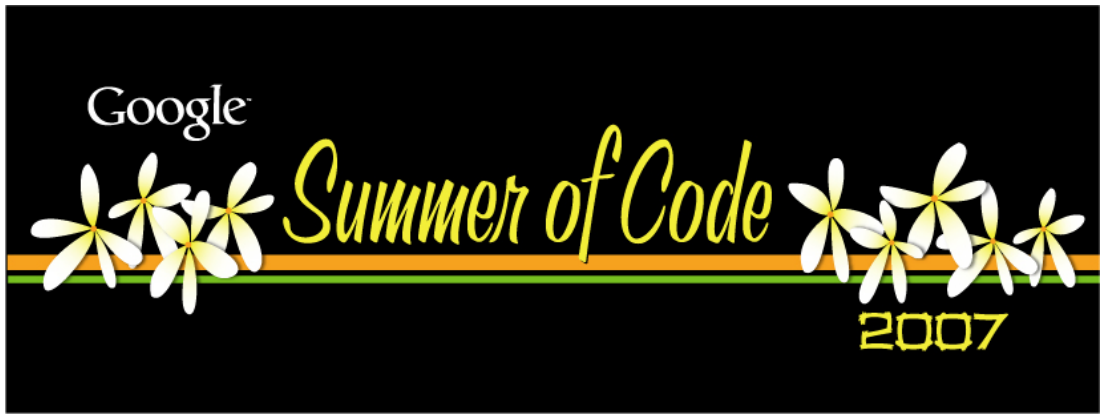
Distributed Storage of Tor Hidden Service Descriptors

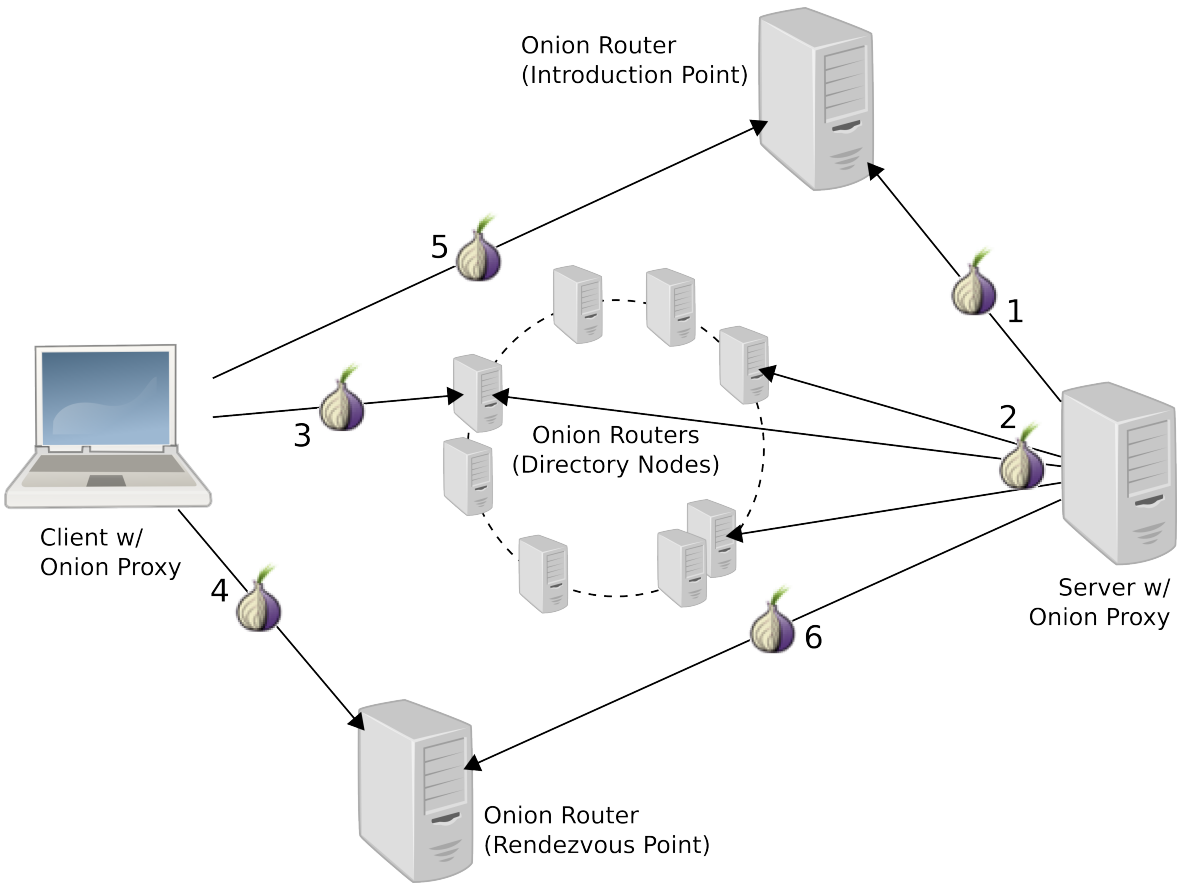
Karsten Loesing

University of Bamberg, Germany



Project objectives

- 3 directory nodes store and serve all hidden service descriptors
- Distribute among large subset of all onion routers ($\approx$ 1000)



Improve security

- Directories learn about service activity and usage
  - Store descriptors under frequently and unpredictably changing IDs
  - PhD thesis: what if services represented people, not machines?
- Directories learn about location of introduction points
  - Encrypt introduction points for clients using cookie (remaining descriptor content stays unencrypted for verification purpose)

Rendezvous Service Descriptor (V2?)

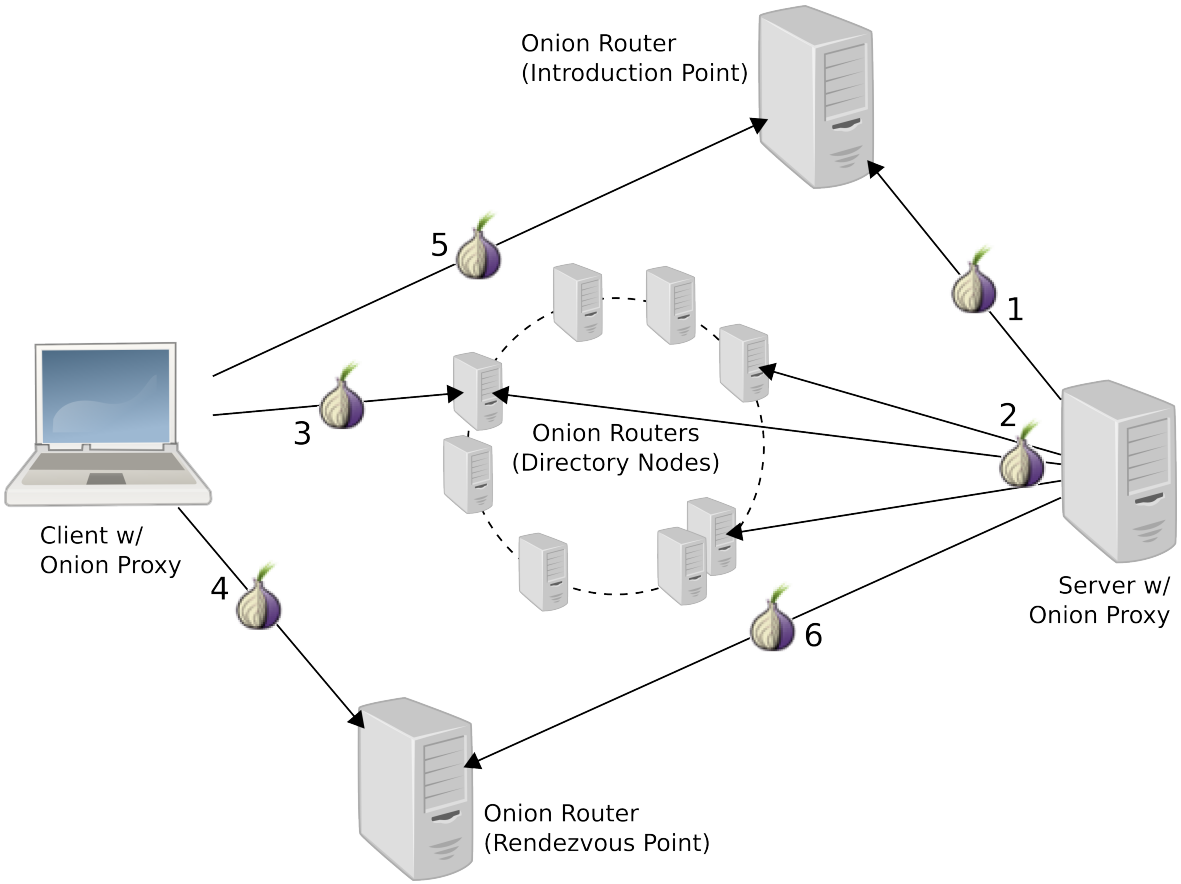
```
onion-address = h(public-key) + cookie
descriptor-id = h(h(public-key) + h(date + cookie))
descriptor-content = {
  public-key,
  h(date + cookie),
  timestamp,
  { introduction-points } encrypted with cookie
} signed with private-key
```

Improve scalability

- Current load:
  - Average of 1000 descriptors at a time
  - Average of 360 publish and 30 fetch requests per 15 minutes
  - Assumed to increase when hidden-service performance gets better
- Apply DHT-like structure
  - Equally distribute load from 3 directory nodes to $\approx$ 1000 onion routers
  - Routing table based on existing Tor router list (avoids maintenance messages)
- Use replication
  - Limit server-initiated replication to fixed number, e.g. 4, independent from possibly growing number of directory nodes
  - Replicate on consecutive nodes to resist node failures, and
  - Replicate on non-consecutive nodes to avoid attacks, e.g. black hole

Questions?

Distributed Storage of Tor Hidden Service Descriptors



Questions...