Tor: The Second-Generation Onion Router

Roger Dingledine, Nick Mathewson, Paul Syverson
Introduction

• Second Generation of Onion Routing
  – Focus on deployability
• Perfect forward secrecy
• Separation of “protocol cleaning” from anonymity
• No mixing, padding or traffic shaping
• TCP Streams can share on circuit
• Leaky-pipe circuit topology
Introduction

- Congestion control
- Directory servers
- Exit policies
- Integrity checking
- Hidden services
Design Goals

• Deployability
• Usability
• Flexibility
• Simple Design
Non-Goals

• Not P2P
• Not secure against end to end attacks
• No protocol normalization
• Not steganographic
Threat Model

• Does not protect against a global passive adversary

• Adversary can:
  – Generate, modify, delay and delete traffic
  – Operate onion routers
  – Compromise many onion routers

• Aim is to project gains traffic analysis attacks not traffic confirmation attacks
  – What do you all think of this distinction? Is it valid?
Design

- Overlay network
- Onion routers route traffic
- Onion Proxy fetches directories and creates circuits on the network
- Uses TCP
- All data is sent in fixed size cells

<table>
<thead>
<tr>
<th>CircID</th>
<th>CMD</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CircID</th>
<th>Relay</th>
<th>StreamID</th>
<th>Digest</th>
<th>Len</th>
<th>CMD</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Circuits

• Describes the Onion Routers on the path
• Can be used by many TCP streams
• Built incrementally
Building a circuit

Create $c_1$, $E(g^{x_1})$

Created $c_1$, $g^{y_1}, H(K_1)$

Relay $c_1$(Extended, $g^{y_2}, H(K_2)$)

Create $c_2$, $E(g^{x_2})$

Created $c_2$, $g^{y_2}, H(K_2)$
Last onion router should get the IP address of Bob’s website to protect Alice’s anonymity.
Additional functionality

• Integrity checking
  – Only done at the edges of a stream
  – SHA-1 digest of data sent and received
  – First 4 bytes of digest are sent with each message for verification

• Rate limiting
  – Uses token bucket approach
  – Interactive streams get preferential treatment
Congestion Control

• There is some concern about OR-to-OR congestion

• Circuit Level throttling
  – 2 windows keep track of relay data to be transmitted to other ORs and data transmitted out of the network
  – Windows are decremented after forwarding packets and increments on a *relay sendme* message
  – When a window reaches 0, no messages are forwarded
Congestion Control

• Stream Level Throttling
  – Streams have packaging windows associated with them
  – The window is decremented as messages are sent and incremented when *relay sendme* are received
  – *relay sendme* messages are sent after the TCP stream has flushed a certain number of bytes

• This congestion control method is pretty primitive. Why not leverage existing work here?
Hidden Service and Rendezvous Points

• Tor accommodates receiver anonymity by allowing location hidden services

• Design goals for location hidden services
  – Access Control
  – Robustness
  – Smear-resistance
  – Application transparency

• Location hidden service leverage rendezvous points
Creating and connecting to a Location hidden service
Other design decisions

• DoS
  – CPU consumption attacks are possible
  – Crashing routers also causes a DoS

• Exit Policies and Abuse
  – As with other systems, abuse is a big deal
  – Routers can specify exit policies restricting how they are used

• Directory Services
  – Advertises current network states and routers
Attacks and Defenses

• Passive Attacks
  – Observing user content
  – End-to-end timing correlation

• Active Attacks
  – Compromising Keys
  – Run a hostile OR

• Attacks on the Directory Service
  – Destroy directory service
  – Subvert 1 or more directory servers

• Attacks against rendezvous points
  – Make many introduction requests
  – Compromise a rendezvous point
Tor in the wild

• There is a current deployment of Tor
• Currently ~ 350 Tor routers
• ~ 40MB read and write at any given time
• Performance
  – 42% increase in time for large file
  – Varied for interactive sessions
Discussion Questions

• In Tor, if your entry node and exit node are compromised, you are sunk. If this is the case, what is the point of circuits with more than 2 hops?

• One of the tensions for Tor (and other anonymity systems) is that they need a good user base to improve the system. However, the anonymity the offer isn’t great. How do you get people to use such a system?