

Tor:
**Anonymous Communications for
the Dept of Defense...and you.**

**Roger Dingledine
The Free Haven Project**

<http://tor.eff.org/>

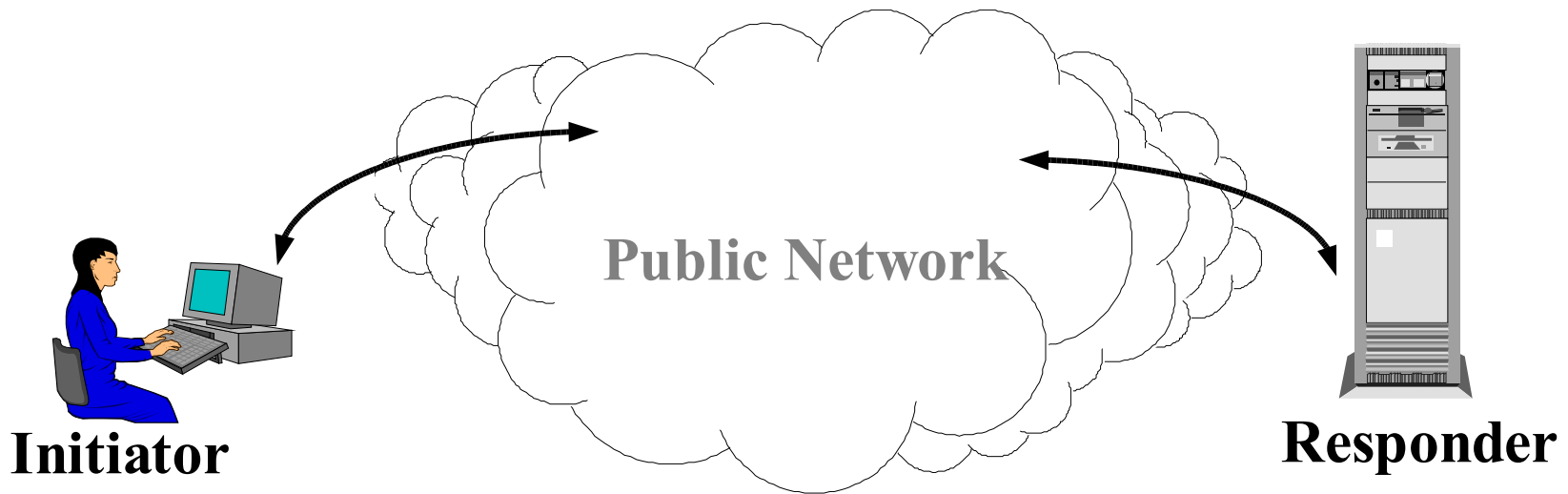
April 12, CFP 2005

Talk Outline

- ◆ Motivation: Why anonymous communication?
 - Myth 1: This is only for privacy nuts.
 - Myth 2: This stuff enables criminals.
- ◆ Tor design overview
- ◆ Hidden servers and rendezvous points
- ◆ Policy issues raised
- ◆ Open technical issues and hard problems

Public Networks are Vulnerable to Traffic Analysis

- ◆ In a Public Network (Internet):
- ◆ Packet (message) headers identify recipients
- ◆ Packet routes can be tracked



Encryption does *not* hide routing information.

Who Needs Anonymity?

- ◆ Journalists, Political Dissidents, Whistleblowers
- ◆ Censorship resistant publishers/readers
- ◆ Socially sensitive communicants:
 - Chat rooms and web forums for abuse survivors, people with illnesses
- ◆ Law Enforcement:
 - Anonymous tips or crime reporting
 - Surveillance and honeypots (sting operations)
- ◆ Corporations:
 - Hiding collaborations of sensitive business units or partners
 - Hiding procurement suppliers or patterns
 - Competitive analysis

Who Needs Anonymity?

- ◆ You:
 - Where are you sending email (who is emailing you)
 - What web sites are you browsing
 - Where do you work, where are you from
 - What do you buy, what kind of physicians do you visit, what books do you read, ...

Who Needs Anonymity?

- ◆ Government

Government Needs Anonymity?

Yes, for...

- ◆ Open source intelligence gathering
 - Hiding individual analysts is not enough
 - That a query was from a govt. source may be sensitive
- ◆ Defense in depth on open and *classified* networks
 - Networks with only cleared users (but a million of them)
- ◆ Dynamic and semitrusted international coalitions
 - Network can be shared without revealing existence or amount of communication between all parties

Anonymity Loves Company

- ◆ You can't be anonymous by yourself
 - *Can* have confidentiality by yourself
- ◆ A network that protects only DoD network users won't hide that connections from that network are from Defense Dept.
- ◆ You must carry traffic for others to protect yourself
- ◆ But those others don't want to trust their traffic to just one entity either. Network needs *distributed trust*.
- ◆ Security depends on diversity and dispersal of network.

Who Needs Anonymity?

- ◆ And yes criminals

Who Needs Anonymity?

- ◆ And yes criminals

But they already have it.

We need to protect everyone else.

Anonymous From Whom? Adversary Model

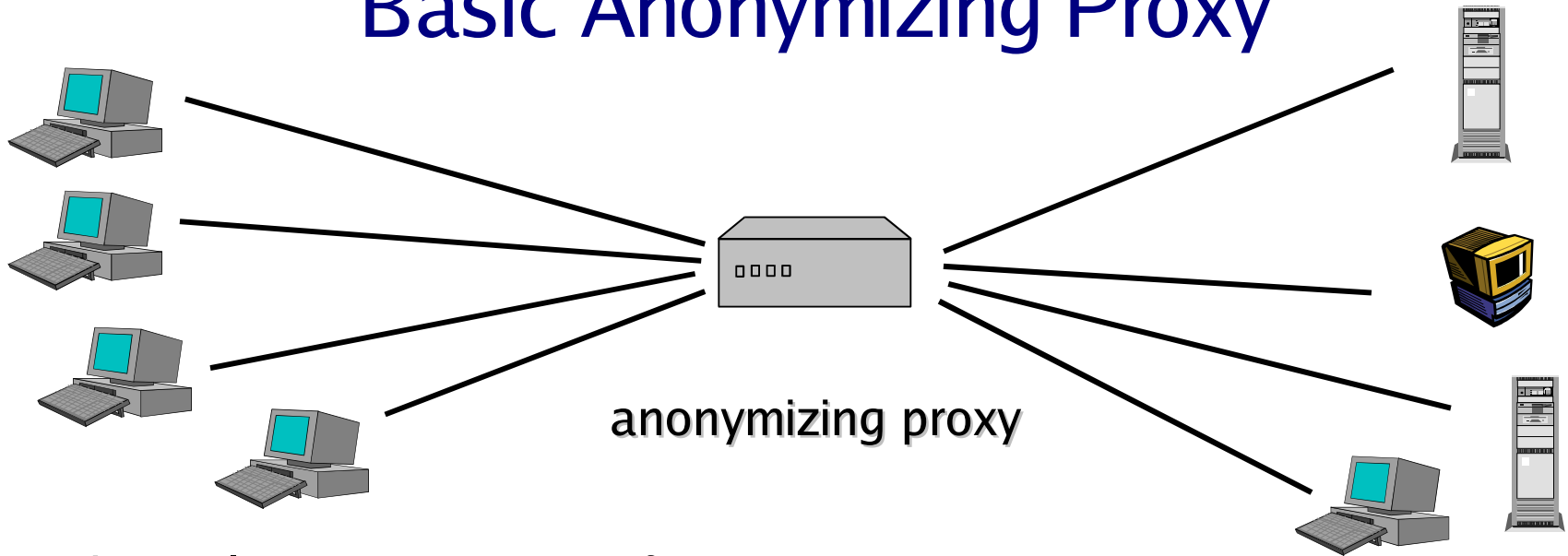
- ◆ Recipient of your message
- ◆ Sender of your message
- => Need Channel and Data Anonymity

- ◆ Observer of network from outside
- ◆ Network Infrastructure (Insider)
- => Need Channel Anonymity

- ◆ Note: Anonymous authenticated communication makes perfect sense
- ◆ Communicant identification should be inside the basic channel, not a property of the channel

Focus of Tor is anonymity of the
communication pipe,
not what goes through it

Basic Anonymizing Proxy



- Channels appear to come from proxy, **not** true originator
- Appropriate for Web connections, etc.:
SSL, TLS, SSH (lower cost symmetric encryption)
- Example: The Anonymizer
- Advantages: Simple, Focuses lots of traffic for more(?) anonymity
- **Main Disadvantage: Single point of failure, compromise, attack**

Onion Routing

Traffic Analysis Resistant Infrastructure

- ◆ Main Idea: Combine Advantages of mixes and proxies
- ◆ Use (expensive) public-key crypto to establish circuits
- ◆ Use (cheaper) symmetric-key crypto to move data
 - Like SSL/TLS based proxies
- ◆ Distributed trust like mixes
- ◆ Related Work (some implemented, some just designs):
 - ISDN Mixes
 - Crowds, JAP Webmixes, Freedom Network
 - Tarzan, Morphmix

Tor

Tor

The Onion Routing

Tor

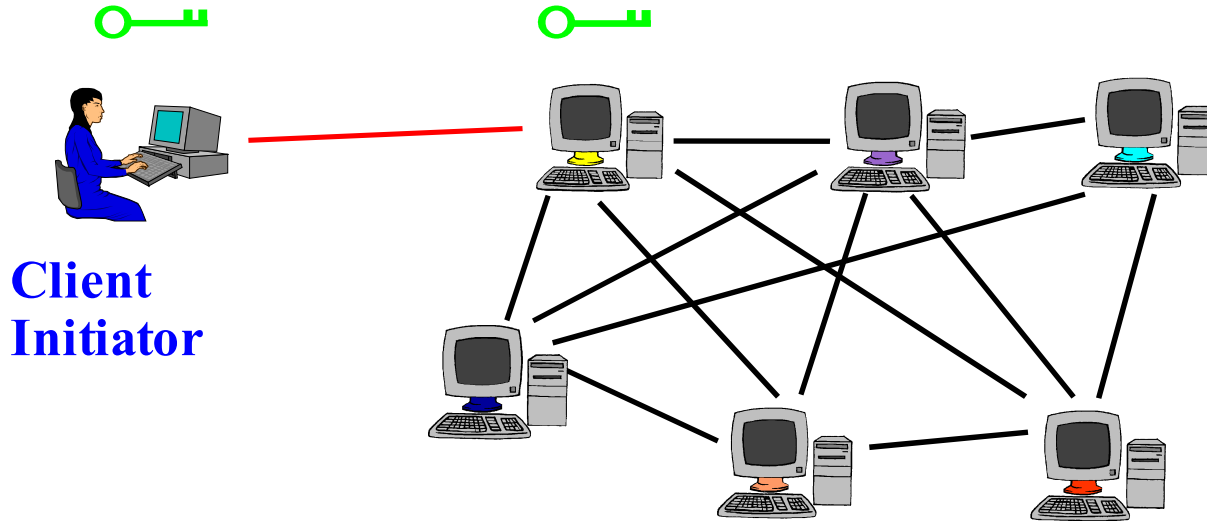
Tor's Onion Routing

Numbers and Performance

- ◆ Running since October 2003
- 100+ nodes on four continents (North America, Europe, Asia, Australia)
- Ten thousand+ (?) users
- Nodes process 1-90 GB / day application cells
- Network has never been down

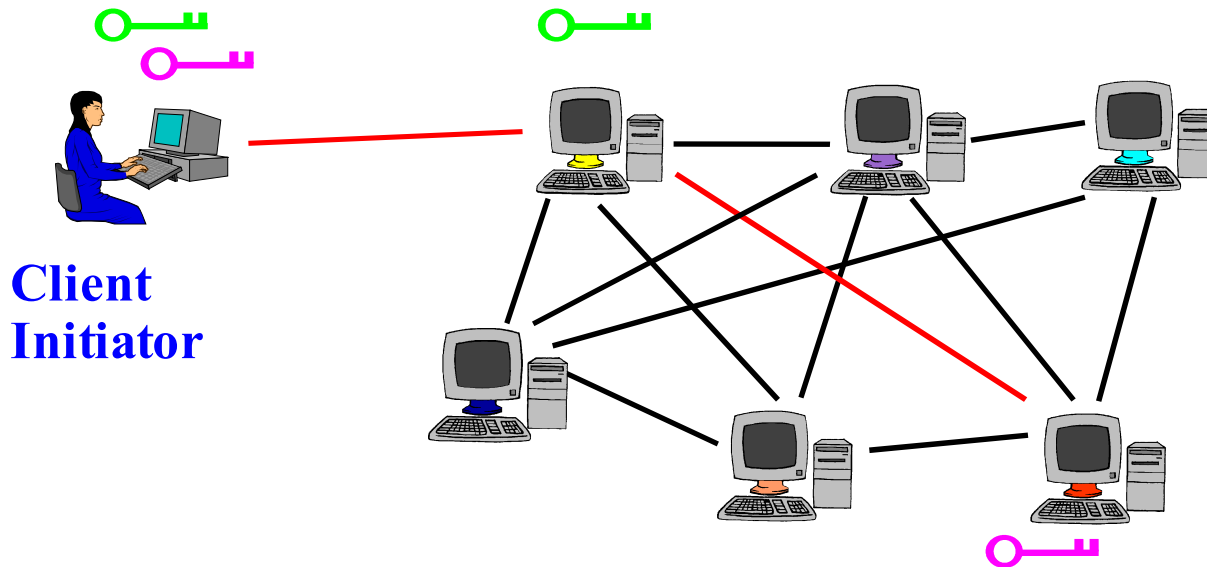
Tor Circuit Setup

- Client Proxy establishes session key + circuit w/ **Onion Router 1**



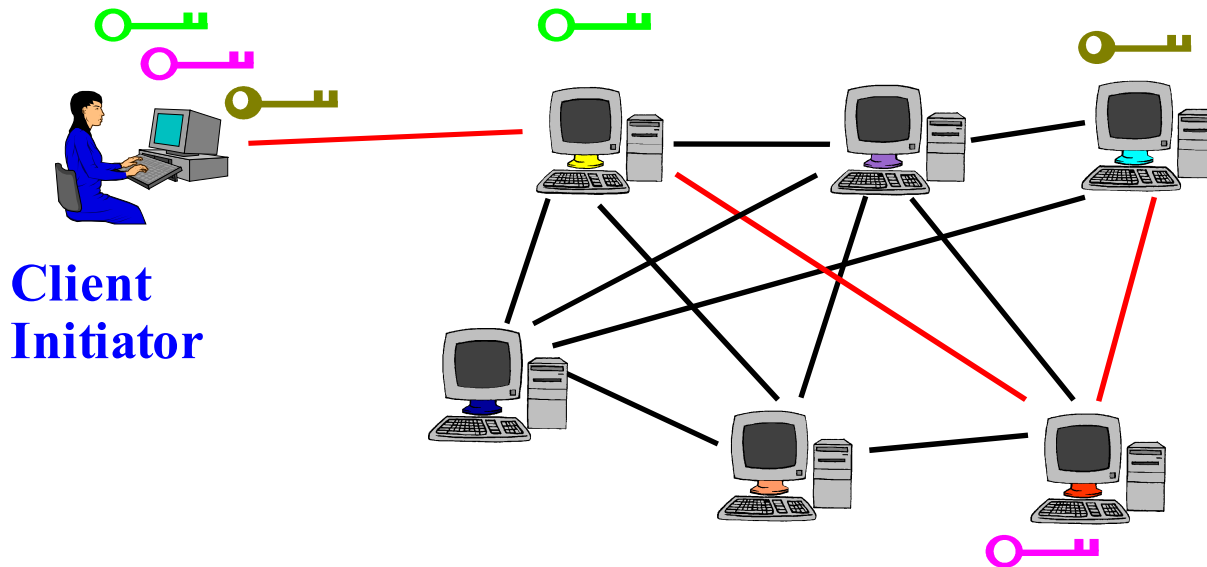
Tor Circuit Setup

- Client Proxy establishes session key + circuit w/ **Onion Router 1**
- Proxy tunnels through that circuit to extend to **Onion Router 2**



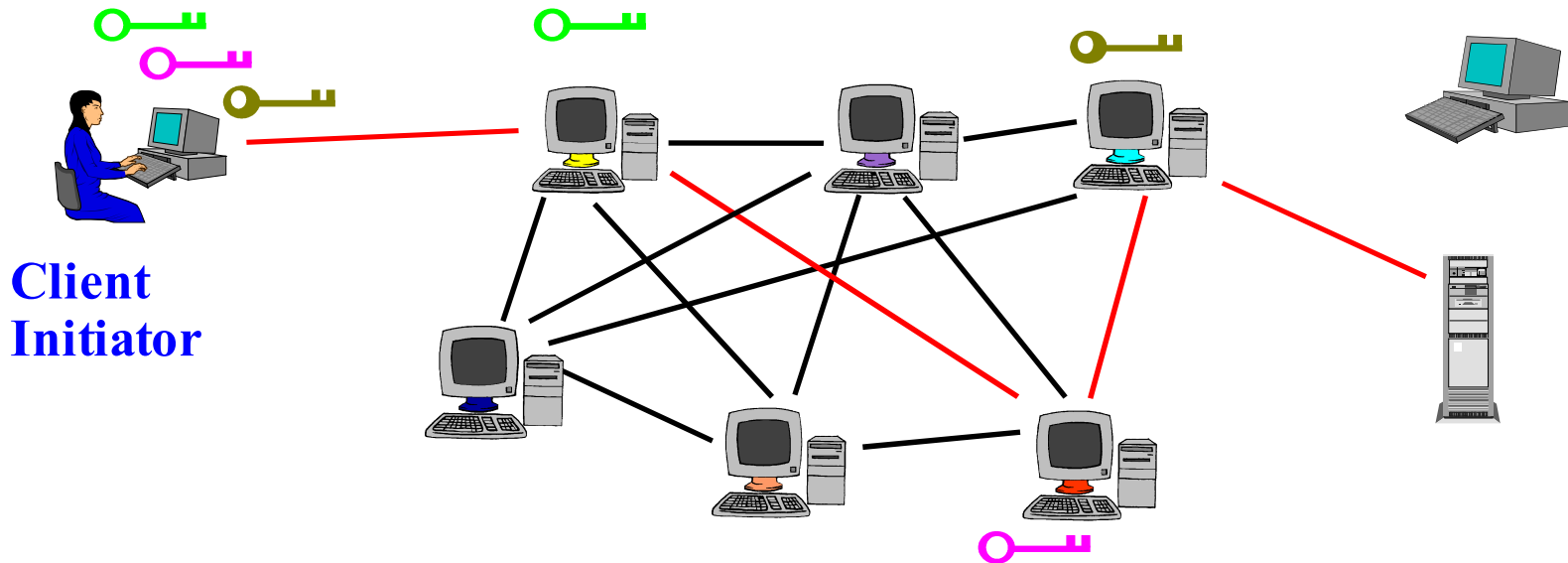
Tor Circuit Setup

- Client Proxy establishes session key + circuit w/ **Onion Router 1**
- Proxy tunnels through that circuit to extend to **Onion Router 2**
- Etc



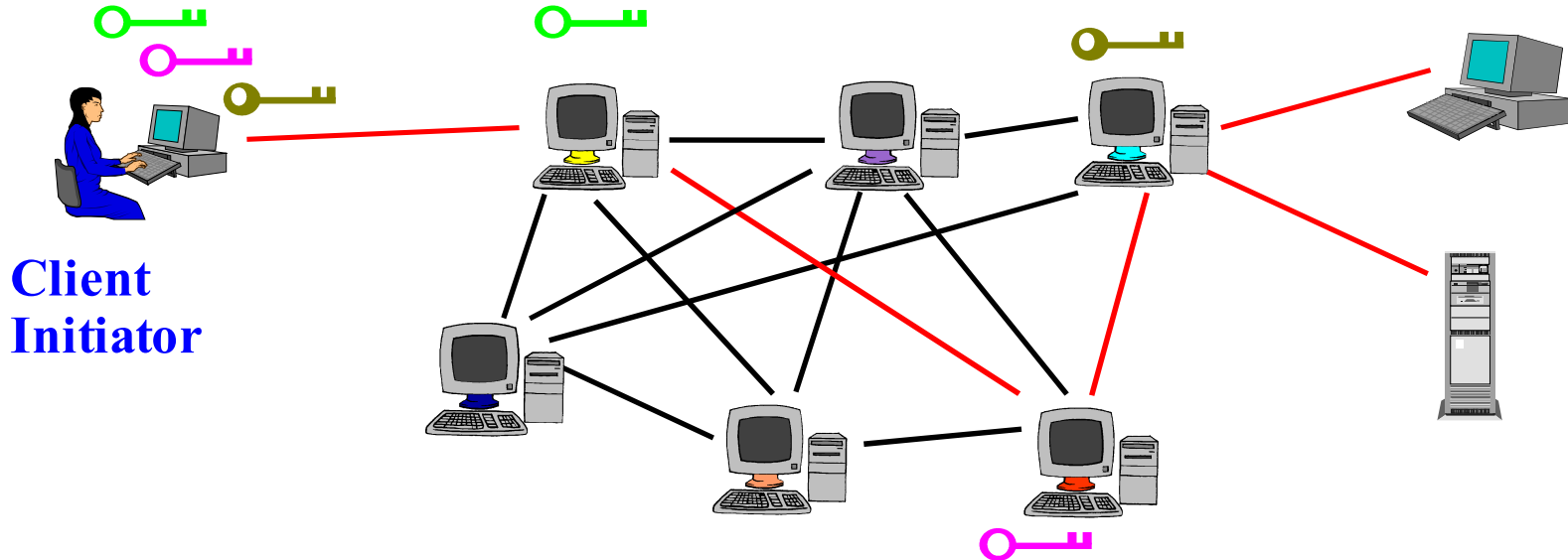
Tor Circuit Usage

- Client Proxy establishes session key + circuit w/ **Onion Router 1**
- Proxy tunnels through that circuit to extend to **Onion Router 2**
- Etc
- Client applications connect and communicate over Tor circuit



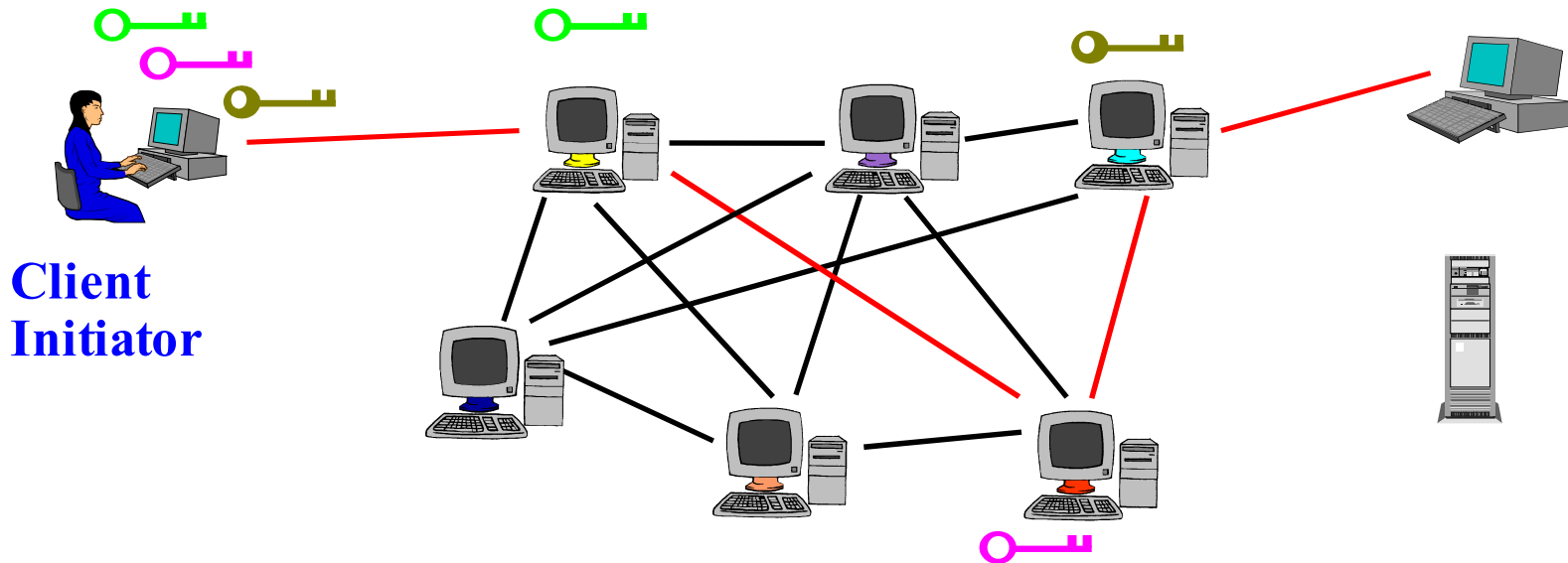
Tor Circuit Usage

- Client Proxy establishes session key + circuit w/ **Onion Router 1**
- Proxy tunnels through that circuit to extend to **Onion Router 2**
- Etc
- Client applications connect and communicate over Tor circuit



Tor Circuit Usage

- Client Proxy establishes session key + circuit w/ **Onion Router 1**
- Proxy tunnels through that circuit to extend to **Onion Router 2**
- Etc
- Client applications connect and communicate over Tor circuit



Where do I go to connect to the network?

- ◆ Directory Servers
 - Maintain list of which onion routers are up, their locations, current keys, exit policies, etc.
 - Directory server keys ship with the code
 - Control which nodes can join network
 - Important to guard against Sybil attack and related problems
 - These directories are cached and served by other servers, to reduce bottlenecks

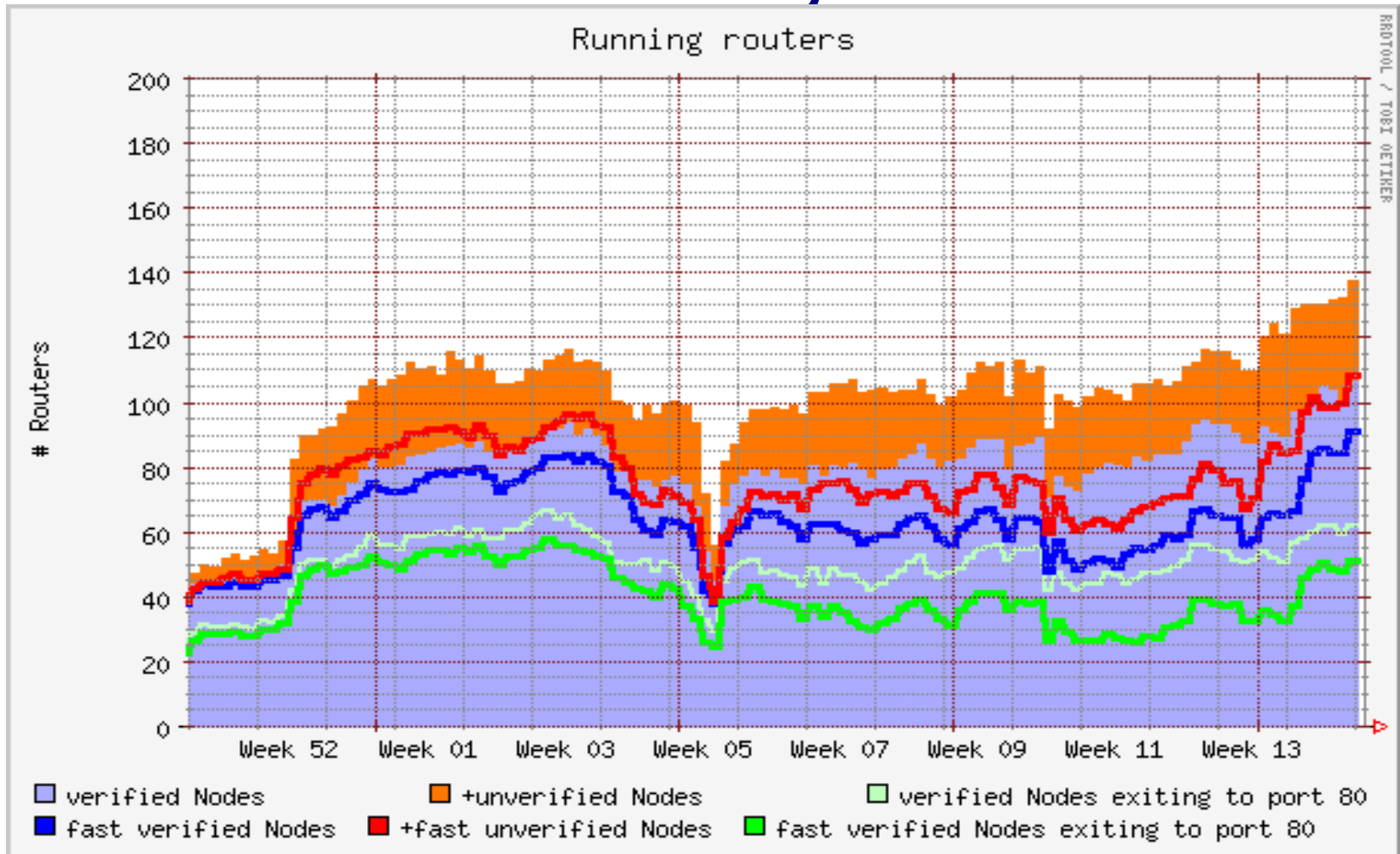
Some Tor Properties

- ◆ Simple modular design, restricted ambitions.
 - ~30K lines of C code
 - Even servers run in user space, no need to be root
 - Flexible exit policies, each node chooses what applications/destinations can emerge from it

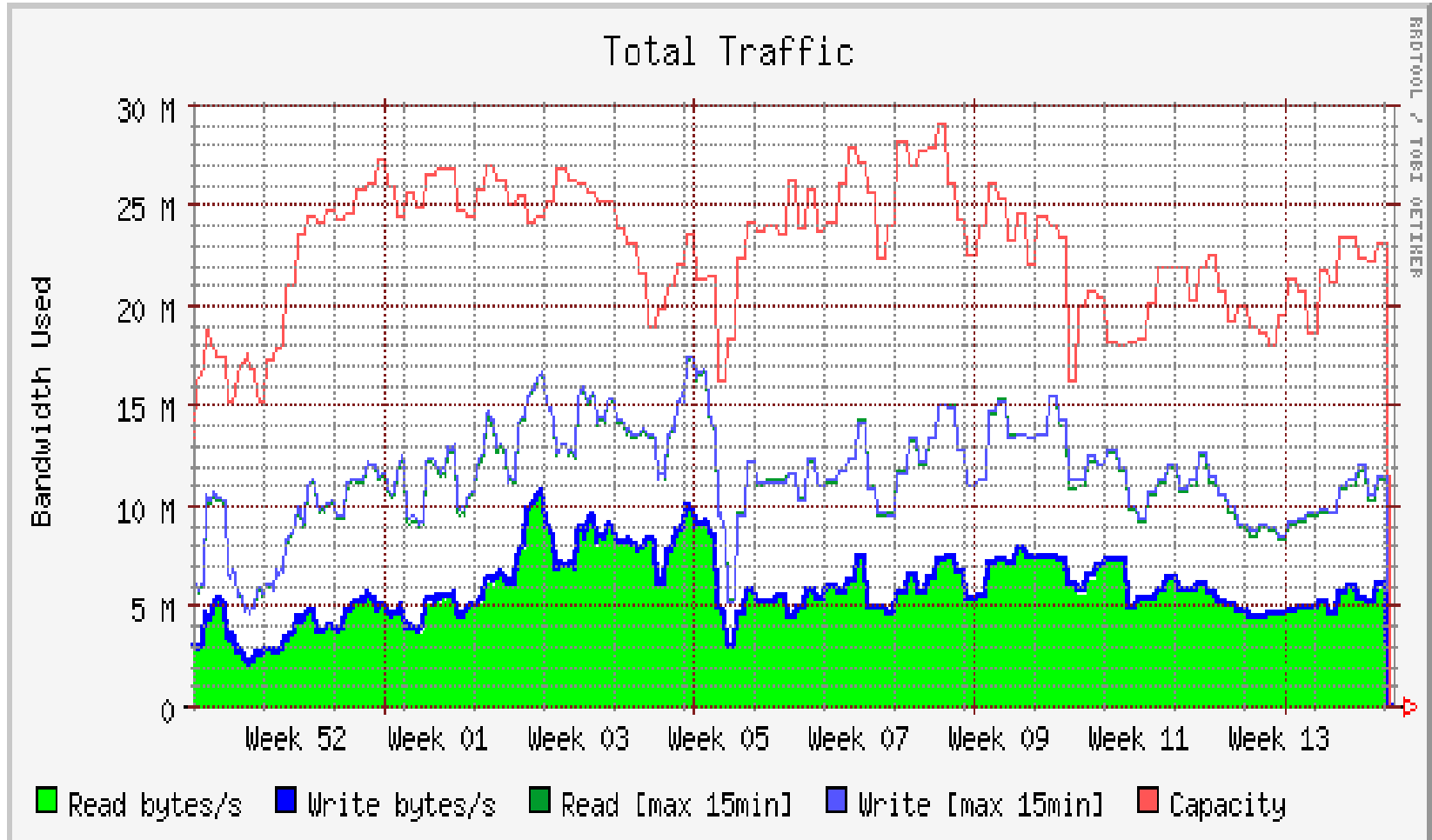
Some Tor Properties

- ◆ Lots of supported platforms:
 - Linux, BSD, MacOS X, Solaris, Windows, ...
- ◆ Deployment paradigm:
 - Volunteer server operators
 - No payments, not proprietary
 - Moving to a P2P incentives model

Number of running Tor servers



Total traffic through Tor network



Location Hidden Servers

- ◆ Alice can connect to Bob's server without knowing where it is or possibly who he is
- ◆ Can provide servers that
 - Are accessible from anywhere
 - Resist censorship
 - Require minimal redundancy for resilience in denial of service (DoS) attack
 - Can survive to provide selected service even during full blown distributed DoS attack
 - Resistant to physical attack (you can't find them)

Get the Code, Run a Node! (or just surf the web anonymously)

- ◆ Current code freely available (3-clause BSD license)
- ◆ Comes with a specification – the JAP team in Dresden implemented a compatible Tor client in Java
- ◆ Design paper, system spec, code, see the list of current nodes, etc.
- ◆ <http://tor.eff.org/>

Policy issues

- ◆ Spam / spam blacklists
- ◆ Wikipedia
- ◆ Internet Relay Chat (IRC)

- ◆ Good time for anonymous credentials?

Tradeoffs

- ◆ Low-latency (Tor) vs. high-latency (Mixminion)
- ◆ Packet-level vs stream-level capture
- ◆ Padding vs. no padding (mixing, traffic shaping)
- ◆ UI vs. no UI
- ◆ AS-level paths and proximity issues
- ◆ Incentives to run servers / allow exits
- ◆ Enclave-level onion routers / proxies / helper nodes
- ◆ Path length? (3 hops, don't reuse nodes)
- ◆ China?
- ◆ P2P network vs. static network