U.S. NAVAL
RESEARCH
LABORATORY

The Tor Network: Freedom and Privacy Online

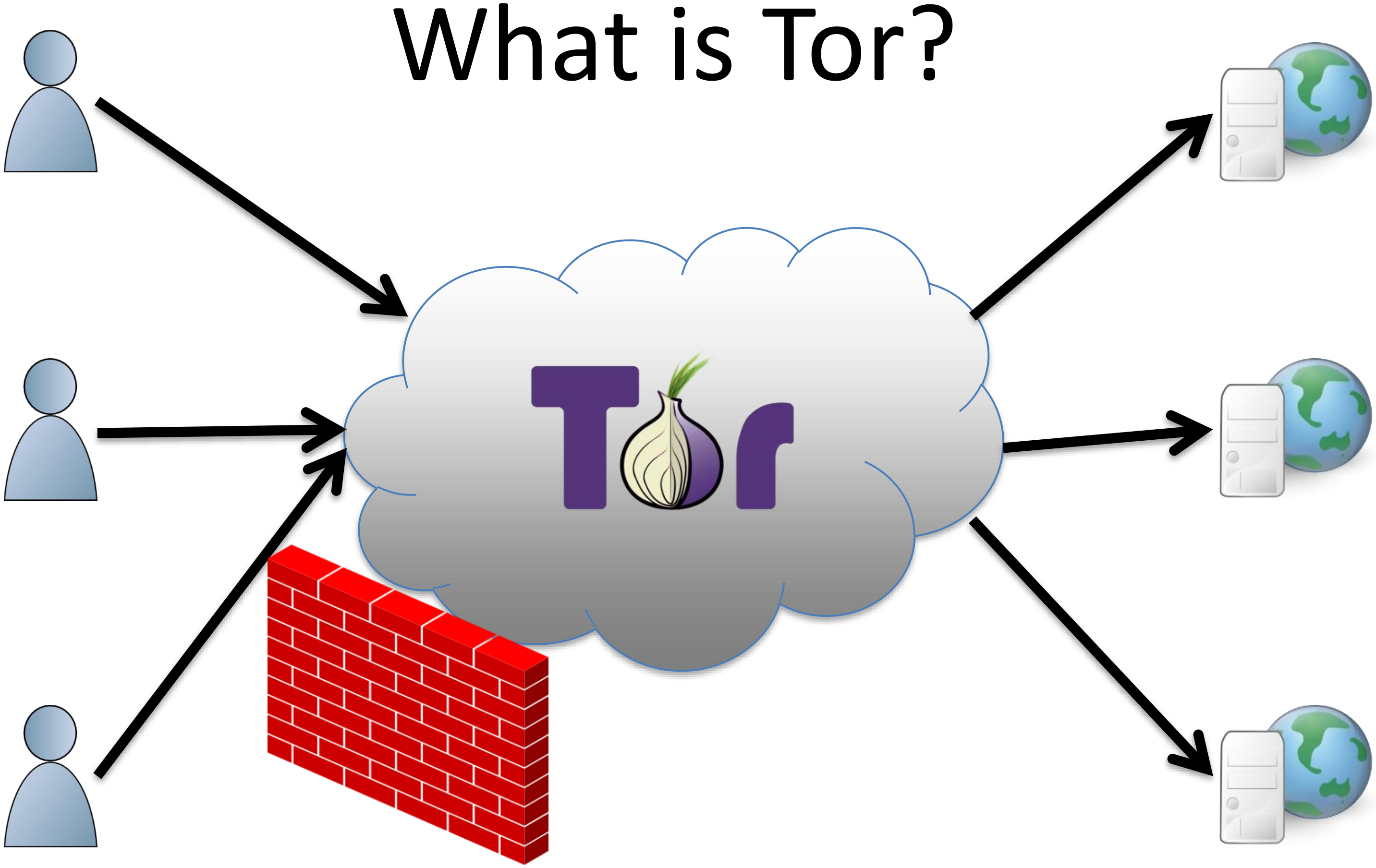
Aaron Johnson

U.S. Naval Research Laboratory

June 19th, 2018

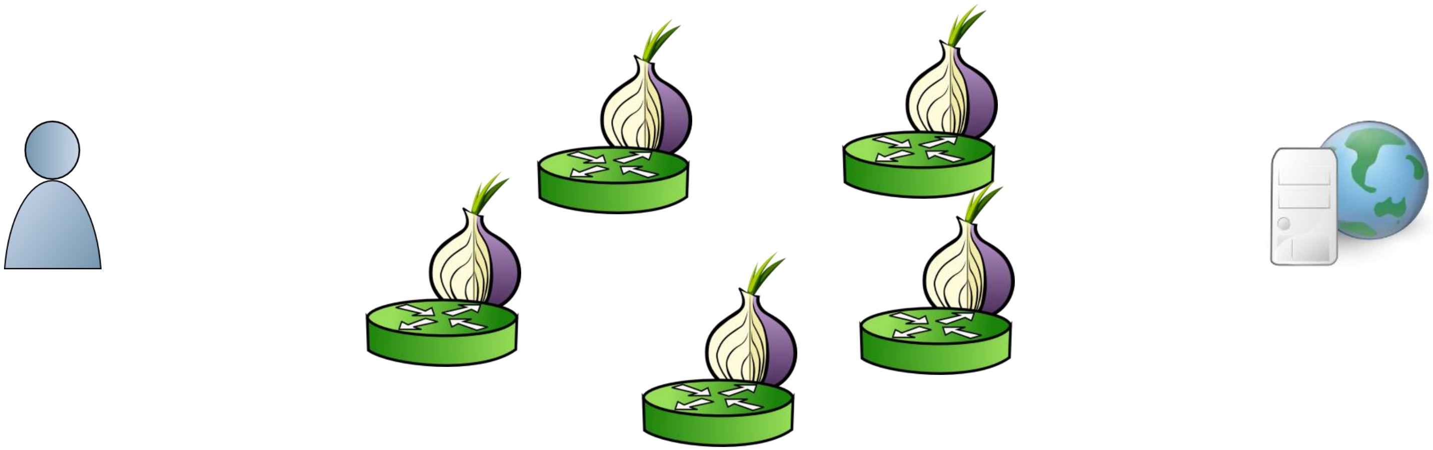
Overview

What is Tor?



Tor is a system for anonymous communication and censorship circumvention.

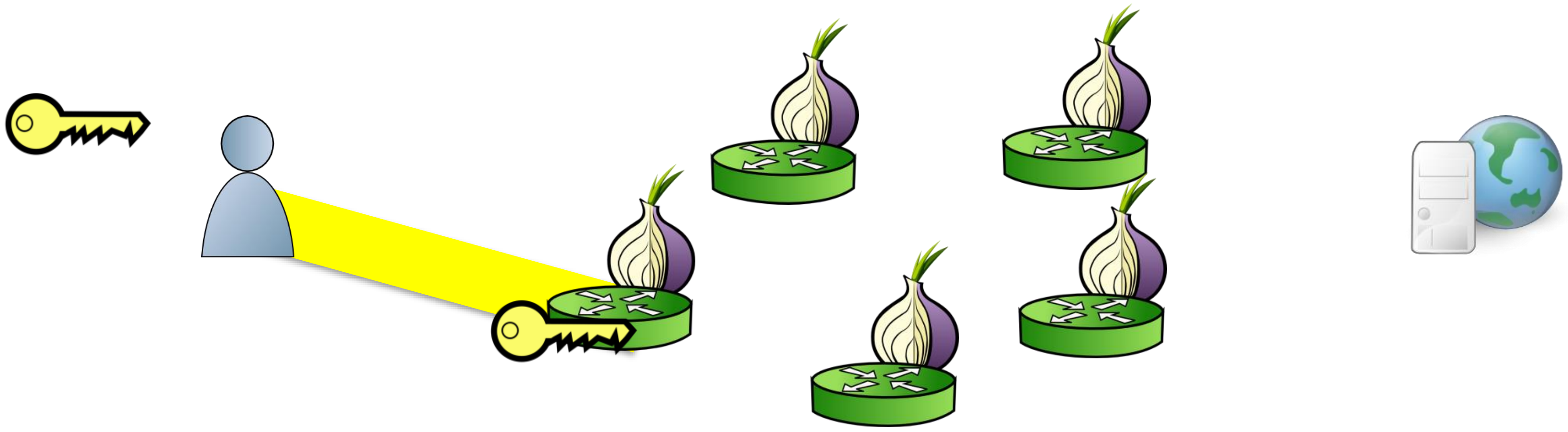
What is Tor?



Users　　Onion Routers　　Destinations

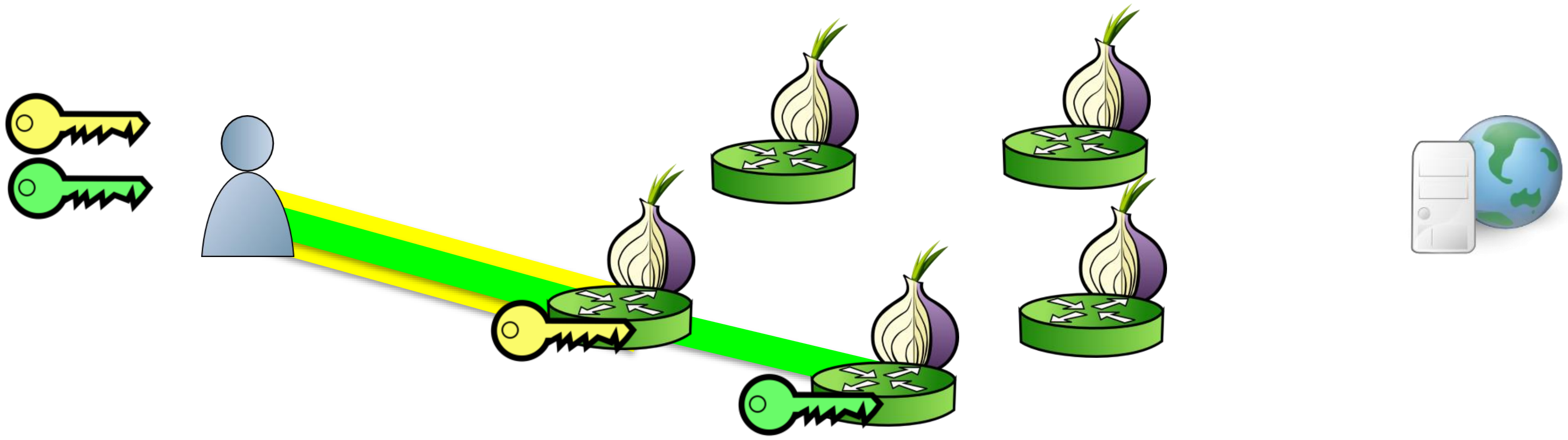
Tor is based on *onion routing*.

What is Tor?



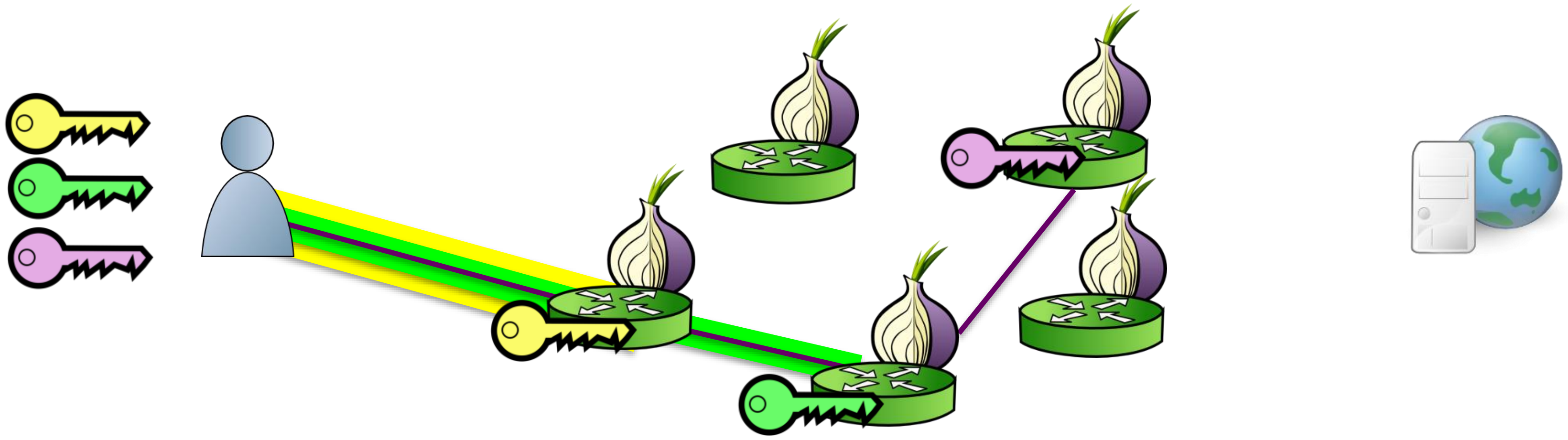
Tor is based on *onion routing*.

What is Tor?



Tor is based on *onion routing*.

What is Tor?



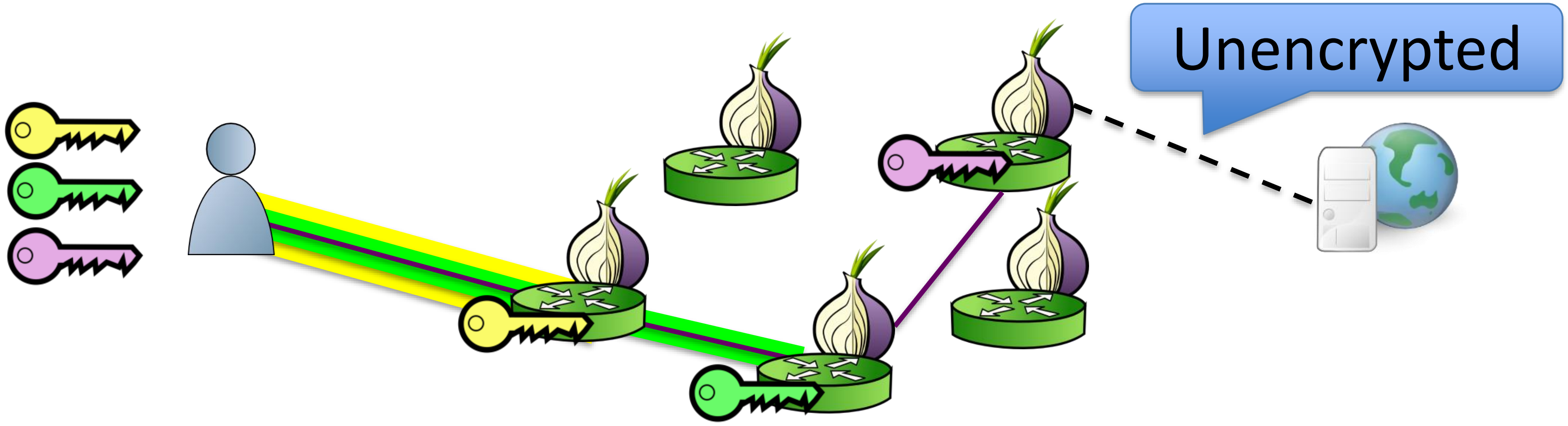
Users

Onion Routers

Destinations

Tor is based on *onion routing*.

What is Tor?



Users Onion Routers Destinations

Tor is based on *onion routing*.

Tor Projects

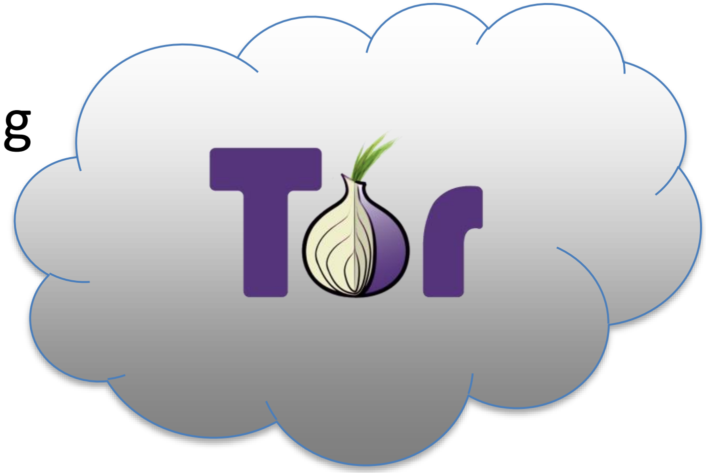
<https://www.torproject.org/projects>

- tor – Tor client, relay, and onion service
- Tor Browser – Web browser over Tor
- Orbot – Tor on Android
- The Amnesic Incognito Live System (Tails)
- Open Observatory of Network Interference

Motivation

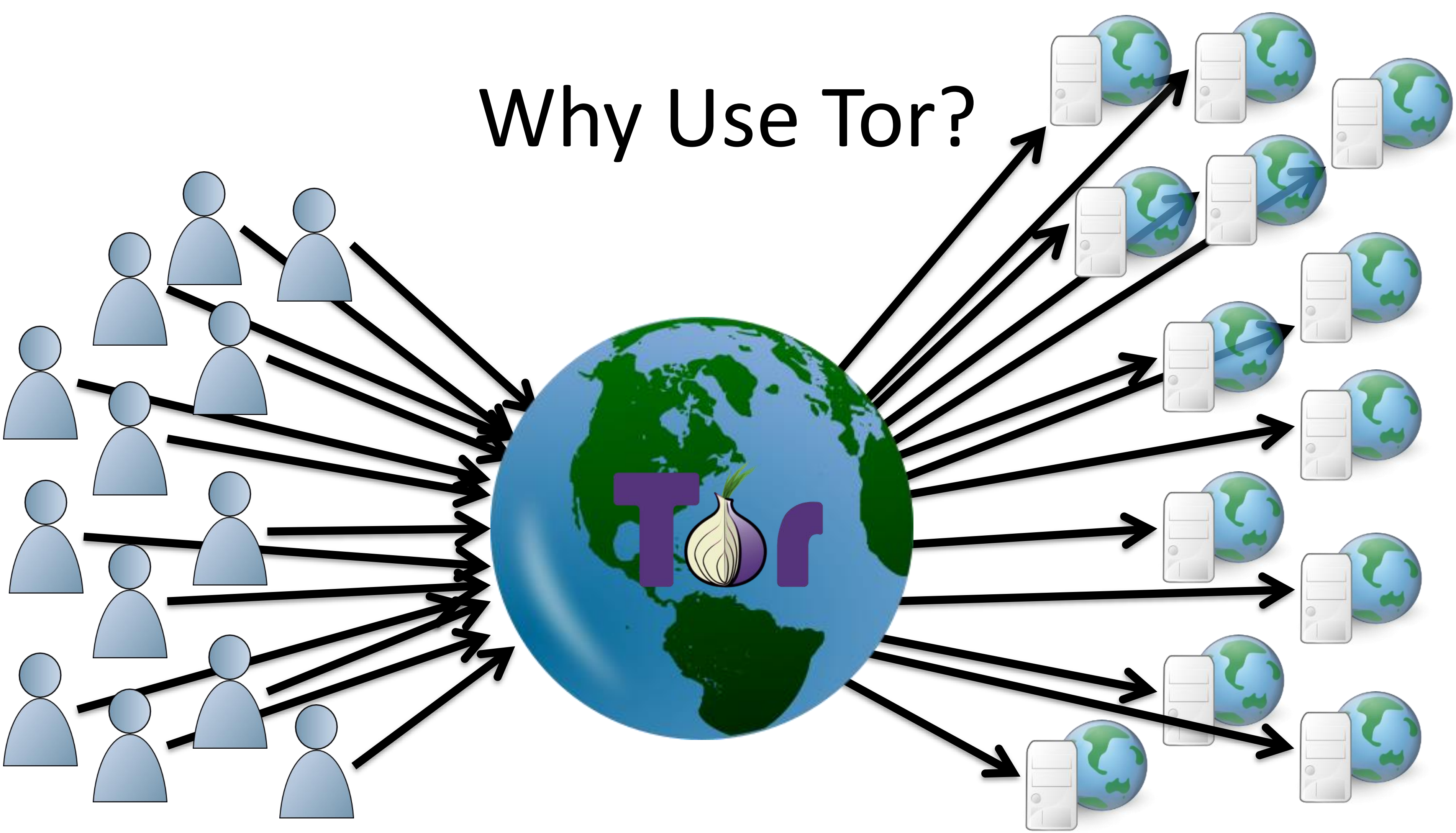
Why Use Tor?

- Individuals avoiding censorship
- Individuals avoiding surveillance



- Journalists protecting themselves or sources
- Law enforcement during investigations
- Intelligence analysts for gathering data

Why Use Tor?



- Over 2,000,000 daily users
- 100Gbps aggregate traffic

- Over 6000 relays in over 75 countries

Tor History

1996: “Hiding Routing Information” by David M. Goldschlag, Michael G. Reed, and Paul F. Syverson. *Information Hiding: First International Workshop*.

1997: "Anonymous Connections and Onion Routing," Paul F. Syverson, David M. Goldschlag, and Michael G. Reed. *IEEE Security & Privacy Symposium*.

1998: Distributed network of 13 nodes at NRL, NRAD, and UMD.

2000: “Towards an Analysis of Onion Routing Security” by Paul Syverson, Gene Tsudik, Michael Reed, and Carl Landwehr. *Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability.*

2003: Tor network is deployed (12 US nodes, 1 German), and Tor code is released by Roger Dingledine and Nick Mathewson under the free and open MIT license.

2004: “Tor: The Second-Generation Onion Router” by Roger Dingledine, Nick Mathewson, and Paul Syverson. *USENIX Security Symposium*.

2006: The Tor Project, Inc. incorporated as a non-profit.

Tor Today

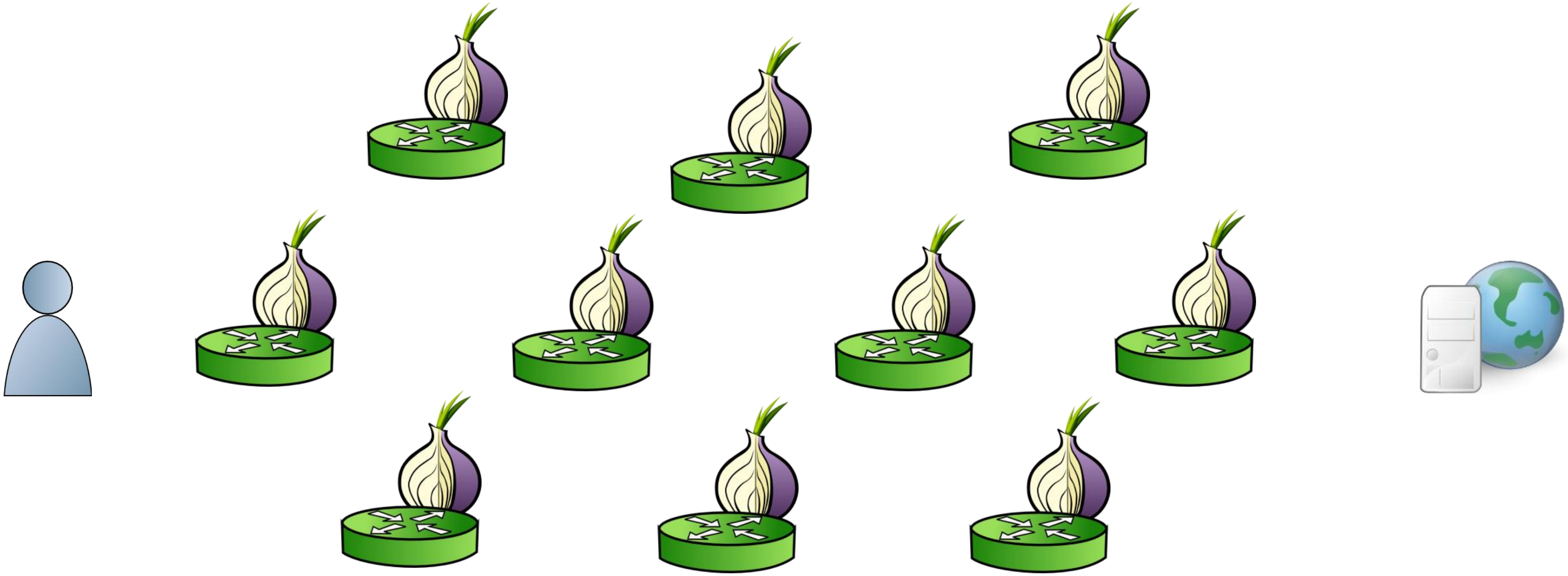
- Funding levels at $2-3 million in 2015 (current funders include Fastly, individuals, NSF, Mozilla, Open Tech. Fund, US State Dept.)
- The Tor Project, Inc. employs a team (30+ paid employees) for software development, research, office, funding, community outreach, and user support
- Much bandwidth, research, development, and outreach still contributed by third parties

Other anonymous communication designs and systems

- Dining Cryptographers network: Dissent, Herbivore
- Mix networks: MixMinion, MixMaster, BitLaundry, Riffle, Vuvuzela
- Onion routing: Aqua, Crowds, Freedom, I2P, Java Anon Proxy, PipeNet
- Privacy-focused VPNs: anonymizer.com, anonymouse.org
- Private Information Retrieval: Pynchon Gate, Pung, Riposte
- Others: Anonymous buses, XOR trees, broadcast

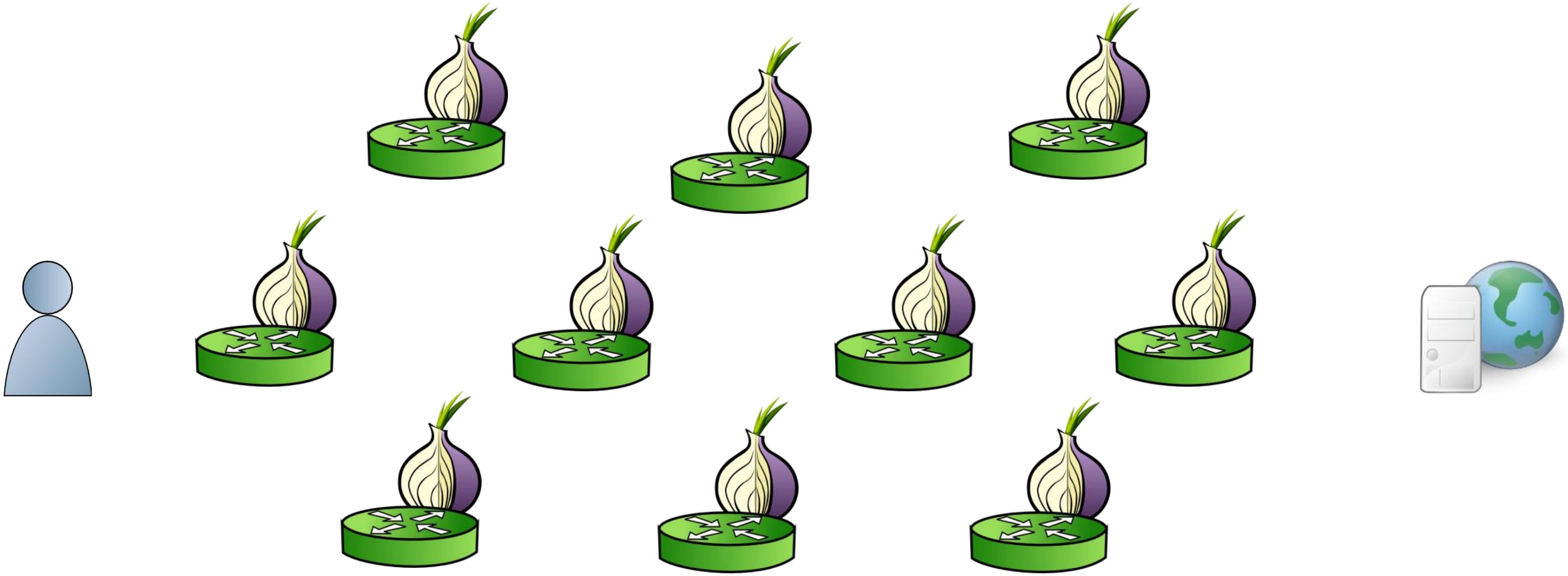
Security Model

Threat Model



Adversary is local and active.

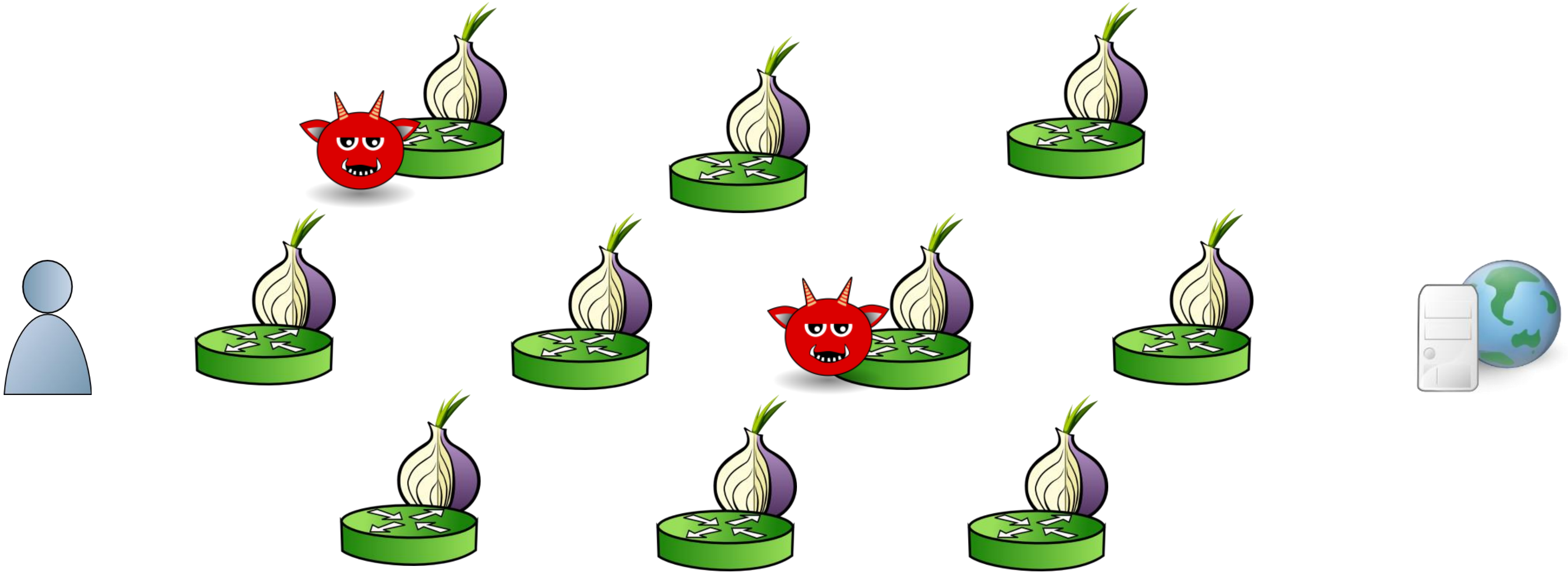
Threat Model



Adversary is local and active.

Not global

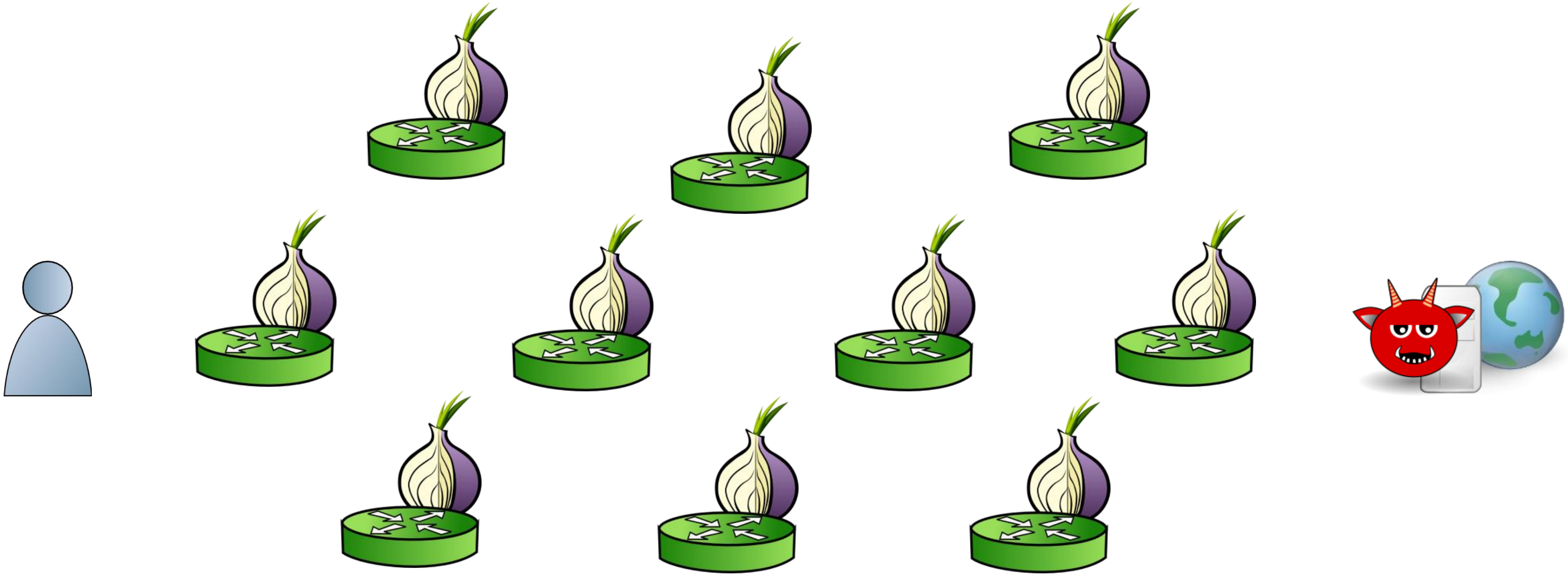
Threat Model



Adversary is local and active.

- Adversary may run relays

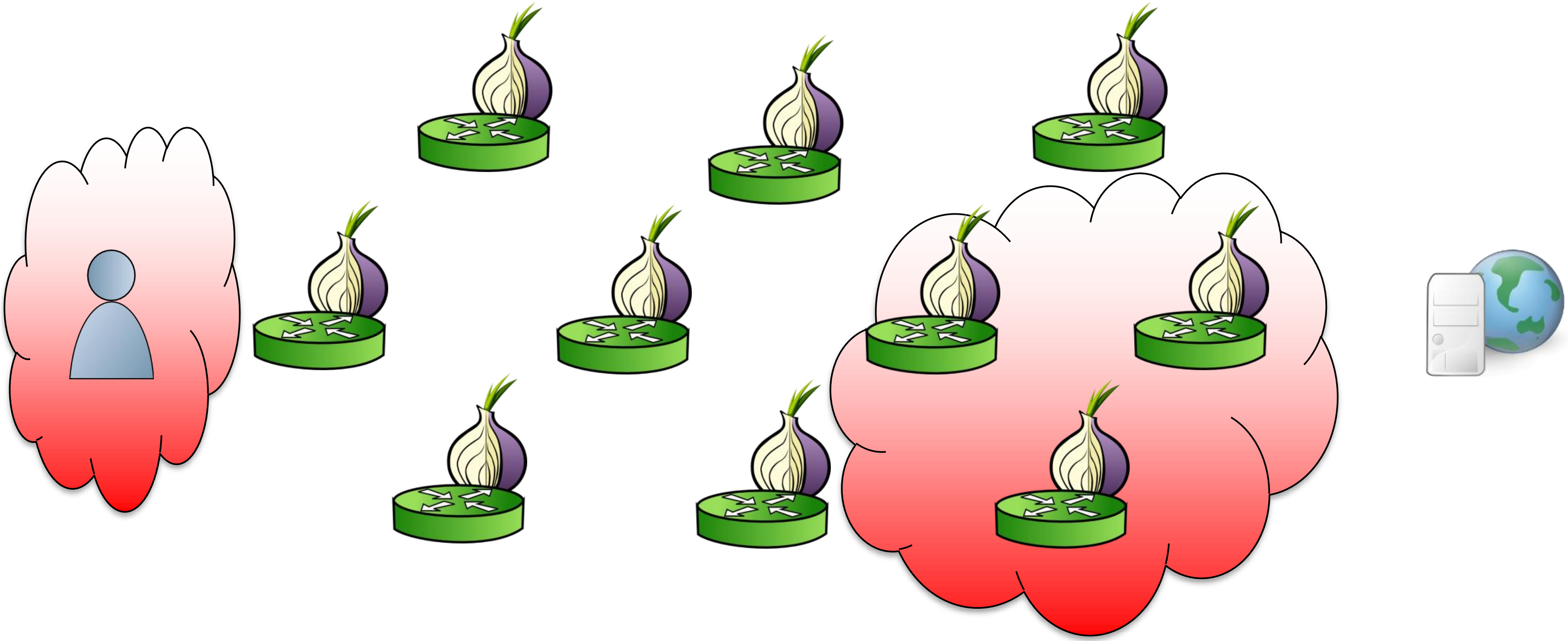
Threat Model



Adversary is local and active.

- Adversary may run relays
- Destination may be malicious

Threat Model



Adversary is local and active.

- Adversary may run relays
- Destination may be malicious
- Adversary may observe some ISPs

Security Definitions

- Identity is primarily IP address but can include other identifying information
- *Sender anonymity*: Connection initiator cannot be determined
- *Receiver anonymity*: Connection recipient cannot be determined
- *Unobservability*: It cannot be determined who is using the system.

Design

General Tor Functionality

- Provides connection-oriented bidirectional communication
- Only makes TCP connections
- Provides standard SOCKS interface to applications
- Provides application-specific software for some popular applications (e.g. HTTP)

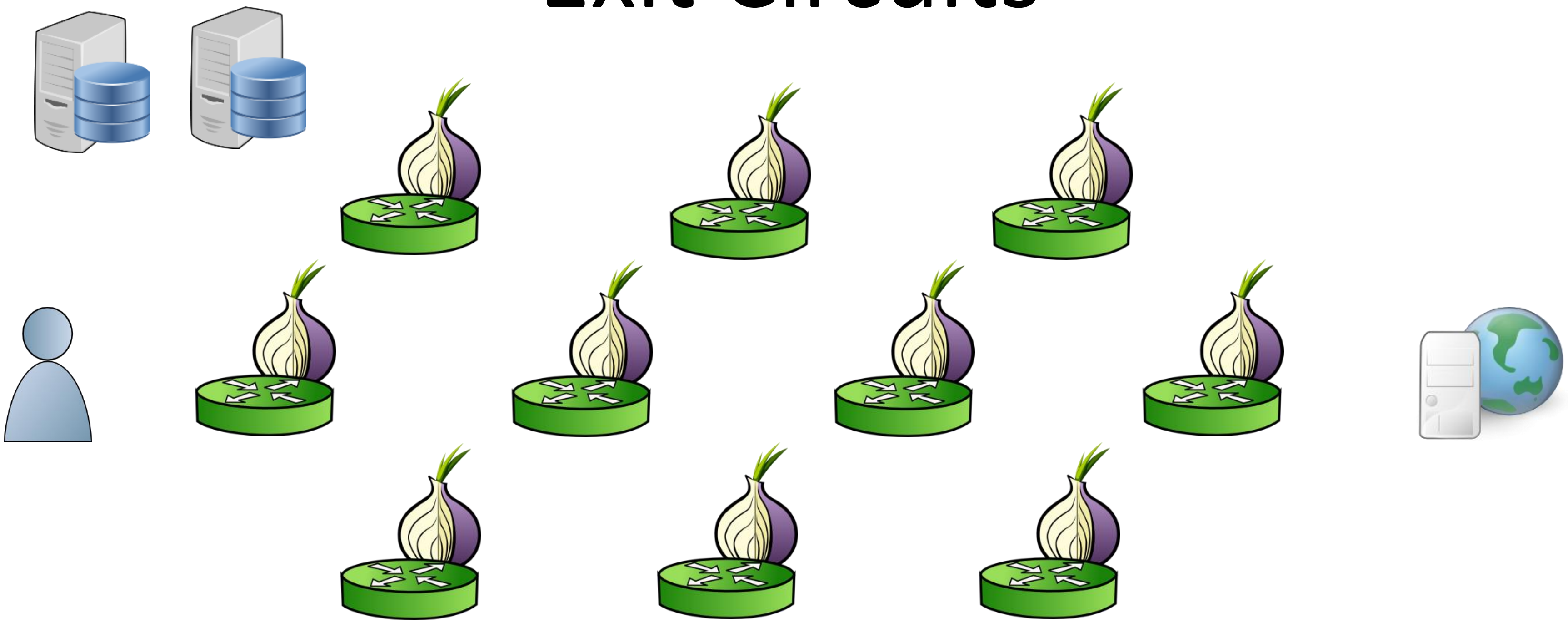
Tor Protocols

1. Exit circuits (*anonymity wrt all but sender*)
2. Onion services (*anonymity wrt all*)
3. Censorship circumvention (*unobservability*)

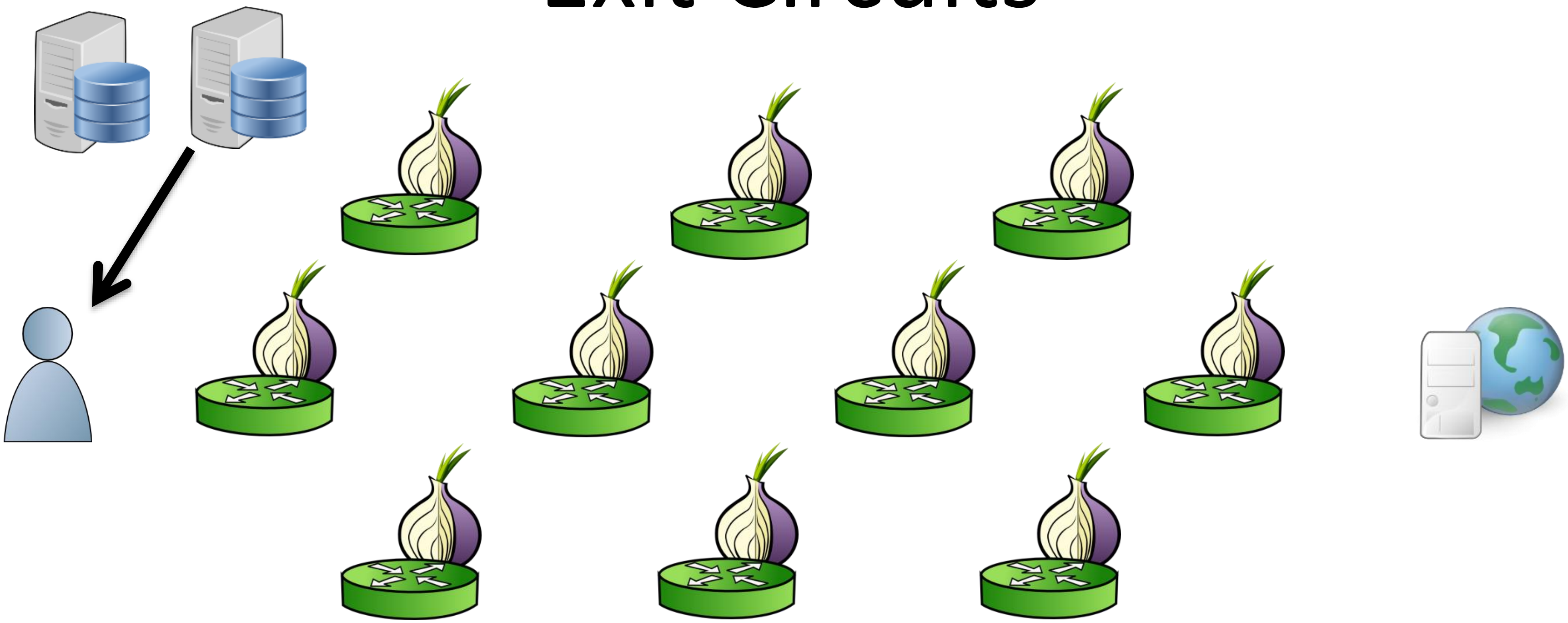
Tor Protocols

1. **Exit circuits (*anonymity wrt all but sender*)**
2. Onion services (*anonymity wrt all*)
3. Censorship circumvention (*unobservability*)

Exit Circuits

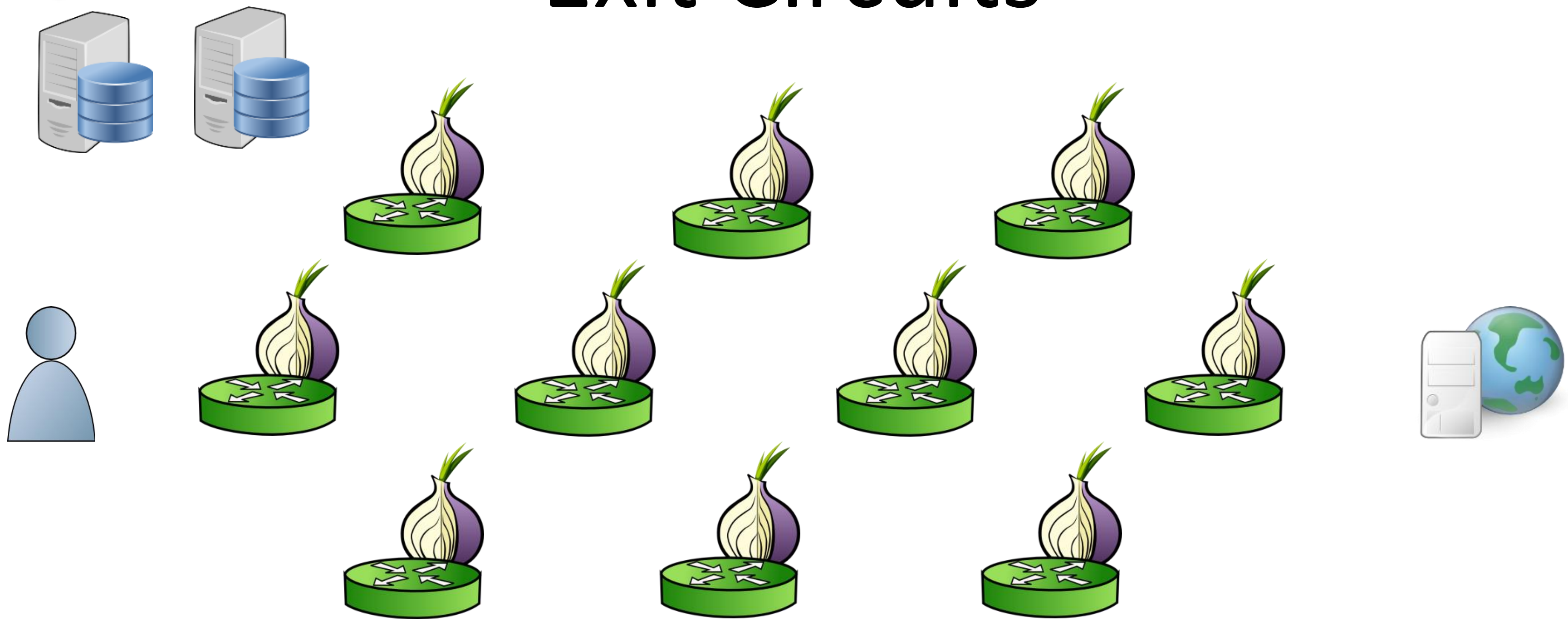


Exit Circuits



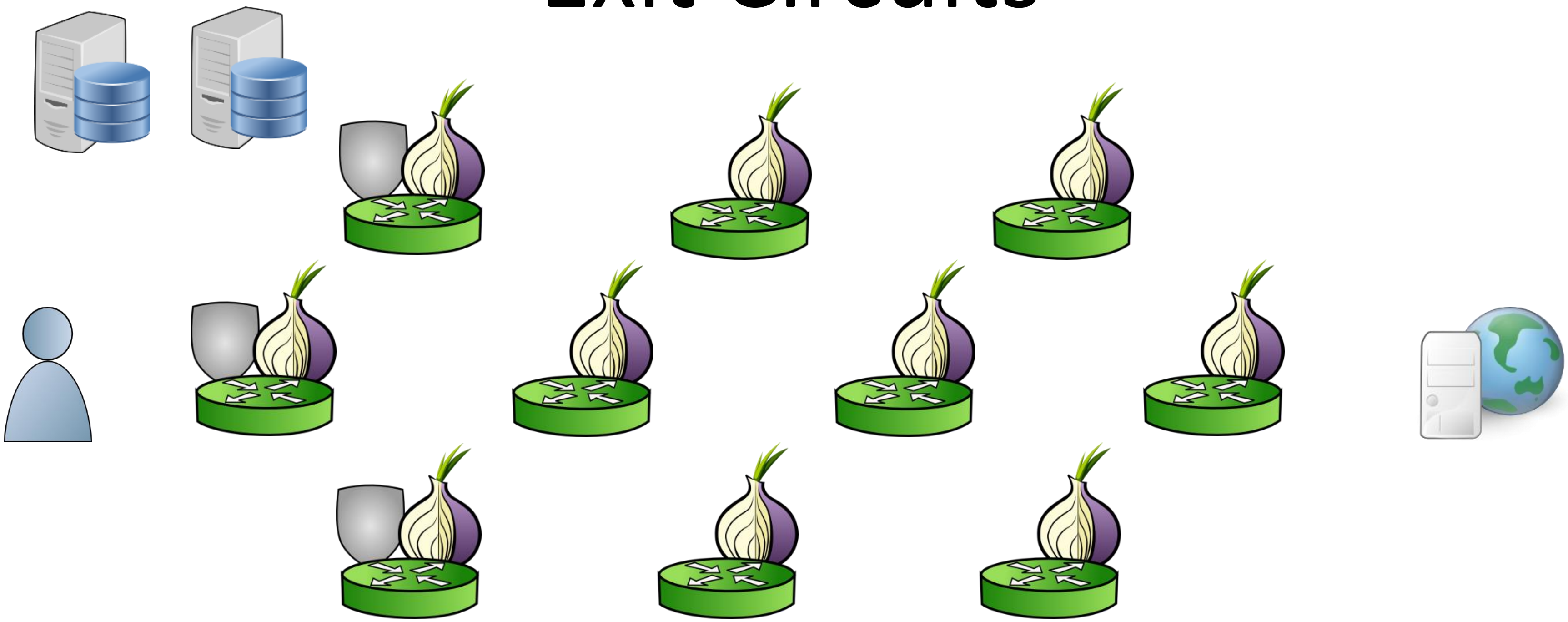
1. Client learns about relays from a directory server.

Exit Circuits



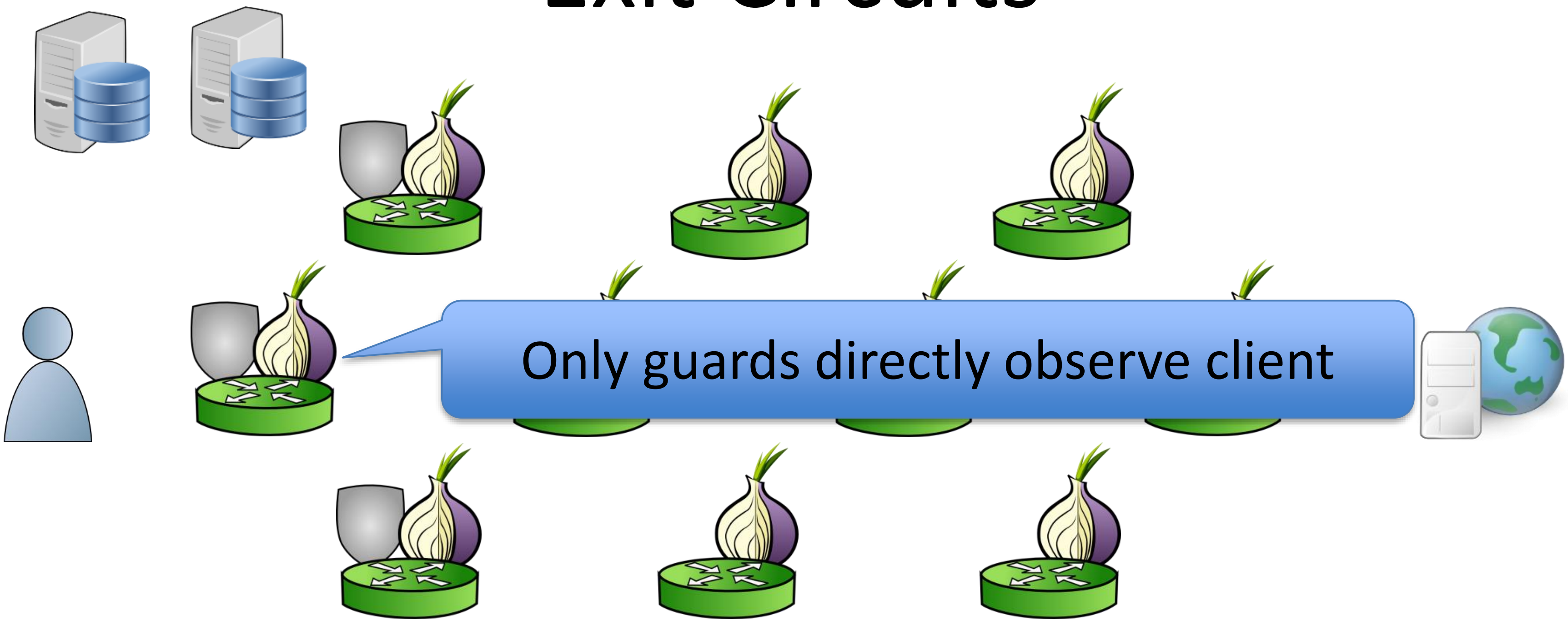
1. Client learns about relays from a directory server.

Exit Circuits



1. Client learns about relays from a directory server.
2. Clients begin all *circuits* with a selected *guard*.

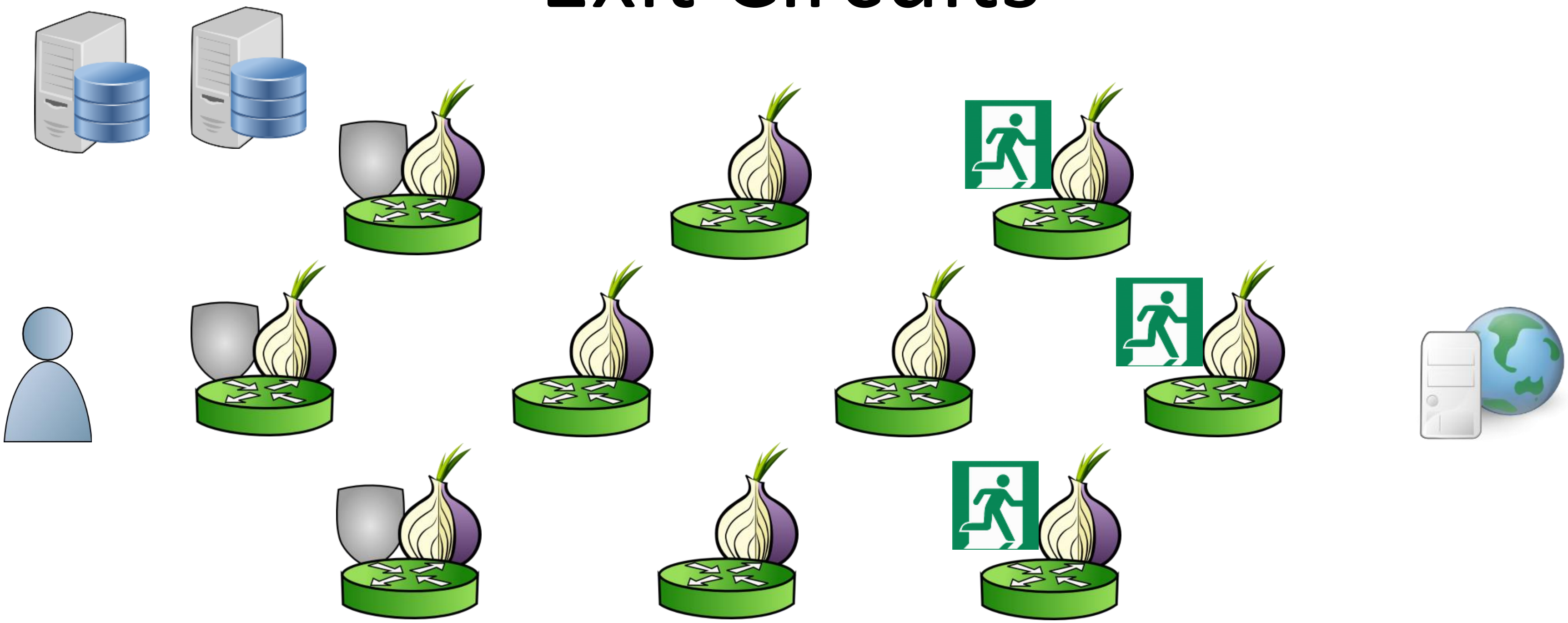
Exit Circuits



Only guards directly observe client

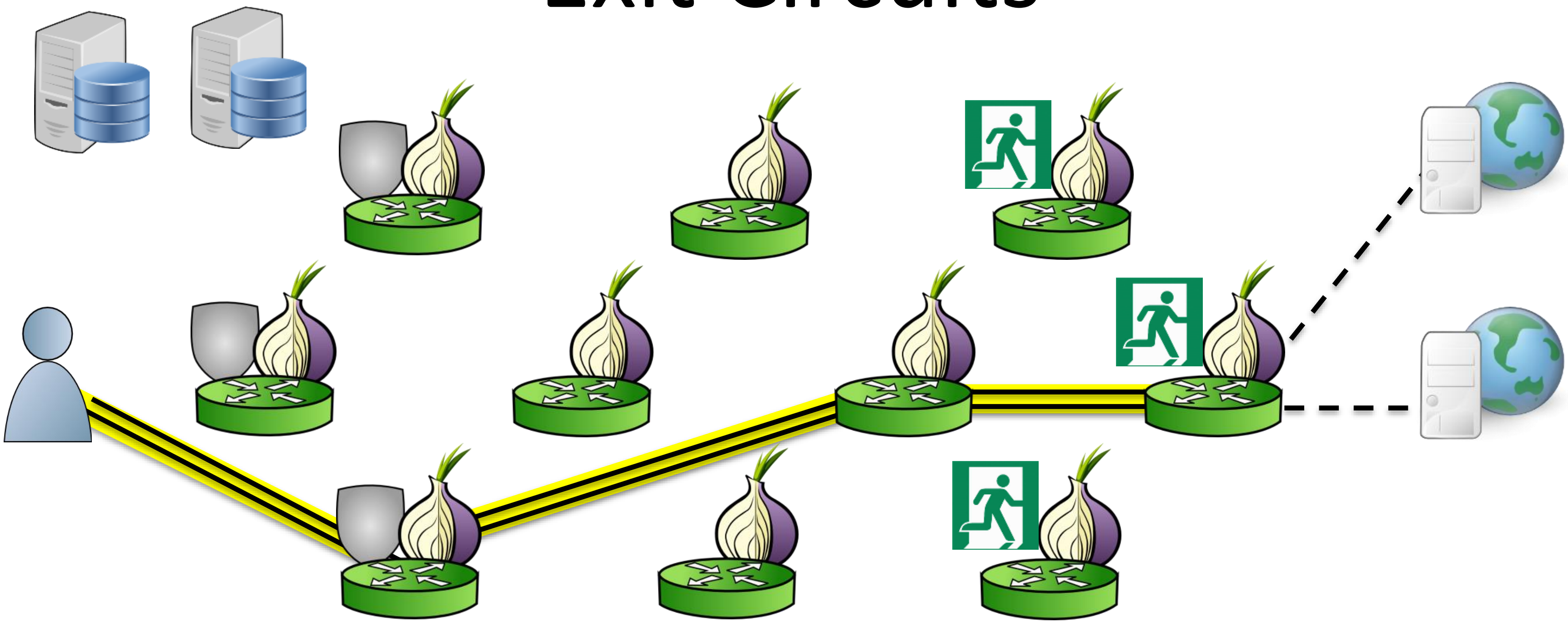
1. Client learns about relays from a directory server.
2. Clients begin all *circuits* with a selected *guard*.

Exit Circuits



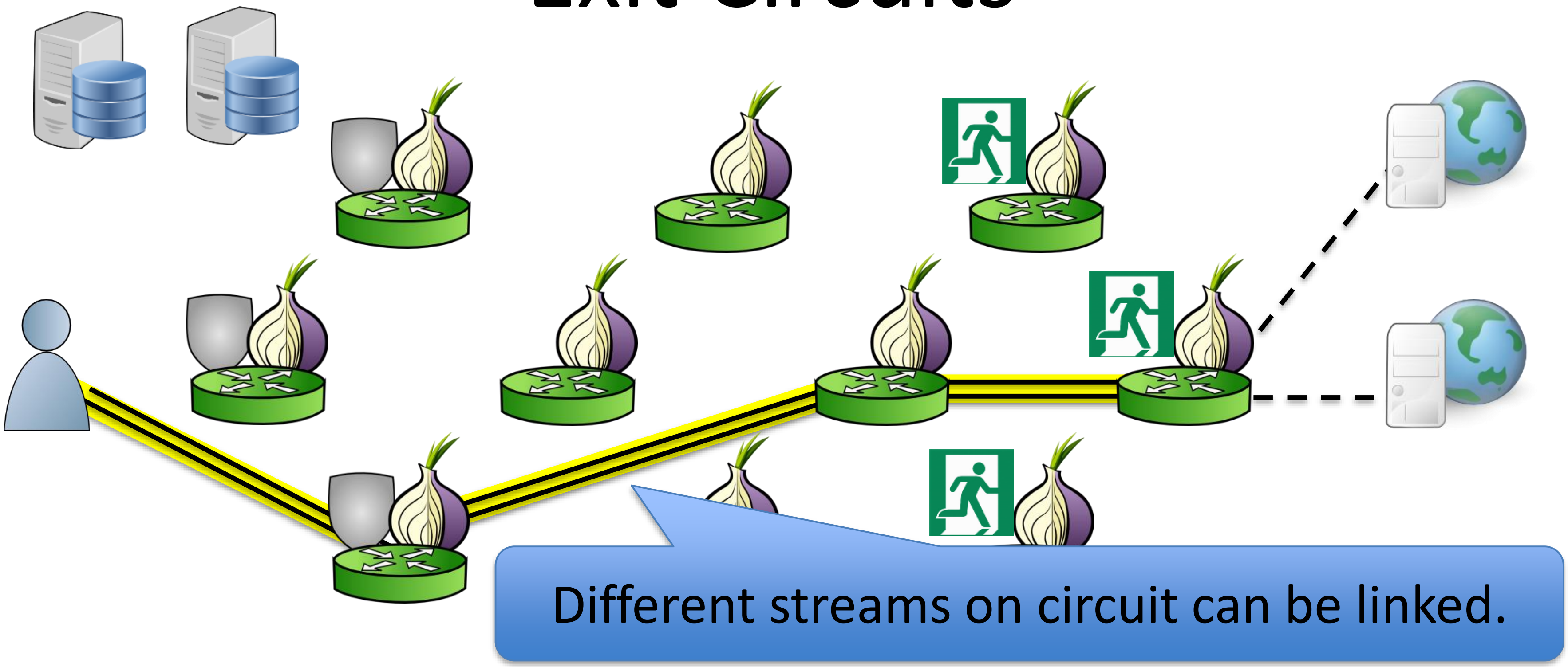
1. Client learns about relays from a directory server.
2. Clients begin all *circuits* with a selected *guard*.
3. Relays define individual *exit policies*.

Exit Circuits



1. Client learns about relays from a directory server.
2. Clients begin all *circuits* with a selected *guard*.
3. Relays define individual *exit policies*.
4. Clients multiplex *streams* over a circuit.

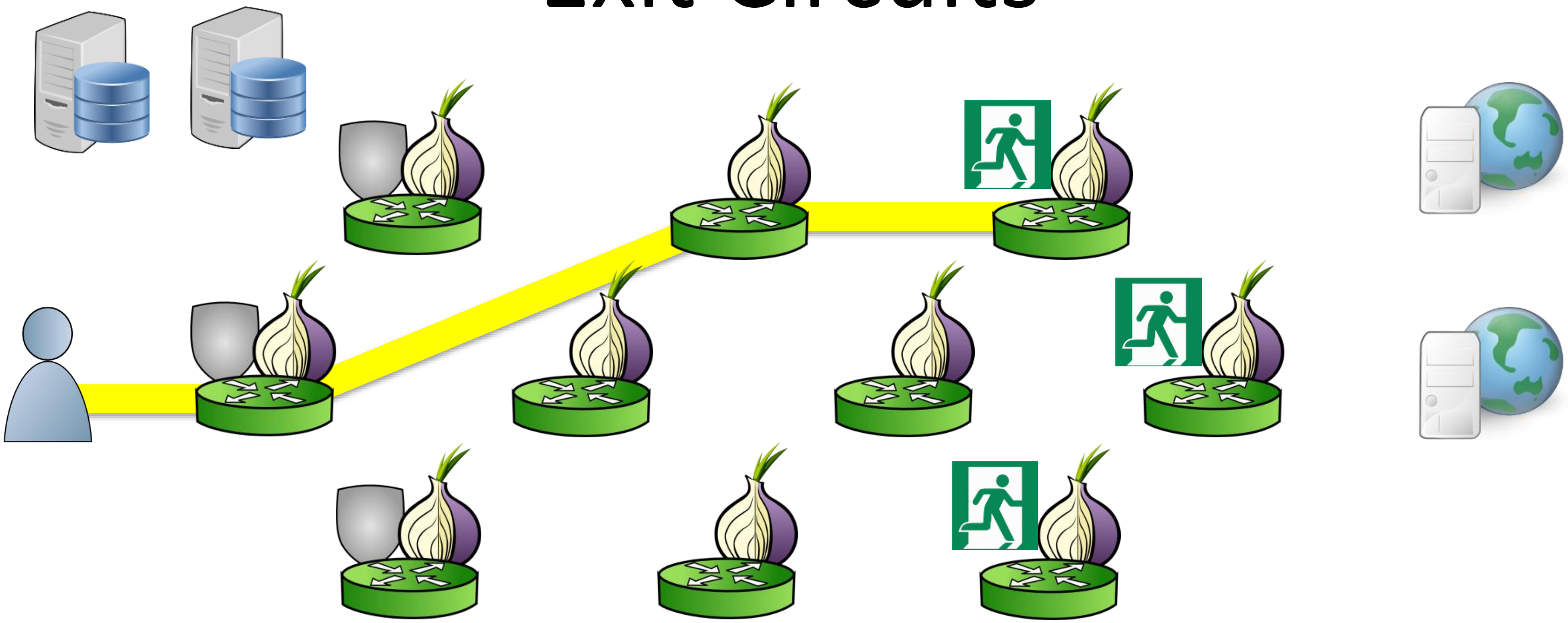
Exit Circuits



Different streams on circuit can be linked.

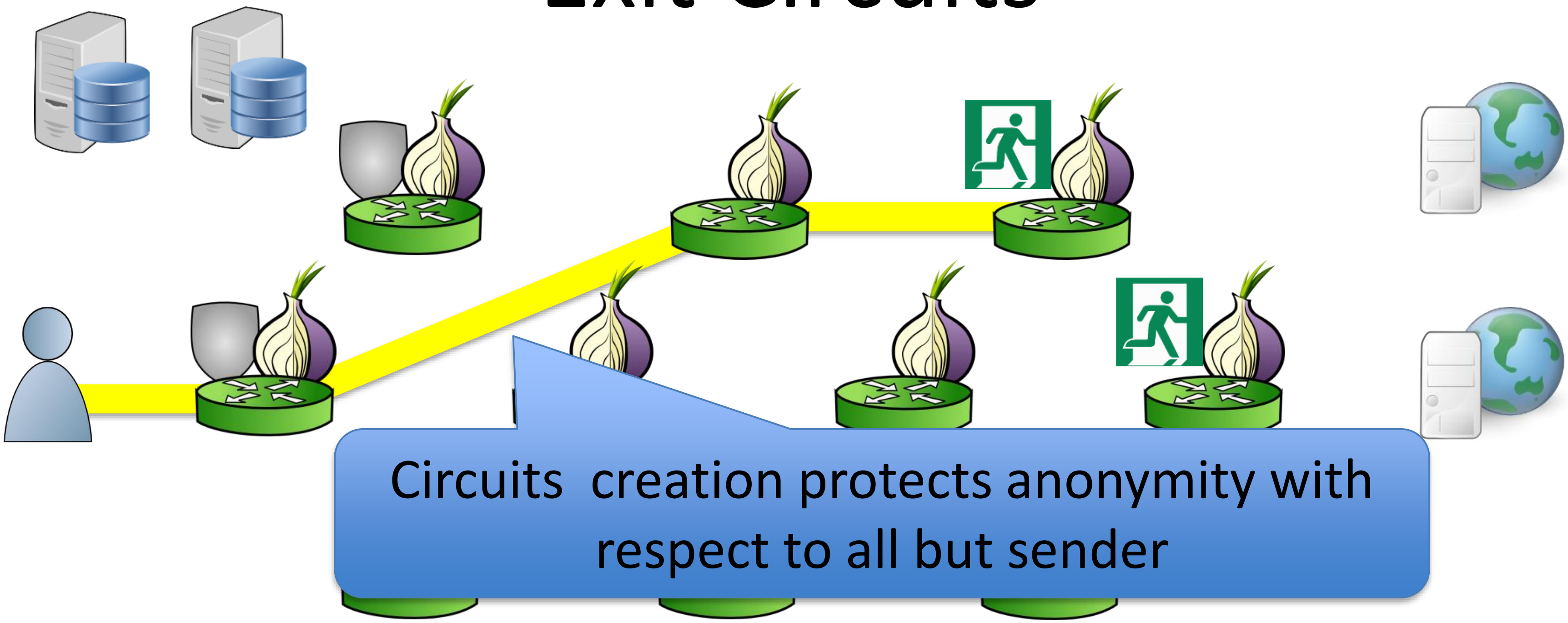
1. Client learns about relays from a directory server.
2. Clients begin all *circuits* with a selected *guard*.
3. Relays define individual *exit policies*.
4. Clients multiplex *streams* over a circuit.

Exit Circuits



1. Client learns about relays from a directory server.
2. Clients begin all *circuits* with a selected *guard*.
3. Relays define individual *exit policies*.
4. Clients multiplex *streams* over a circuit.
5. New circuits replace existing ones periodically.

Exit Circuits



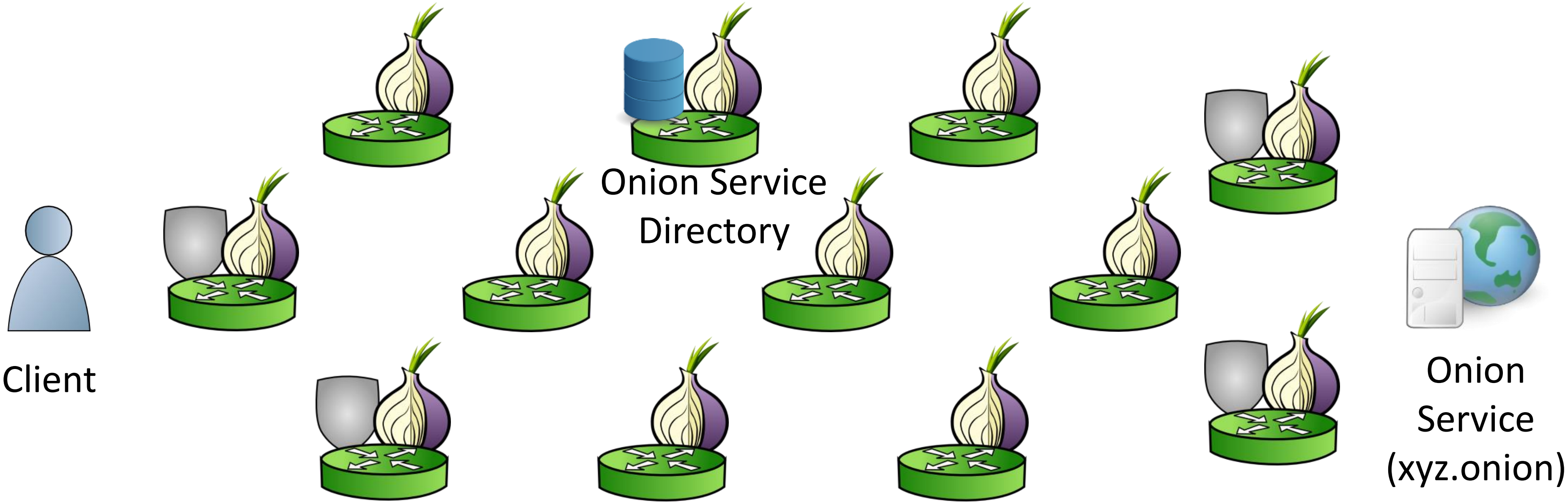
Circuits creation protects anonymity with respect to all but sender

1. Client learns about relays from a directory server.
2. Clients begin all *circuits* with a selected *guard*.
3. Relays define individual *exit policies*.
4. Clients multiplex *streams* over a circuit.
5. New circuits replace existing ones periodically.

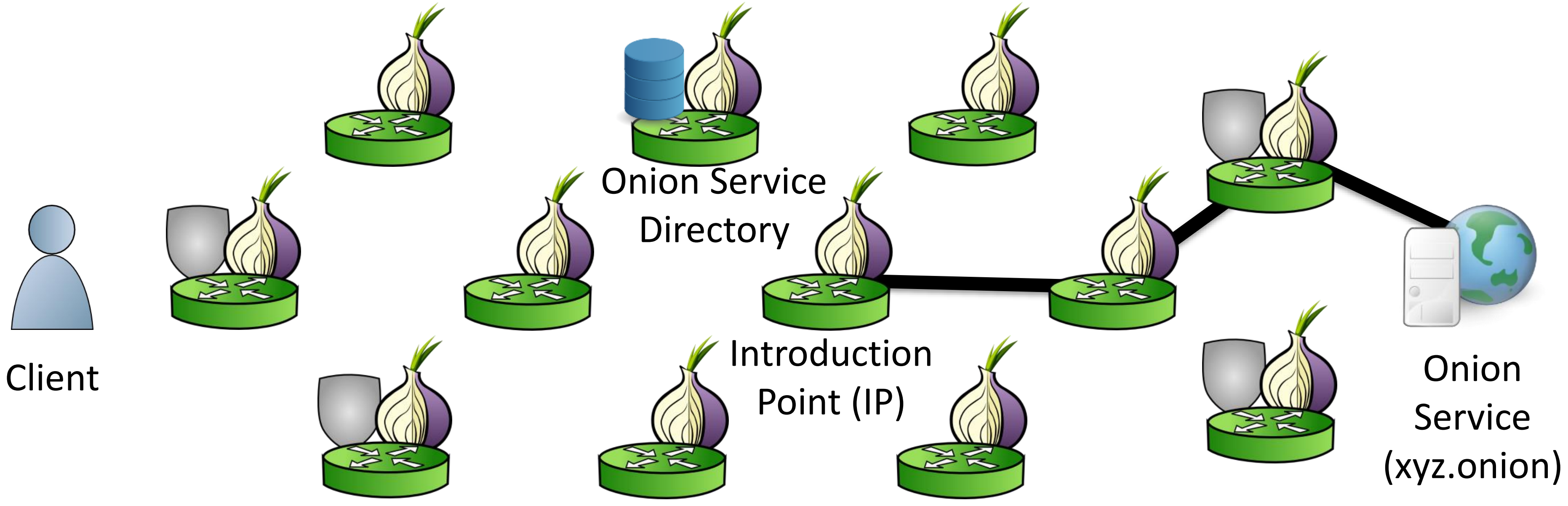
Tor Protocols

1. Exit circuits (*anonymity wrt all but sender*)
2. **Onion services (*anonymity wrt all*)**
3. Censorship circumvention (*unobservability*)

Onion Services

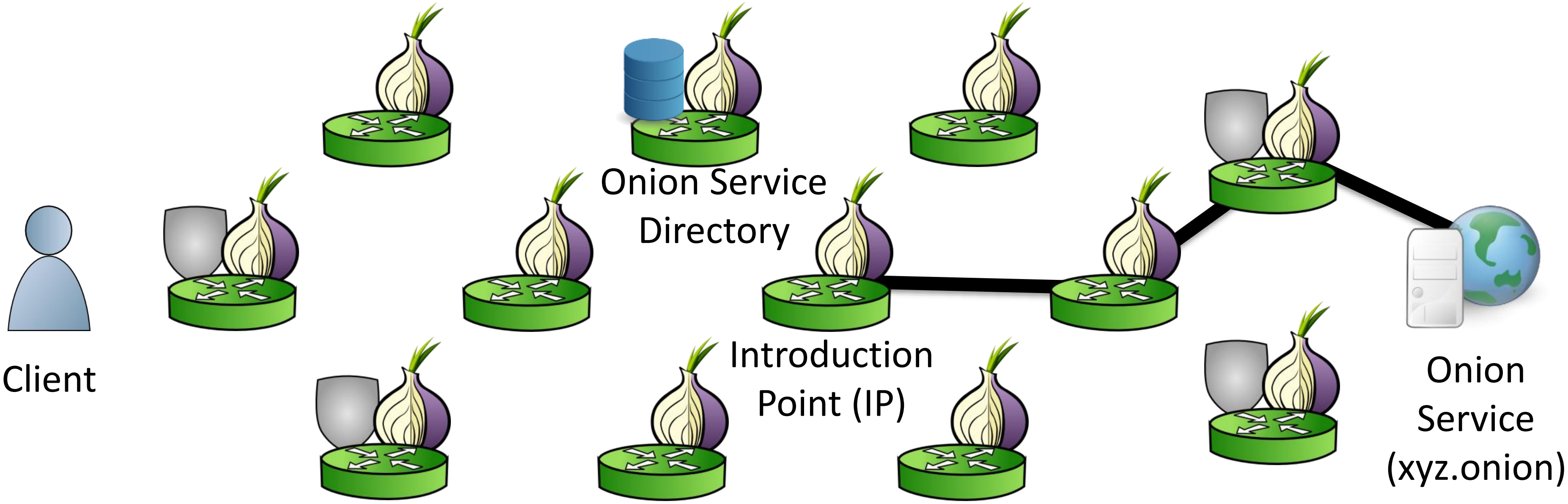


Onion Services



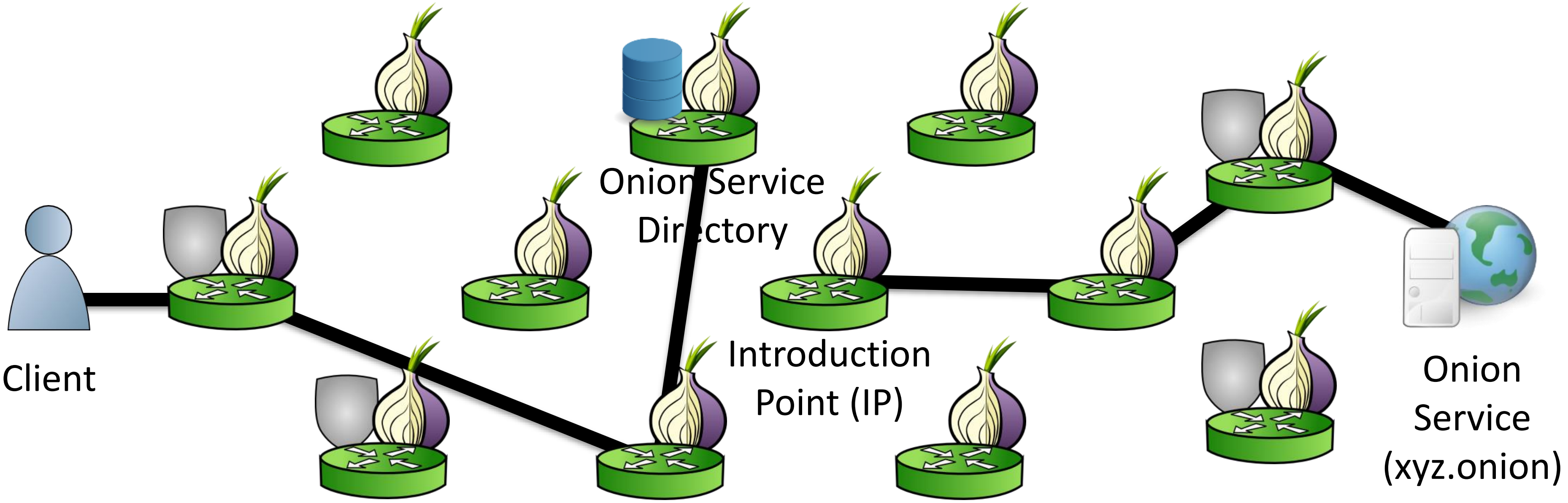
1. Onion service chooses and publishes *Introduction Point* (IP).

Onion Services



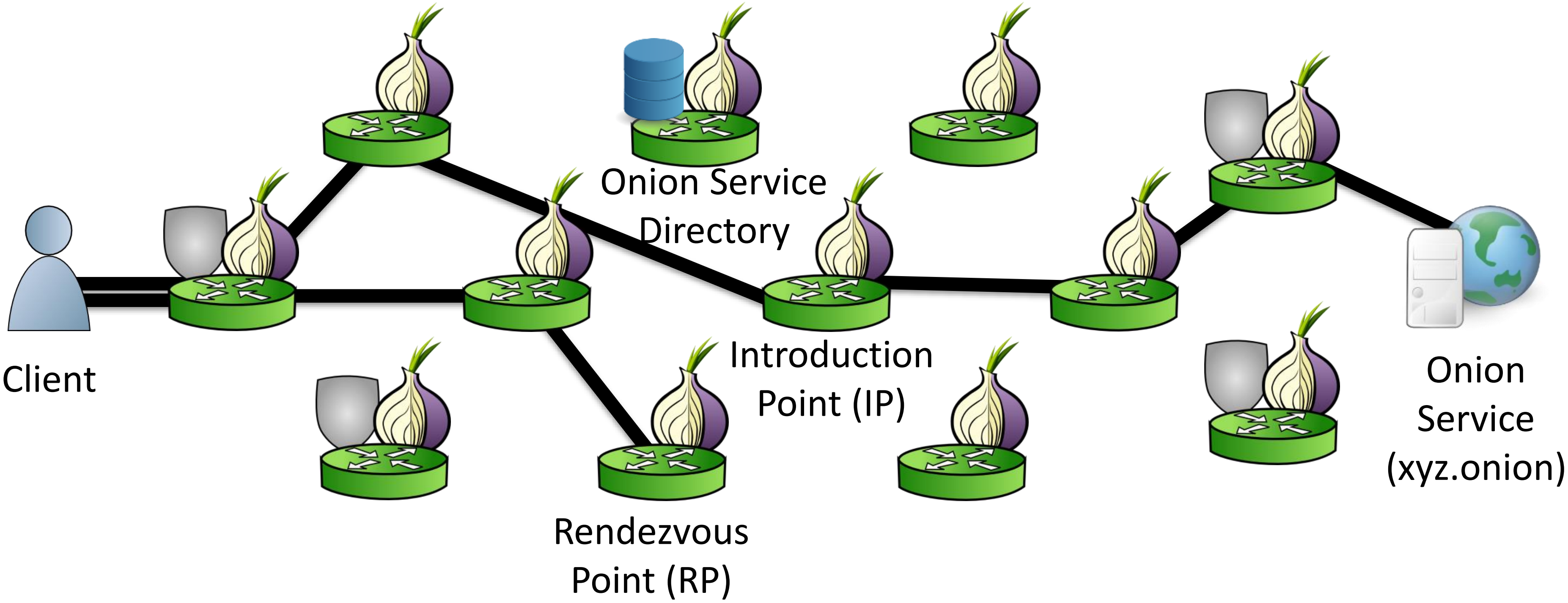
2. Client learns onion address (xyz.onion) out of band.

Onion Services



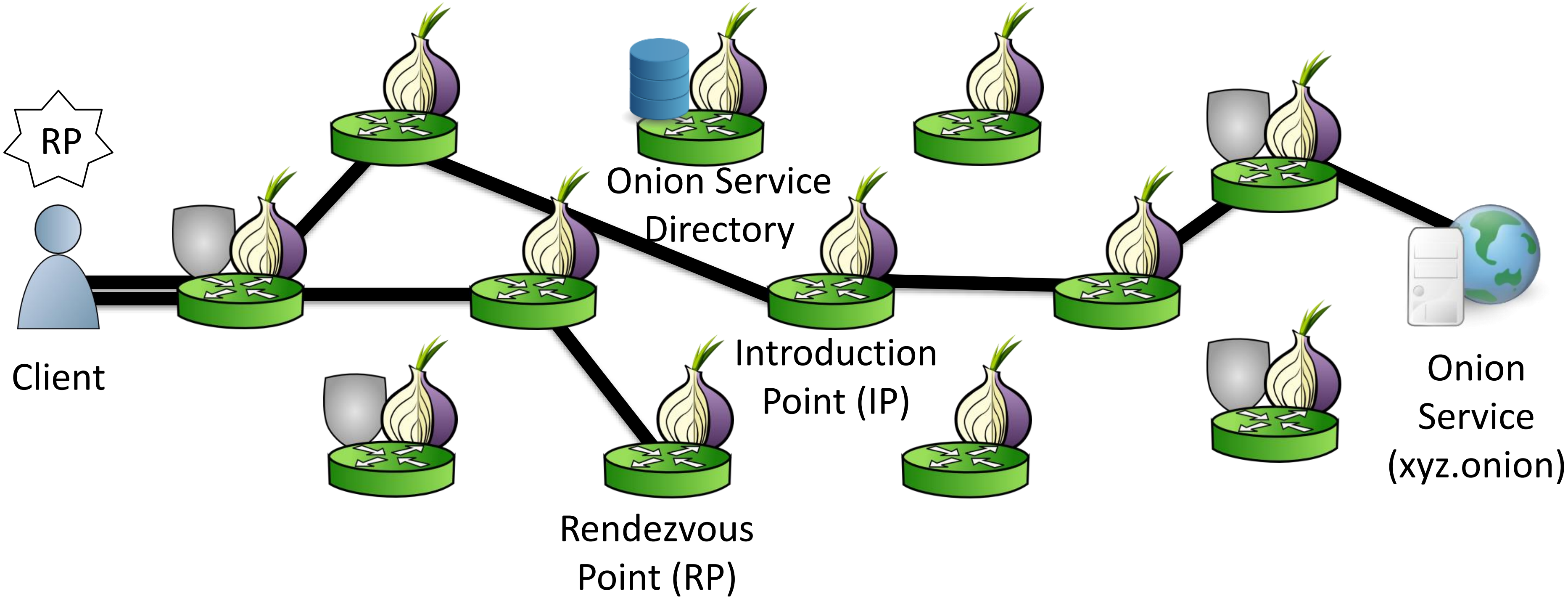
3. Client looks up IP at an Onion Service Directory using .onion address.

Onion Services



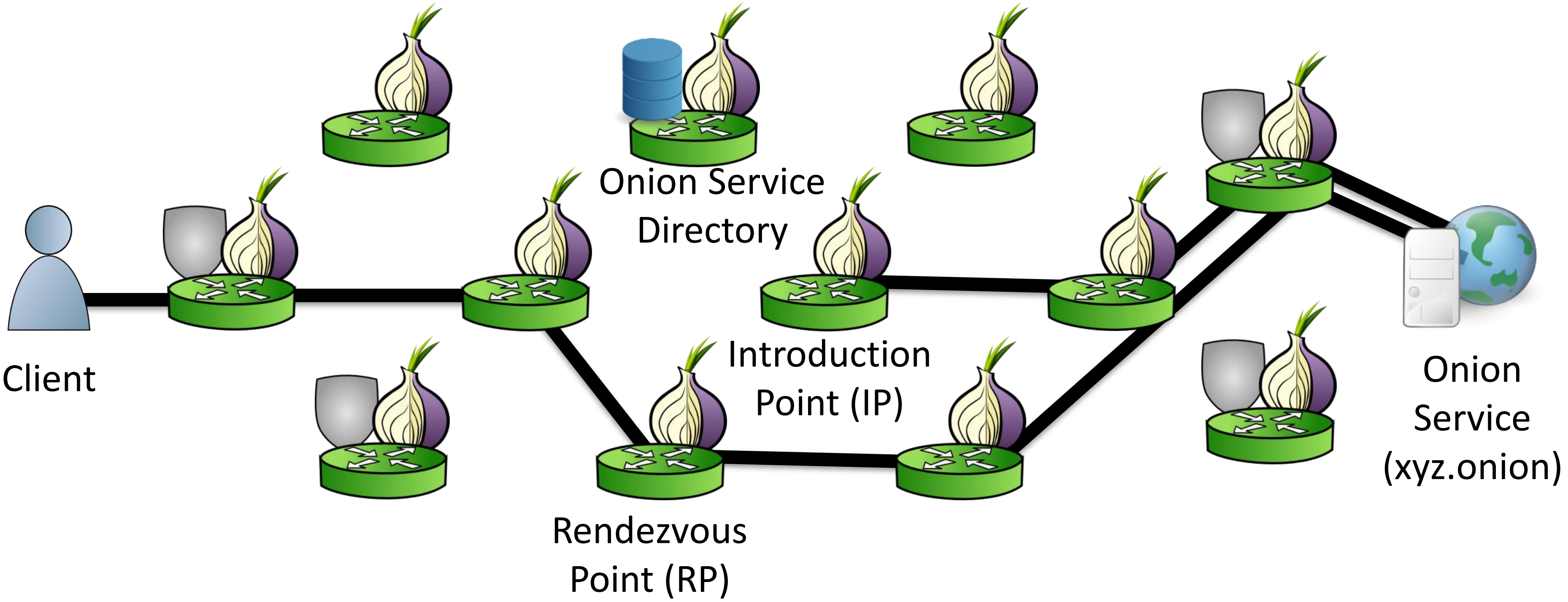
4. Client builds circuits to IP and to chosen Rendezvous Point (RP).

Onion Services



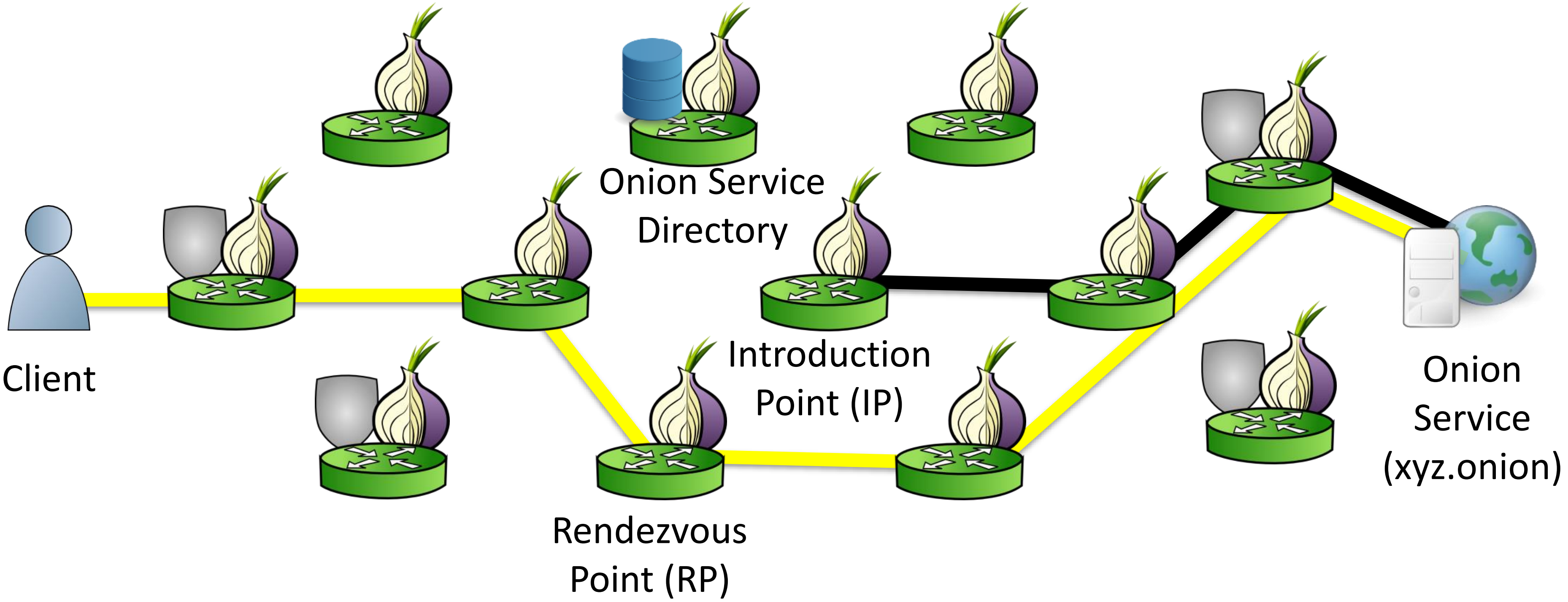
5. Client notifies onion service of RP through IP.

Onion Services



6. Onion service builds new circuit to RP.

Onion Services

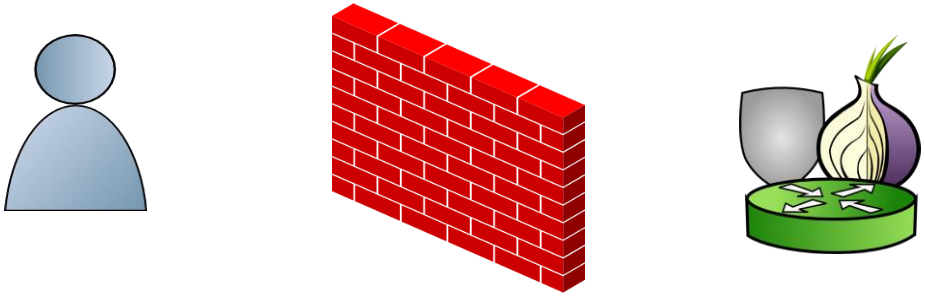


7. Client and onion service communicate through RP circuits.

Tor Protocols

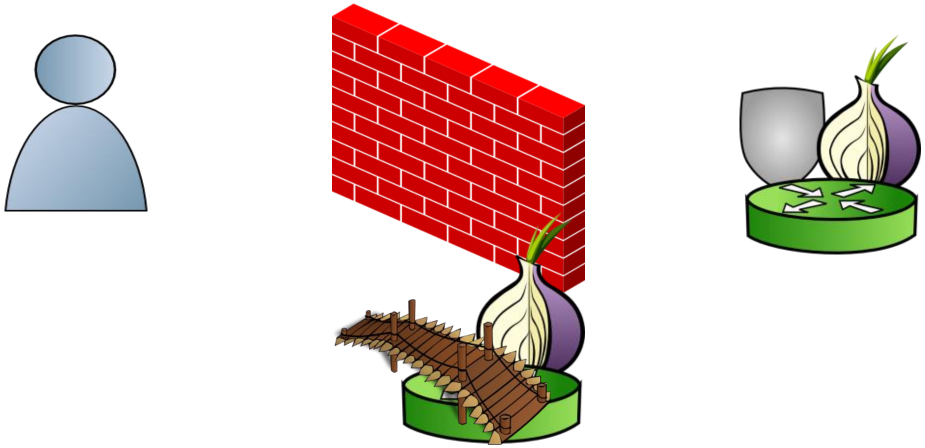
1. Exit circuits (*anonymity wrt all but sender*)
2. Hidden services (*anonymity wrt all*)
3. **Censorship circumvention (*unobservability*)**

Blocking Tor



- Tor directory and relay IPs are public
- Tor connections are made over TLS
- Tor cells have a fixed length

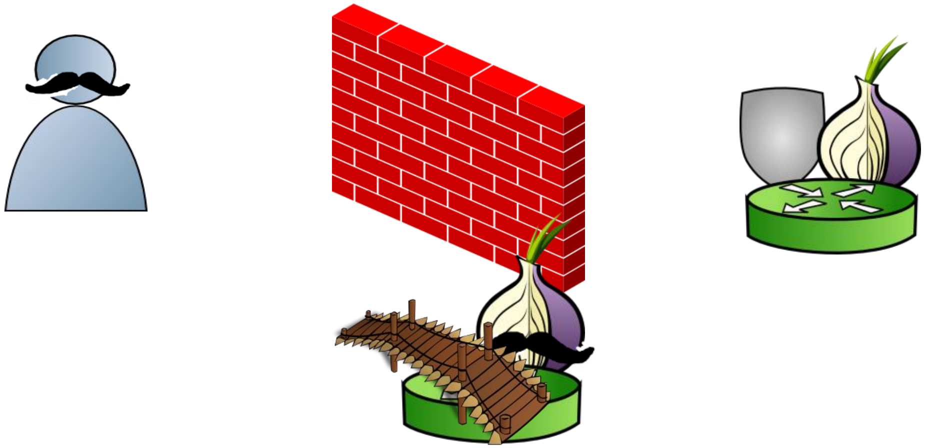
Blocking Tor



- **Tor directory and relay IPs are public**
- Tor connections are made over TLS
- Tor cells have a fixed length

- Private Tor *bridges* released via
  - CAPTCHA
  - Email request
  - Personal communication
- *Meek* uses cloud services (e.g. Azure) and *domain fronting*

Blocking Tor

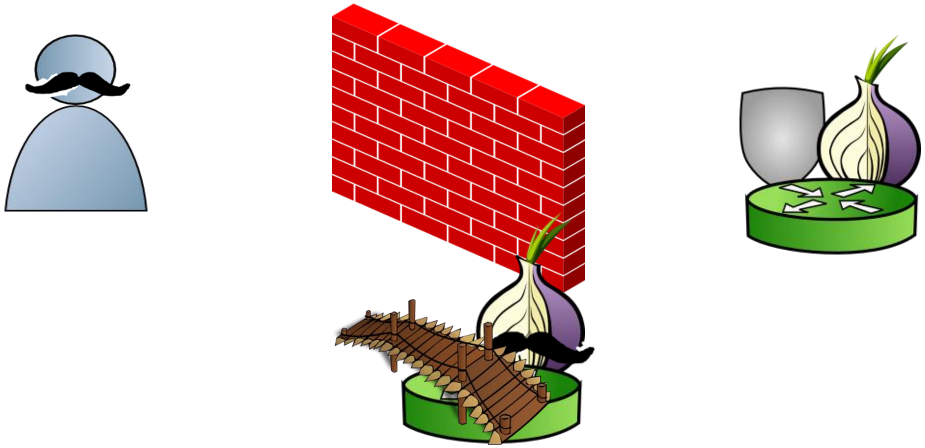


- Tor directory and relay IPs are public
- **Tor connections are made over TLS**
- Tor cells have a fixed length

Pluggable transports

- *obfsproxy4* makes protocol look like strings of random bits
- SkypeMorph/FreeWave; Steganographic VOIP

Blocking Tor

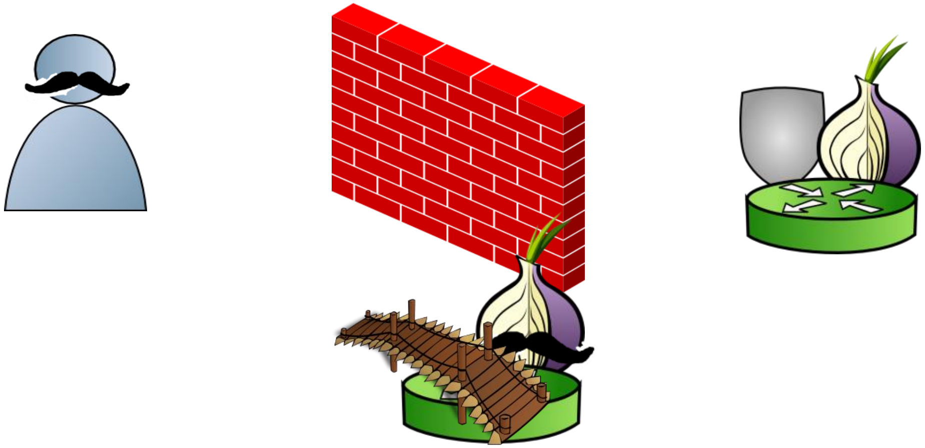


- Tor directory and relay IPs are public
- Tor connections are made over TLS
- **Tor cells have a fixed length**

Defenses

- ScrambleSuit: randomized lengths
- StegoTorus: Steganographic HTTP

Blocking Tor



- Tor directory and relay IPs are public
- Tor connections are made over TLS
- Tor cells have a fixed length

Weaknesses

- Countries have manpower to enumerate bridges
- Network surveillance used to detect possible Tor connections, followup scans confirm

Attacks

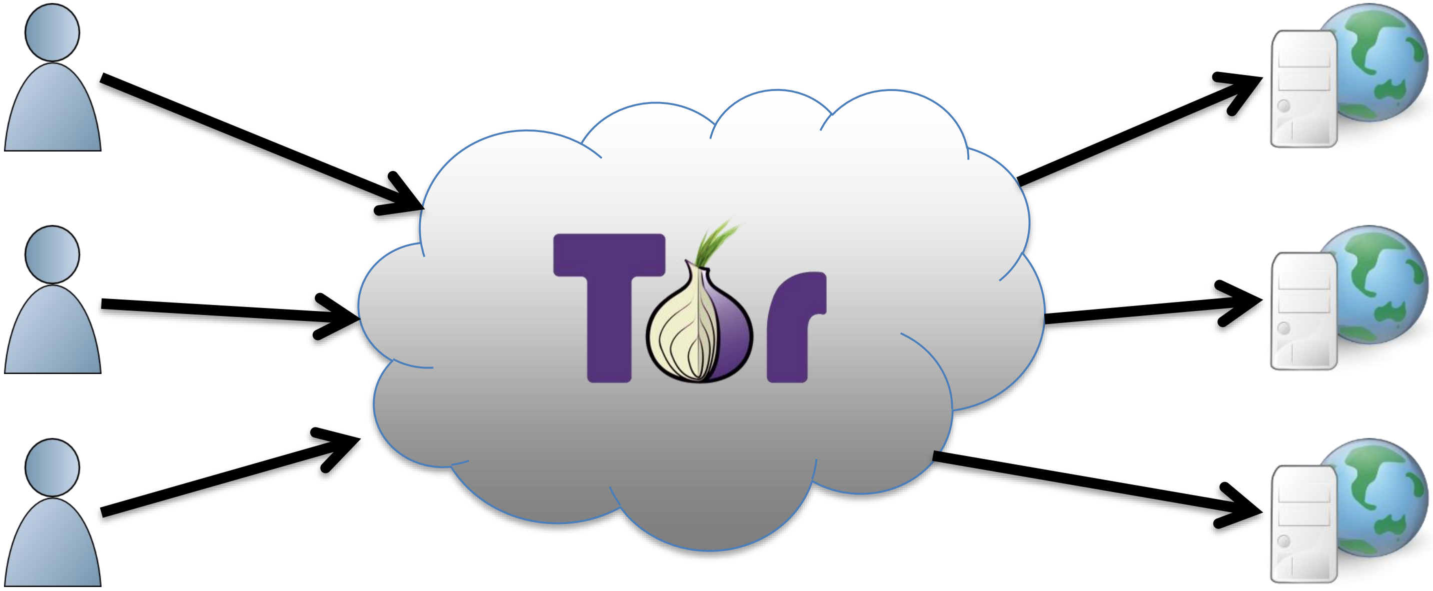
Attacks on Tor

1. Application-layer attacks
2. Bandwidth manipulation
3. Congestion/throughput attack
4. Correlation attack
5. Denial-of-service attacks
6. Guard discovery & compromise
7. Latency attack
8. Route hijacking/interception
9. Sniper attack
10. Website fingerprinting

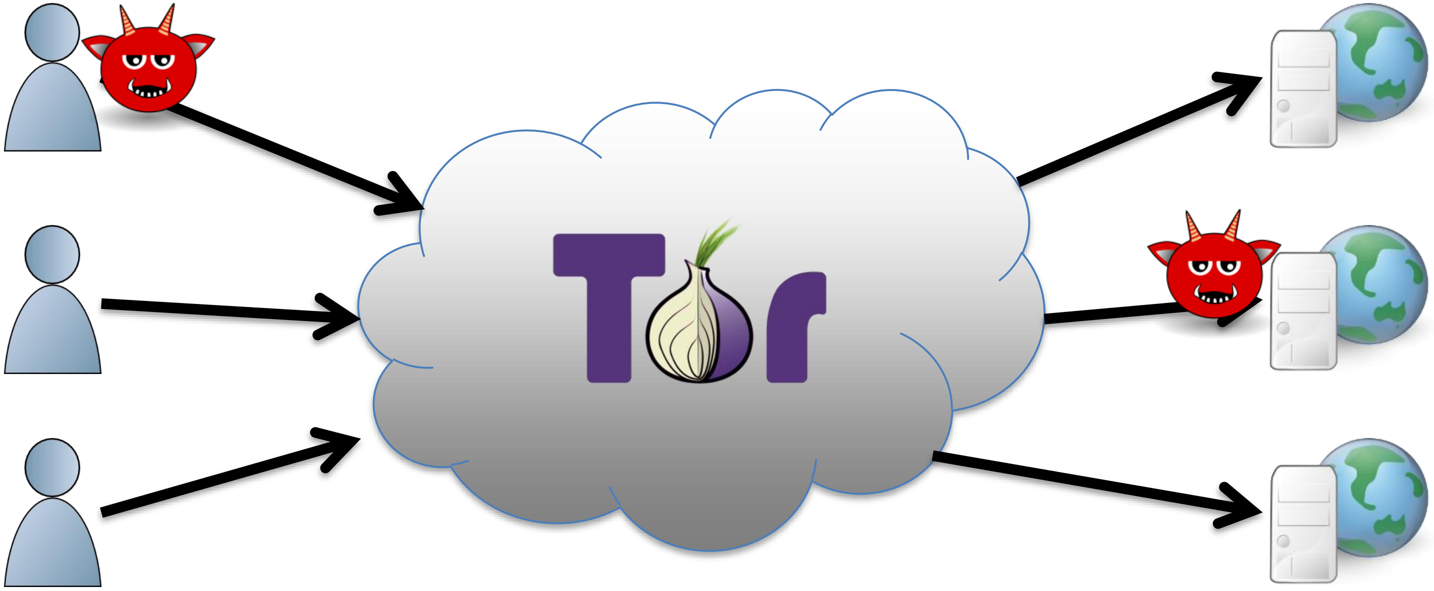
Attacks on Tor

1. Application-layer attacks
2. Bandwidth manipulation
3. Congestion/throughput attack
4. **Correlation attack**
5. Denial-of-service attacks
6. Guard discovery & compromise
7. Latency attack
8. Route hijacking/interception
9. Sniper attack
10. Website fingerprinting

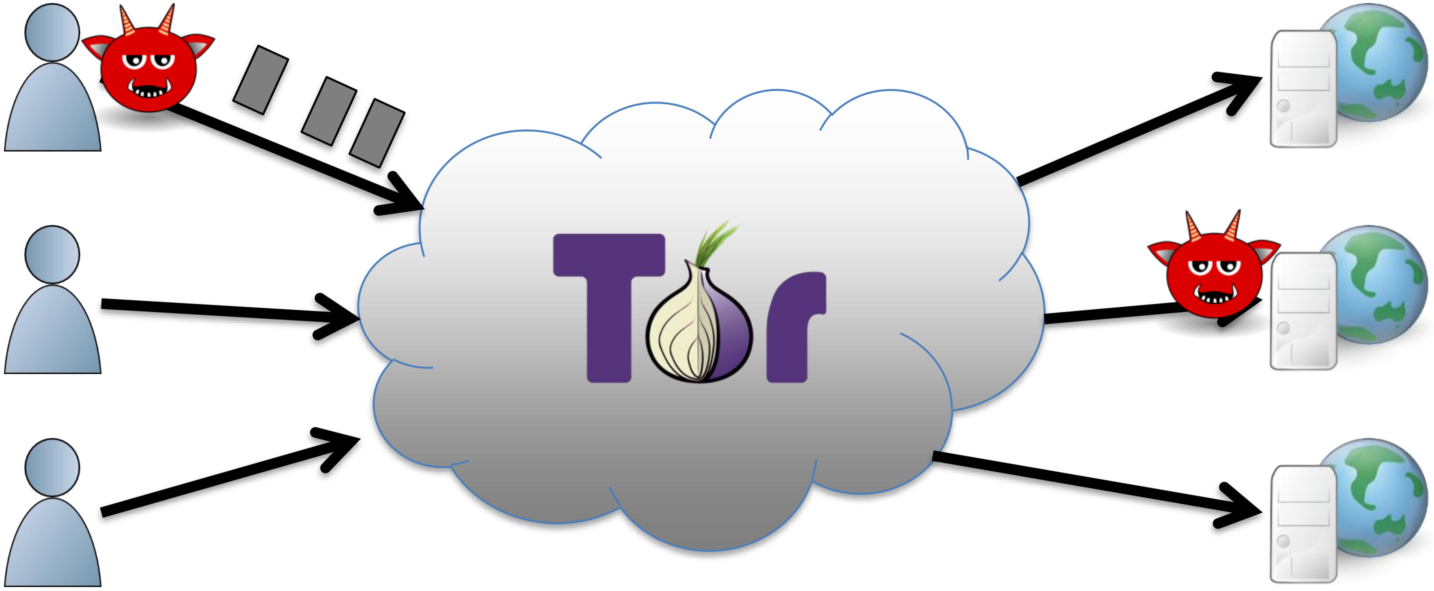
Correlation Attack



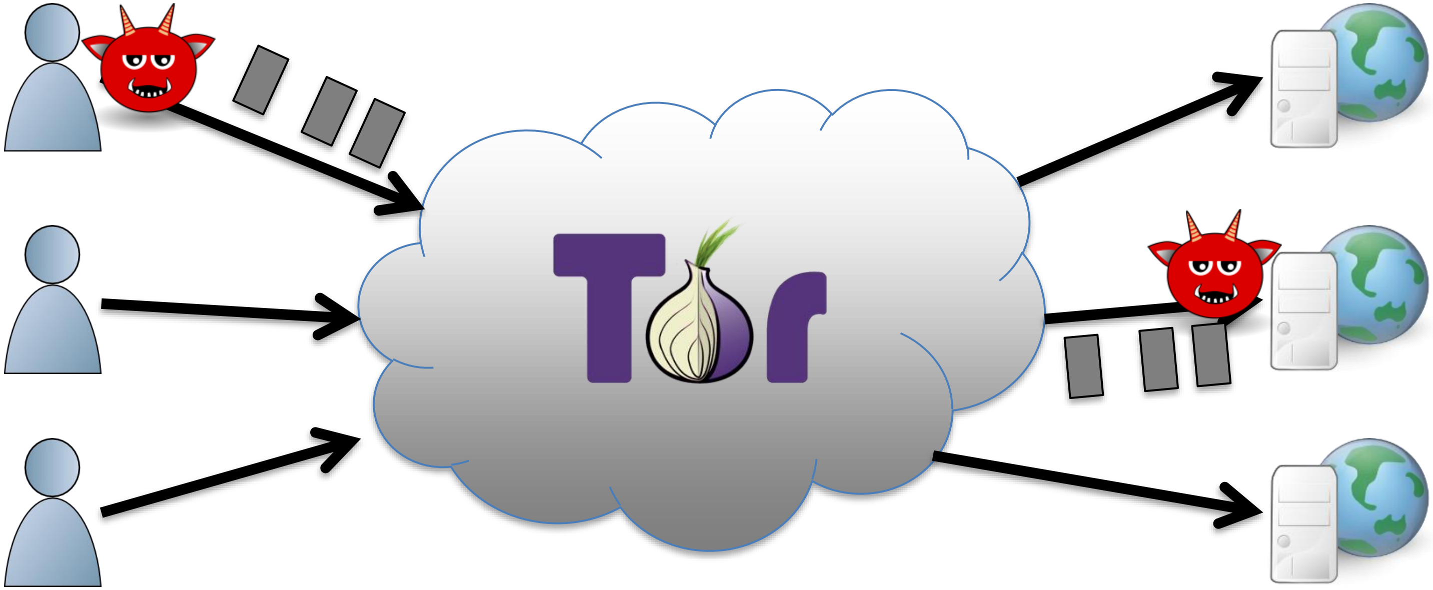
Correlation Attack



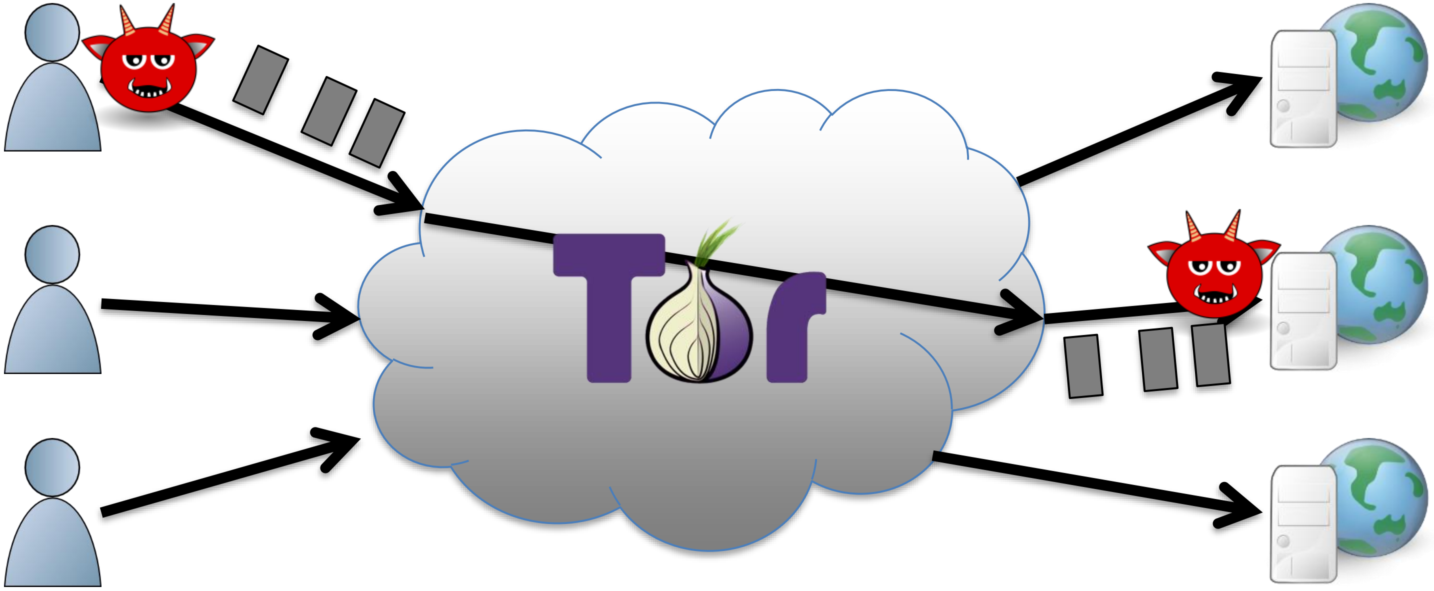
Correlation Attack



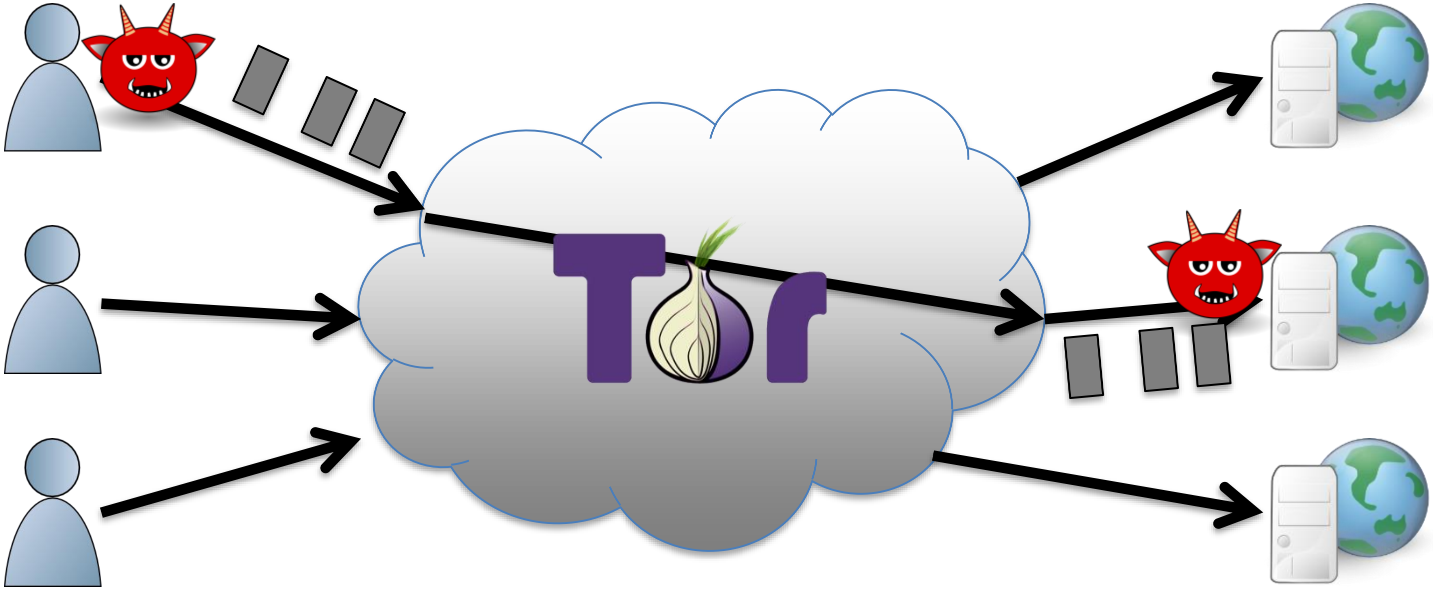
Correlation Attack



Correlation Attack

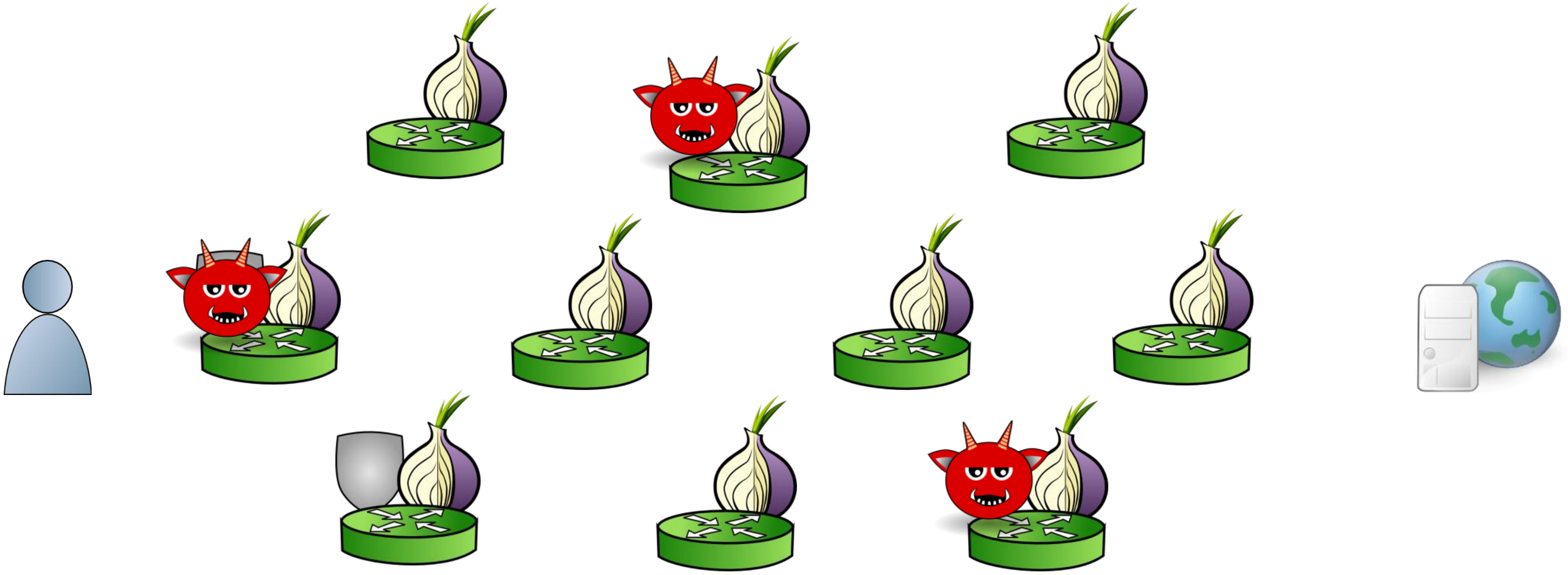


Correlation Attack



Possible when:

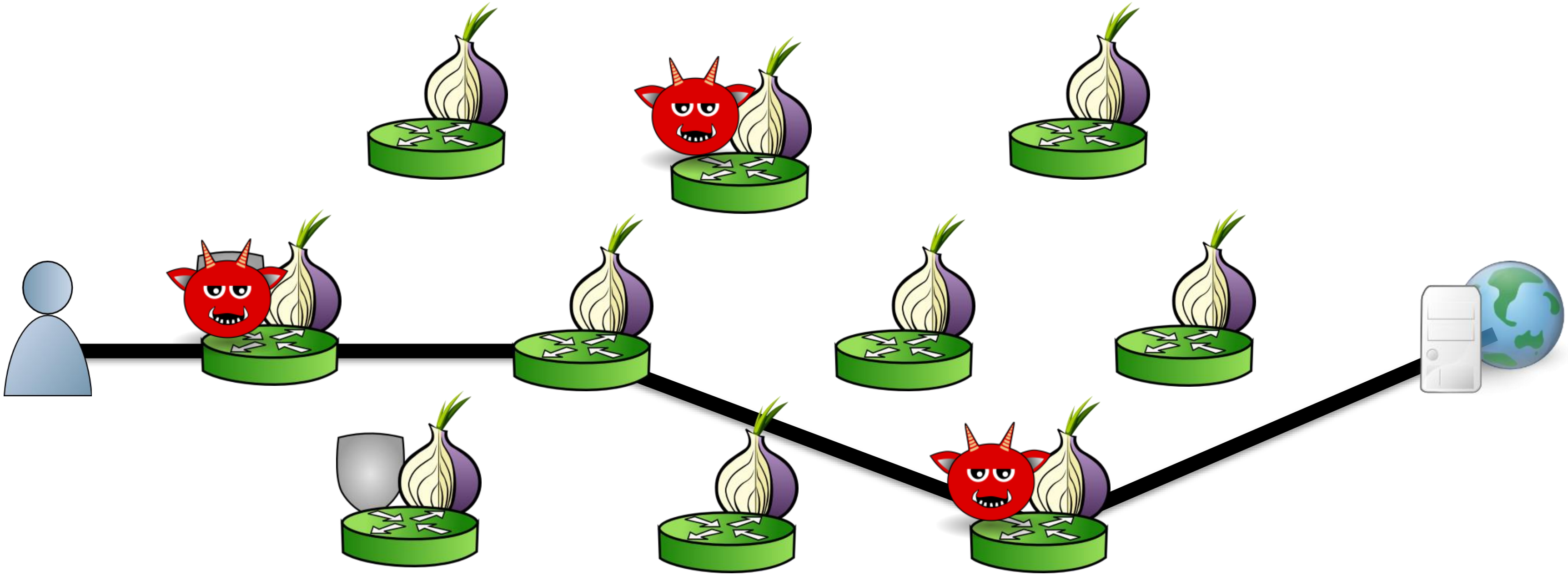
Correlation Attack



Possible when:

1. Adversary controls relays.

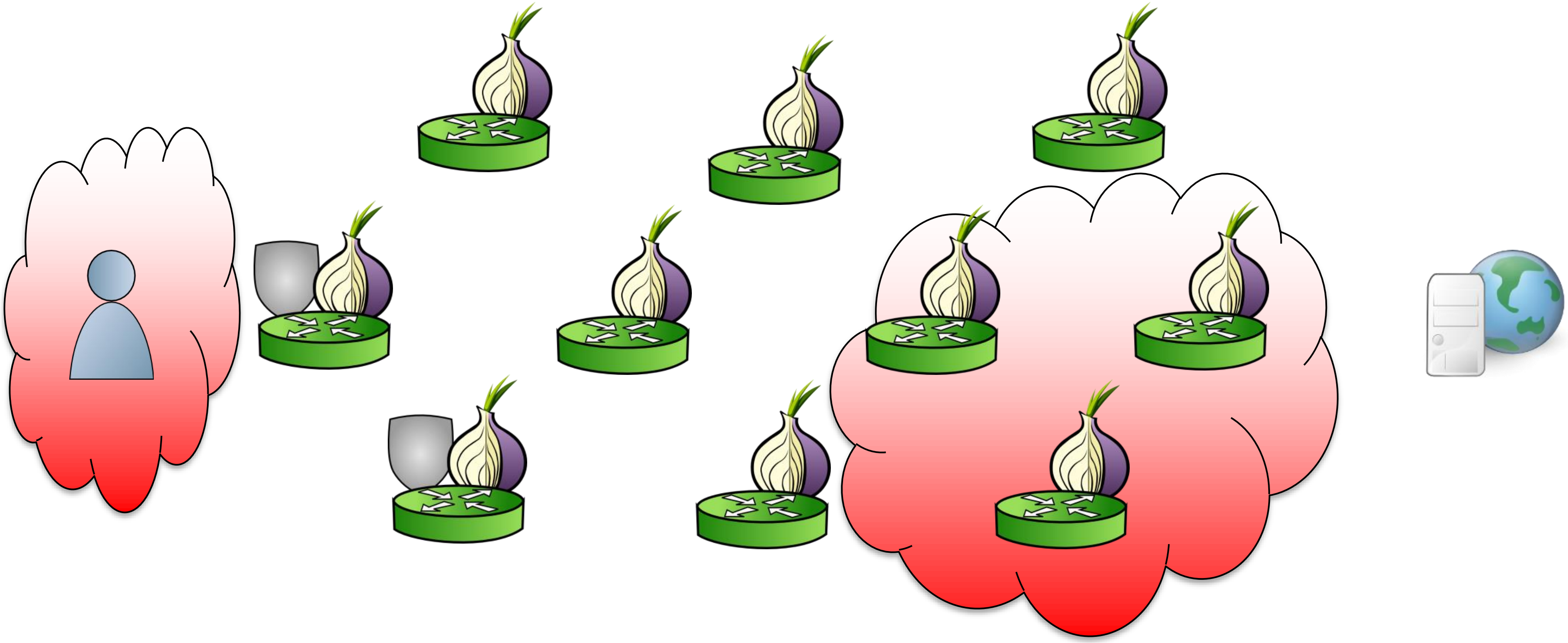
Correlation Attack



Possible when:

1. Adversary controls relays.

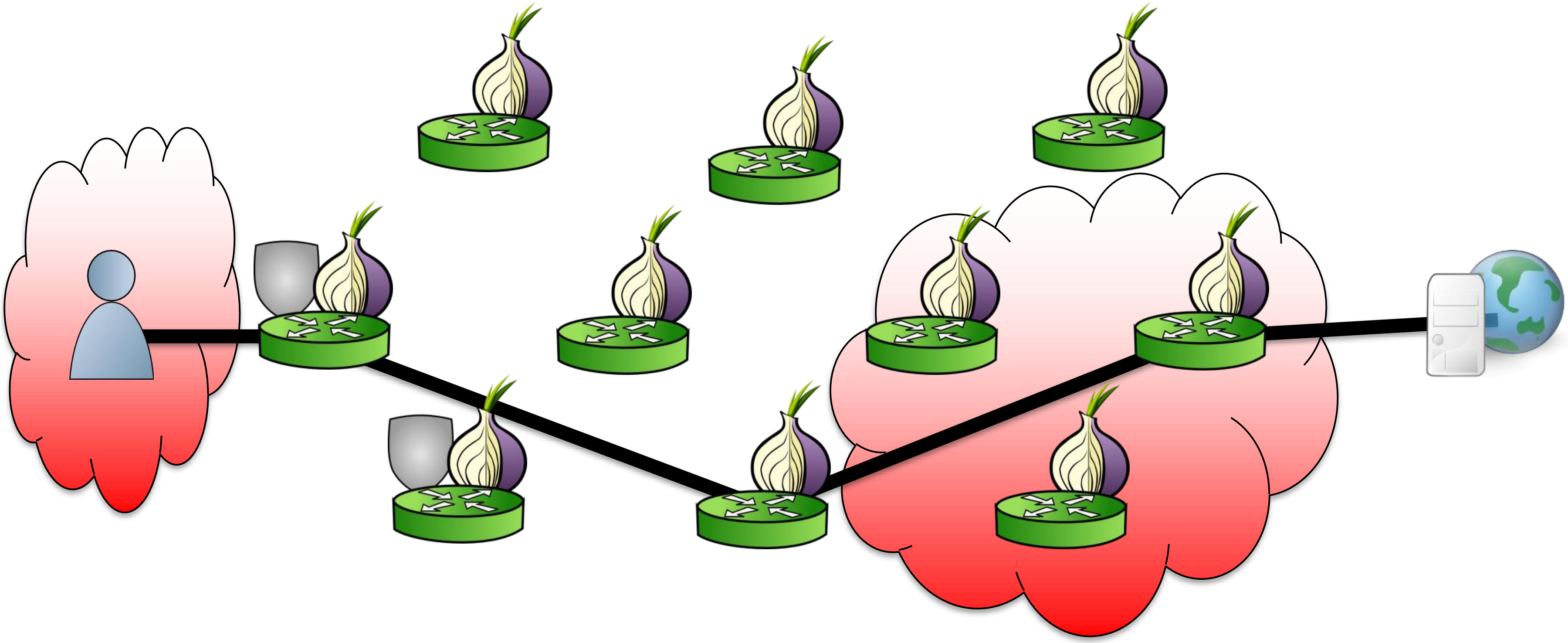
Correlation Attack



Possible when:

1. Adversary controls relays.
2. Adversary observes parts of the network.

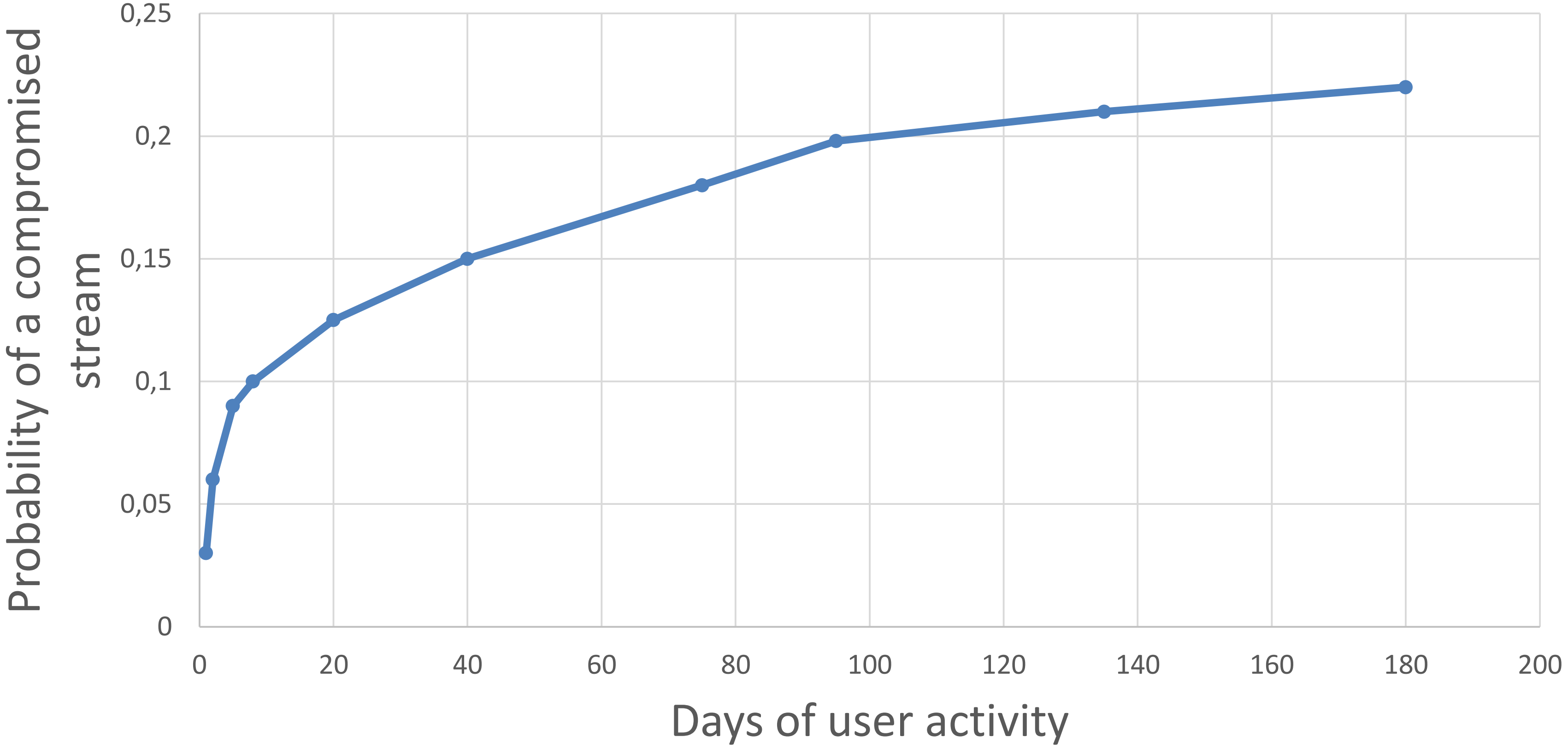
Correlation Attack



Possible when:

1. Adversary controls relays.
2. Adversary observes parts of the network.

Correlation Attack



Adversary relays are

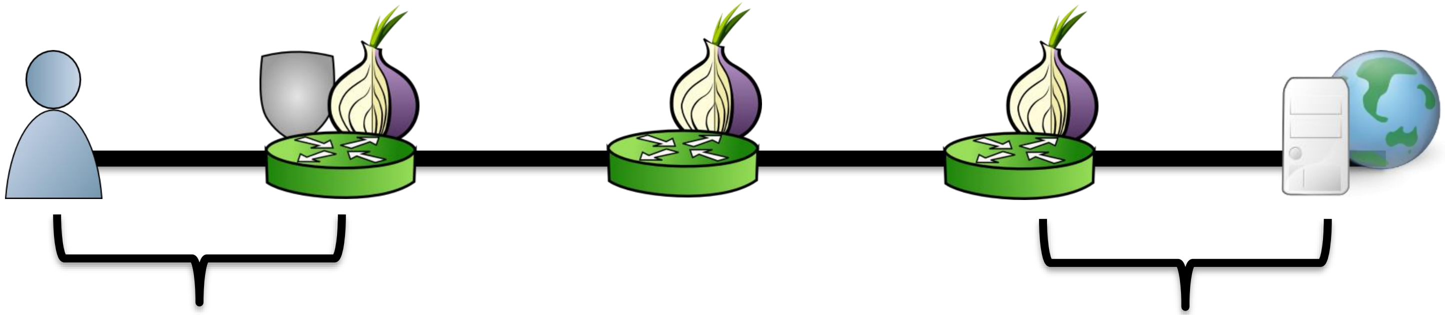
- 10% of guard bandwidth
- 1% of exit bandwidth

[1] *Users Get Routed: Traffic Correlation on Tor by Realistic Adversaries* by Aaron Johnson, Chris Wacek, Rob Jansen, Micah Sherr, and Paul Syverson. CCS 2013.

The Future

The Future of Tor

1. Improved security

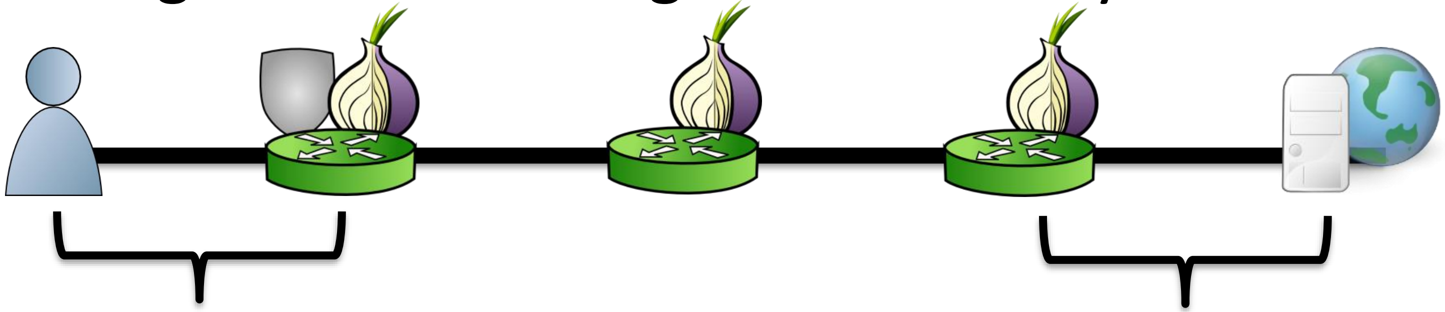


Most critical for security

Difficult to secure – change frequently to prevent tracking

The Future of Tor

1. Improved security
   - Location guards: Prevent traffic correlation
   - Route sentinels: Observe route hijacks
   - Vanguards: Prevent guard discovery



Most critical for security

Difficult to secure – change frequently to prevent tracking

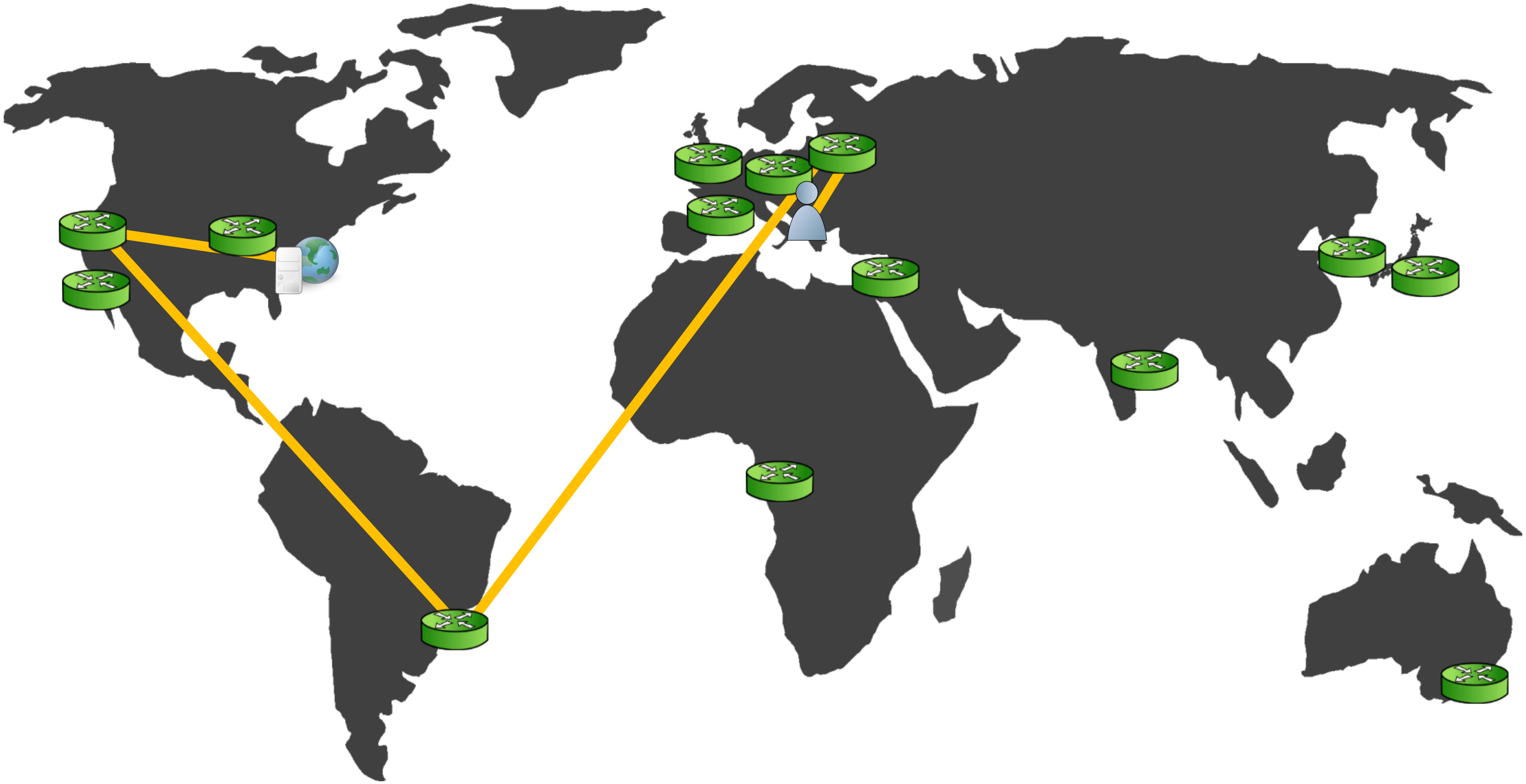
2. *The Sniper Attack: Anonymously Deanonymizing and Disabling the Tor Network*
By Rob Jansen, Florian Tschorsch, Aaron Johnson, and Björn Scheuermann. NDSS 2014.

3. *Avoiding The Man on the Wire: Improving Tor's Security with Trust-Aware Path Selection*
By Aaron Johnson,Rob Jansen, Aaron D. Jaggard,Joan Feigenbaum,and Paul Syverson. NDSS 2017.

4. *Tempest: Temporal Dynamics in Anonymity Systems*
By Ryan Wails, Yixin Sun, Aaron Johnson, Mung Chiang, and Prateek Mittal. PoPETS 2018.

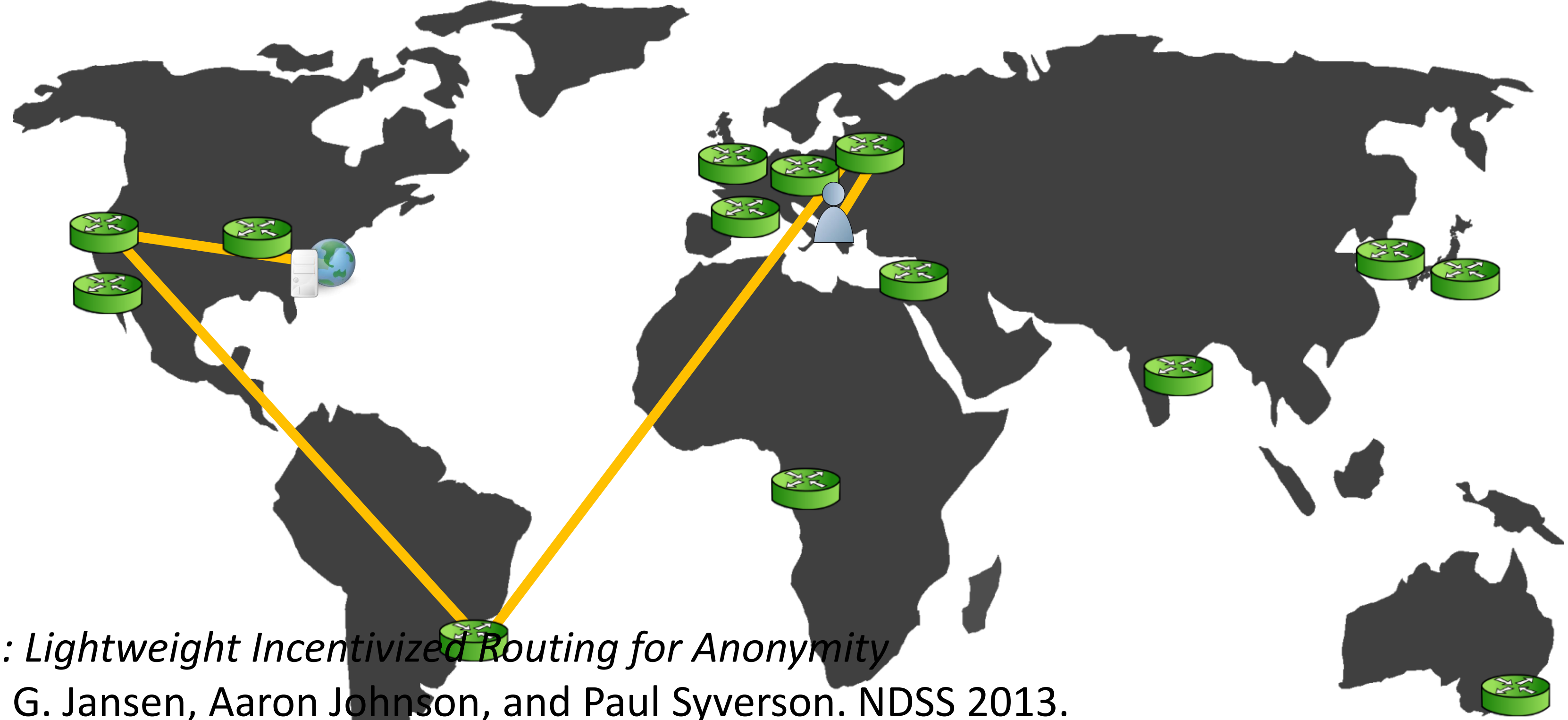
The Future of Tor

2. Improved performance



The Future of Tor

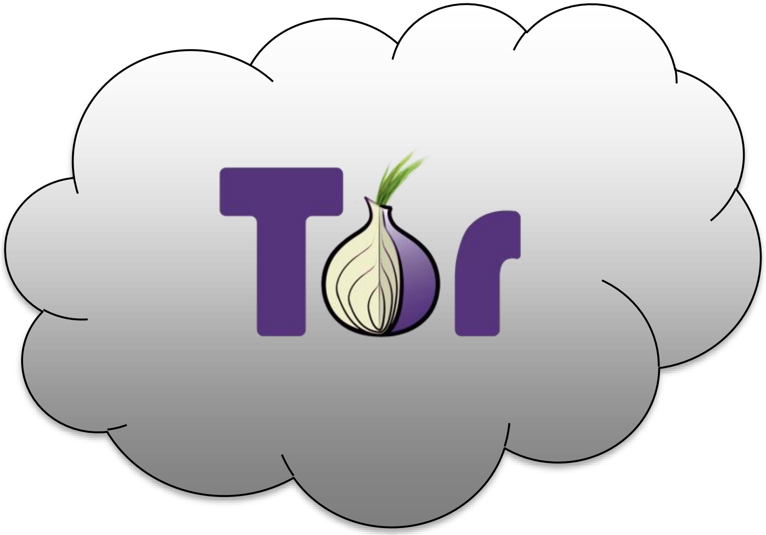
2. Improved performance
   - QUIC/UDP: Enhanced congestion control
   - Scalability: consensus and onion services
   - Secure bandwidth measurement



5. *LIRA: Lightweight Incentivized Routing for Anonymity*
By Rob G. Jansen, Aaron Johnson, and Paul Syverson. NDSS 2013.
6. *PeerFlow: Secure Load Balancing in Tor*
By Aaron Johnson, Rob Jansen, Nicholas Hopper, Aaron Segal, and Paul Syverson. PoPETS 2017.

The Future of Tor

3. Improved transparency *while protecting privacy*

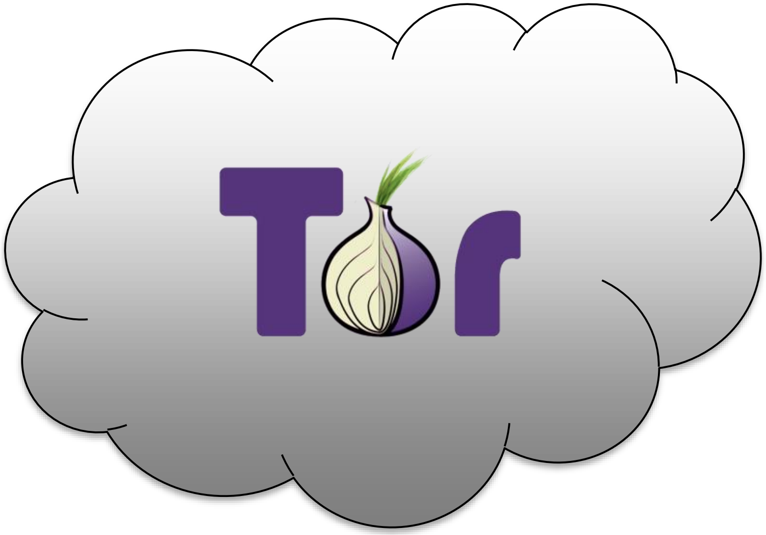


What is going on inside Tor?

- How many users?
- How many onion services?
- How much traffic?
- Where do users visit?
- Any attacks on Tor?
- Any abusers using Tor?

The Future of Tor

3. Improved transparency *while protecting privacy*
   - Publish network statistics
   - Monitor Tor for attacks
   - Detect uses of Tor for abuse



What is going on inside Tor?

- How many users?
- How many onion services?
- How much traffic?
- Where do users visit?
- Any attacks on Tor?
- Any abusers using Tor?

7. *Hidden-service statistics reported by relays*
By David Goulet, Aaron Johnson, George Kadianakis, and Karsten Loesing. Tor TR 2015-04-001.
8. *Safely Measuring Tor*
By Rob Jansen and Aaron Johnson. CCS 2016.
9. *Distributed Measurement with Private Set-Union Cardinality*
By Ellis Fenske, Akshaya Mani, Aaron Johnson, and Micah Sherr. CCS 2017.

The Future of Tor

3. Onionize the Internet
   - Clearweb -> onionspace: single onion services

     Debian.org (http://sejnfjrq6szgca7v.onion/)

     DuckDuckGo (https://3g2upl4pq6kufc4m.onion/)

     Facebook (https://facebookcorewwwi.onion)

     New York Times (https://www.nytimes3xbfgragh.onion/)

     ProPublica (https://www.propub3r6espa33w.onion/)
   - Self-authentication: Invulnerable to Certificate Authority attacks
   - Secure name lookup: Encrypted, authenticated, anonymous (unlike DNS)

10. *Rendezvous Single Onion Services*
By Tim Wilson-Brown, John Brooks, Aaron Johnson, Rob Jansen, George Kadianakis, Paul Syverson, and Roger Dingledine. Tor Proposal 260, 2015.

The Future of Tor

4. Tor Browser = Mozilla Firefox Private Browser
   - Fusion (Firefox USIng ONions)
   - Currently: Tor Uplift Project (https://wiki.mozilla.org/Security/Tor_Uplift)
   - Mass-market anonymity and tracking protection
     - Disable common attack vectors
     - Eliminate supercookies
     - Perform per-tab isolation

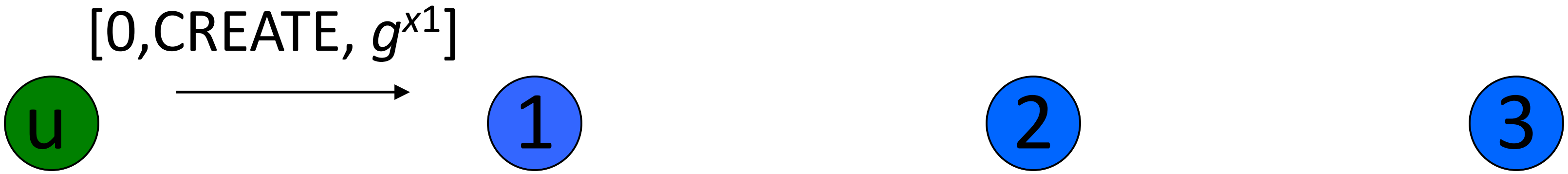
Questions?

Creating a Circuit

$\{m\}_{s_i}$: Encrypted using the DH session key g^{xiyi}

Creating a Circuit

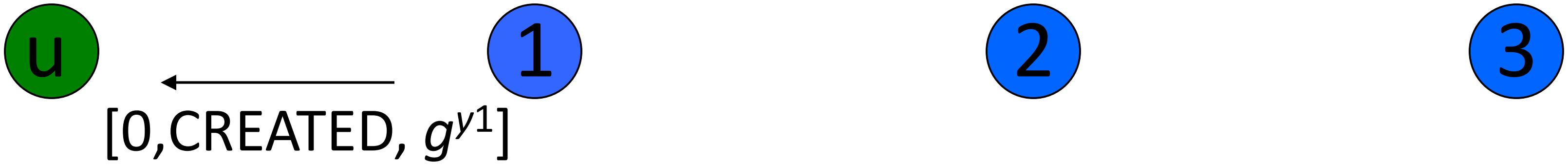
$\{m\}_{s_i}$: Encrypted using the DH session key $g^{x_i y_i}$



1. CREATE/CREATED

Creating a Circuit

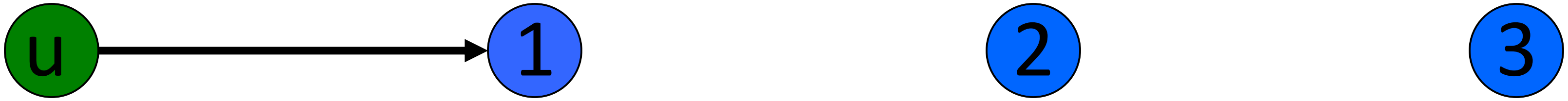
$\{m\}_{s_i}$: Encrypted using the DH session key $g^{x_i y_i}$



1. CREATE/CREATED

Creating a Circuit

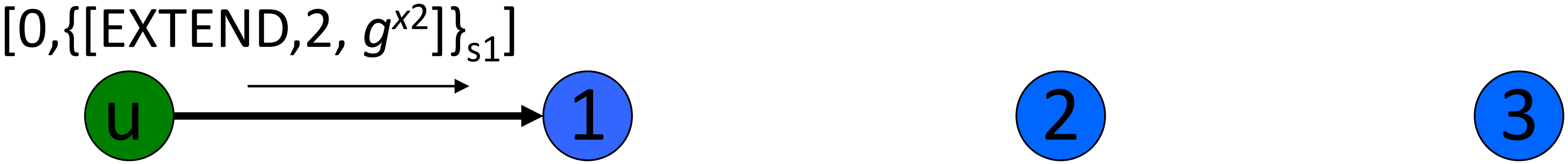
$\{m\}_{s_i}$: Encrypted using the DH session key $g^{x_i y_i}$



1. CREATE/CREATED

Creating a Circuit

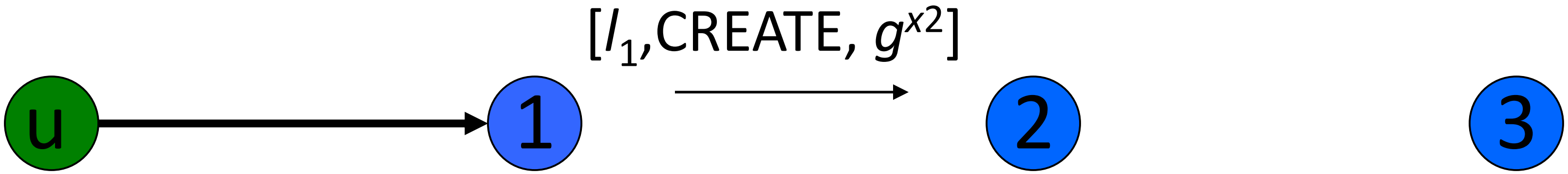
$\{m\}_{s_i}$: Encrypted using the DH session key $g^{x_i y_i}$



1. CREATE/CREATED
2. EXTEND/EXTENDED

Creating a Circuit

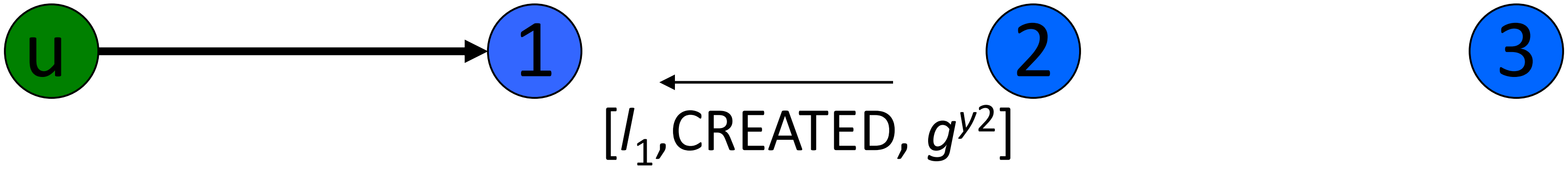
$\{m\}_{s_i}$: Encrypted using the DH session key $g^{x_i y_i}$



1. CREATE/CREATED
2. EXTEND/EXTENDED

Creating a Circuit

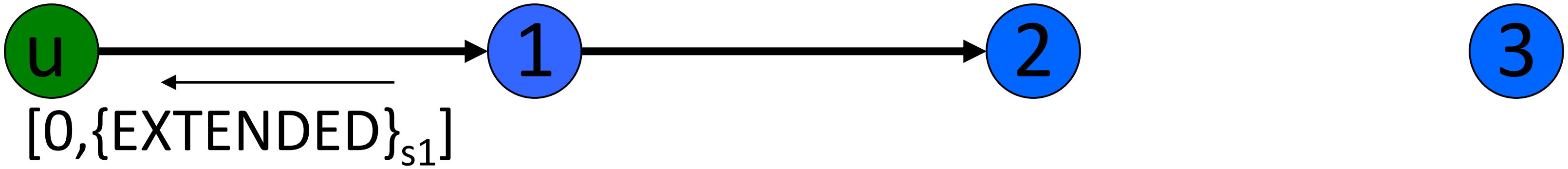
$\{m\}_{s_i}$: Encrypted using the DH session key $g^{x_i y_i}$



1. CREATE/CREATED
2. EXTEND/EXTENDED

Creating a Circuit

$\{m\}_{s_i}$: Encrypted using the DH session key $g^{x_iy_i}$

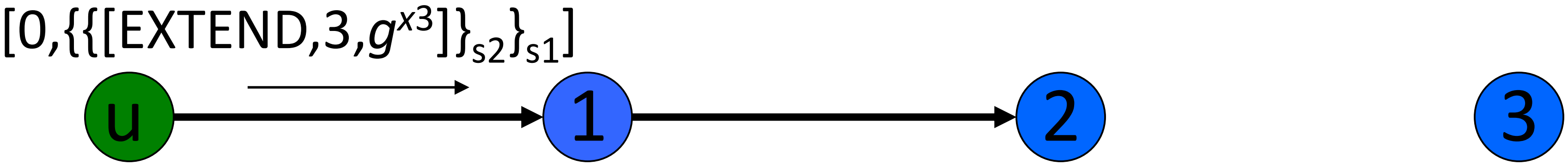


1. CREATE/CREATED

2. EXTEND/EXTENDED

Creating a Circuit

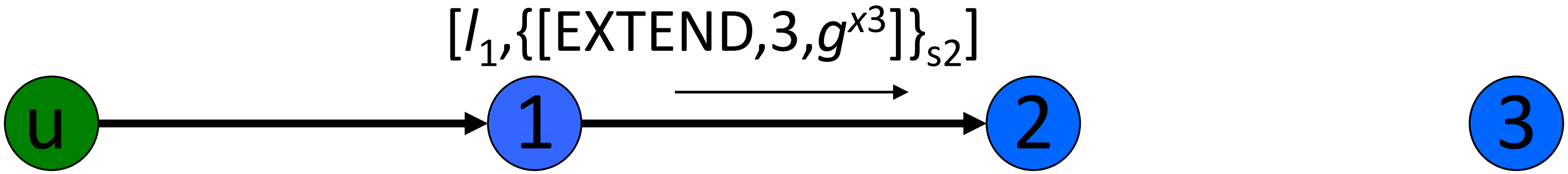
$\{m\}_{s_i}$: Encrypted using the DH session key $g^{x_i y_i}$



1. CREATE/CREATED
2. EXTEND/EXTENDED
3. [Repeat with layer of encryption]

Creating a Circuit

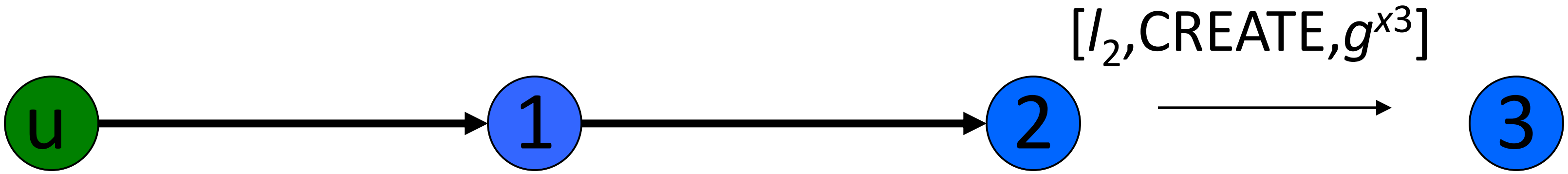
$\{m\}_{s_i}$: Encrypted using the DH session key $g^{x_i y_i}$



1. CREATE/CREATED
2. EXTEND/EXTENDED
3. [Repeat with layer of encryption]

Creating a Circuit

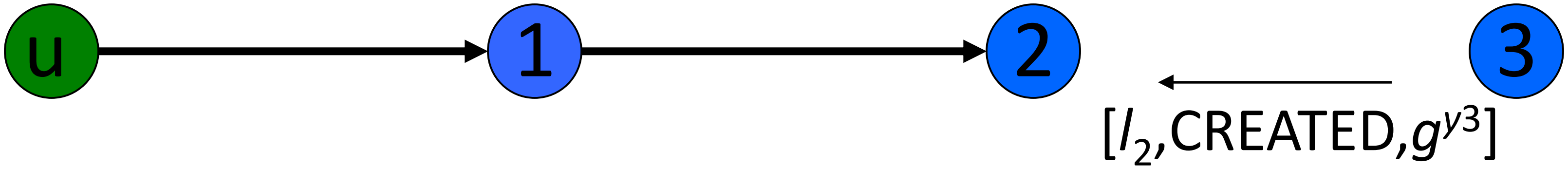
$\{m\}_{s_i}$: Encrypted using the DH session key $g^{x_i y_i}$



1. CREATE/CREATED
2. EXTEND/EXTENDED
3. [Repeat with layer of encryption]

Creating a Circuit

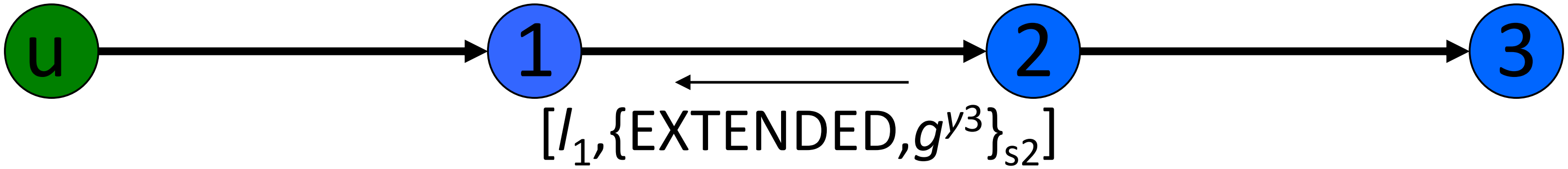
$\{m\}_{s_i}$: Encrypted using the DH session key $g^{x_i y_i}$



1. CREATE/CREATED
2. EXTEND/EXTENDED
3. [Repeat with layer of encryption]

Creating a Circuit

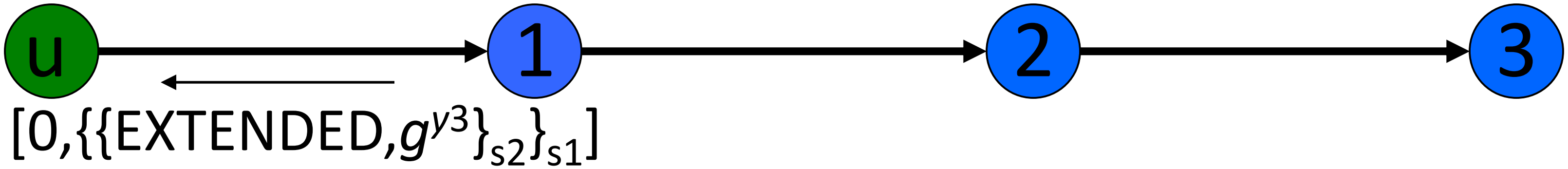
$\{m\}_{s_i}$: Encrypted using the DH session key g^{xiyi}



1. CREATE/CREATED
2. EXTEND/EXTENDED
3. [Repeat with layer of encryption]

Creating a Circuit

$\{m\}_{s_i}$: Encrypted using the DH session key $g^{x_i y_i}$



1. CREATE/CREATED
2. EXTEND/EXTENDED
3. [Repeat with layer of encryption]