U.S. NAVAL RESEARCH LABORATORY

Avoiding The Man on the Wire: Improving Tor's Security with Trust-Aware Path Selection

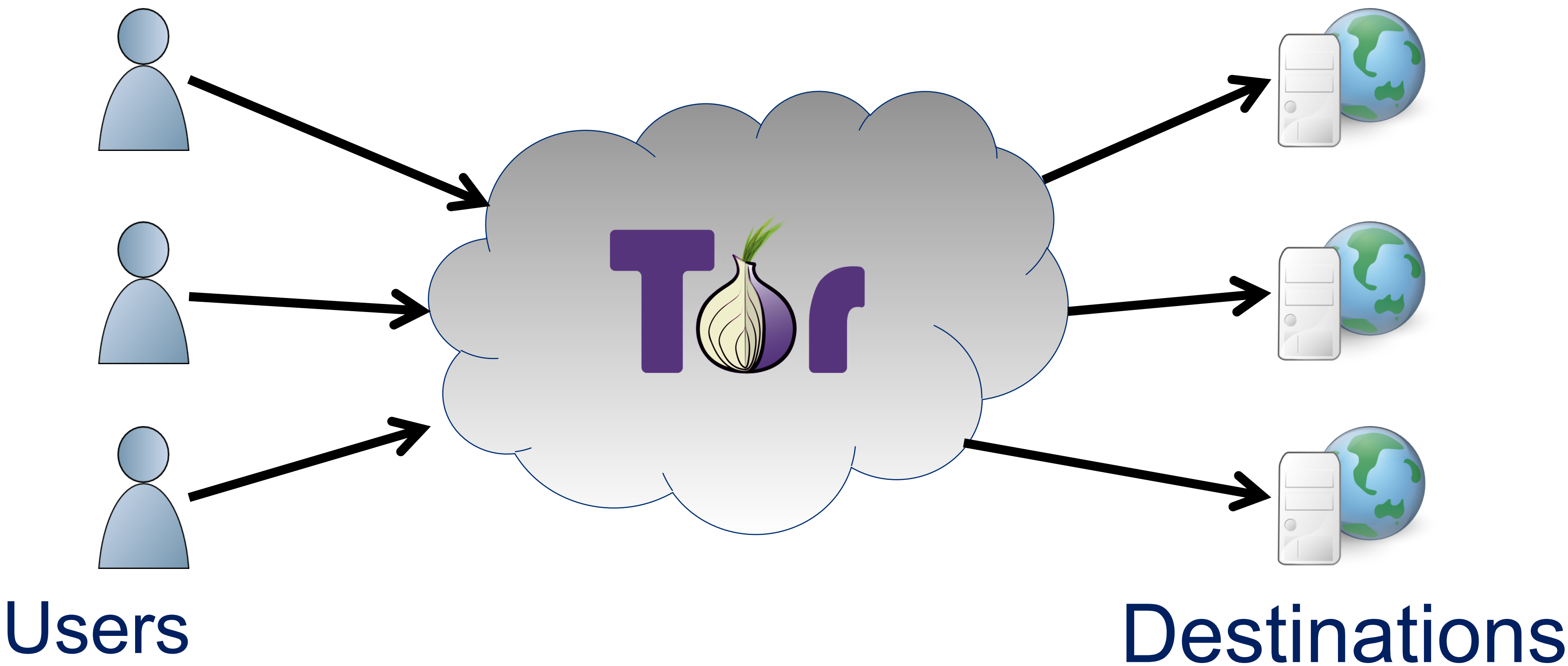
Aaron Johnson (U.S. Naval Research Laboratory)
Rob Jansen (U.S. Naval Research Laboratory)
Aaron D. Jaggard (U.S. Naval Research Laboratory)
Joan Feigenbaum (Yale University)
Paul Syverson (U.S. Naval Research Laboratory)

Febraury 28th, 2017
Network and Distributed System Security Symposium (NDSS 2017)

Talk Overview

1. Problem
2. Background
3. Attack on Prior Approach
4. Solution #1: Use Trust
5. Solution #2: Cluster
6. Trust-Aware Path Selection
7. Conclusion

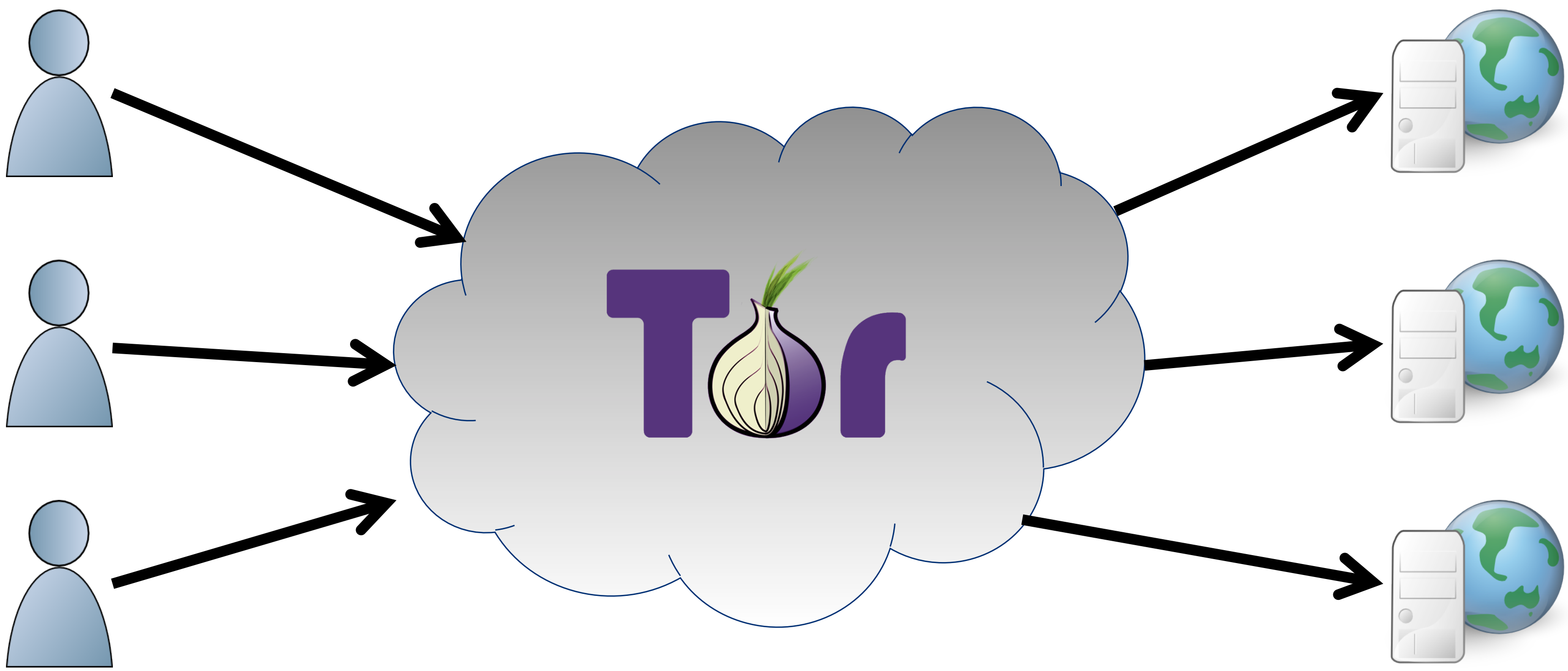
Problem



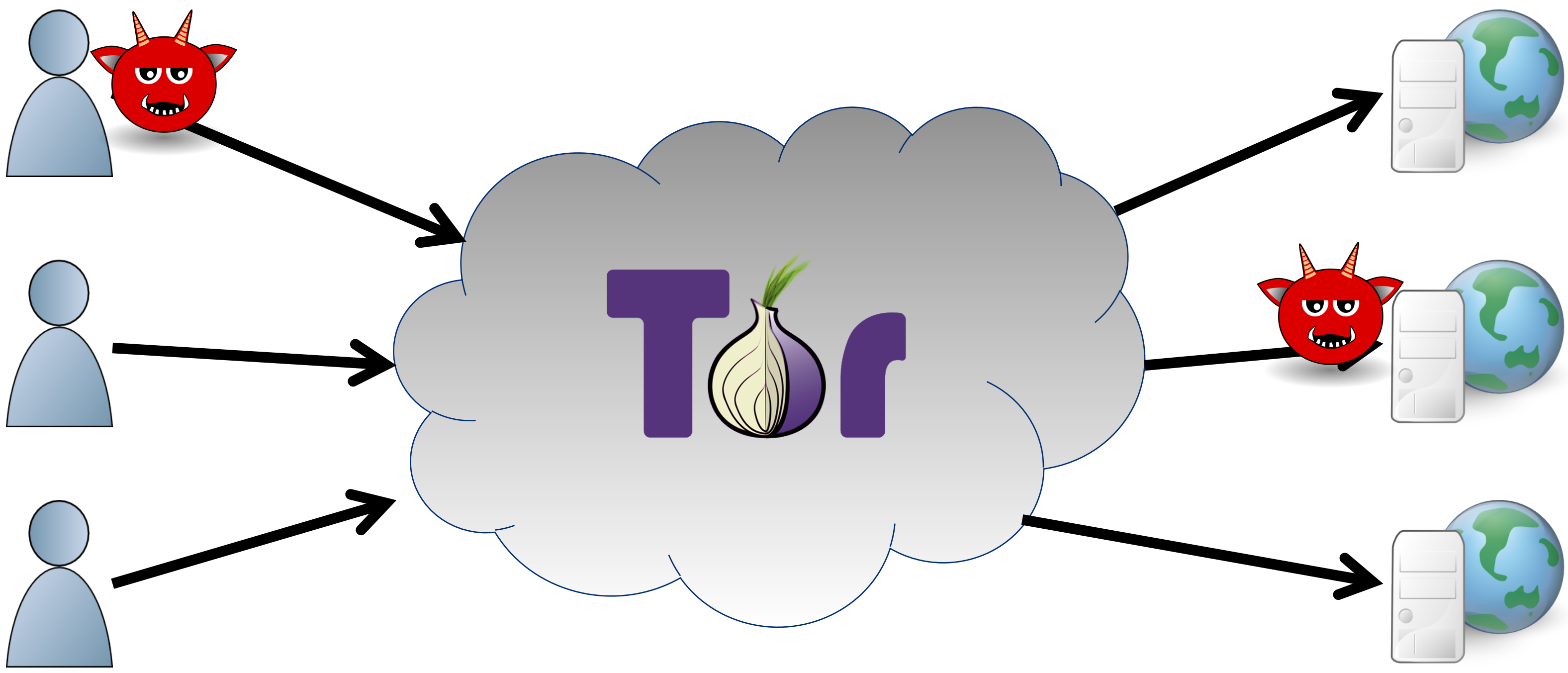
Tor is a popular system for anonymous communication.

- > 1.5 million daily users
- > 80 Gbit/s aggregate traffic

Problem

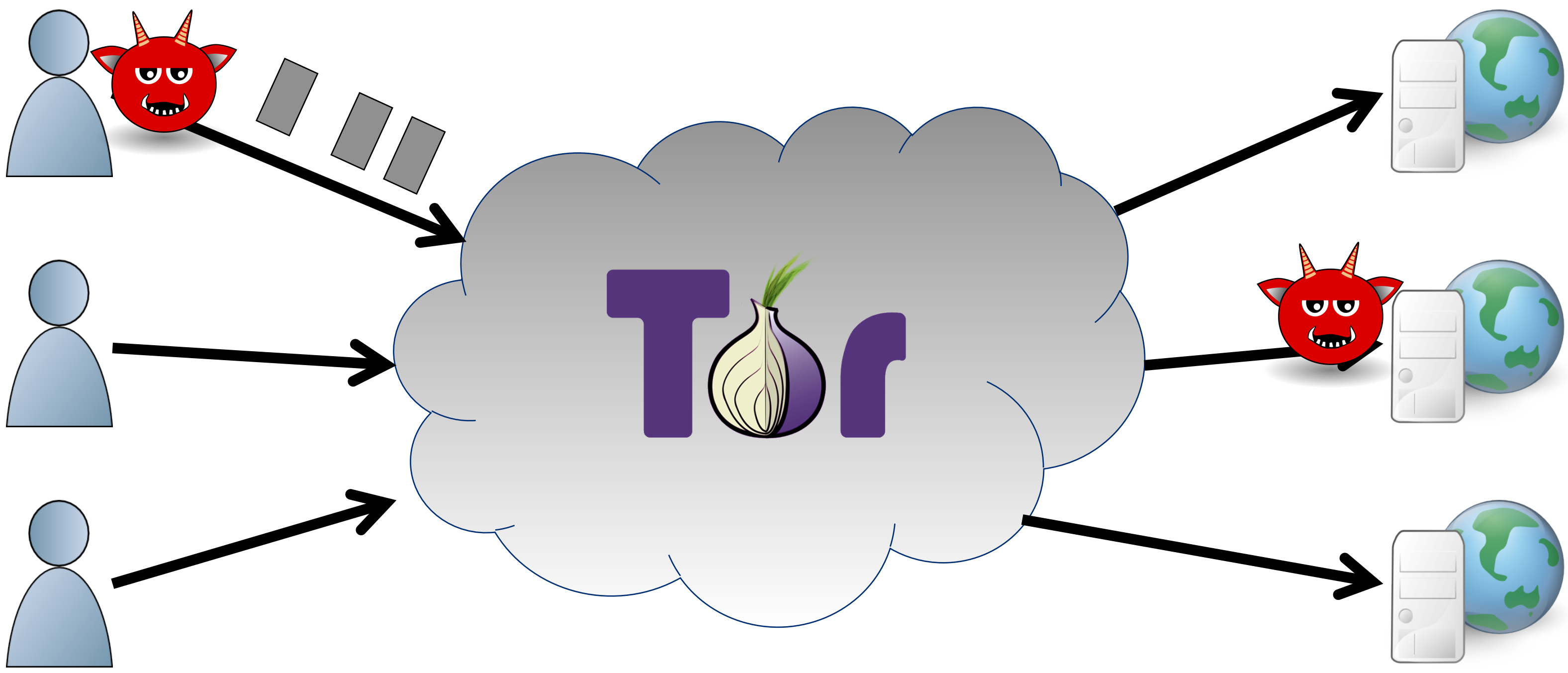


Problem

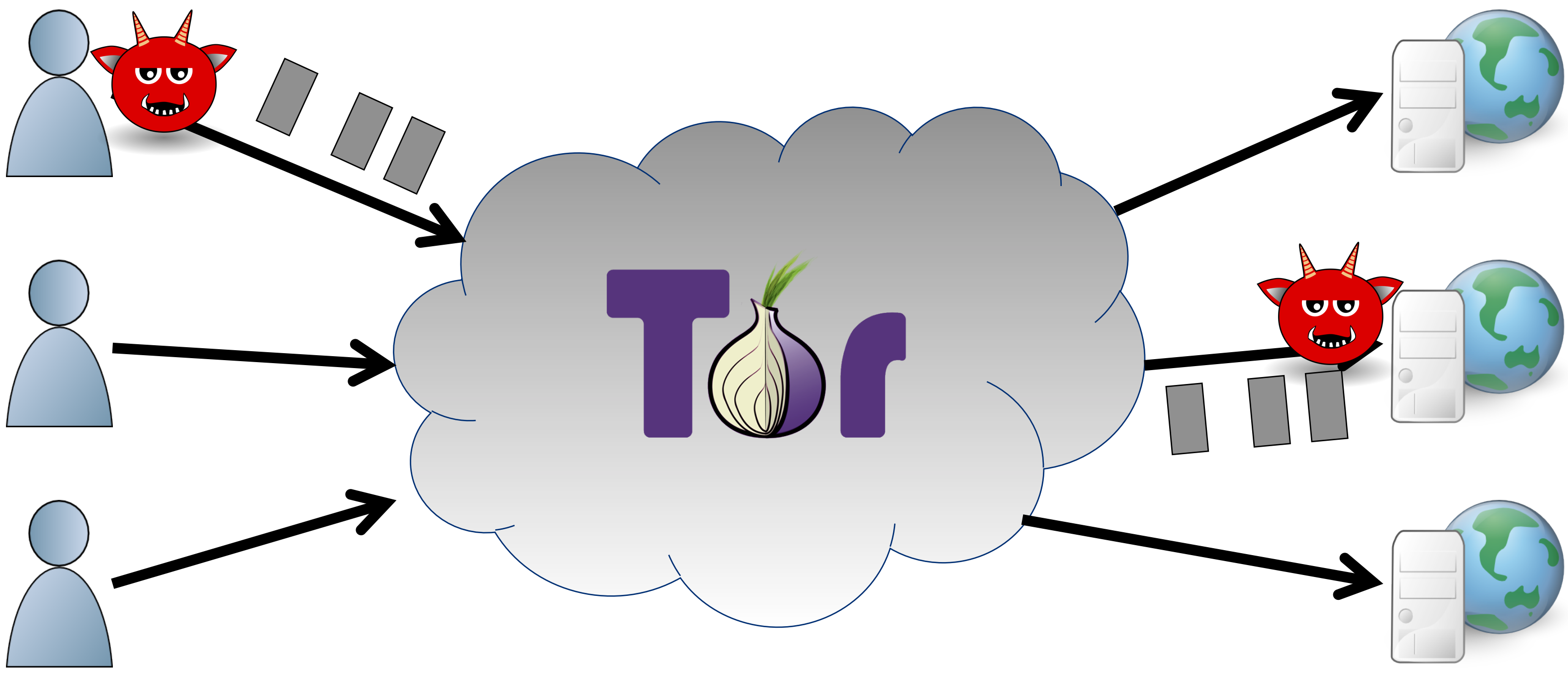


U.S. NAVAL RESEARCH LABORATORY

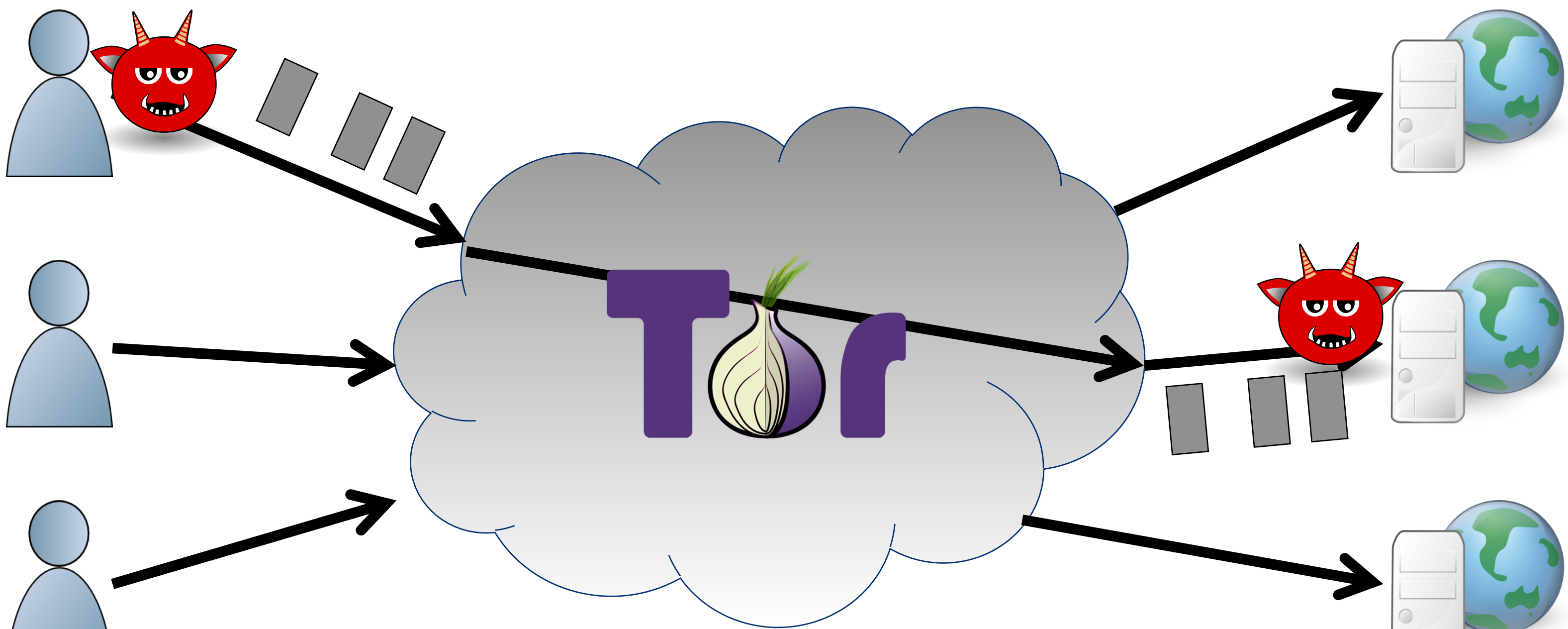
Problem



Problem

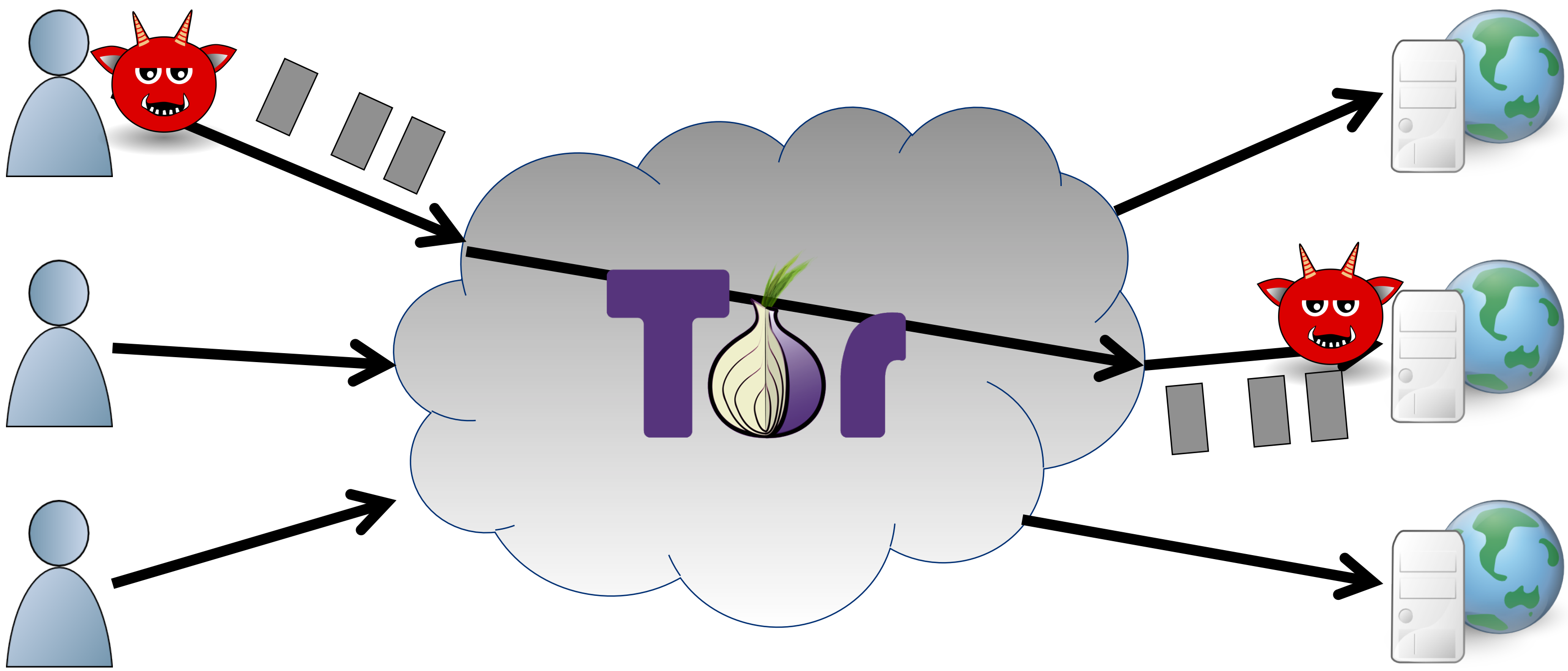


Problem



Traffic Correlation Attack

Problem



Traffic Correlation Attack

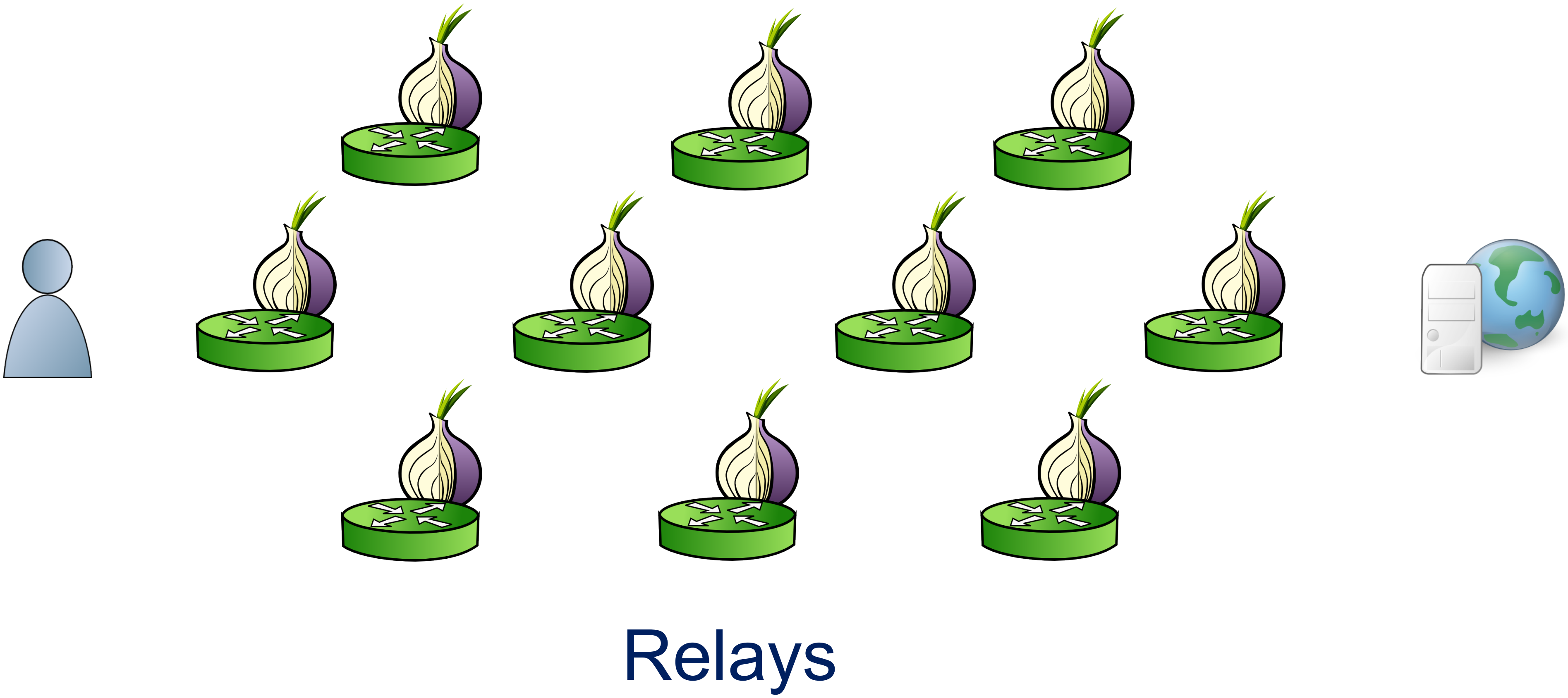
Other attacks

- Website fingerprinting
- Application-layer leaks
- Latency leaks
- Congestion attacks
- Throughput attacks
- Denial-of-Service attacks

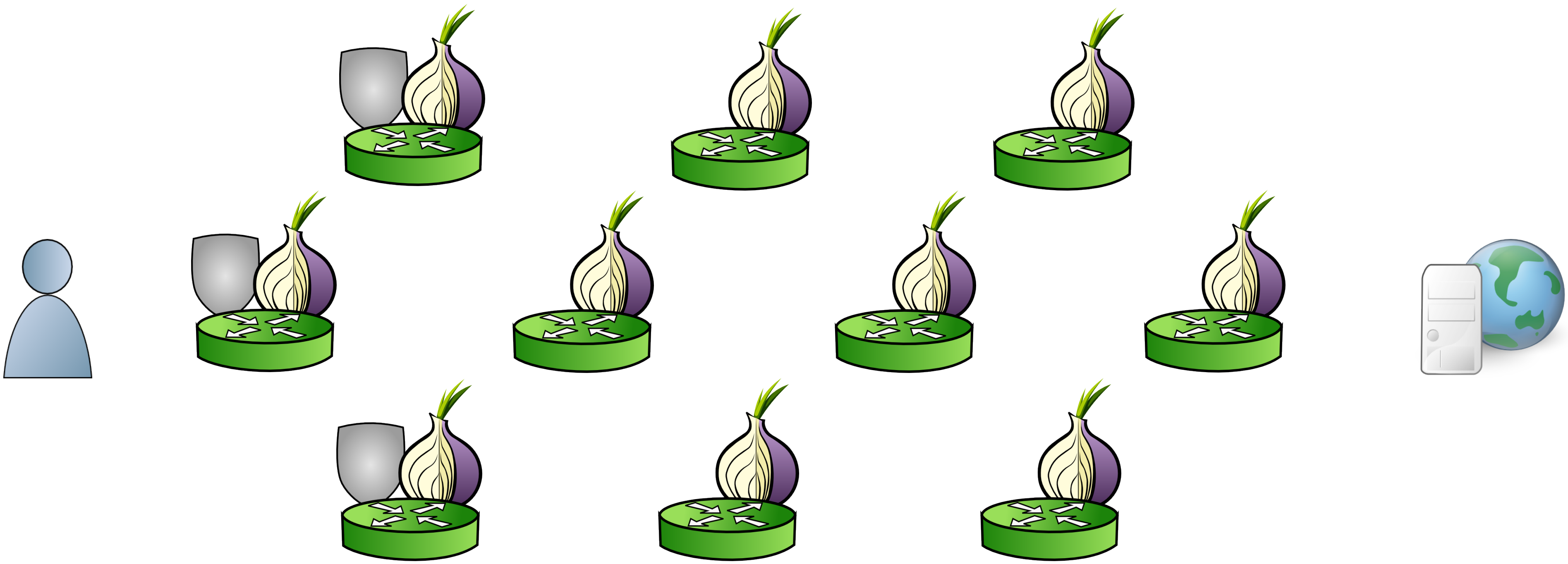
Talk Overview

1. Problem
2. Background
3. Attack on Prior Approach
4. Solution #1: Use Trust
5. Solution #2: Cluster
6. Trust-Aware Path Selection
7. Conclusion

Background: Using Circuits

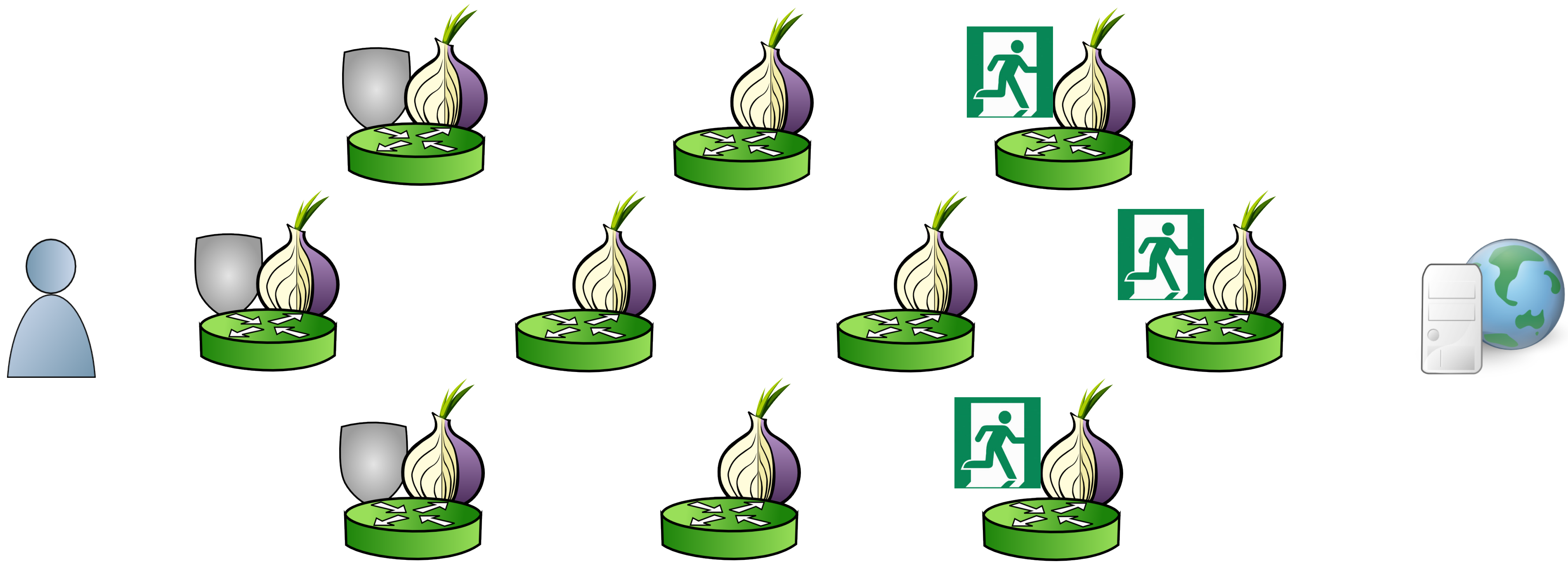


Background: Using Circuits



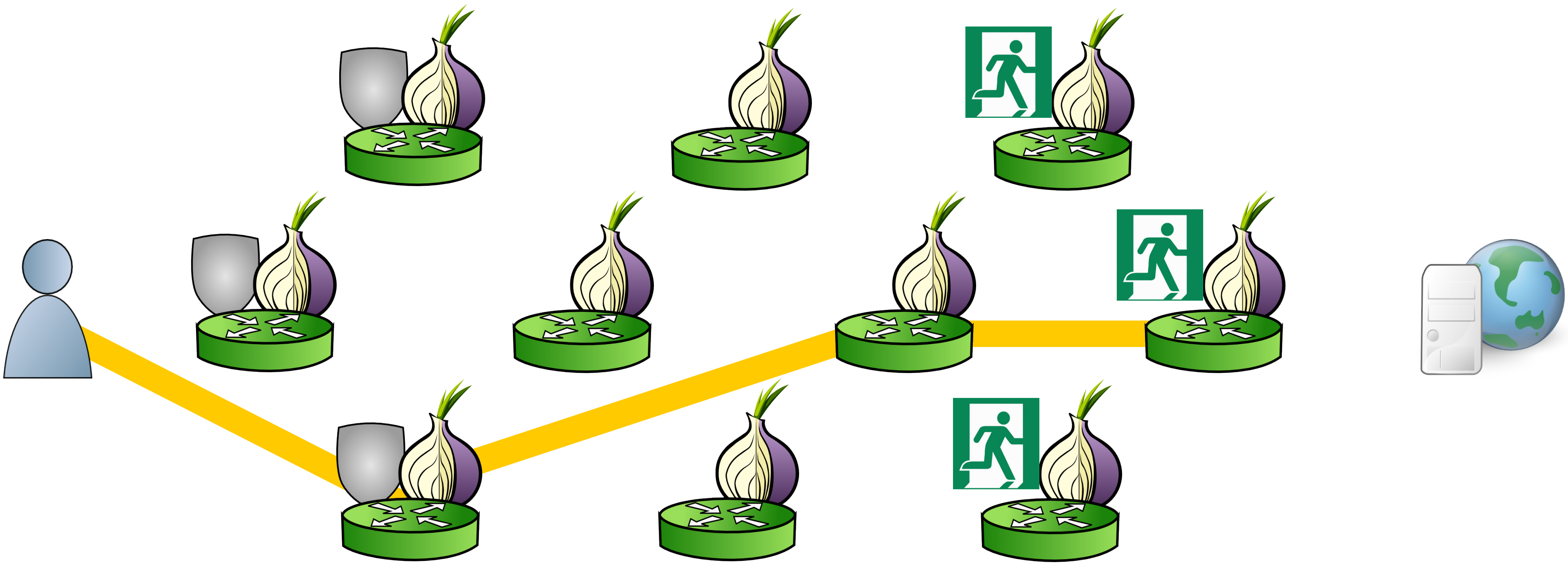
1. Clients begin all connections with a given *guard*.

Background: Using Circuits



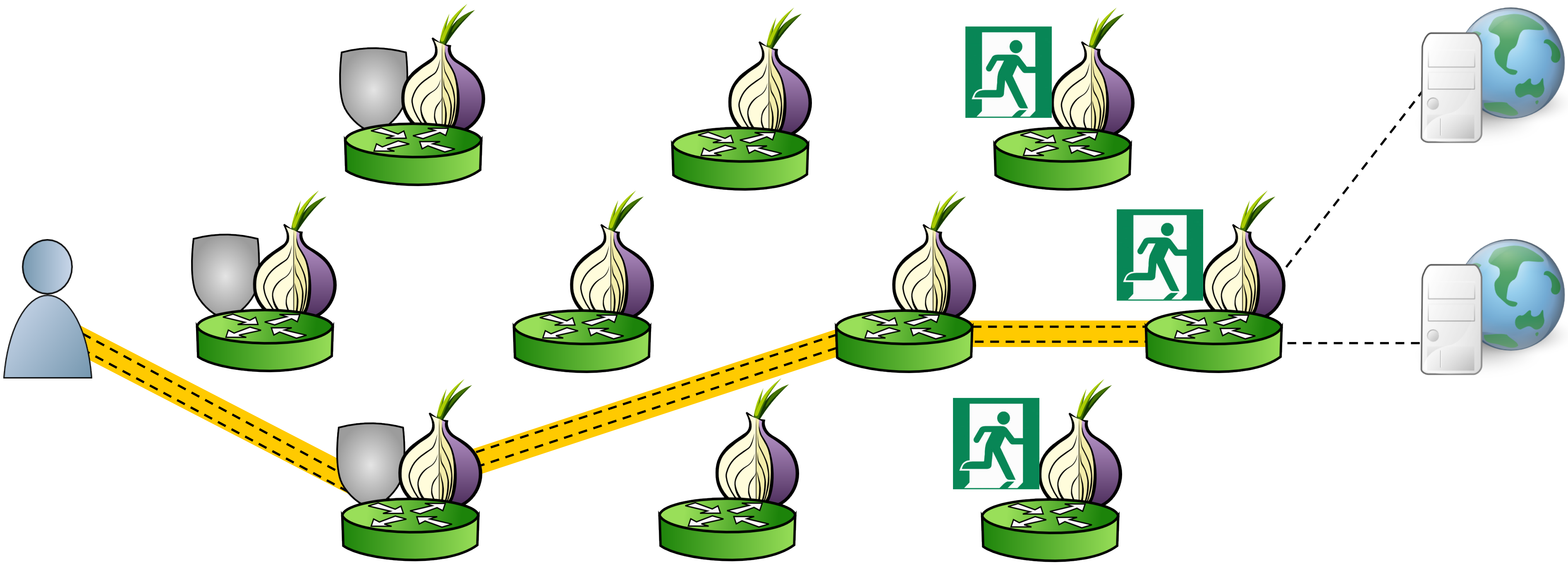
1. Clients begin all connections with a given *guard*.
2. Relays define individual *exit policies*.

Background: Using Circuits



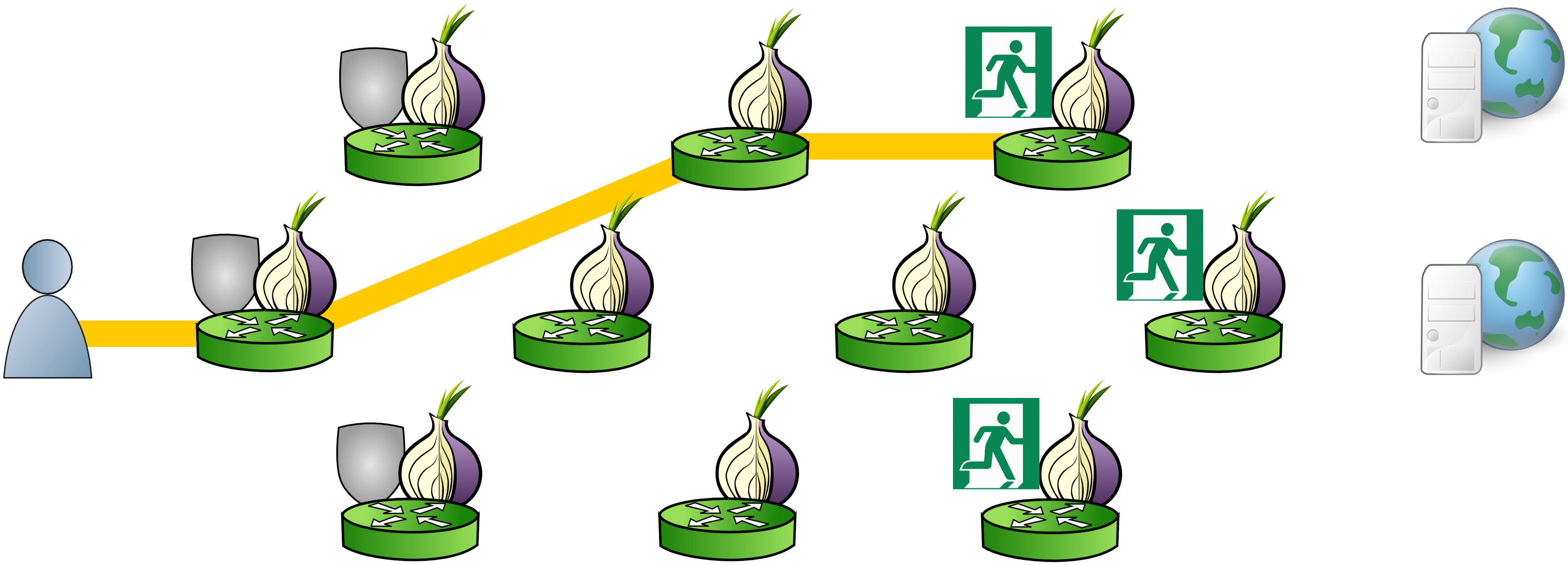
1. Clients begin all connections with a given *guard*.
2. Relays define individual *exit policies*.
3. Clients construct onion-encrypted circuits.

Background: Using Circuits



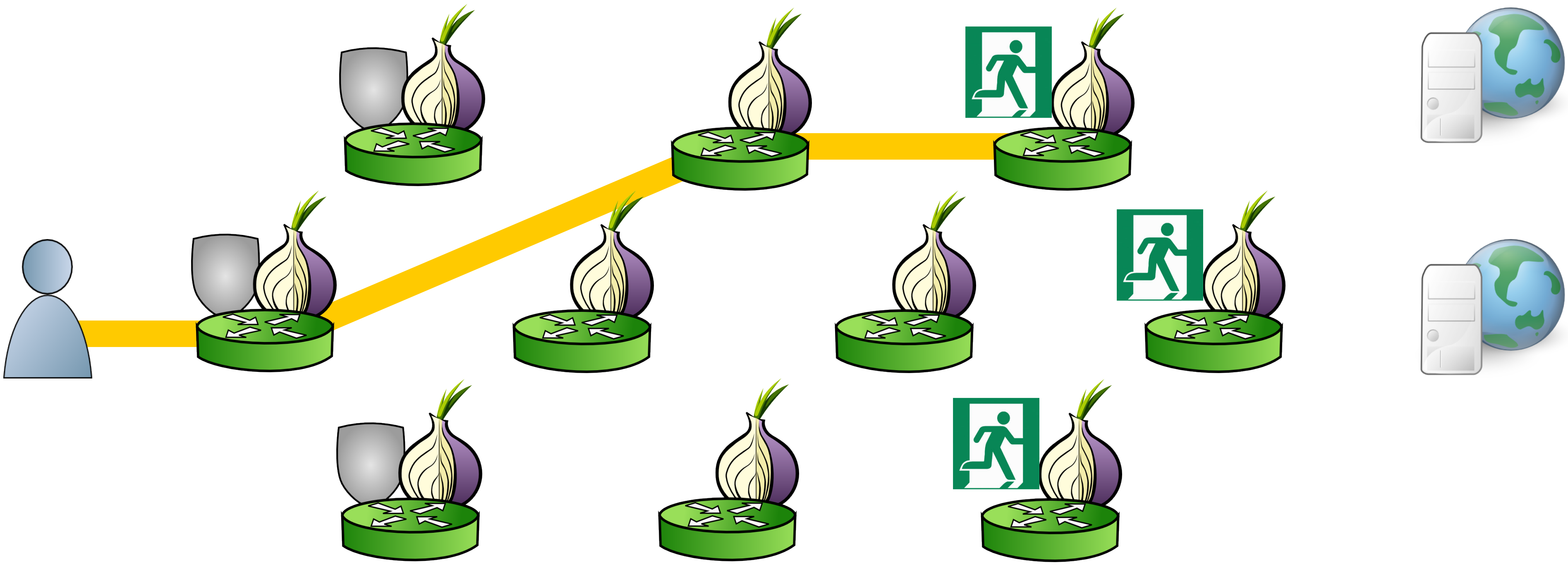
1. Clients begin all connections with a given *guard*.
2. Relays define individual *exit policies*.
3. Clients construct onion-encrypted circuits.
4. Clients multiplex *streams* over a circuit.

Background: Using Circuits



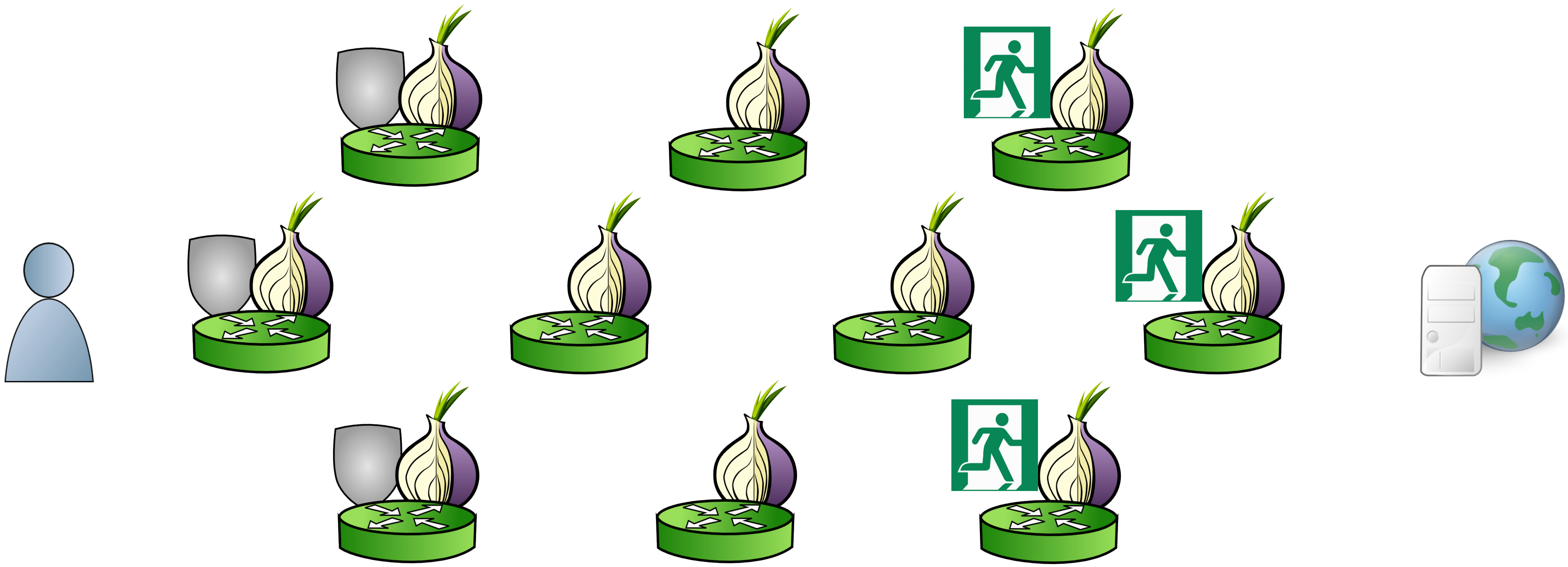
1. Clients begin all connections with a given *guard*.
2. Relays define individual *exit policies*.
3. Clients construct onion-encrypted circuits.
4. Clients multiplex *streams* over a circuit.
5. New circuits replace existing ones periodically.

Background: Using Circuits



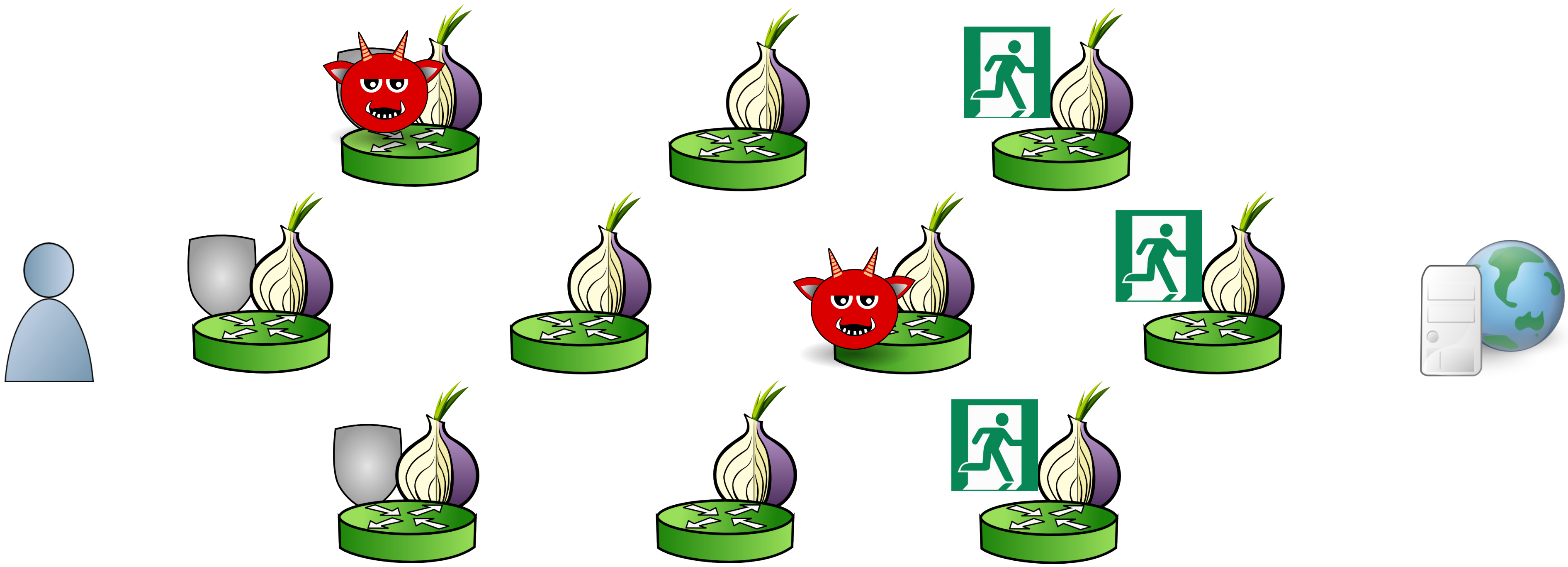
1. Clients begin all connections with a given *guard*.
2. Relays define individual *exit policies*.
3. Clients construct onion-encrypted circuits.
4. Clients multiplex *streams* over a circuit.
5. New circuits replace existing ones periodically.
6. Clients randomly choose proportional to bandwidth.

Background: Threat Model



Adversary is local and active.

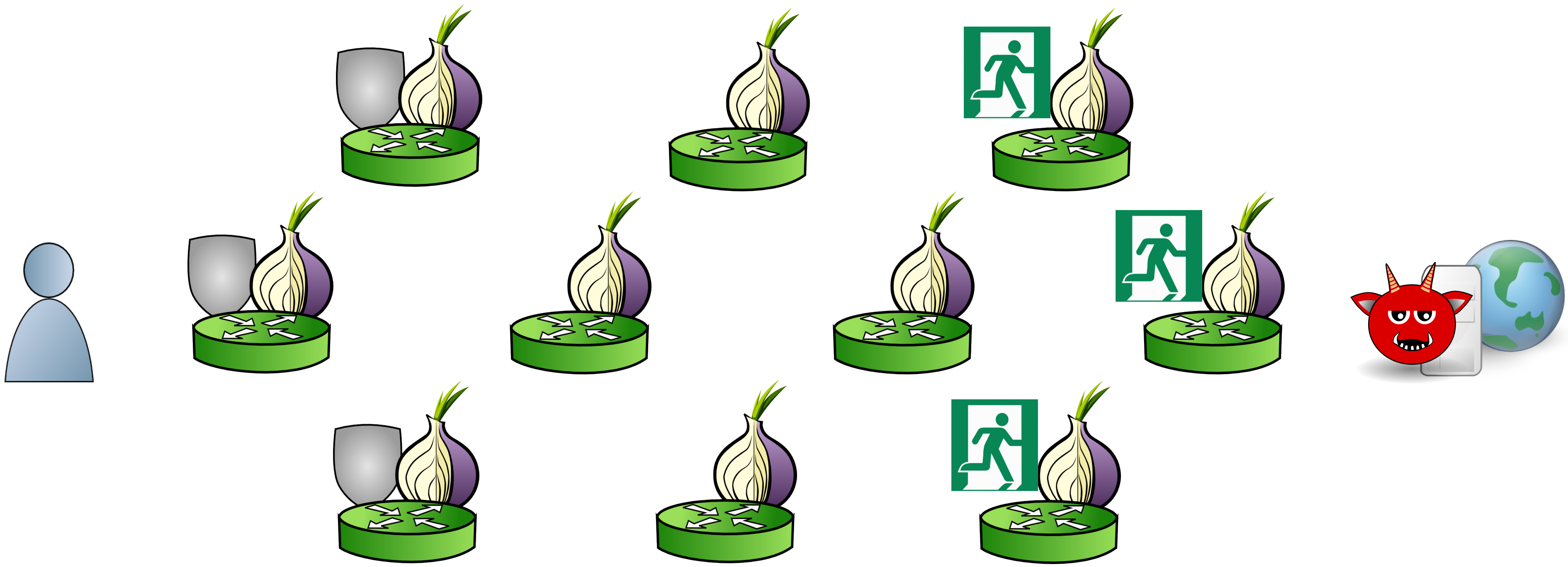
Background: Threat Model



Adversary is local and active.

- Adversary may run relays

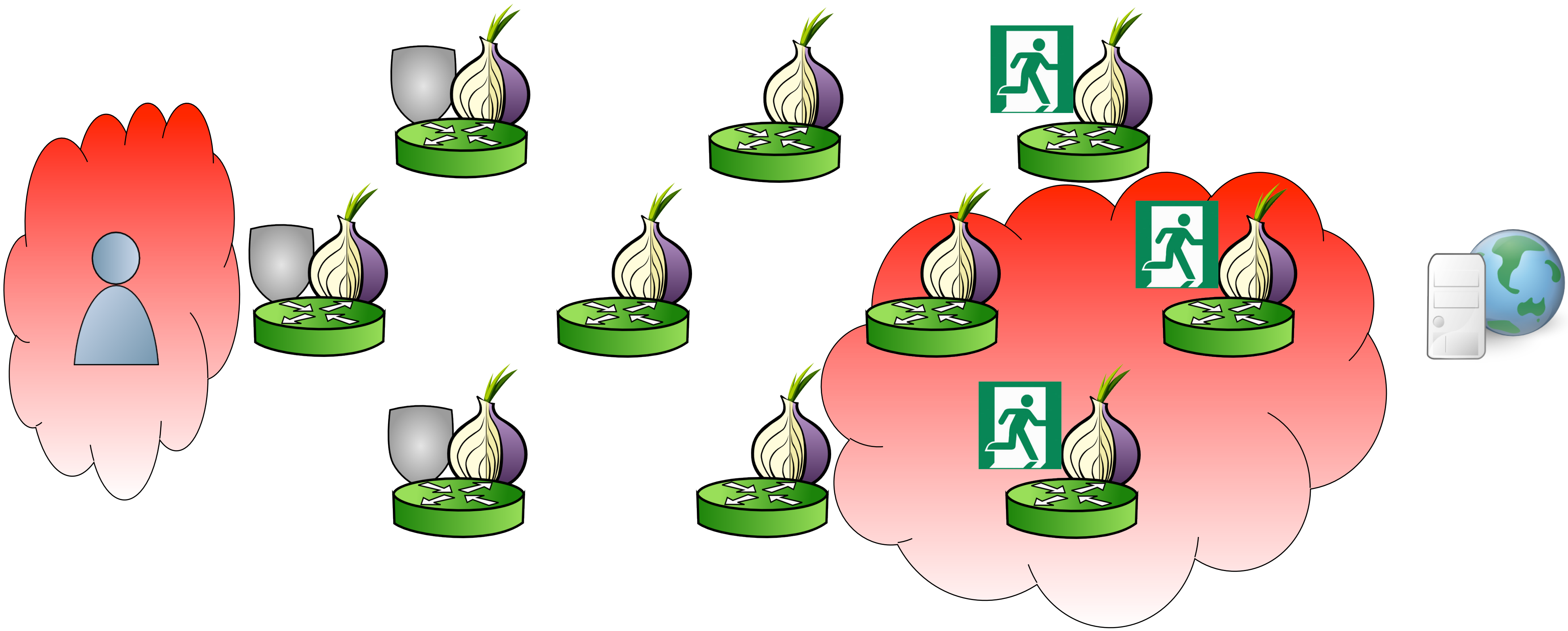
Background: Threat Model



Adversary is local and active.

- Adversary may run relays
- Adversary may run destination

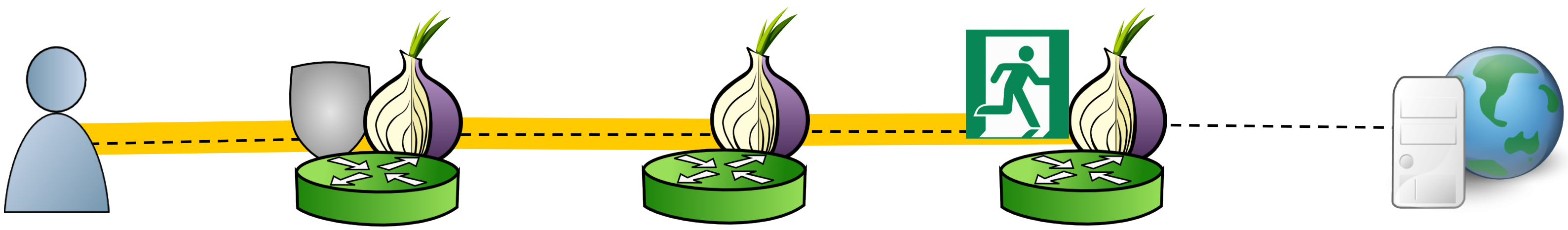
Background: Threat Model



Adversary is local and active.

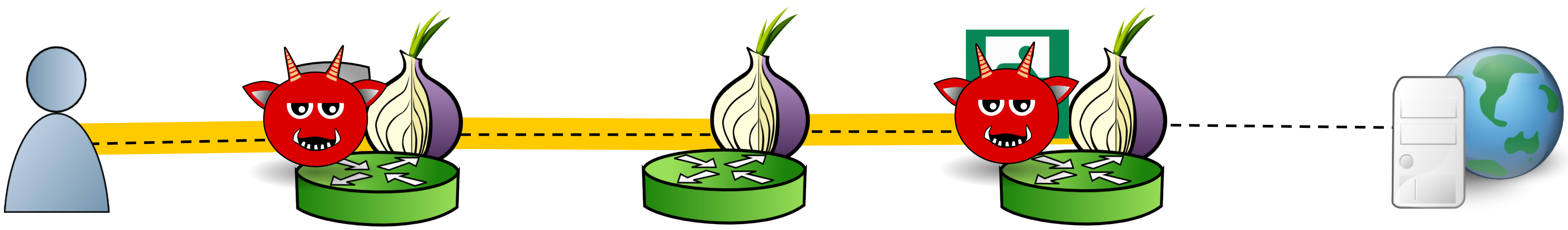
- Adversary may run relays
- Adversary may run destination
- Adversary may observe subnetworks

Background: Traffic Correlation



Traffic-correlation threats

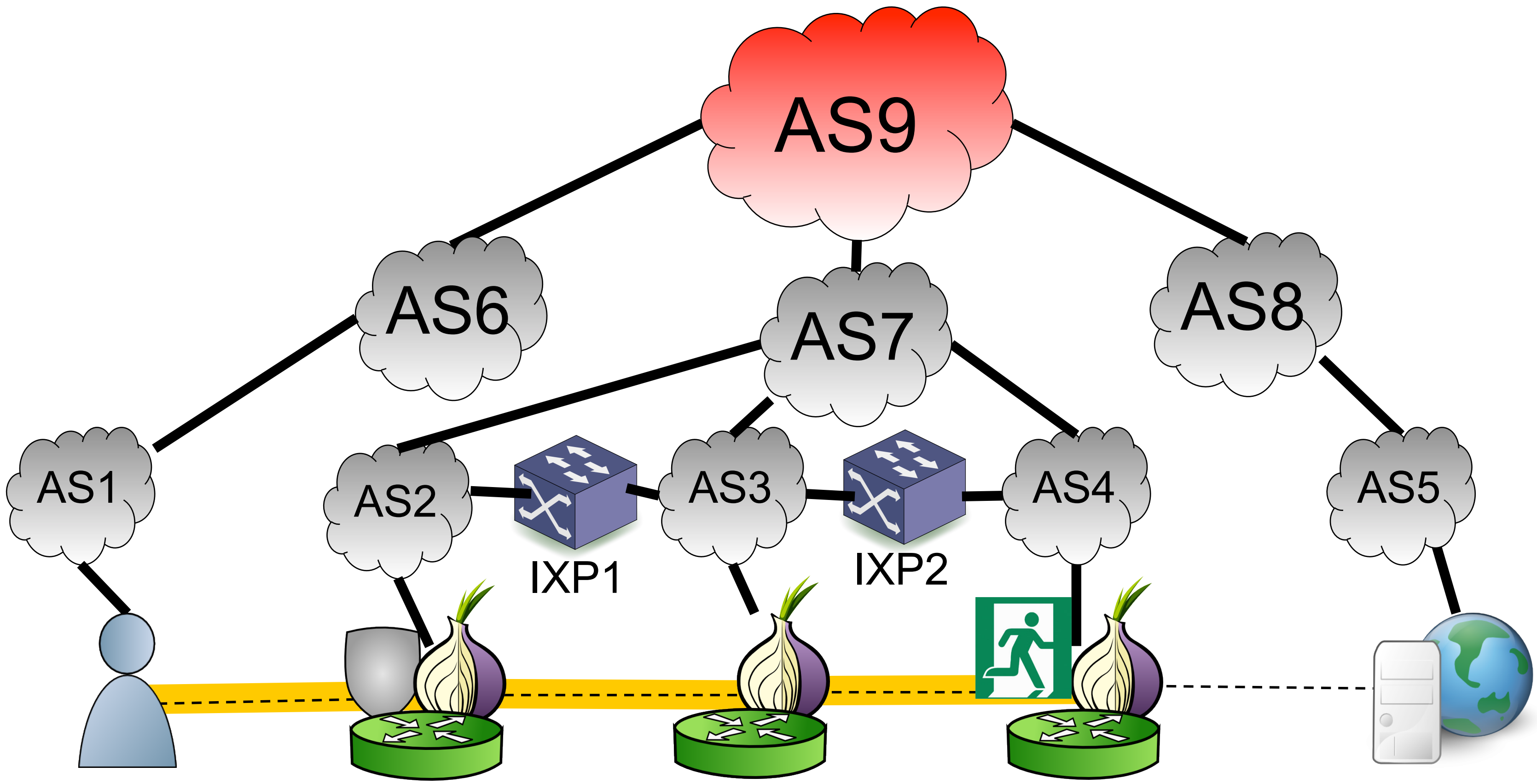
Background: Traffic Correlation



Traffic-correlation threats

- Relays

Background: Traffic Correlation



Traffic-correlation threats

- Relays
- Autonomous Systems (ASes): the networks that compose the Internet
- Internet Exchange Points (IXPs): facilities at which many ASes simultaneously connect

Talk Overview

1. Problem
2. Background
3. Attack on Prior Approach
4. Solution #1: Use Trust
5. Solution #2: Cluster
6. Trust-Aware Path Selection
7. Conclusion

Prior Approach to Prevent Traffic Correlation

Idea: Choose Tor circuits so that no single AS or IXP appears between client and guard and between exit and destination.

U.S. NAVAL RESEARCH LABORATORY

Prior Approach to Prevent Traffic Correlation

Idea: Choose Tor circuits so that no single AS or IXP appears between client and guard and between exit and destination.

1. N. Feamster and R. Dingledine, “Location diversity in anonymity networks,” in Workshop on Privacy in the Electronic Society, 2004.
2. M. Edman and P. Syverson, “AS-awareness in Tor path selection,” in ACM Conference on Computer and Communications Security, 2009.
3. M. Akhoondi, C. Yu, and H. V. Madhyastha, “LASTor: A low-latency AS-aware Tor client,” in IEEE Symposium on Security & Privacy, 2012.
4. R. Nithyanand, O. Starov, P. Gill, A. Zair, and M. Schapira, “Measuring and mitigating AS-level adversaries against Tor,” in Network & Distributed System Security Symposium, 2016.

Prior Approach to Prevent Traffic Correlation

Idea: Choose Tor circuits so that no single AS or IXP appears between client and guard and between exit and destination.

1. N. Feamster and R. Dingledine, “Location diversity in anonymity networks,” in Workshop on Privacy in the Electronic Society, 2004.
2. M. Edman and P. Syverson, “AS-awareness in Tor path selection,” in ACM Conference on Computer and Communications Security, 2009.
3. M. Akhoondi, C. Yu, and H. V. Madhyastha, “LASTor: A low-latency AS-aware Tor client,” in IEEE Symposium on Security & Privacy, 2012.
4. R. Nithyanand, O. Starov, P. Gill, A. Zair, and M. Schapira, “Measuring and mitigating AS-level adversaries against Tor,” in Network & Distributed System Security Symposium, 2016.

U.S. NAVAL RESEARCH LABORATORY

Prior Approach to Prevent Traffic Correlation

Astoria [Nithyanand et al. 2016]:

1. For new circuit, consider all pairs of guards and exits
    a. If pair exists without same AS on both sides, choose randomly among such pairs proportionally to bandwidth
    b. Else, choose pairs to minimize the maximum probability that any given AS can perform traffic correlation
2. Reuse existing circuit created for destination in same AS

Prior Approach to Prevent Traffic Correlation

Astoria [Nithyanand et al. 2016]:

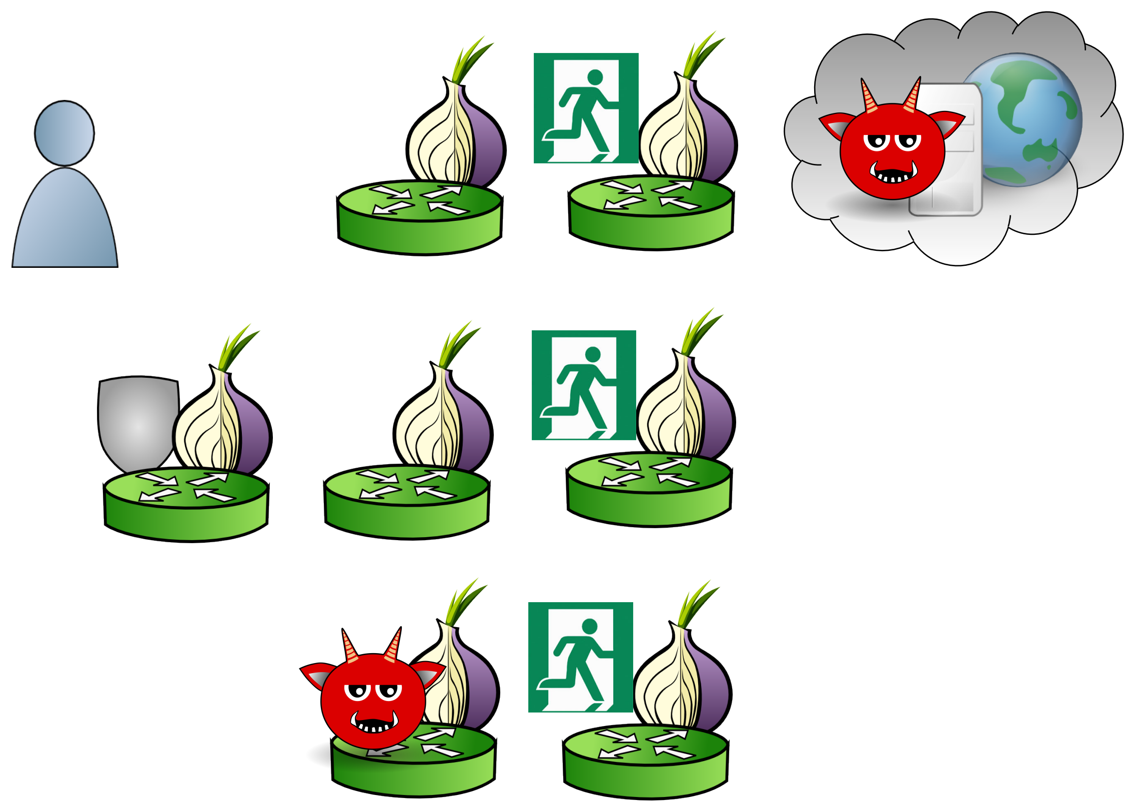
1. For new circuit, consider all pairs of guards and exits
   a. If pair exists without same AS on both sides, choose randomly among such pairs proportionally to bandwidth
   b. Else, choose pairs to minimize the maximum probability that any given AS can perform traffic correlation
2. Reuse existing circuit created for destination in same AS

Problems:

1. Adversaries need not only observe at an AS.
2. Location-based path selection leaks information about client and destination locations.

Chosen-Destination Attack

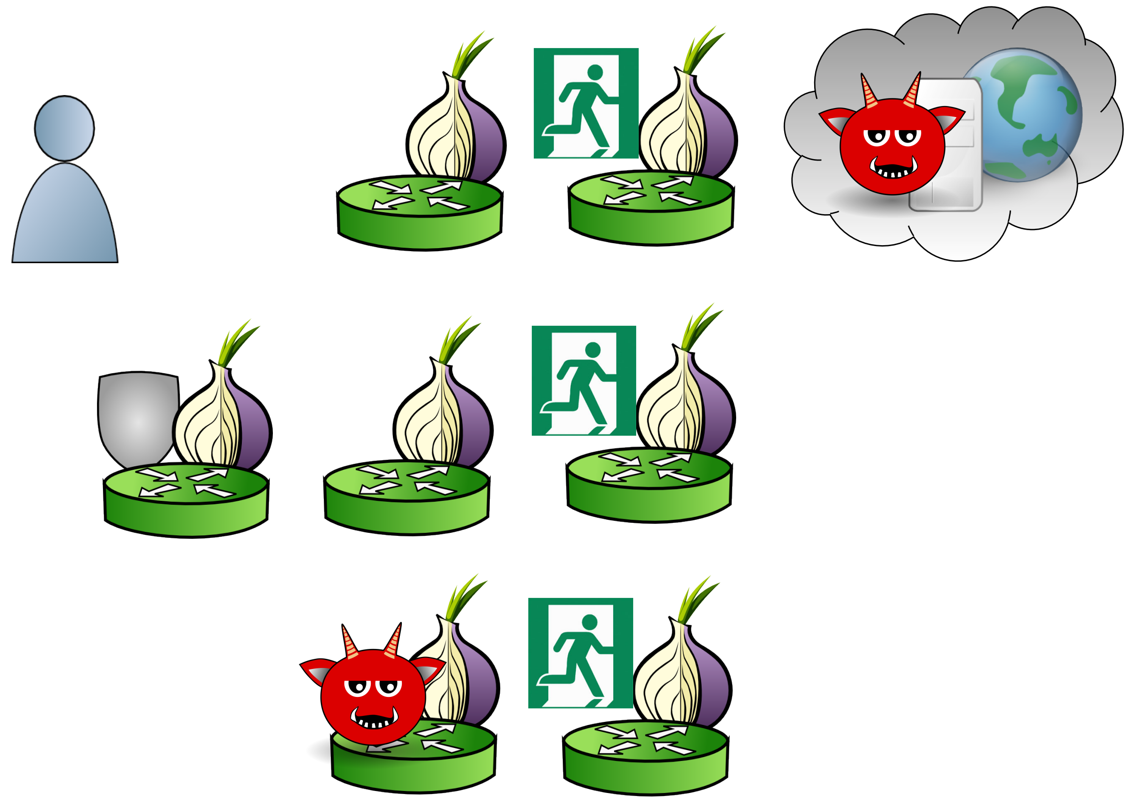
Chosen-Destination Attack on Astoria



Chosen-Destination Attack

Chosen-Destination Attack on Astoria

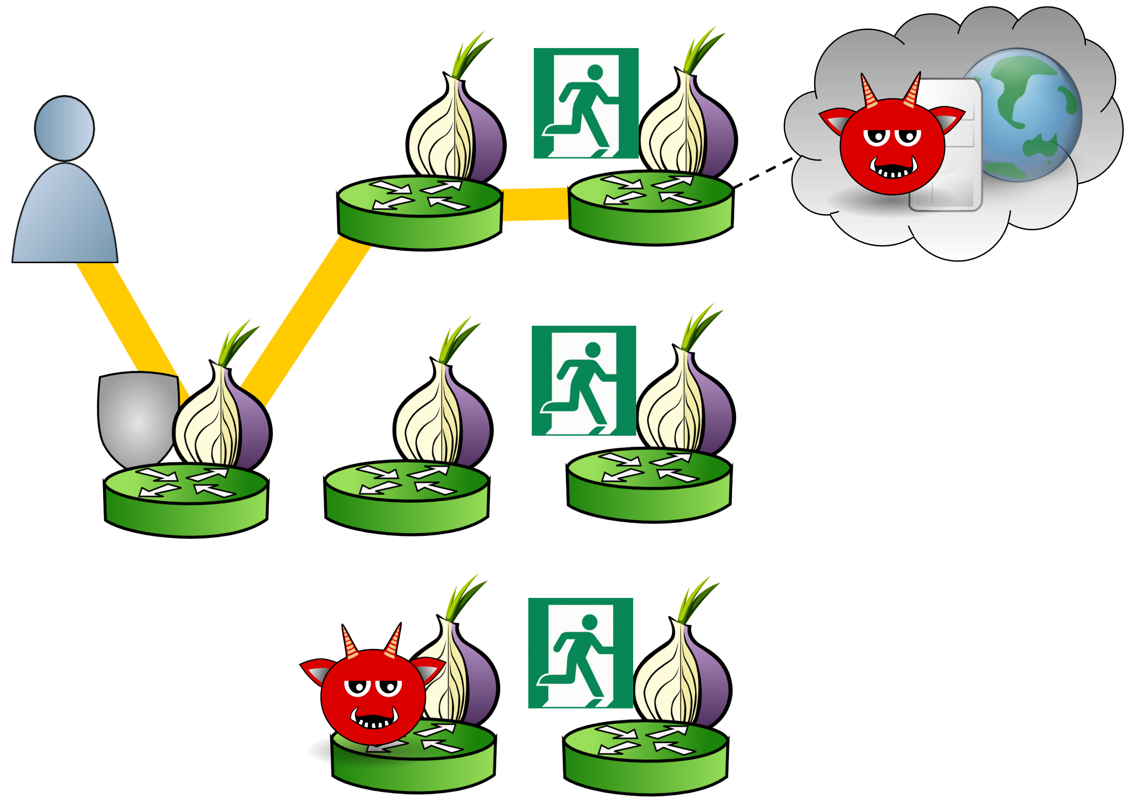
1. Client makes initial connection to malicious website.



Chosen-Destination Attack

Chosen-Destination Attack on Astoria

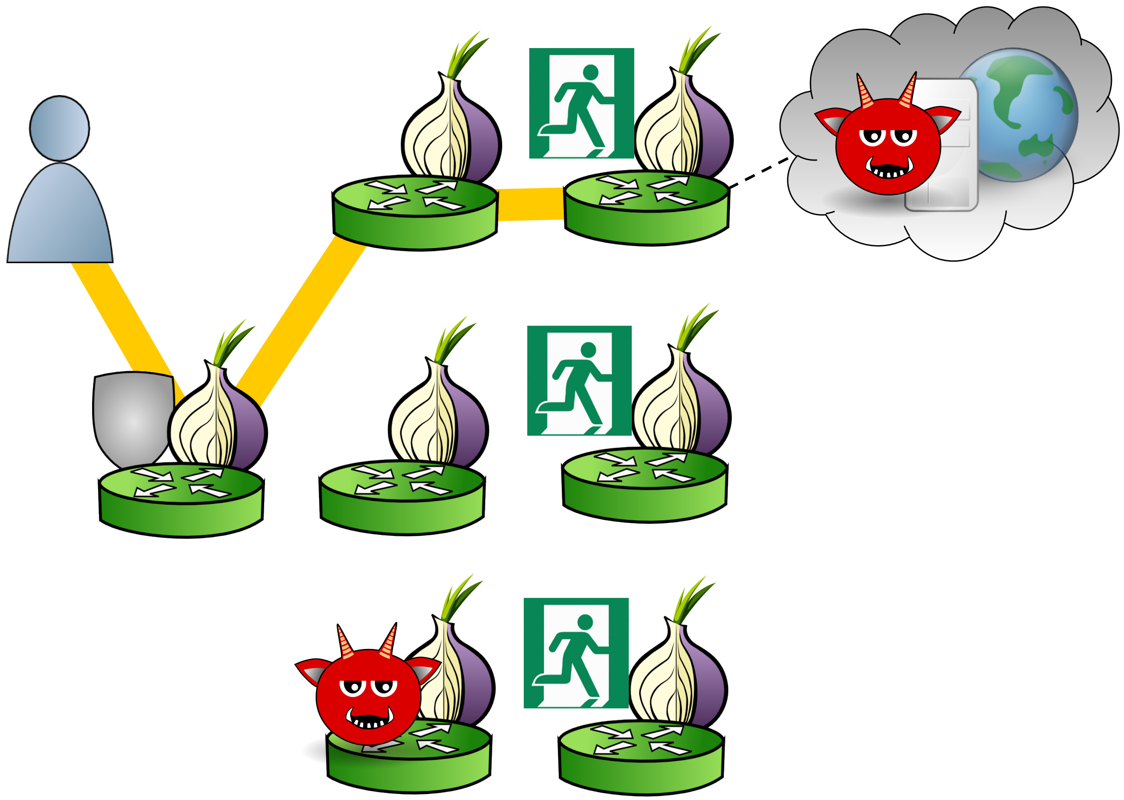
1. Client makes initial connection to malicious website.



Chosen-Destination Attack

Chosen-Destination Attack on Astoria

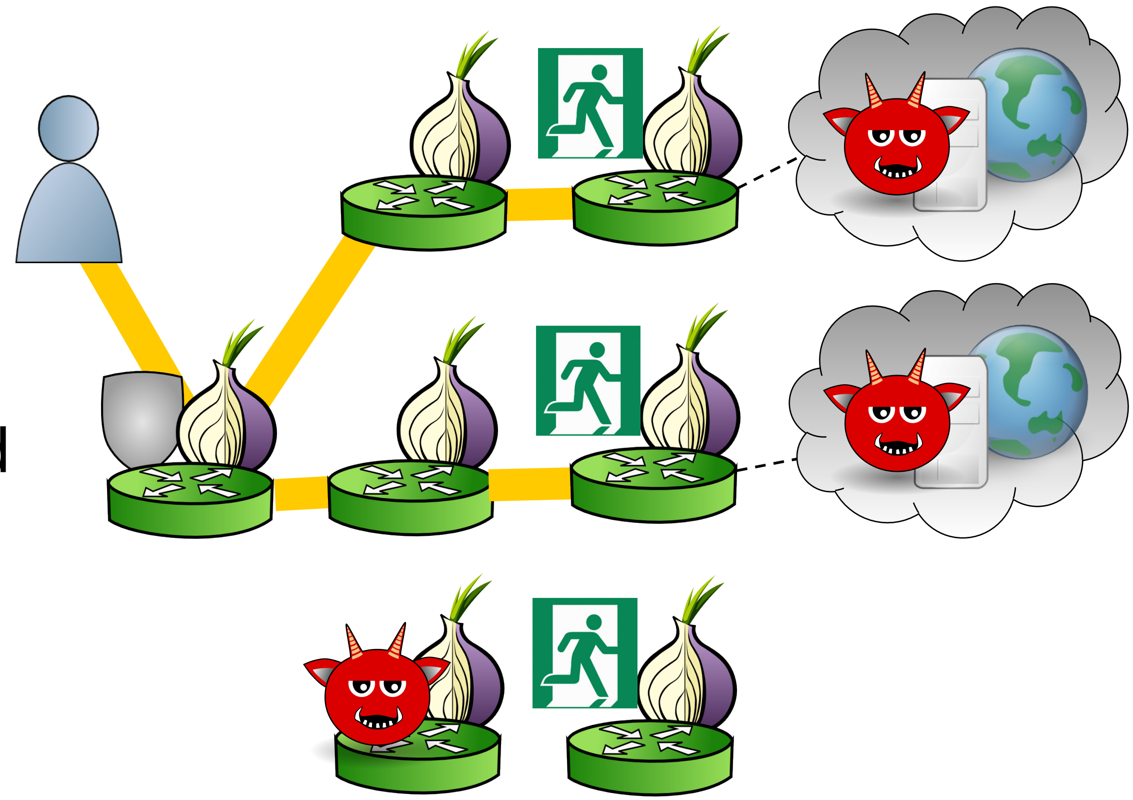
1. Client makes initial connection to malicious website.
2. Client connects to sequence of malicious servers in other ASes to download resources linked in webpage.



Chosen-Destination Attack

Chosen-Destination Attack on Astoria

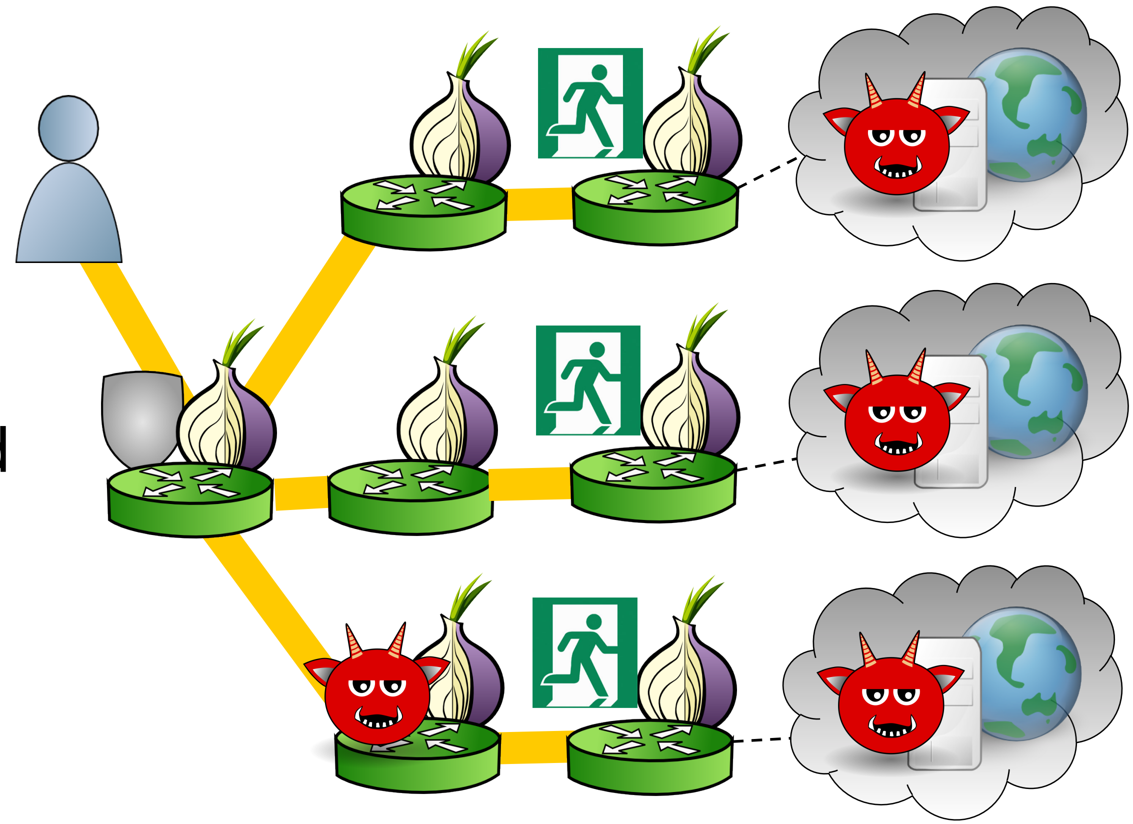
1. Client makes initial connection to malicious website.
2. Client connects to sequence of malicious servers in other ASes to download resources linked in webpage.



Chosen-Destination Attack

Chosen-Destination Attack on Astoria

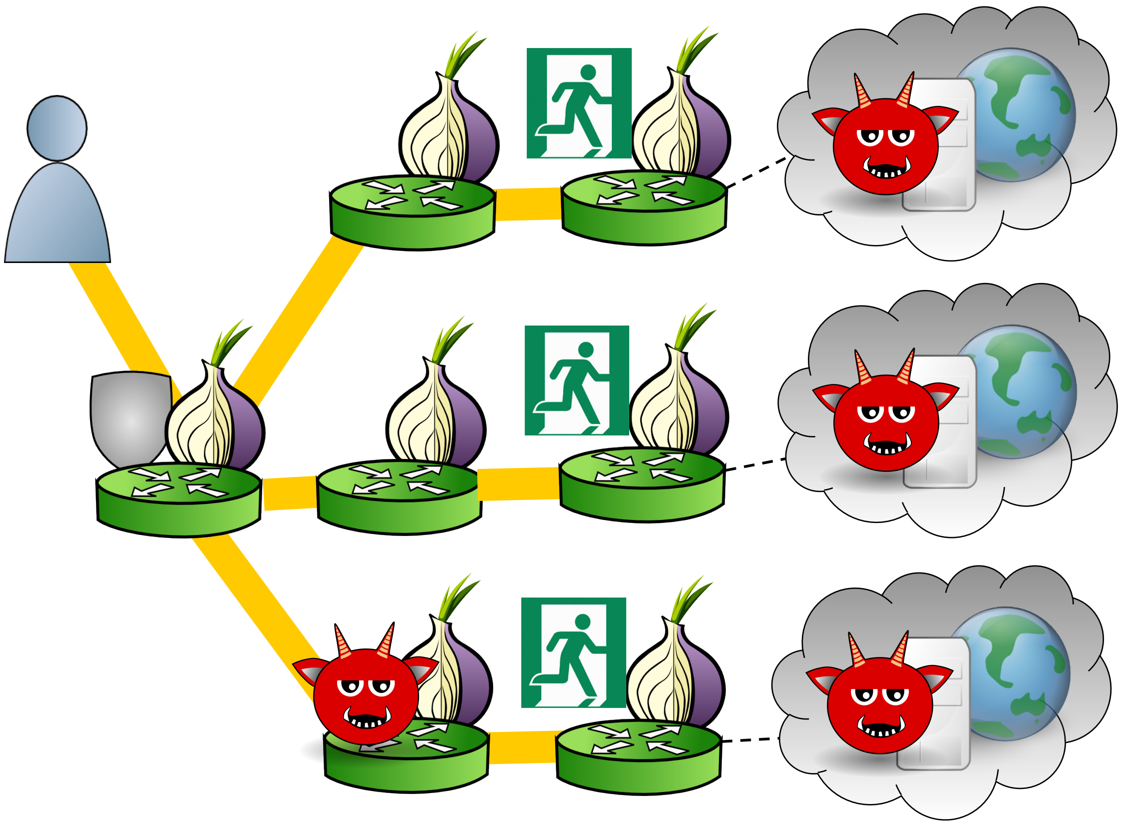
1. Client makes initial connection to malicious website.
2. Client connects to sequence of malicious servers in other ASes to download resources linked in webpage.



Chosen-Destination Attack

Chosen-Destination Attack on Astoria

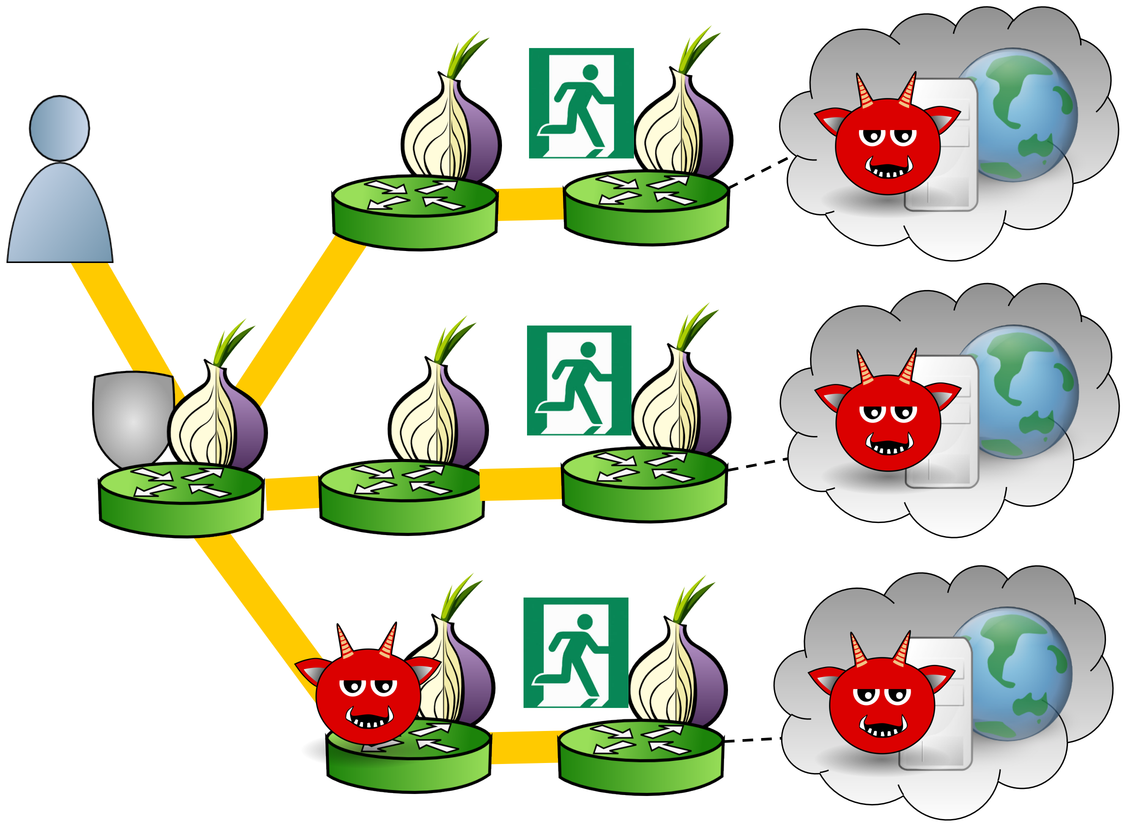
1. Client makes initial connection to malicious website.
2. Client connects to sequence of malicious servers in other ASes to download resources linked in webpage.
3. Client eventually reveals guard(s) by choosing malicious middle relay.



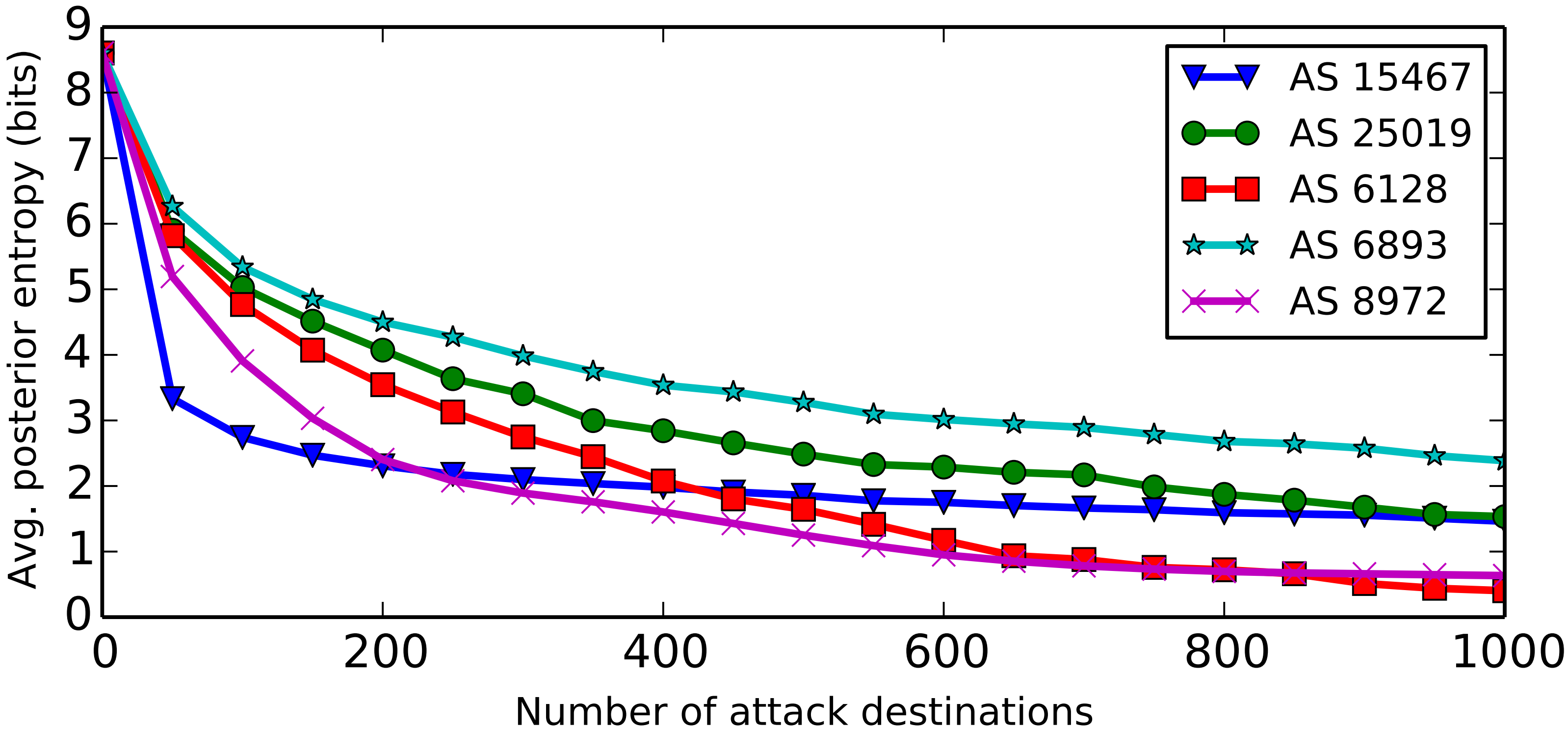
Chosen-Destination Attack

Chosen-Destination Attack on Astoria

1. Client makes initial connection to malicious website.
2. Client connects to sequence of malicious servers in other ASes to download resources linked in webpage.
3. Client eventually reveals guard(s) by choosing malicious middle relay.
4. Guard(s) and pattern of exits leaks client AS.



Chosen-Destination Attack



- 5 popular Tor client ASes
- Entropy over 400 popular Tor client ASes vs. number of random attack destination ASes
- Attack can succeed in seconds

Talk Overview

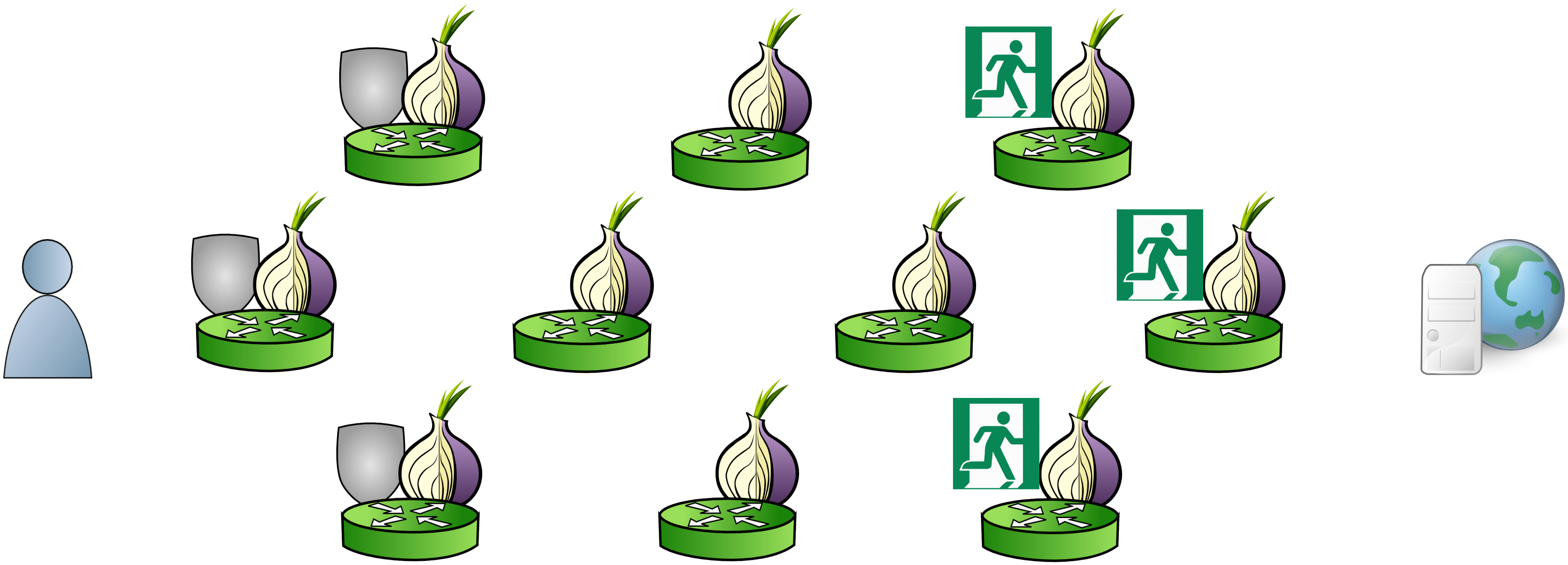
1. Problem
2. Background
3. Attack on Prior Approach
4. Solution #1: Use Trust
5. Solution #2: Cluster
6. Trust-Aware Path Selection
7. Conclusion

Network Trust as a Solution

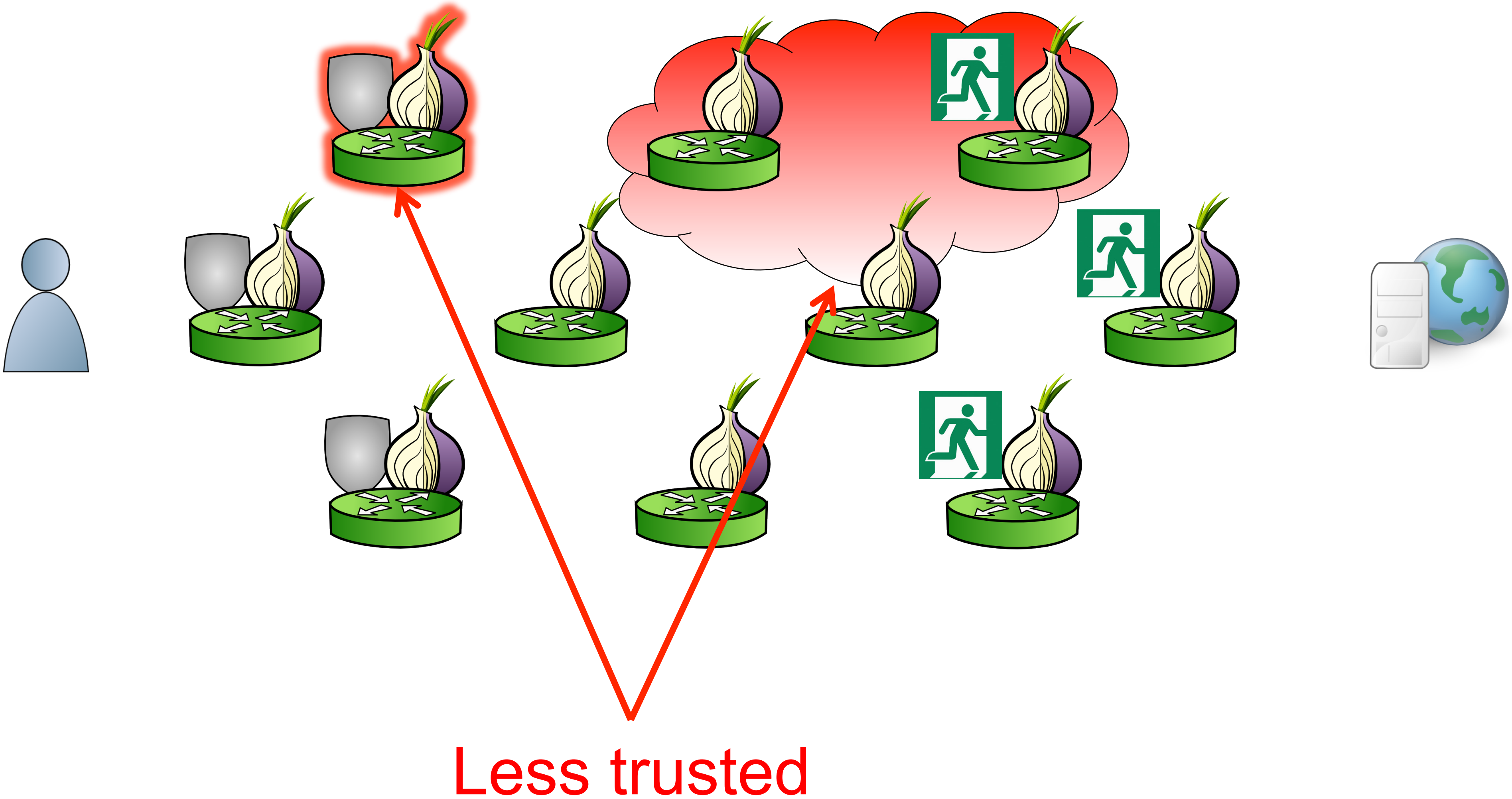
Problem: Adversaries need not only observe at an AS.

U.S. NAVAL RESEARCH LABORATORY

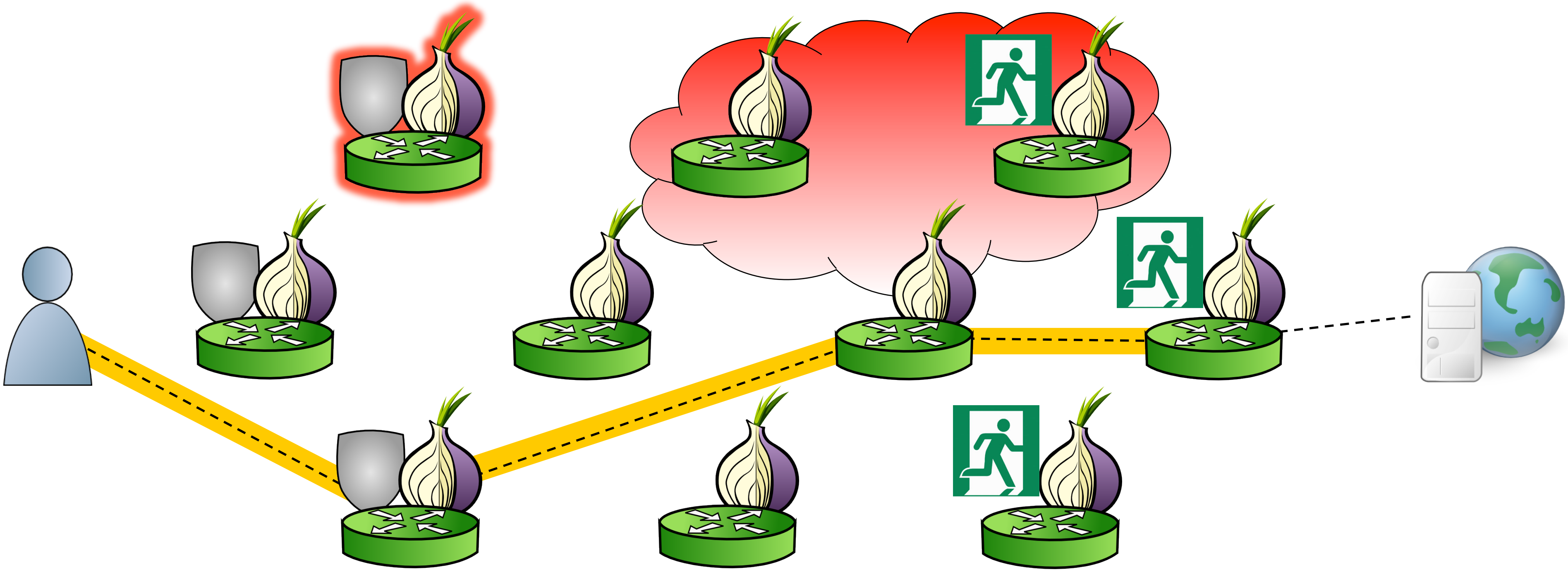
Network Trust as a Solution



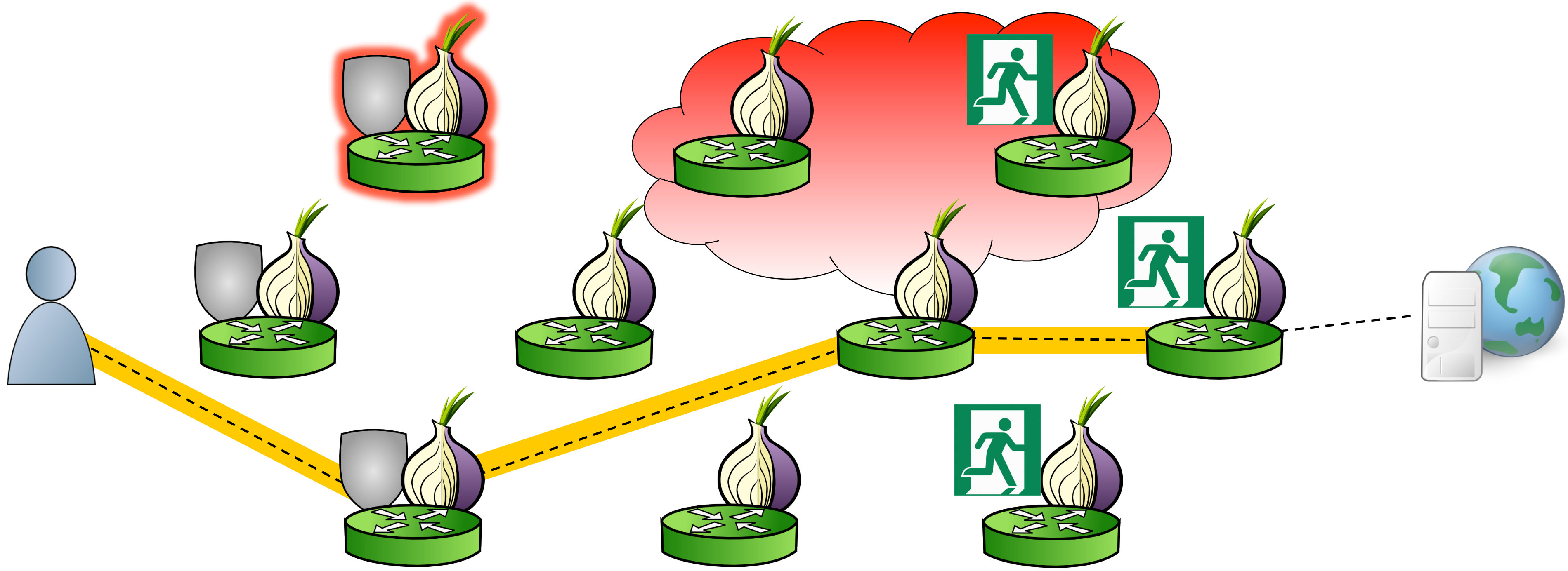
Network Trust as a Solution



Network Trust as a Solution



Network Trust as a Solution



Trust belief: probability distribution on adversary *location*

- Tor relays
- *Virtual links*: client-guard and destination-exit links

Trust policy:

- Trust belief per adversary
- Weight per adversary indicating concern level

Network Trust as a Solution

A.D. Jaggard, A. Johnson, S. Cortes, P. Syverson, and J. Feigenbaum, "20,000 In League Under the Sea: Anonymous Communication, Trust, MLATs, and Undersea Cables", In Proceedings on Privacy Enhancing Technologies, Vol. 2015, Number 1, April 2015.

Trust Factors

- Relays: operator, uptime, country
- Links: AS, IXP, undersea cable, country

Trust Sources

- Default (provided by Tor)
- Trusted authorities (e.g. EFF)
- Social networks

Talk Overview

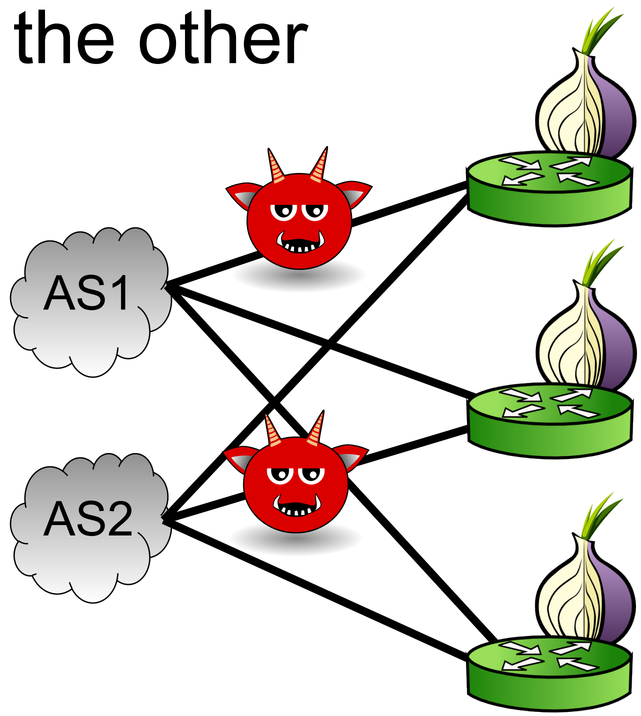
1. Problem
2. Background
3. Attack on Prior Approach
4. Solution #1: Use Trust
5. Solution #2: Cluster
6. Trust-Aware Path Selection
7. Conclusion

Cluster Locations

Problem: Location-based path selection leaks information about client and destination locations.

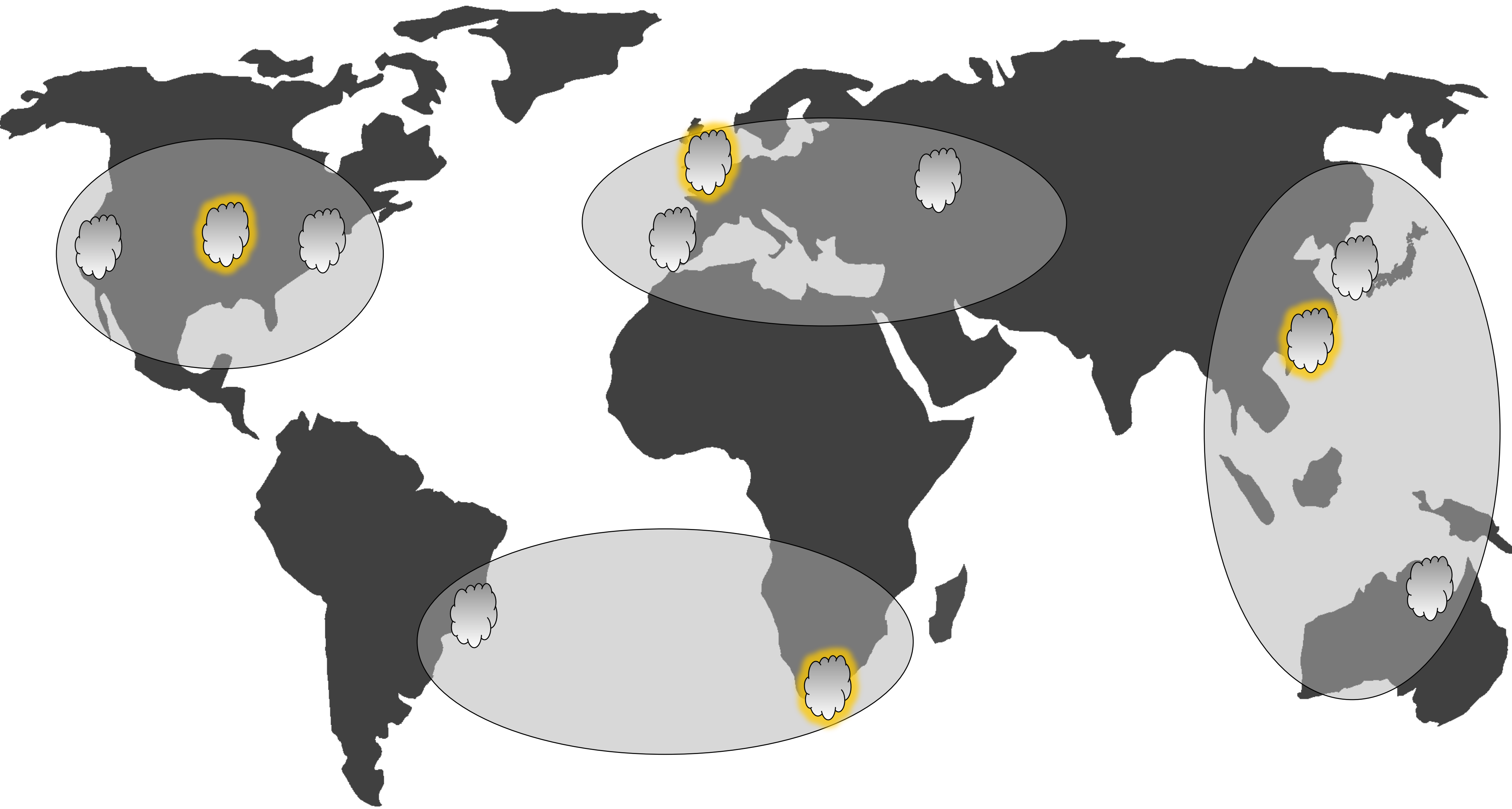
Cluster Locations

- Locations are ASes (could also be IP prefixes)
- Tor clusters client and destination locations
- Cluster members act like the cluster representative
- Distance between locations is sum over guards/exits of expected weight of adversaries that appear on one virtual link but not the other



Clustering algorithm

Modified k-means to choose balanced clusters



Talk Overview

1. Problem
2. Background
3. Attack on Prior Approach
4. Solution #1: Use Trust
5. Solution #2: Cluster
6. Trust-Aware Path Selection
7. Conclusion

Trust-Aware Path Selection (TAPS)

TrustAll

- All users use TAPS.

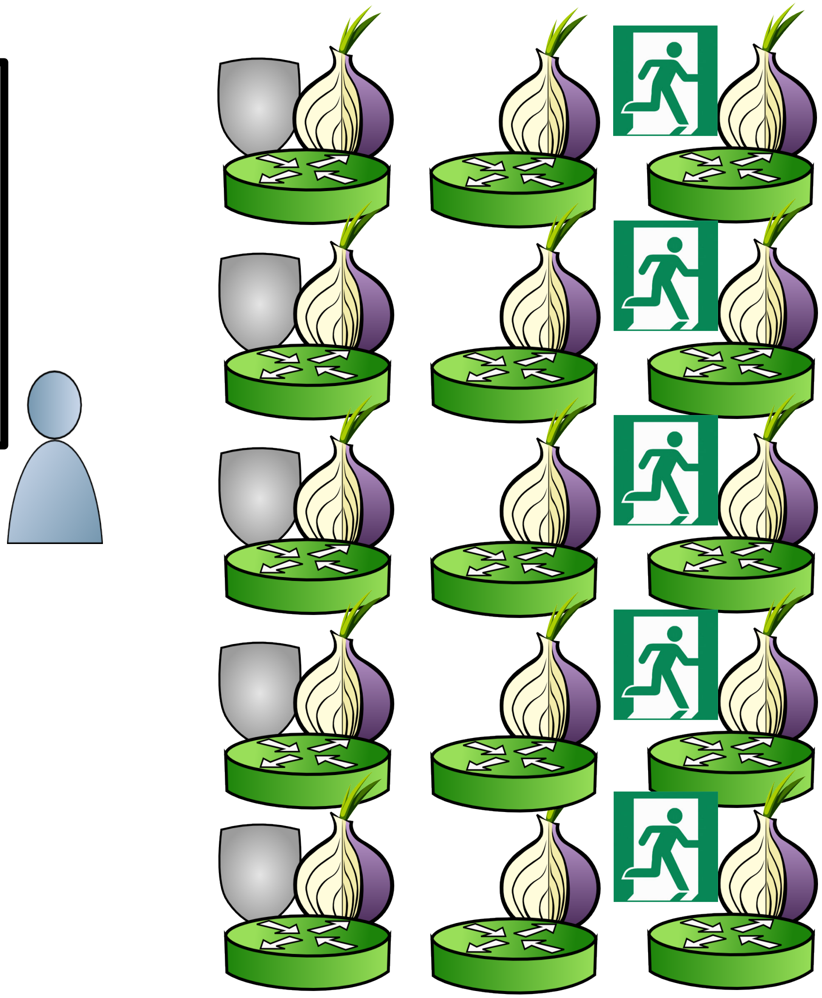
TrustOne

- Most users use “vanilla” Tor instead of TAPS.
- Exits may be chosen as in vanilla Tor to blend in (guards are chosen much less frequently).
- Tighter security parameters because load-balancing won’t be as affected.

Trust-Aware Path Selection (TAPS)

Guard selection

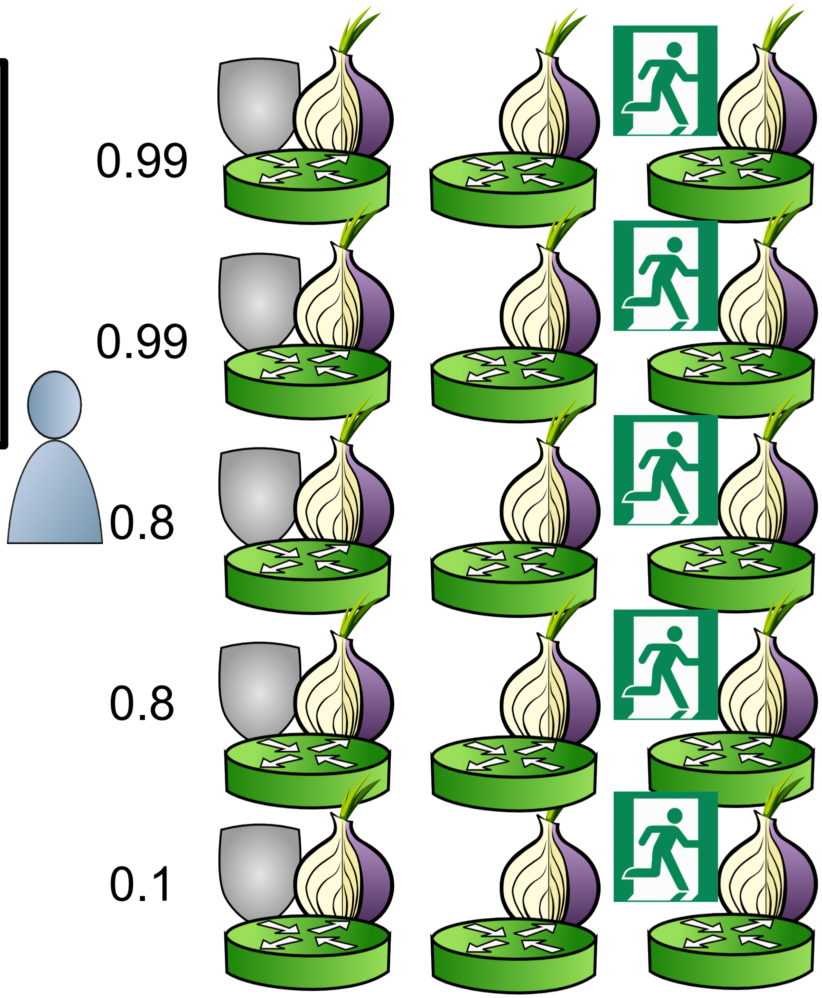
1. Score guards.
2. Randomly choose guard with score close enough to highest.



Trust-Aware Path Selection (TAPS)

Guard selection

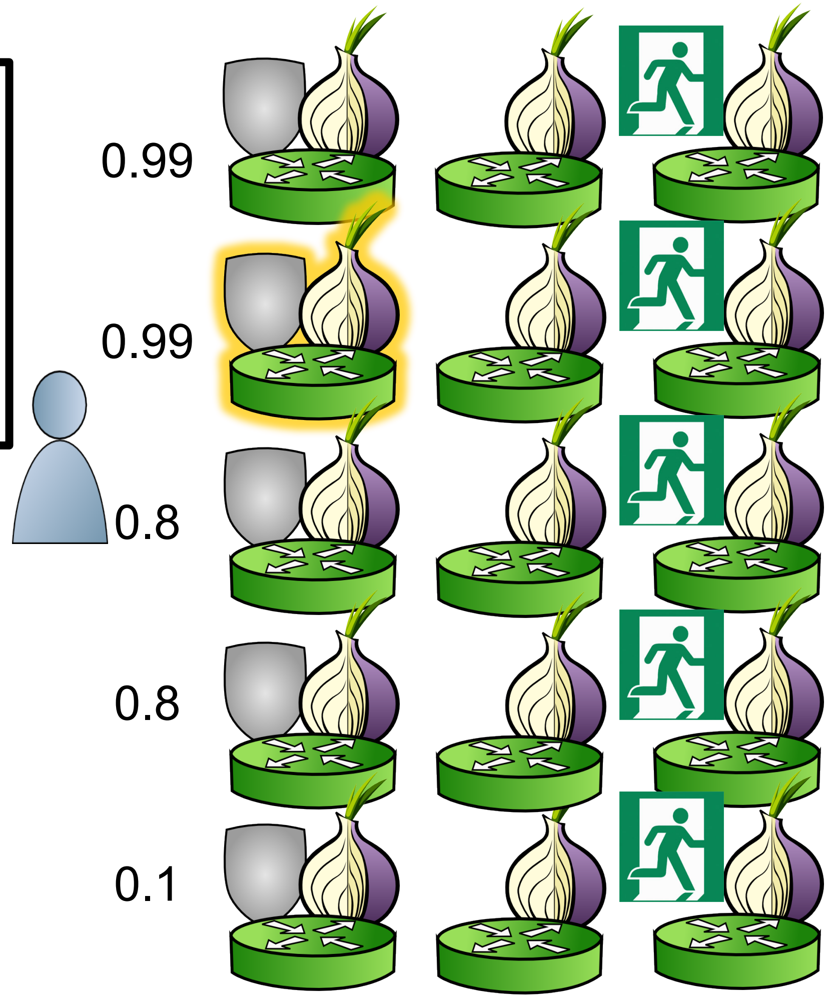
1. Score guards.
2. Randomly choose guard with score close enough to highest.



Trust-Aware Path Selection (TAPS)

Guard selection

1. Score guards.
2. Randomly choose guard with score close enough to highest.



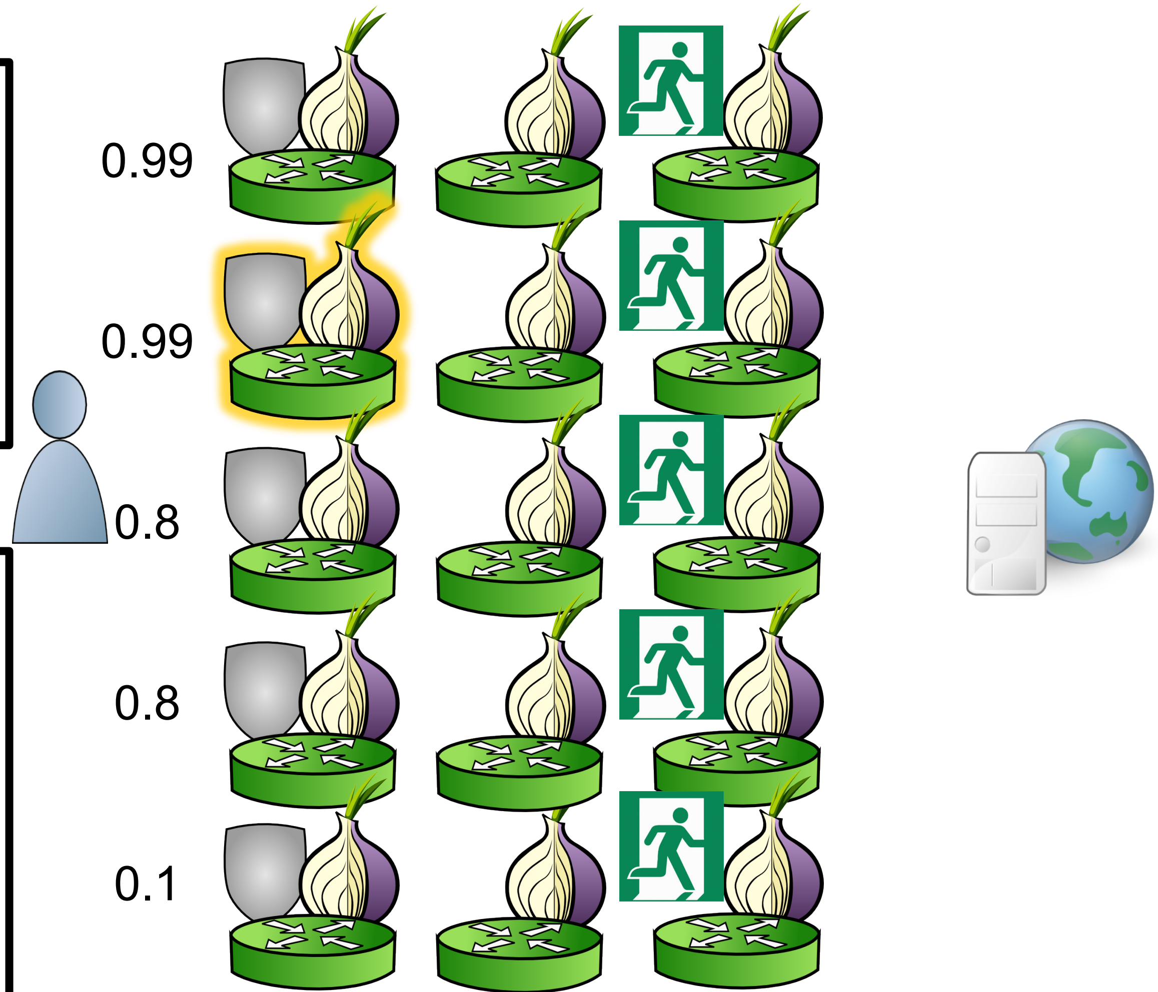
Trust-Aware Path Selection (TAPS)

Guard selection

1. Score guards.
2. Randomly choose guard with score close enough to highest.

Circuit reuse/creation

1. Score exit routers.
2. Reuse circuit with exit score close enough to highest, else randomly choose such an exit.
3. If needed, randomly choose middle and construct circuit.



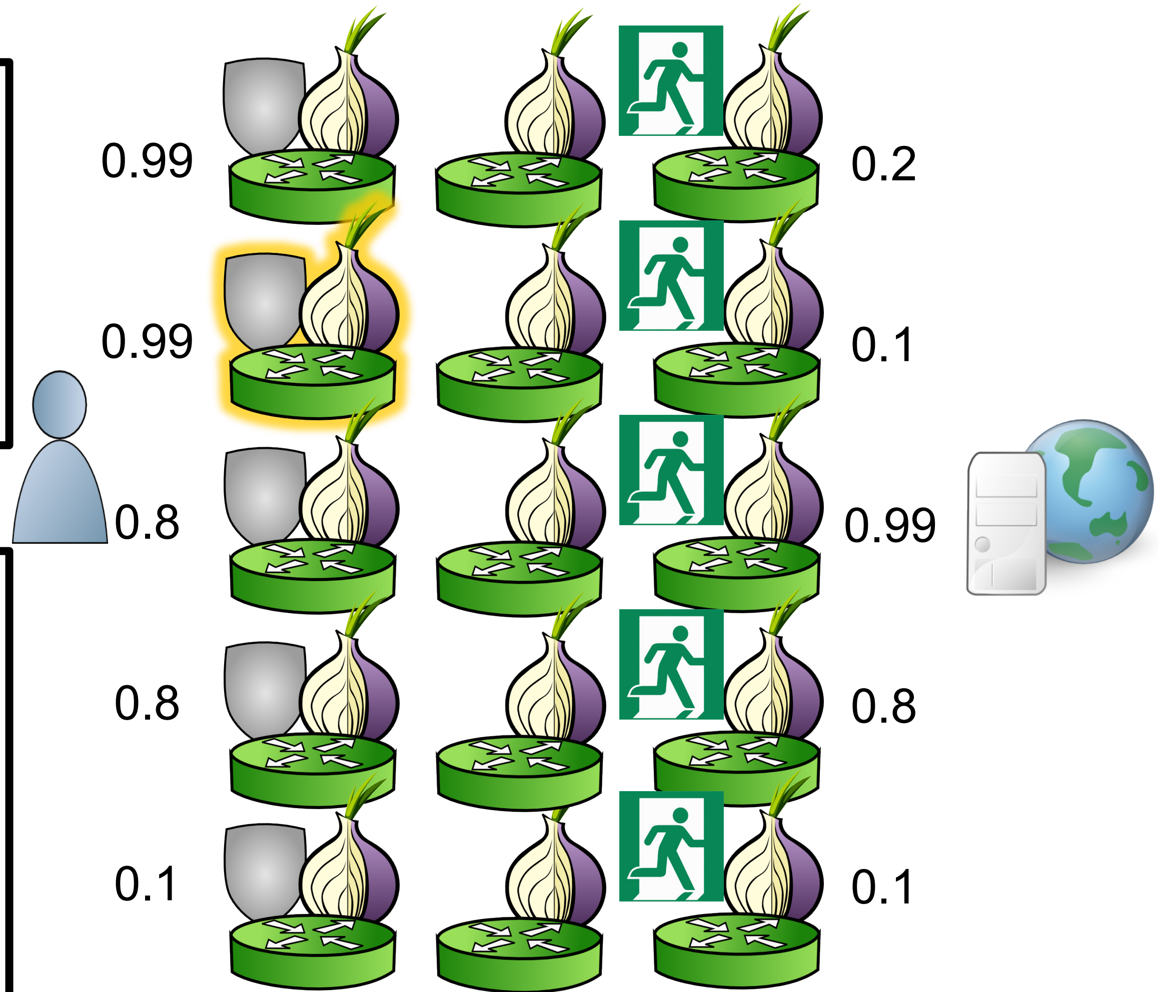
Trust-Aware Path Selection (TAPS)

Guard selection

1. Score guards.
2. Randomly choose guard with score close enough to highest.

Circuit reuse/creation

1. Score exit routers.
2. Reuse circuit with exit score close enough to highest, else randomly choose such an exit.
3. If needed, randomly choose middle and construct circuit.



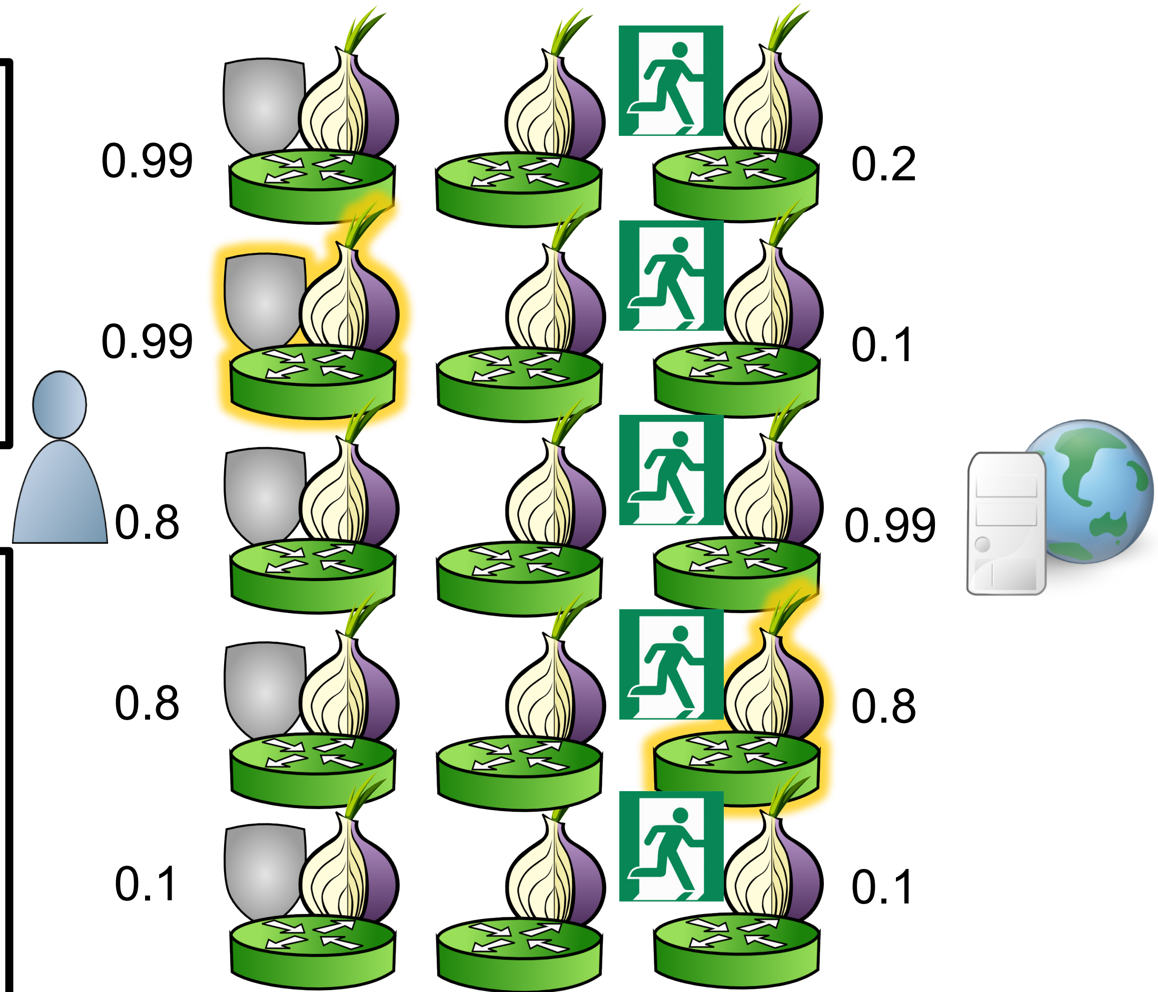
Trust-Aware Path Selection (TAPS)

Guard selection

1. Score guards.
2. Randomly choose guard with score close enough to highest.

Circuit reuse/creation

1. Score exit routers.
2. Reuse circuit with exit score close enough to highest, else randomly choose such an exit.
3. If needed, randomly choose middle and construct circuit.



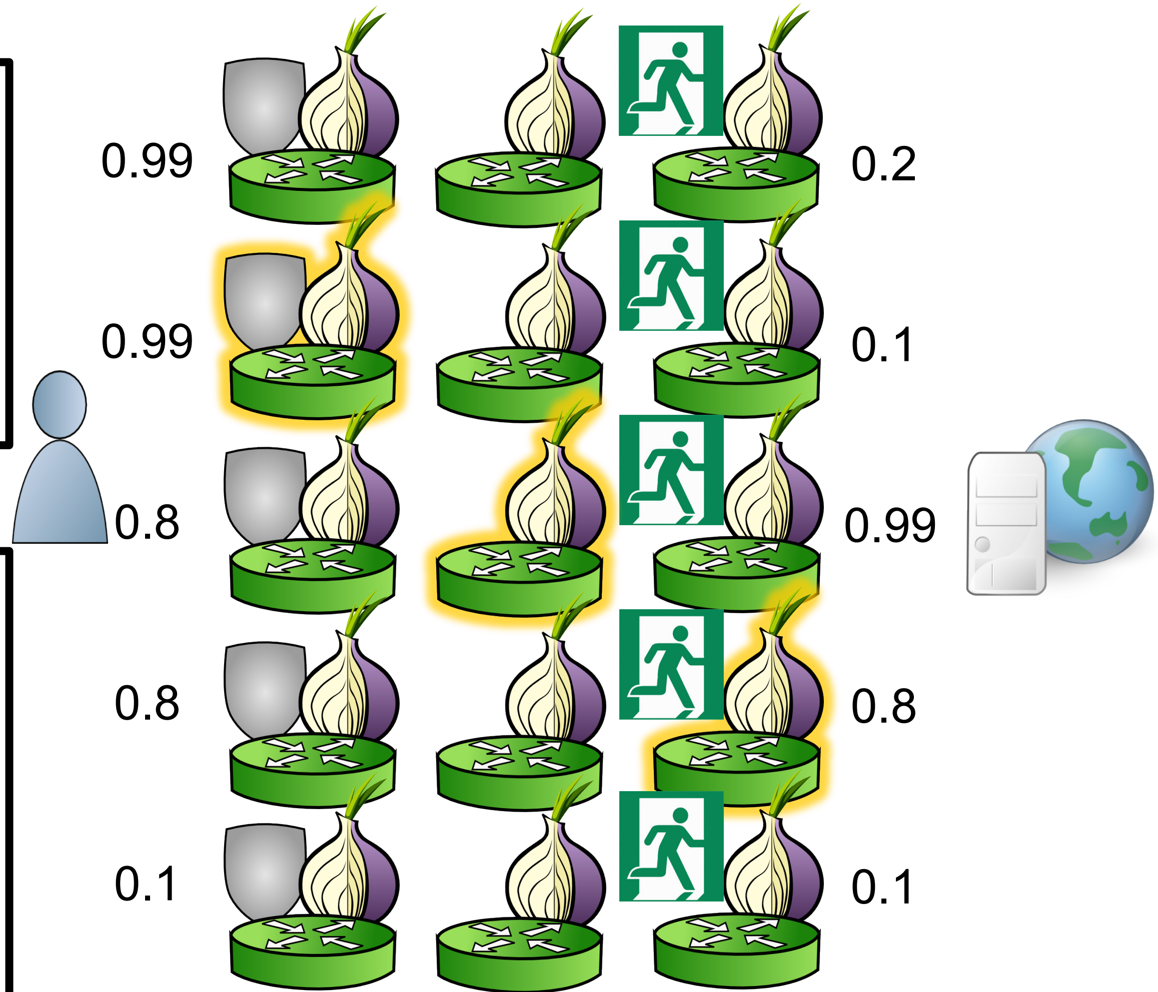
Trust-Aware Path Selection (TAPS)

Guard selection

1. Score guards.
2. Randomly choose guard with score close enough to highest.

Circuit reuse/creation

1. Score exit routers.
2. Reuse circuit with exit score close enough to highest, else randomly choose such an exit.
3. If needed, randomly choose middle and construct circuit.



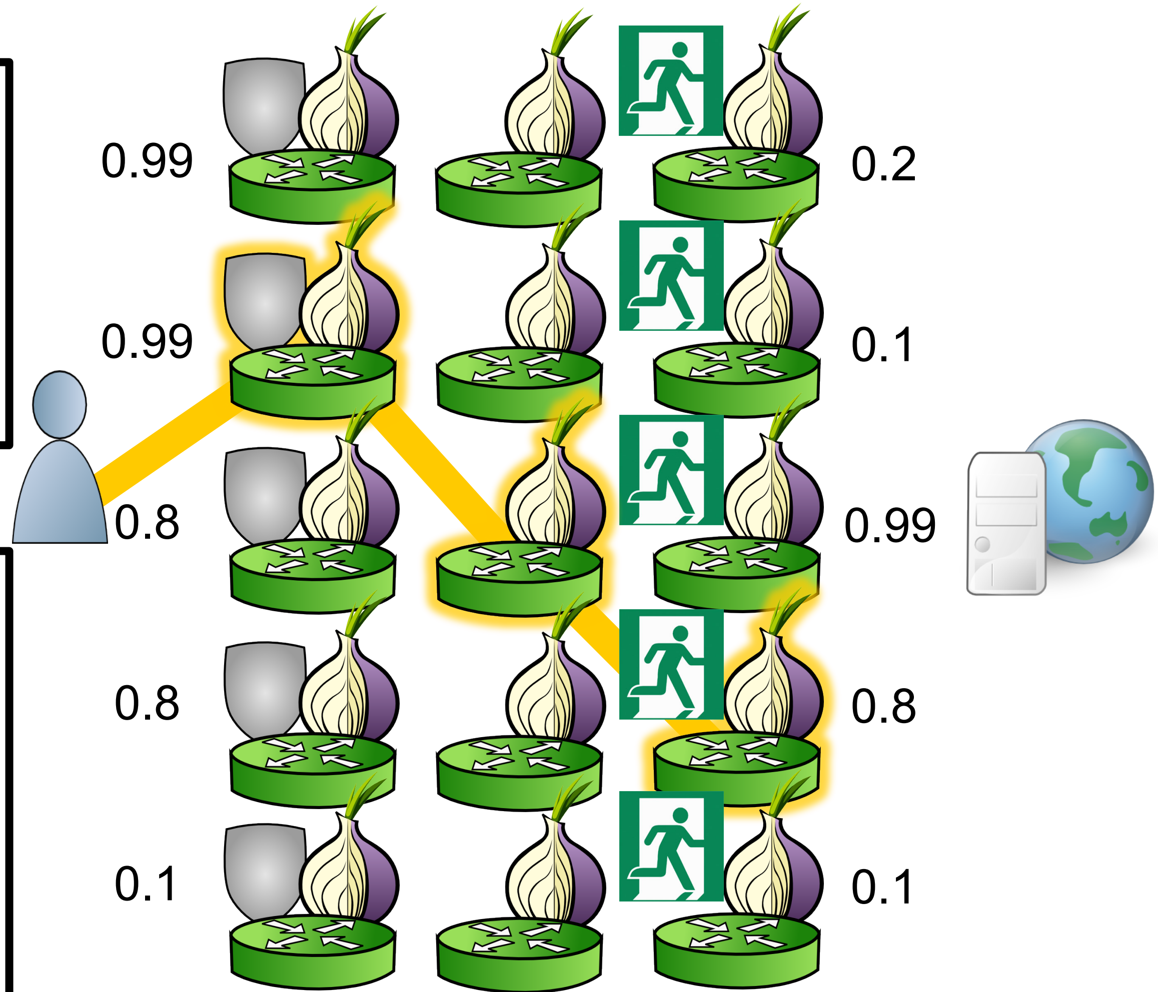
Trust-Aware Path Selection (TAPS)

Guard selection

1. Score guards.
2. Randomly choose guard with score close enough to highest.

Circuit reuse/creation

1. Score exit routers.
2. Reuse circuit with exit score close enough to highest, else randomly choose such an exit.
3. If needed, randomly choose middle and construct circuit.



TAPS Experiments: Path Simulations

TrustAll

- Users engage in typical Web behavior (browse, search, social network, etc.), accessing 135 destination IPs

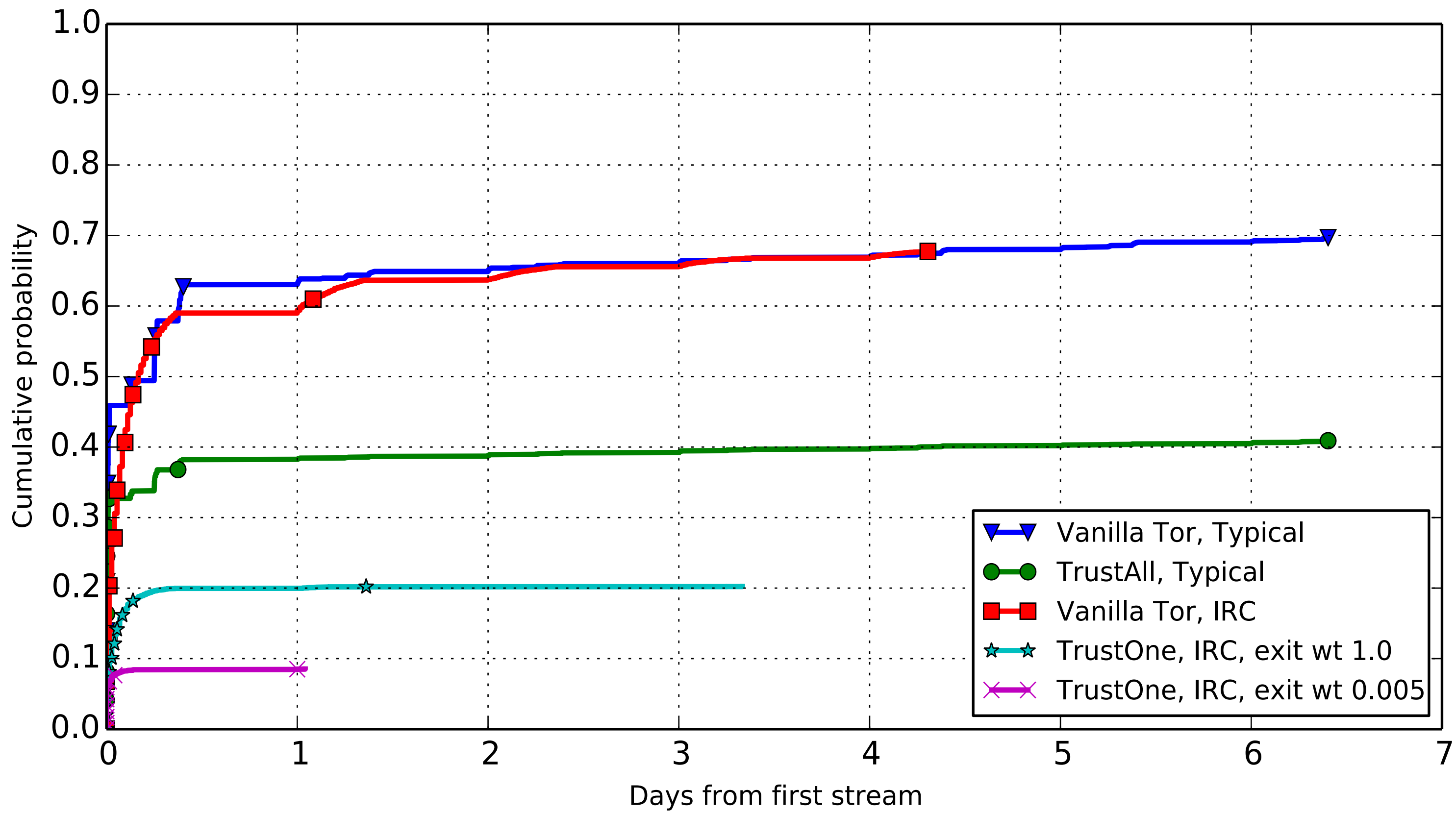
TrustOne

- User visits a single IRC chat server

Pervasive adversary “The Man” (possible default)

- Each AS/IXP organization independently compromised with probability 0.1
- Each relay family compromised with probability $.02 \leq p \leq .1$ decreasing with uptime of relays

TAPS Experiments: Path Simulations



Time to first compromised connection from most popular client AS (6128) over 7 days

TAPS Experiments: Shadow Simulations

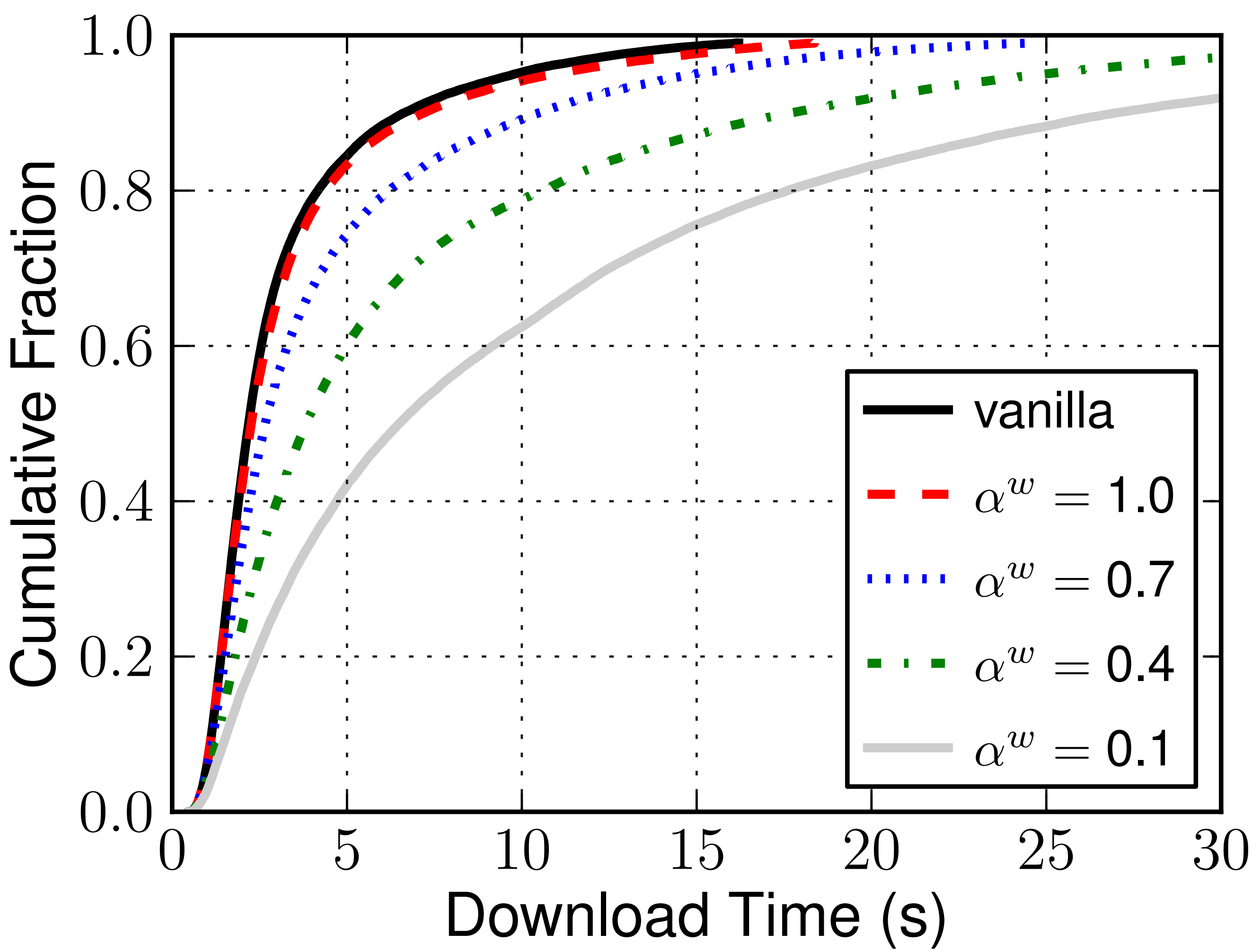
Simulated network

- 400 relays
- 1380 clients: 1080 Web, 120 bulk, 180 ShadowPerf
- 500 file servers
- 1 simulated hour

TAPS simulation

- Implemented TAPS in Tor
- TrustAll algorithm
- The Man trust policy
- Varied α^{ω} parameter of bandwidth fraction of highest-scoring relays to select from (α^{ω}=0.2 in path simulations)

TAPS Experiments: Shadow Simulations



320KiB file download

U.S. NAVAL RESEARCH LABORATORY

Talk Overview

1. Problem
2. Background
3. Attack on Prior Approach
4. Solution #1: Use Trust
5. Solution #2: Cluster
6. Trust-Aware Path Selection
7. Conclusion

Conclusion

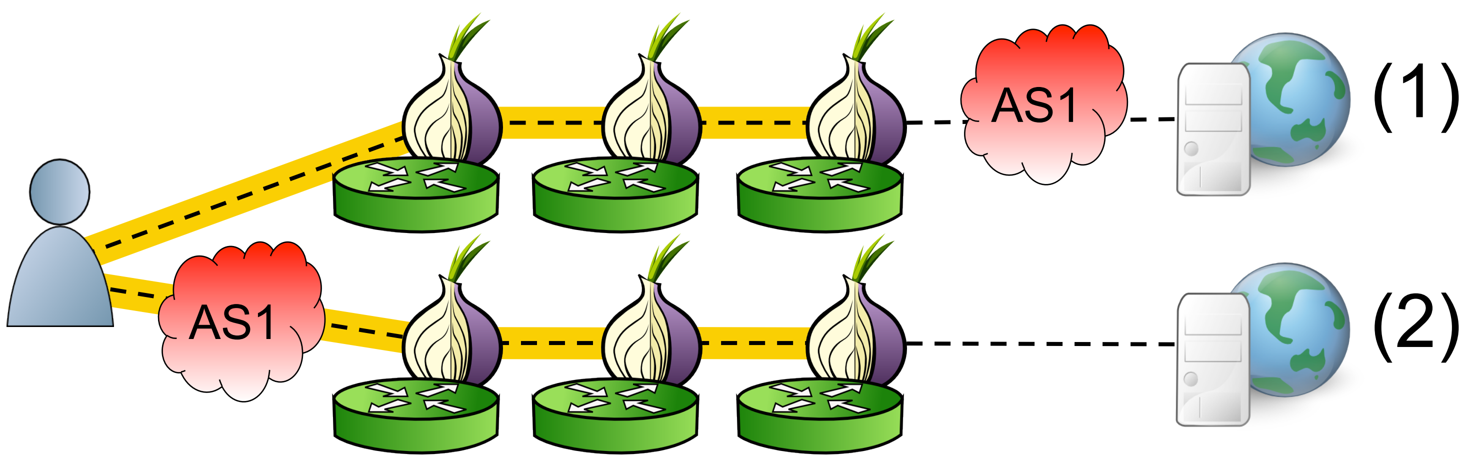
Conclusion

- Tor can be deanonymized via timing correlation.
- We present an attack on previous defense.
- We propose the Trust-Aware Path Selection (TAPS) algorithm that is not vulnerable to our attack.
- We demonstrate TAPS can improve user security without major cost in performance.

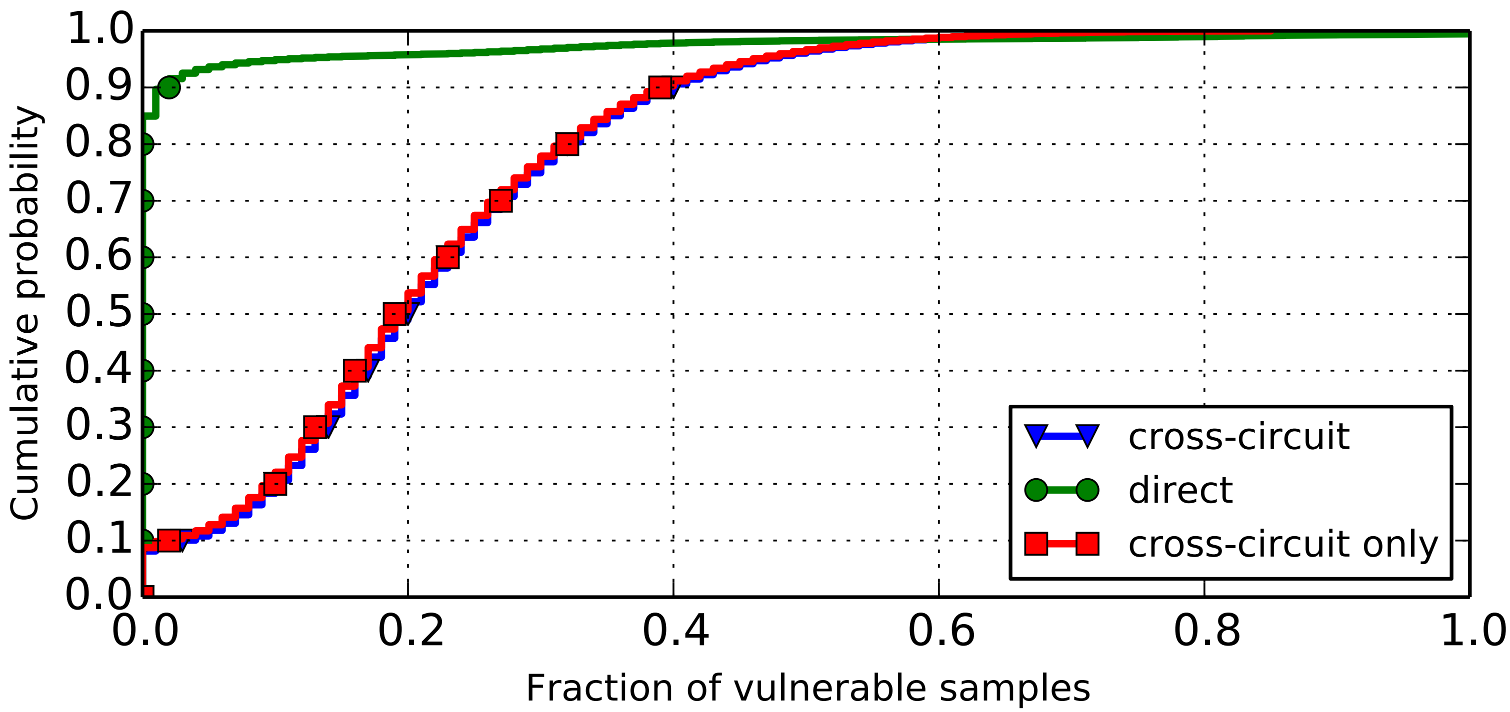
Cross-Circuit Attack

Cross-Circuit Attack on Astoria

1. Client makes initial connection to honest website (1).
2. Client downloads linked resource from other server. Needs to use different guard for (2) than used for (1).
3. Malicious AS can perform correlation attack across circuits using known download pattern for website.



Cross-Circuit Attack

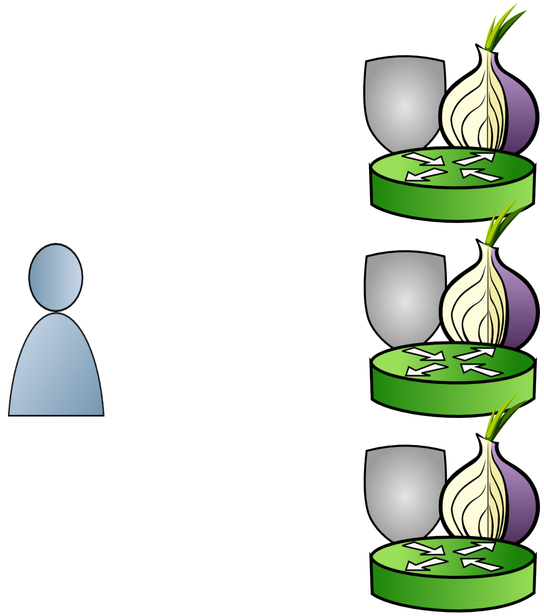


- Repeatedly simulated Astoria visits to Alexa top 5000 websites from top 400 Tor client Ases
- Median frequency cross-circuit attack: 0.2
- Median frequency of direct-circuit attack: 0.03

Trust-Aware Path Selection (TAPS)

Relay security scoring

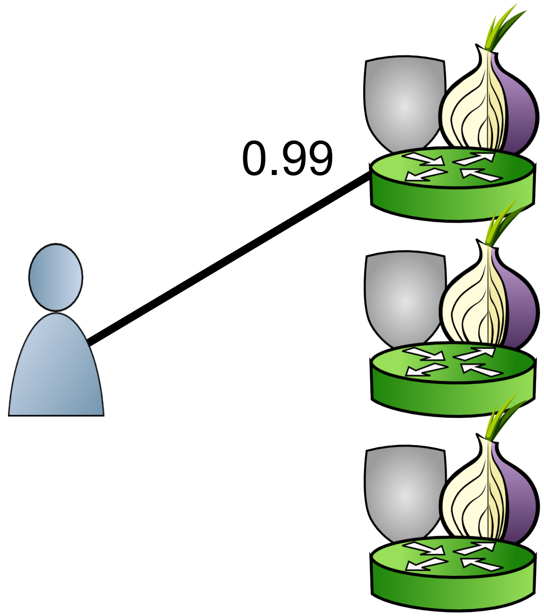
- GUARDSECURITY(*client_loc*, *guard*): Expected weight of adversaries not between *client_loc* and *guard*



Trust-Aware Path Selection (TAPS)

Relay security scoring

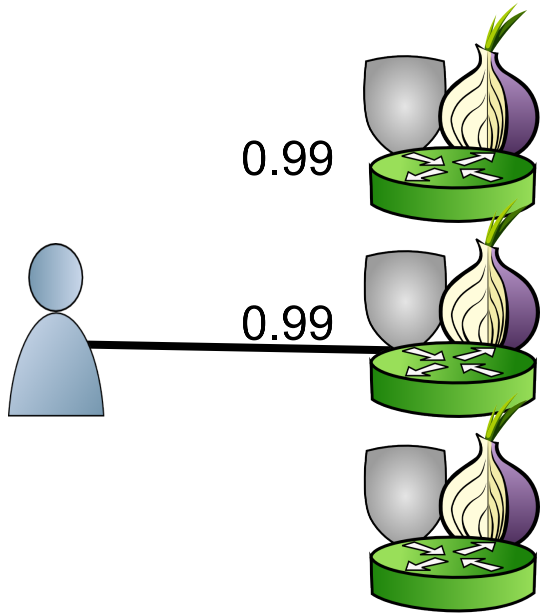
- GUARDSECURITY(*client_loc*, *guard*): Expected weight of adversaries not between *client_loc* and *guard*



Trust-Aware Path Selection (TAPS)

Relay security scoring

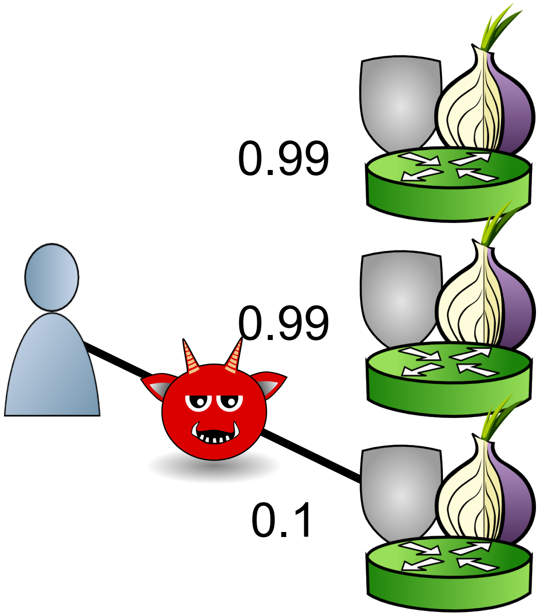
- GUARDSECURITY(*client_loc*, *guard*): Expected weight of adversaries not between *client_loc* and *guard*



Trust-Aware Path Selection (TAPS)

Relay security scoring

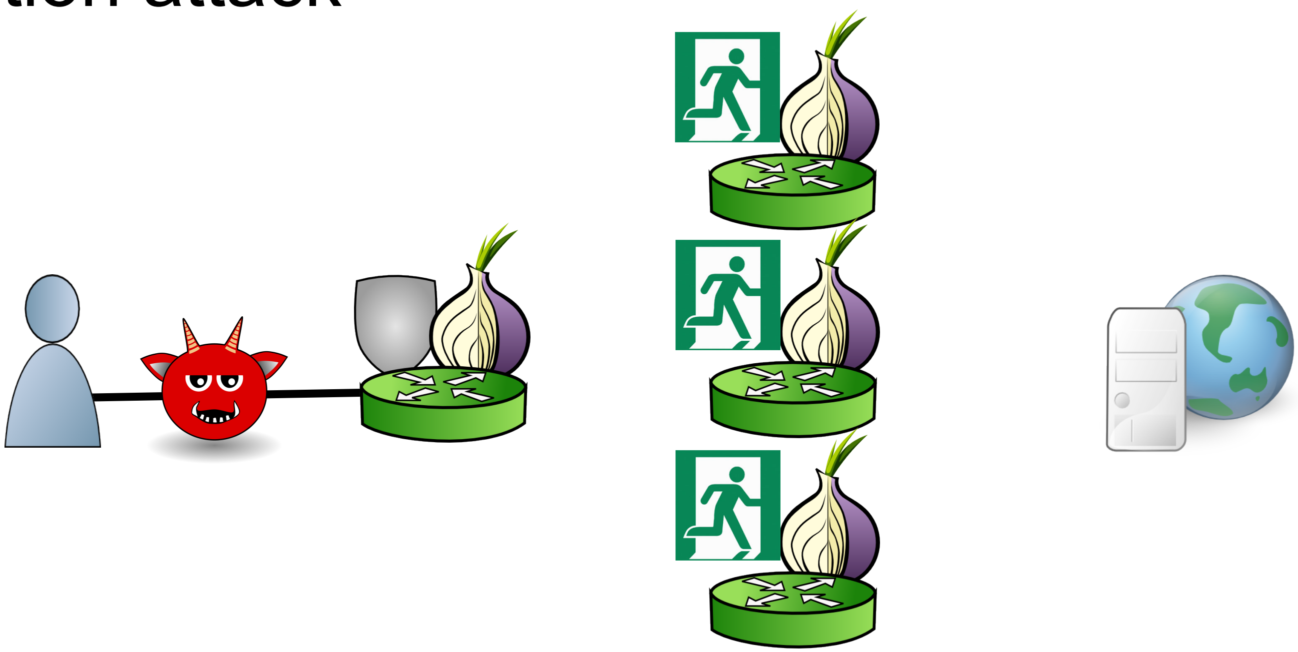
- GUARDSECURITY(*client_loc*, *guard*): Expected weight of adversaries not between *client_loc* and *guard*



Trust-Aware Path Selection (TAPS)

Relay security scoring

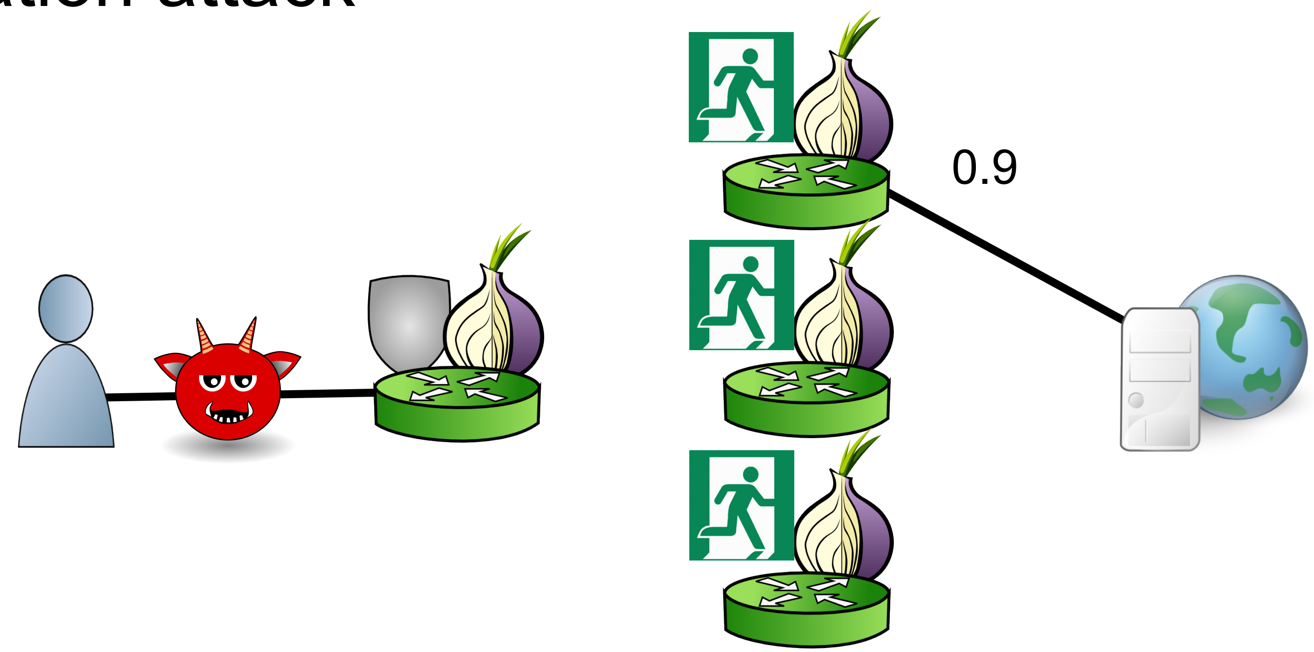
- GUARDSECURITY(*client_loc*, *guard*): Expected weight of adversaries not between *client_loc* and *guard*
- EXITSECURITY(*client_loc*, *dst_loc*, *guard*, *exit*): Expected weight of adversaries unable to perform correlation attack



Trust-Aware Path Selection (TAPS)

Relay security scoring

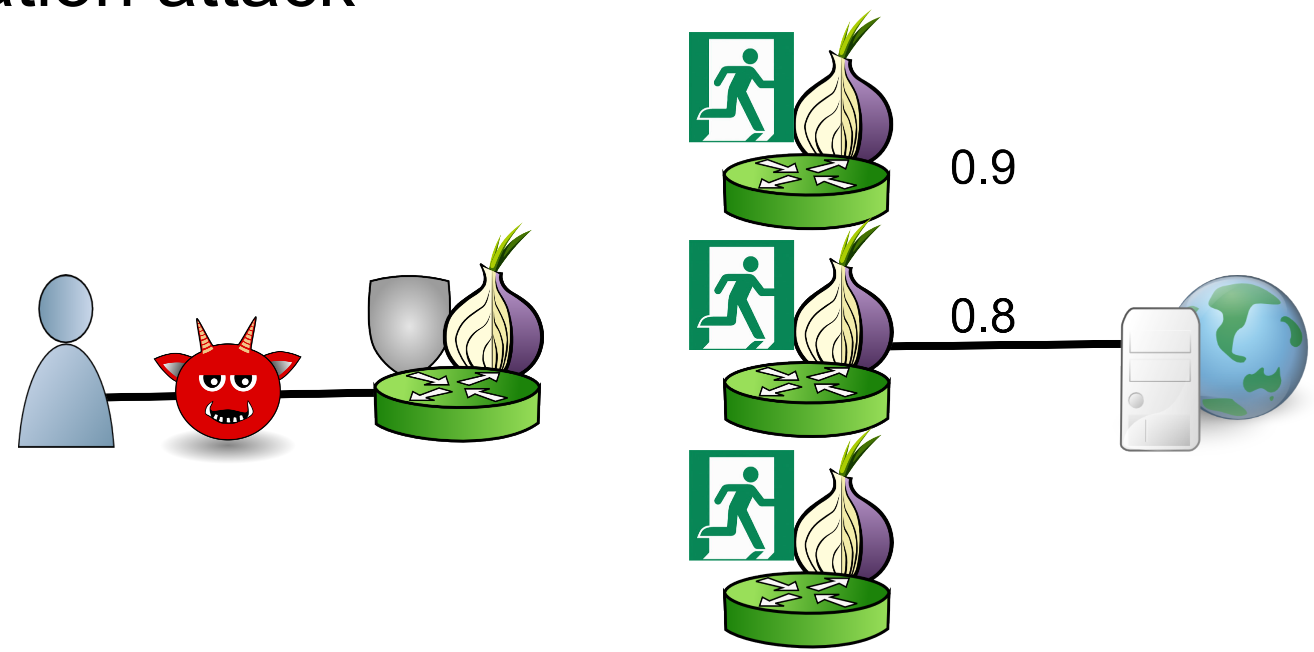
- GUARDSECURITY(*client_loc*, *guard*): Expected weight of adversaries not between *client_loc* and *guard*
- EXITSECURITY(*client_loc*, *dst_loc*, *guard*, *exit*): Expected weight of adversaries unable to perform correlation attack



Trust-Aware Path Selection (TAPS)

Relay security scoring

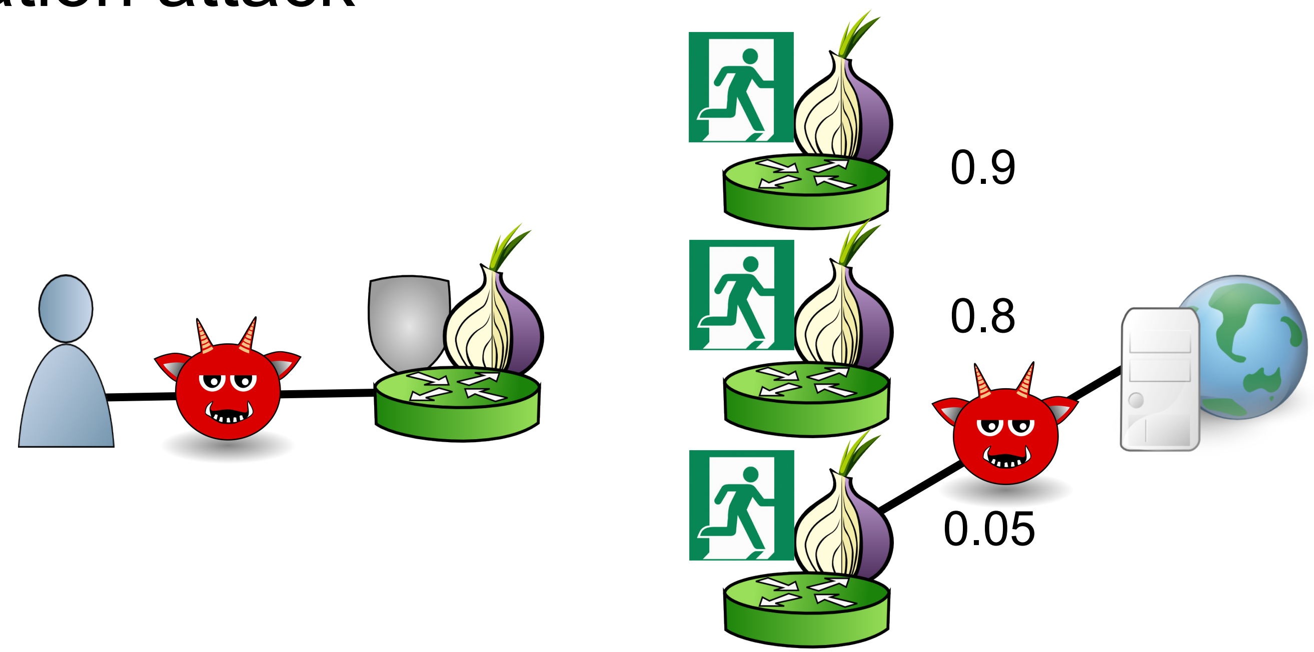
- GUARDSECURITY(*client_loc*, *guard*): Expected weight of adversaries not between *client_loc* and *guard*
- EXITSECURITY(*client_loc*, *dst_loc*, *guard*, *exit*): Expected weight of adversaries unable to perform correlation attack



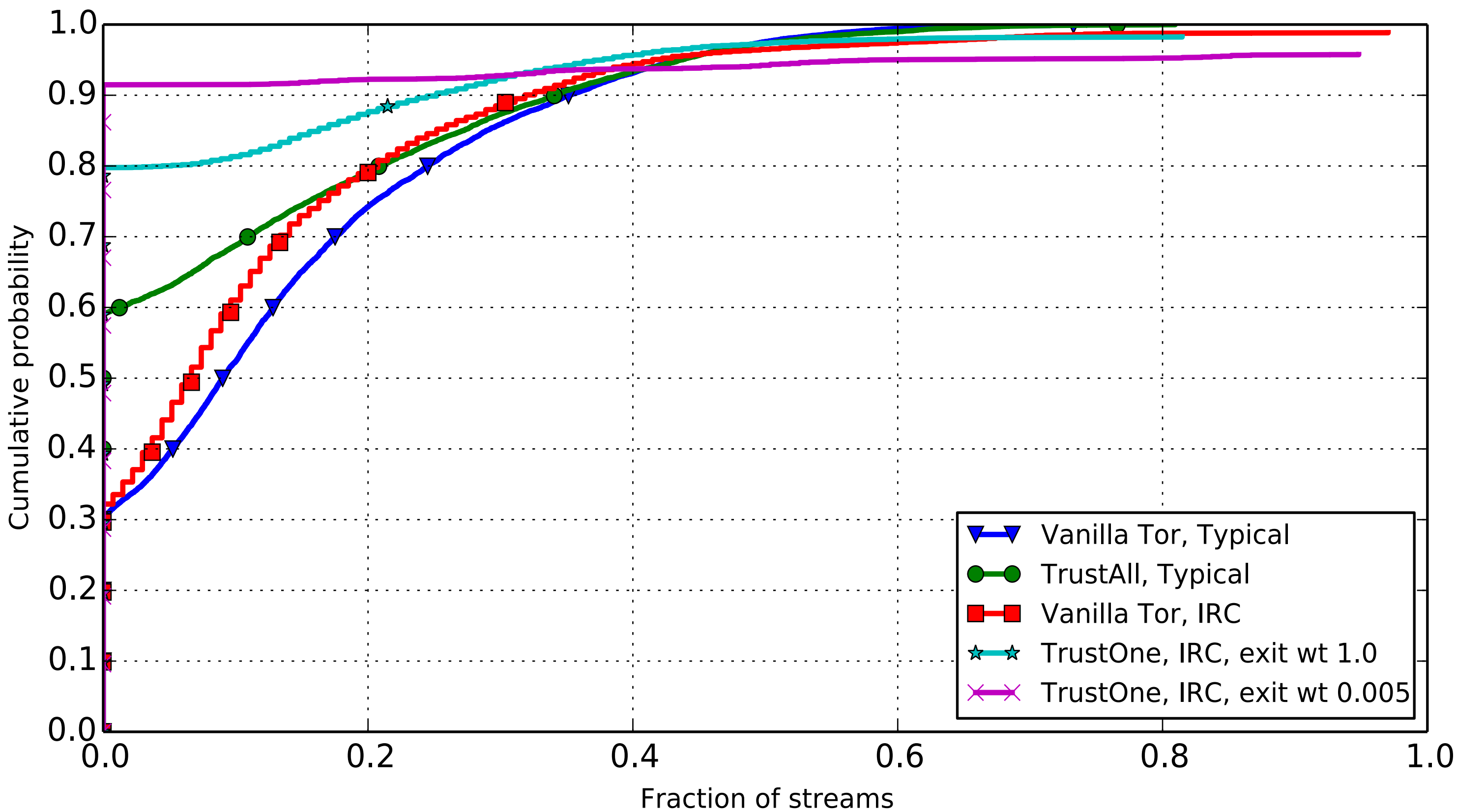
Trust-Aware Path Selection (TAPS)

Relay security scoring

- GUARDSECURITY(*client_loc*, *guard*): Expected weight of adversaries not between *client_loc* and *guard*
- EXITSECURITY(*client_loc*, *dst_loc*, *guard*, *exit*): Expected weight of adversaries unable to perform correlation attack

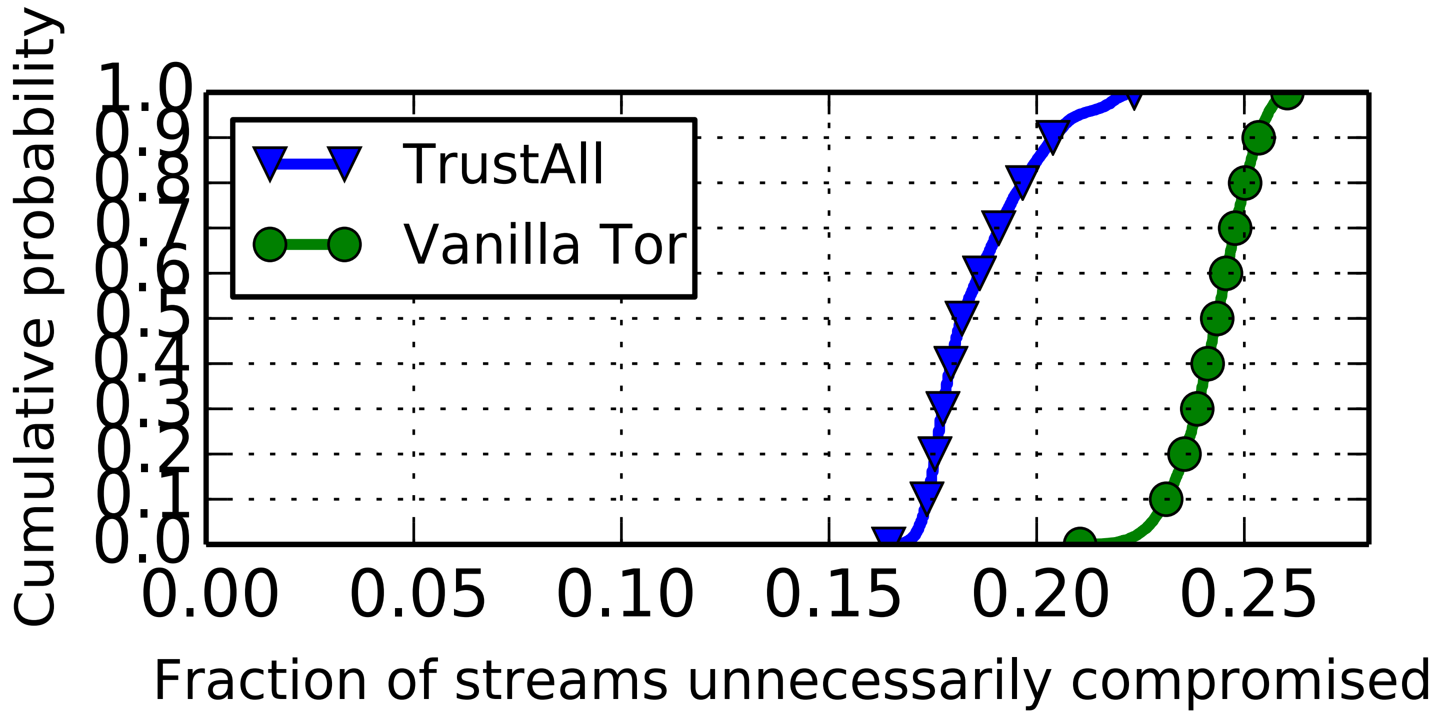


TAPS Experiments: Path Simulations



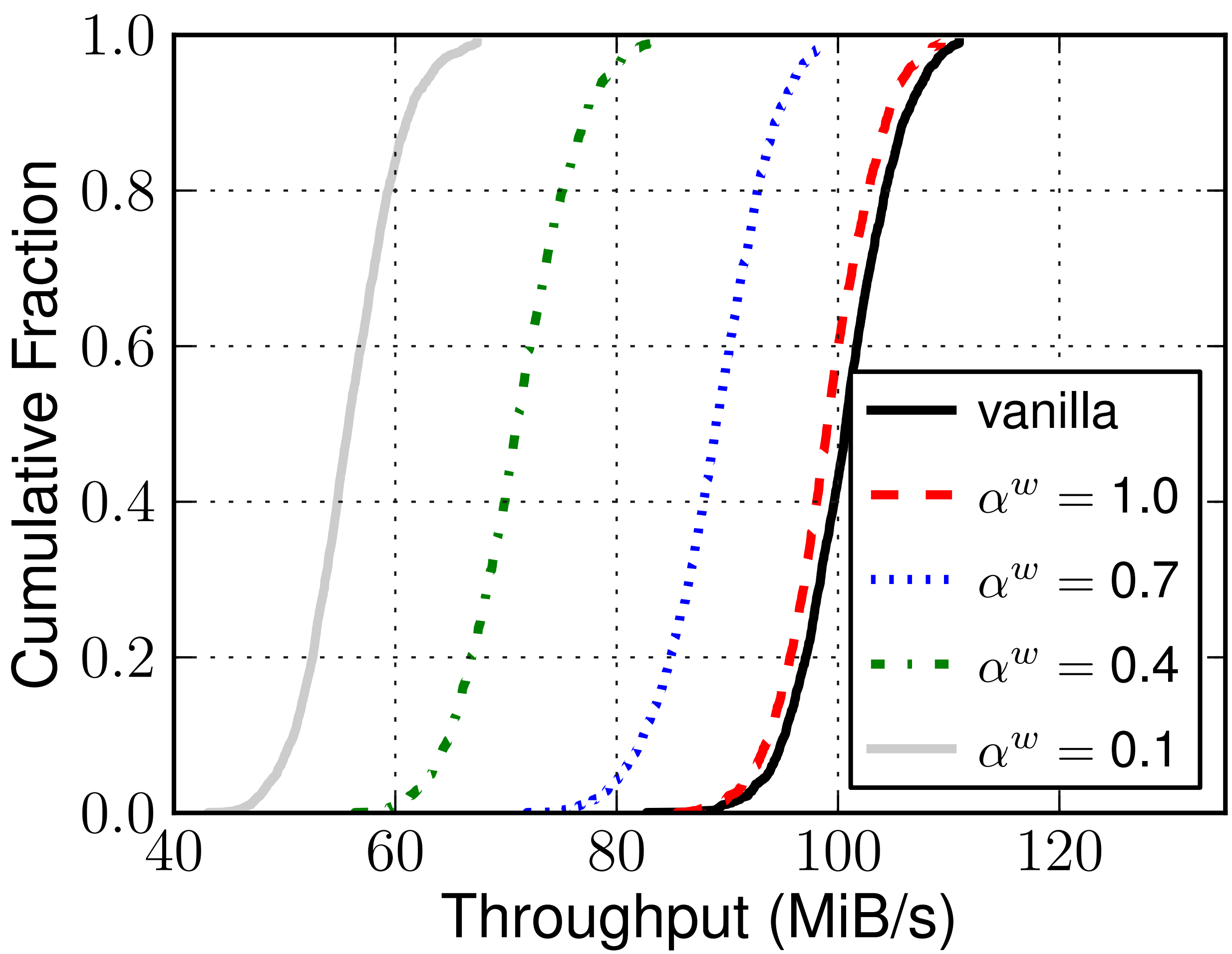
Fraction of compromised connections from most popular AS (6128) over 7 days

TAPS Experiments: Countries



Streams compromised by any country for typical usage over 7 days from most popular AS (6128) (except from US where AS 6128 is).

TAPS Experiments: Shadow Simulations



Aggregate relay throughput