CS 6431

Anonymity Networks and Censorship Resistance

Vitaly Shmatikov

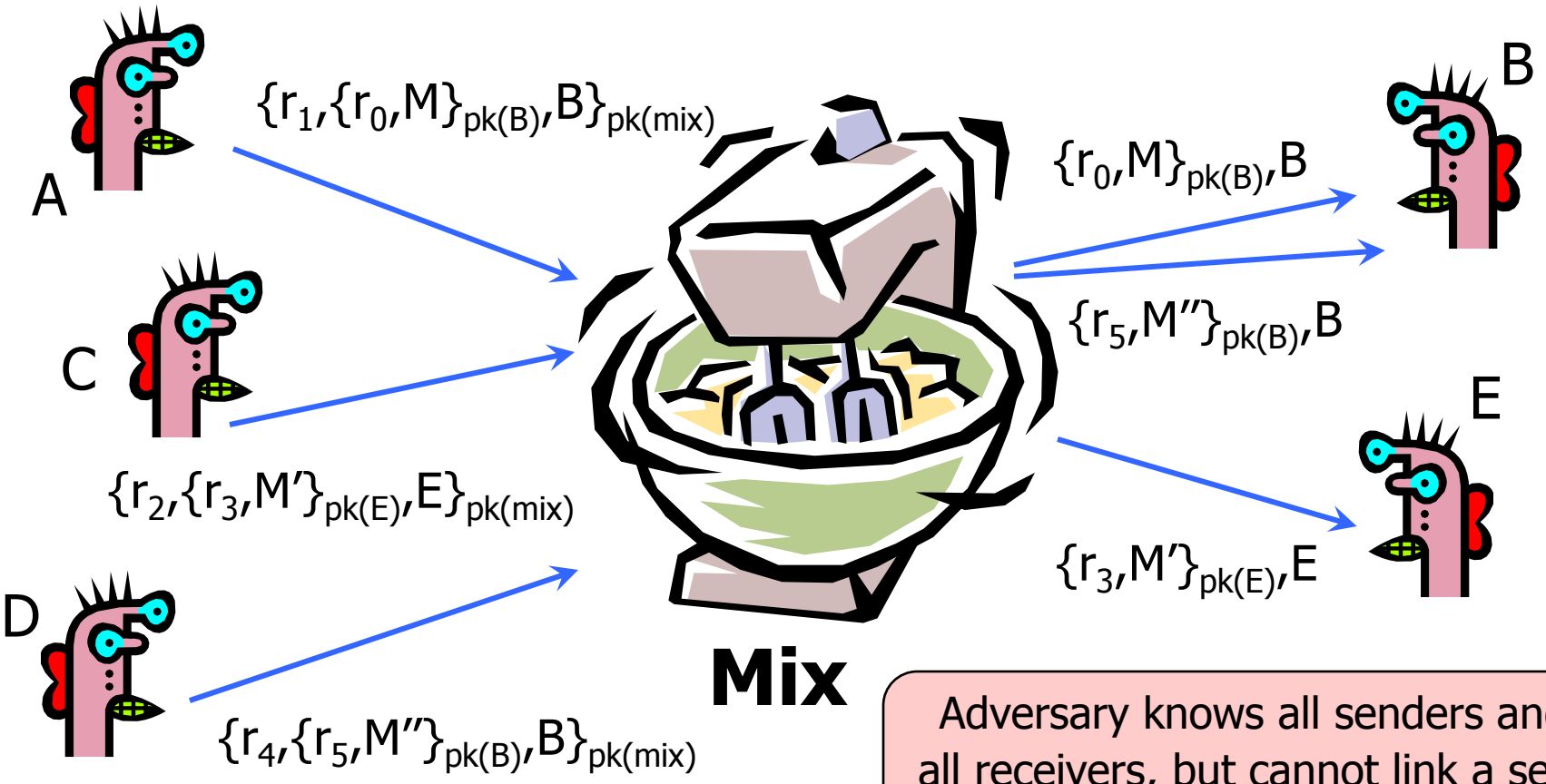
Privacy on Public Networks

- Internet is designed as a public network
- Routing information is public
  - IP packet headers identify source and destination
  - Even a passive observer can easily figure out who is talking to whom
- Encryption does not hide identities
  - Encryption hides payload, but not routing headers
  - Even IP-level encryption (VPNs, tunnel-mode IPsec) reveals IP addresses of gateways

Chaum's Mix

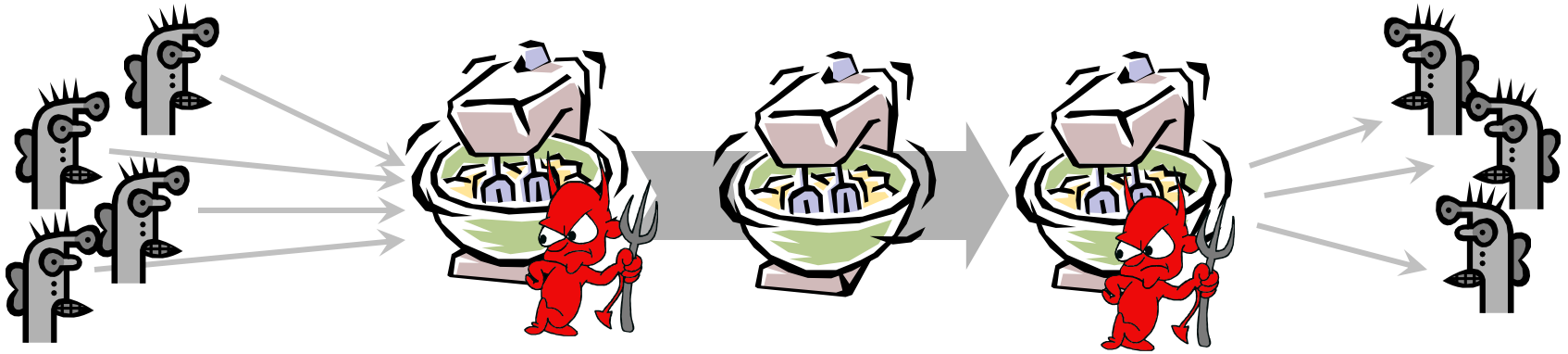
- Early proposal for anonymous email
  - David Chaum. "Untraceable electronic mail, return addresses, and digital pseudonyms". Communications of the ACM, February 1981.
- Public-key crypto + trusted re-mailer (Mix)
  - Untrusted communication medium
  - Public keys used as persistent pseudonyms
- Modern anonymity systems use Mix as the basic building block

Basic Mix Design



Adversary knows all senders and all receivers, but cannot link a sent message with a received message

Mix Cascades and Mixnets



- Messages are sent through a sequence of mixes
  - Can also form an arbitrary network of mixes ("mixnet")
- Some of the mixes may be controlled by attacker, but even a single good mix ensures anonymity
- Pad and buffer traffic to foil correlation attacks

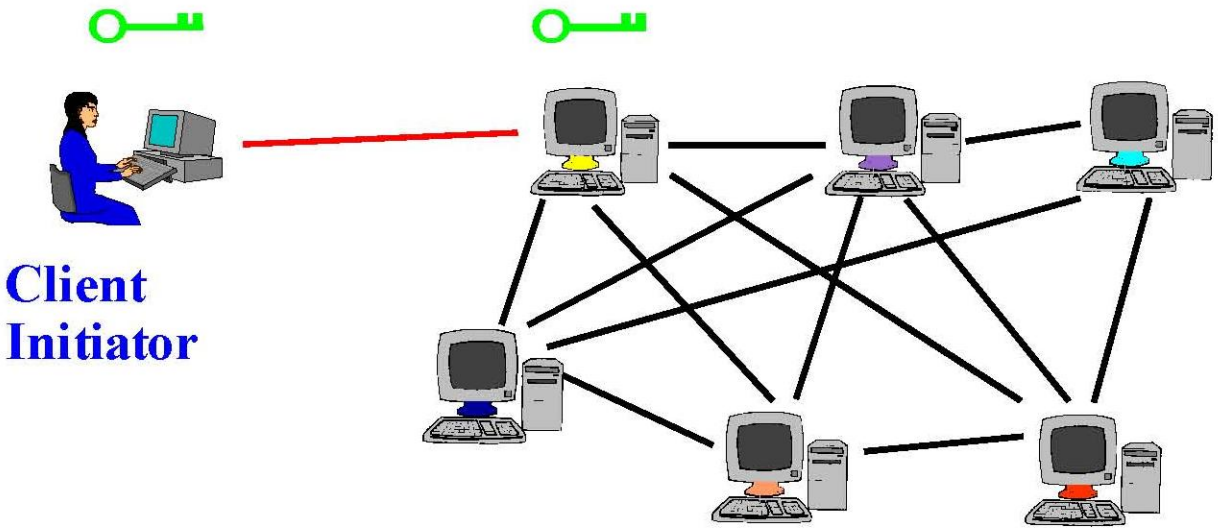
Disadvantages of Basic Mixnets

- Public-key encryption and decryption at each mix are computationally expensive
- Basic mixnets have high latency
  - Ok for email, but not for Web browsing
- Challenge: low-latency anonymity network
  - Use public-key crypto to establish a "circuit" with pairwise symmetric keys between hops
  - Then use symmetric decryption and re-encryption to move data along the established circuits

- Second-generation onion routing network
  - http://tor.eff.org
  - Specifically designed for low-latency anonymous Internet communications (e.g., Web browsing)
  - Running since October 2003
- Hundreds of nodes on all continents
- Over 2,500,000 users
- "Easy-to-use" client
  - Freely available, can use it for anonymous browsing

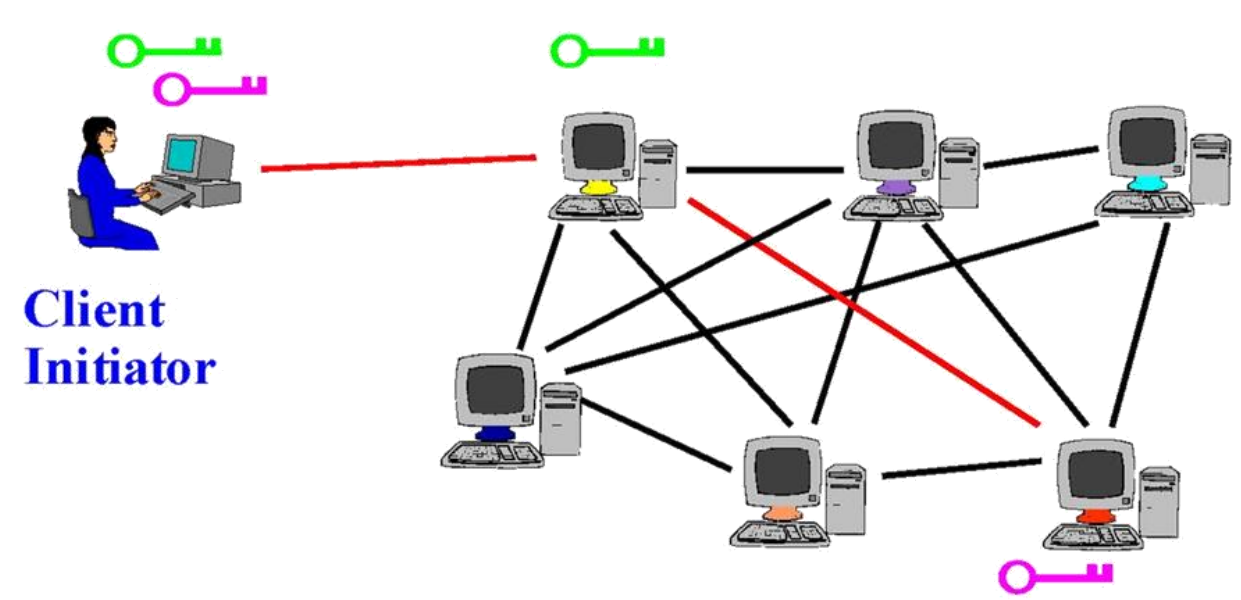
Tor Circuit Setup (1)

- Client proxy establishes a symmetric session key and circuit with Onion Router #1



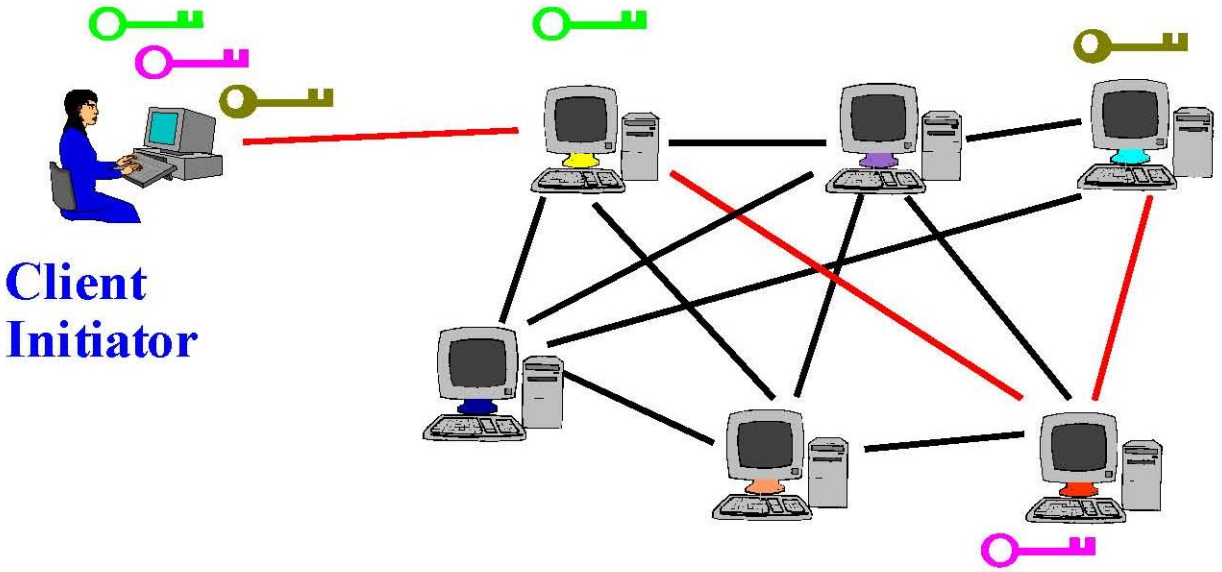
Tor Circuit Setup (2)

- Client proxy extends the circuit by establishing a symmetric session key with Onion Router #2
  - Tunnel through Onion Router #1



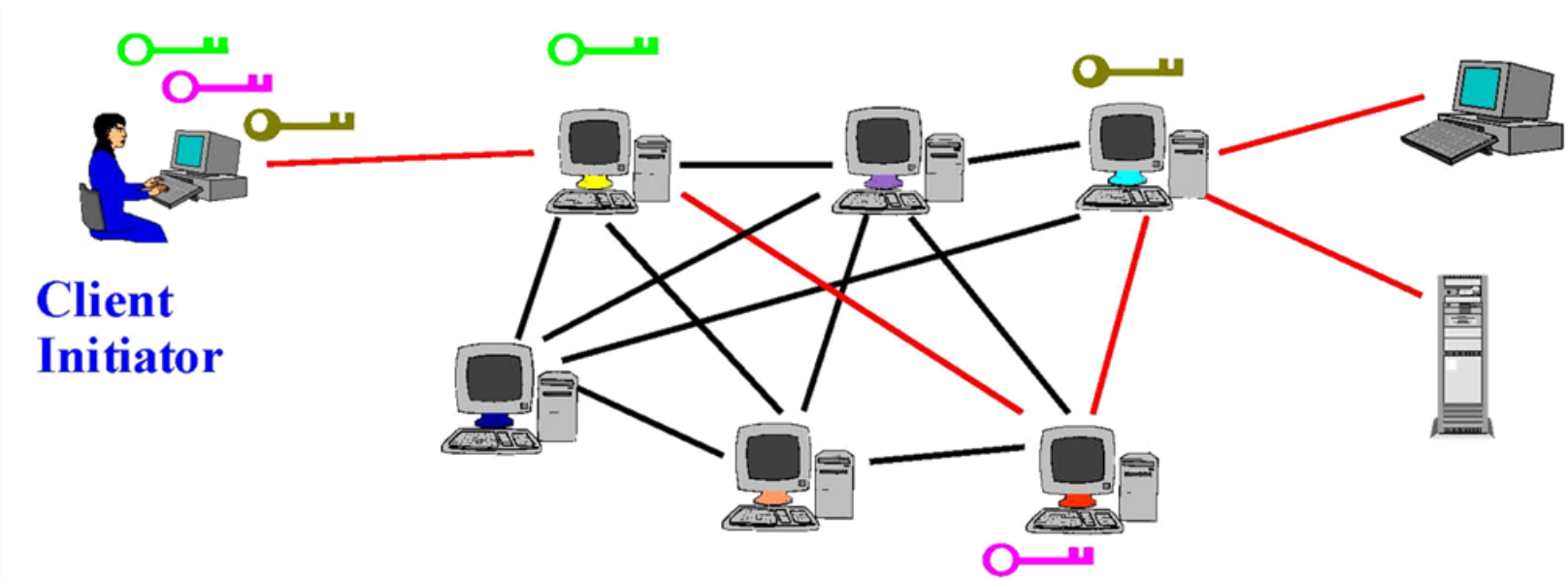
Tor Circuit Setup (3)

- Client proxy extends the circuit by establishing a symmetric session key with Onion Router #3
  - Tunnel through Onion Routers #1 and #2



Using a Tor Circuit

- Client applications connect and communicate over the established Tor circuit
  - Datagrams decrypted and re-encrypted at each link



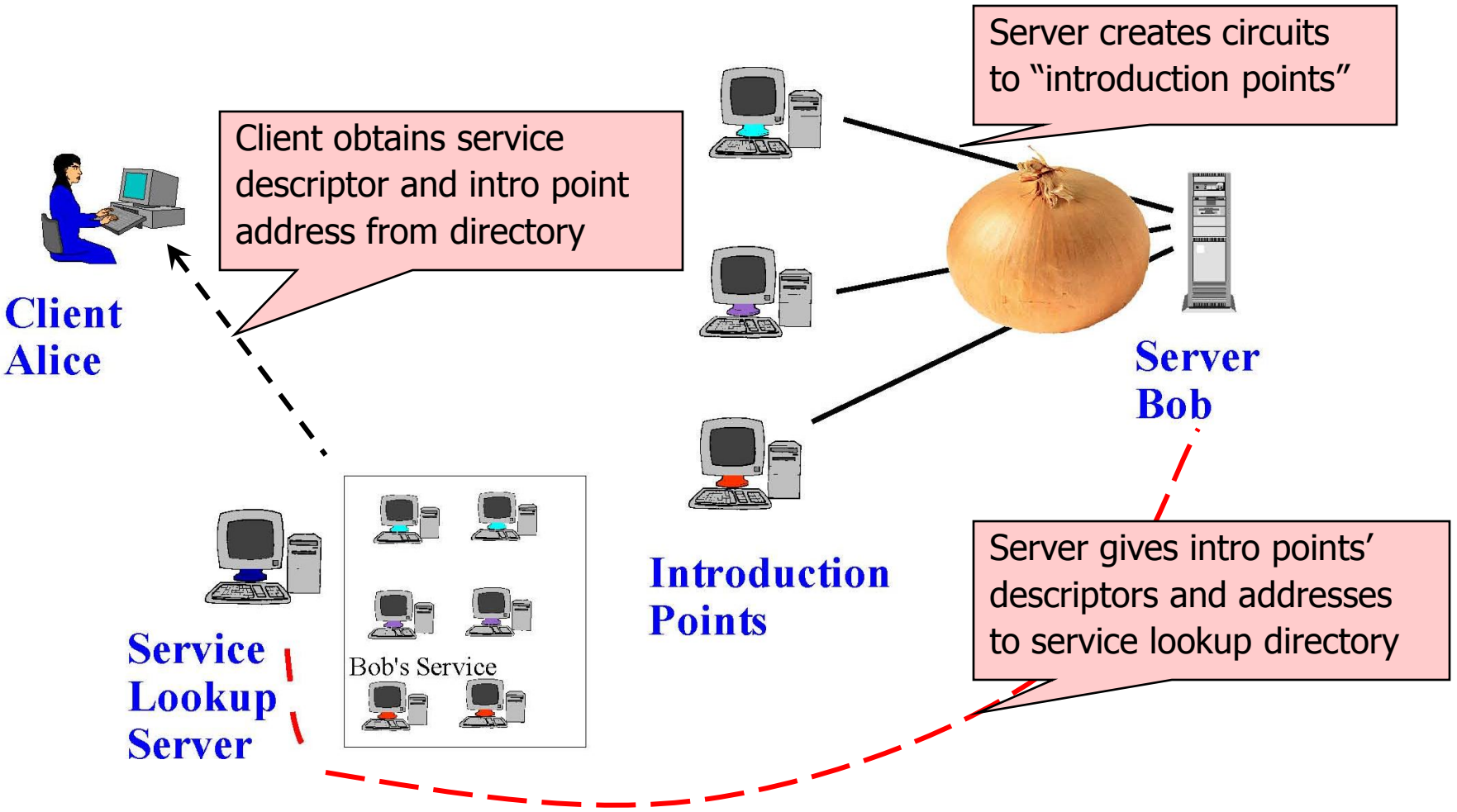
Tor Management Issues

- Many TCP connections can be "multiplexed" over one anonymous circuit
- Directory servers
  - Lists of active onion routers, their locations, current public keys, etc.
  - Control how new routers join the network
    - "Sybil attack": attacker creates a large number of routers
  - Directory servers' keys ship with Tor code

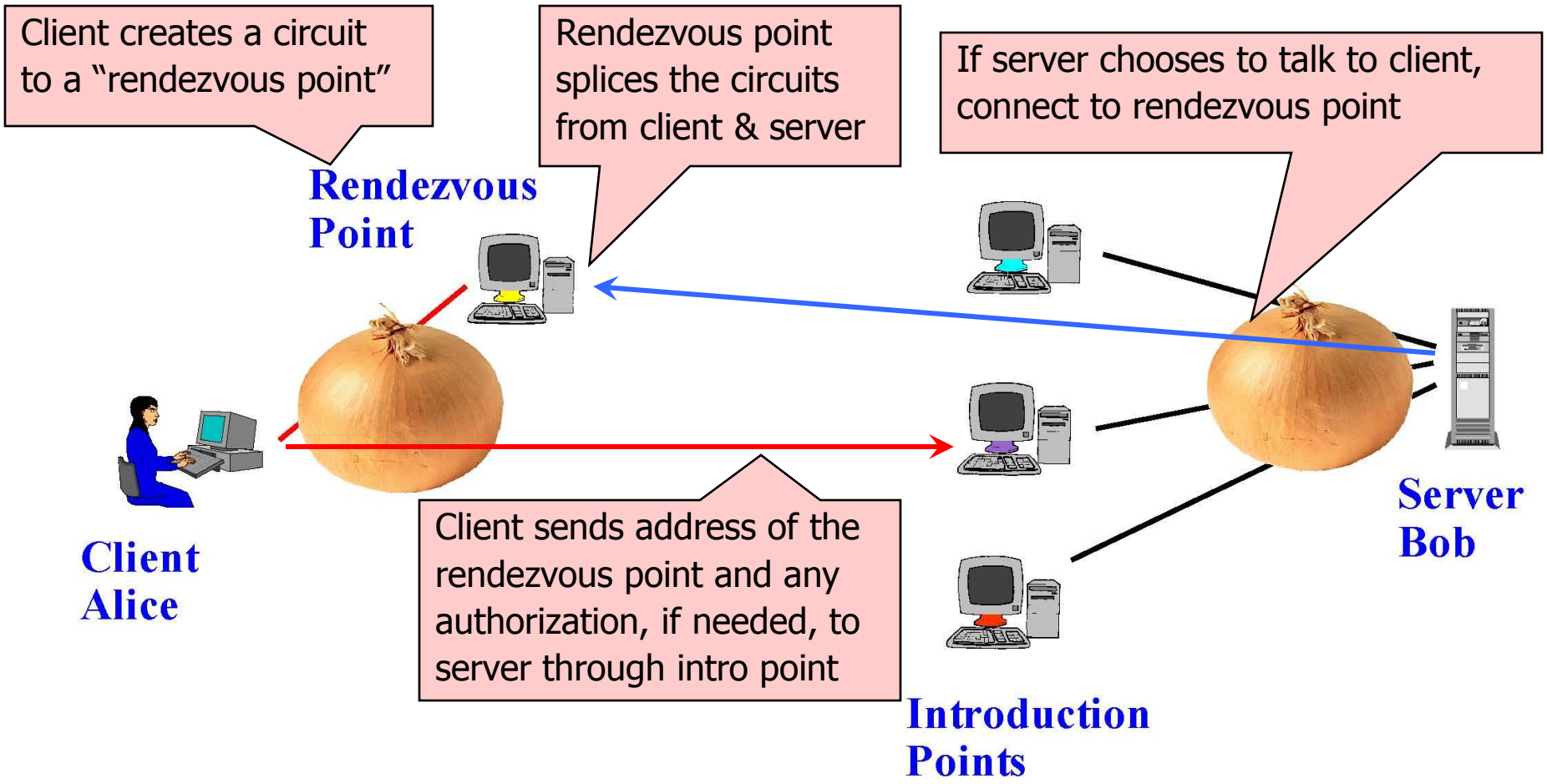
Location Hidden Services

- Goal: deploy a server on the Internet that anyone can connect to without knowing where it is or who runs it
- Accessible from anywhere
- Resistant to censorship
- Can survive a full-blown DoS attack
- Resistant to physical attack
  - Can’t find the physical server!

Deploying a Hidden Service



Using a Hidden Service



Silk Road

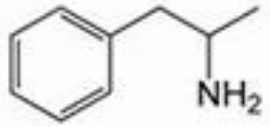
anonymous marketplace

Wel
messages(0) | orders(0) | account(฿0.00)

Shop by category:

- Drugs(1582)
  - Cannabis(271)
  - Dissociatives(33)
  - Ecstasy(217)
  - Opioids(106)
  - Other(65)
  - Prescription(274)
  - Psychedelics(306)
  - Stimulants(190)
- Apparel(37)
- Art(1)
- Books(300)
- Computer equipment(9)
- Digital goods(218)
- Drug paraphernalia(33)
- Electronics(13)

10 Grams high grade MDMA 80+%
฿61.17



Amphetamines sulfate / Speed freebase...
฿28.59

2g Jack Frost (weed) *420 SALE****
฿8.54

5 Grams of pure MDMA crystals
฿42.04

100 red Y tablets 111mg (lab tested)...
฿97.77

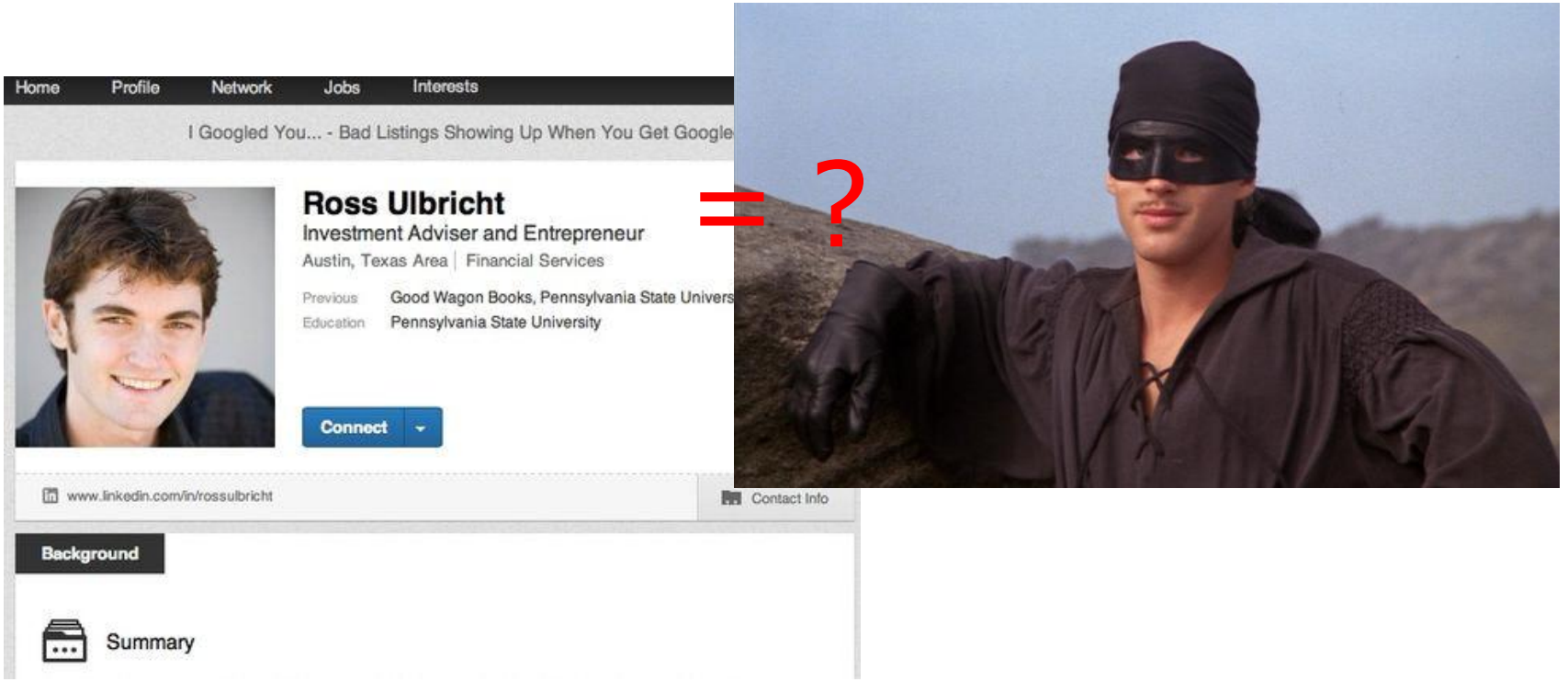
Michael Jackson Discography 1971-2009...
฿2.52

New

- Th or
- W fa
- Ac H
- A m A
- St A

Silk Road Shutdown

- Ross Ulbricht, alleged operator of the Silk Road Marketplace, arrested by the FBI on Oct 1, 2013



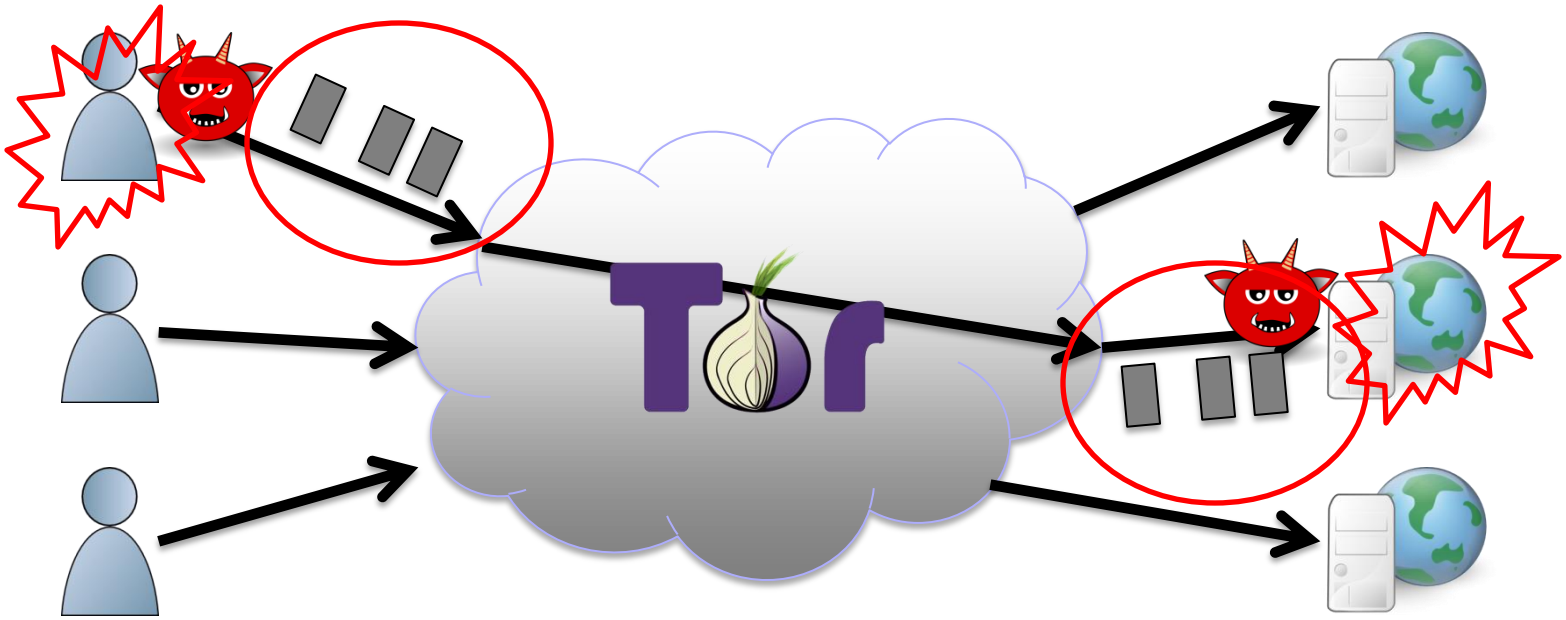
Silk Road Shutdown Theories

- A package of fake IDs from Canada traced to an apartment to San Francisco?
- A fake murder-for-hire arranged by DPR?
- A Stack Overflow question accidentally posted by Ulbricht under his real name?
  - "How can I connect to a Tor hidden service using curl in php?"
  - … a few seconds later, changed username to "frosty"
  - … oh, and the encryption key on the Silk Road server ends with the substring "frosty@frosty"
- Probably not weaknesses in Tor

How Was Silk Road Located?

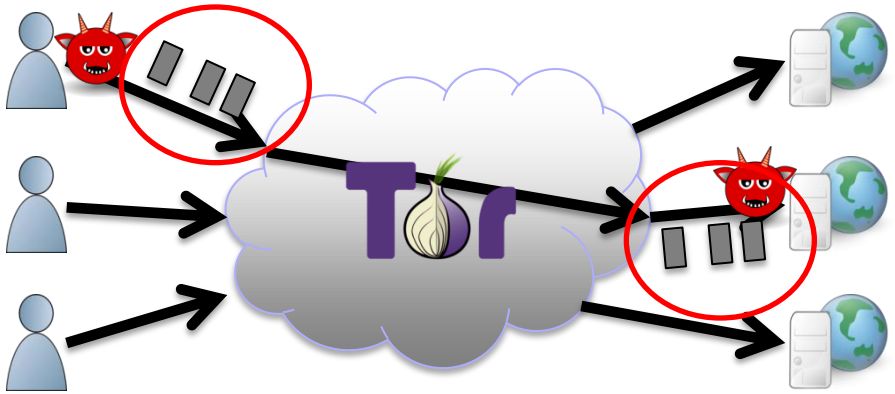
- FBI agent Tarbell's testimony:
  - Agents examined the headers of IP packets as they interacted with the Silk Road's login screen, noticed an IP address not associated with any Tor nodes
  - As they typed this address into the browser, Silk Road's CAPTCHA prompt appeared
  - Address led to rented server in a data center in Iceland
- Common problem: misconfigured software does not send all traffic via Tor, leaks IP address
  - Is this really what happened with the Silk Road server?

Main (?) Tor Problem



Traffic correlation and confirmation

Traffic Confirmation Techniques



- Congestion and denial-of-service attacks
  - Attack a Tor relay, see if circuit slows down
- Throughput attacks
- Latency leaks
- Website fingerprinting

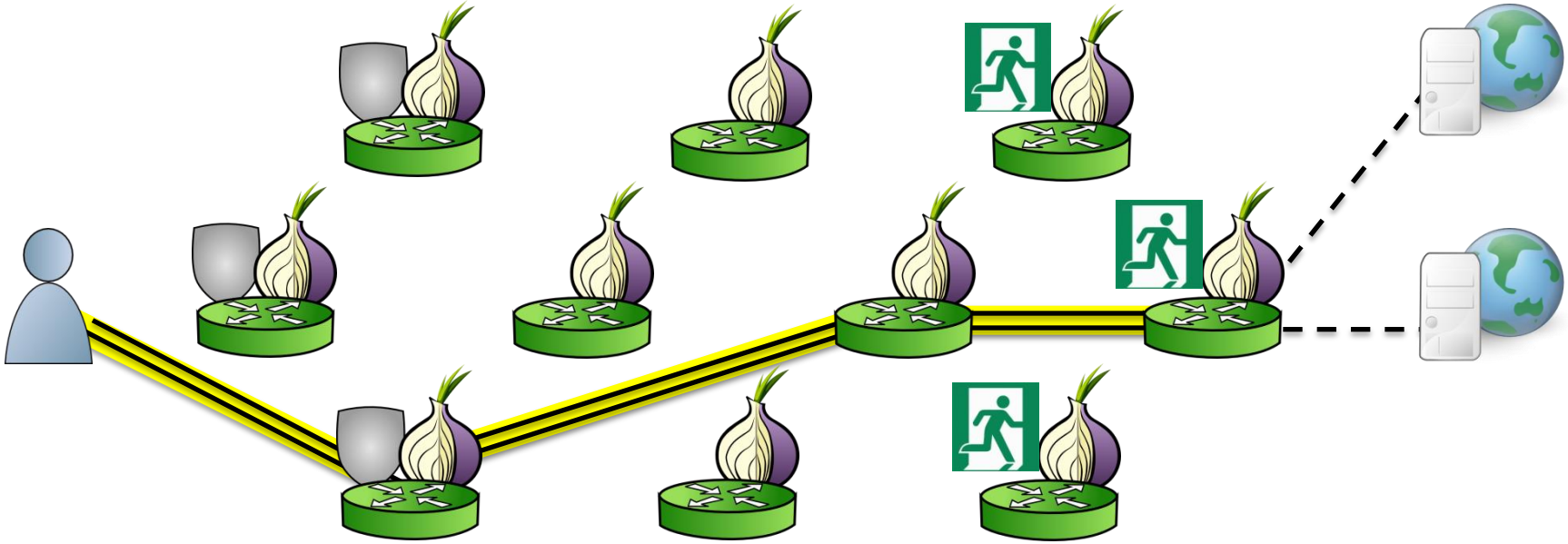
Tor Adversaries

[Johnson et al. "Users Get Routed". CCS 2013]

A realistic model of Tor adversaries needs to incorporate:

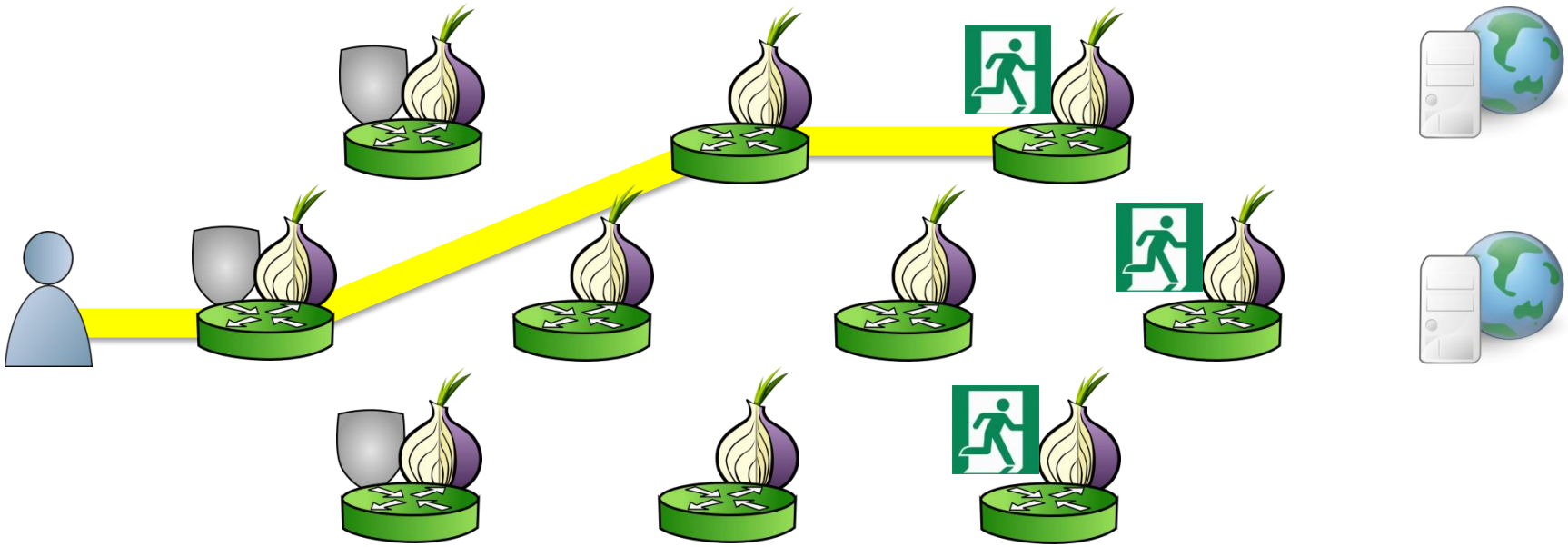
- Autonomous systems and Internet exchange points
- Evolution of Internet topology over time
- Traffic generated by typical applications over time

Using Tor Circuits



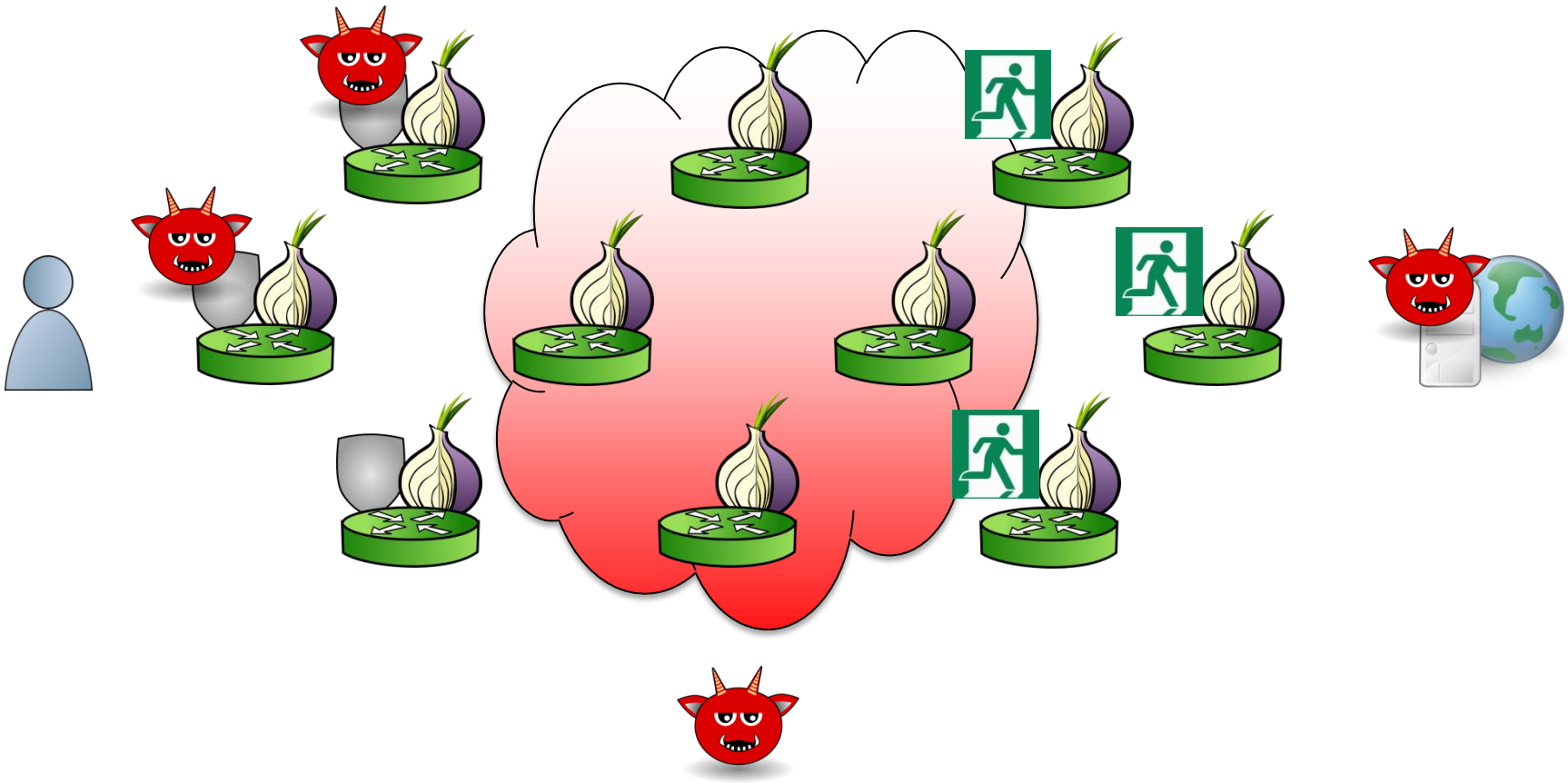
1. Clients begin all circuits with a selected guard
2. Relays define individual exit policies
3. Clients multiplex streams over a circuit

Using Tor Circuits



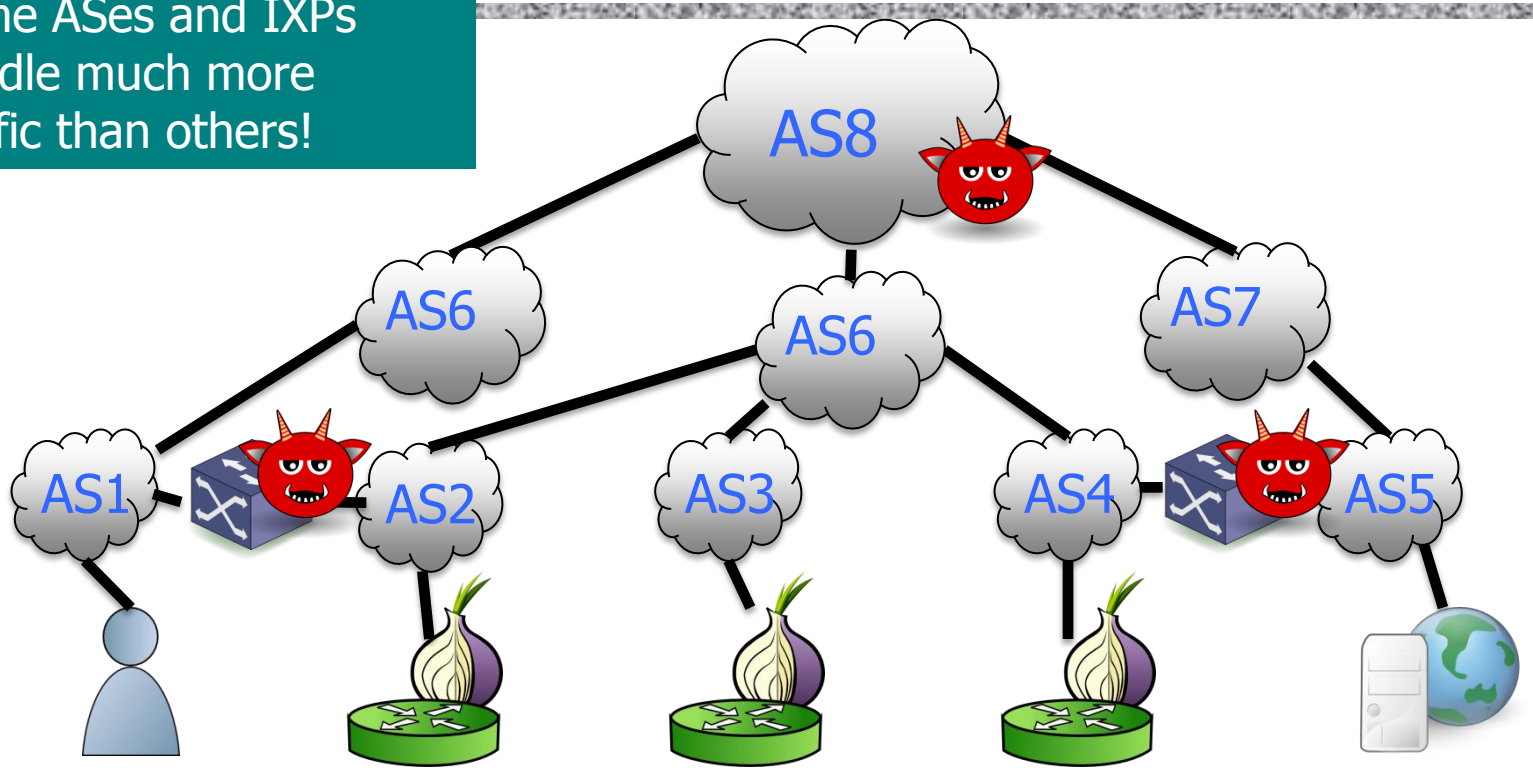
1. Clients begin all circuits with a selected guard
2. Relays define individual exit policies
3. Clients multiplex streams over a circuit
4. New circuits replace existing ones periodically

Node Adversaries



Link Adversaries

Some ASes and IXPs handle much more traffic than others!



Adversary has fixed location, may control one or more autonomous systems or Internet exchange points (IXP)

Modeling User Behavior


Gmail/GChat


Gcal/GDocs


Facebook


Web search

One session at
9:00, 12:00, 15:00, and 18:00
Su-Sa


IRC

Repeated sessions
8:00-17:00, M-F


BitTorrent

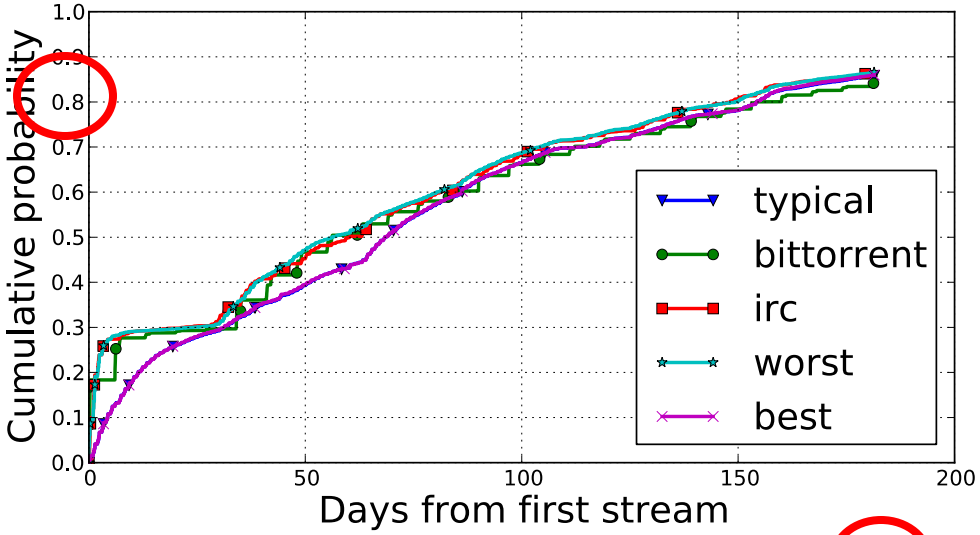
Repeated sessions
0:00-6:00, Sa-Su

20-minute traces

TorPS: The Tor Path Simulator

- Realistic client software model based on the current Tor
- Reimplemented path selection in Python
- Major path selection features:
  - Bandwidth weighting
  - Exit policies
  - Guards and guard rotation
  - Hibernation
  - /16 and family conflicts

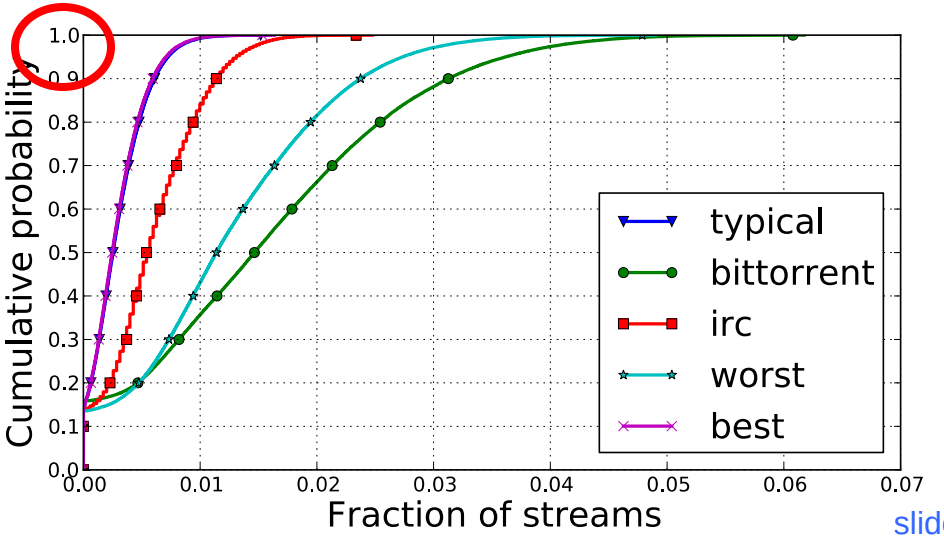
Node Adversary Success



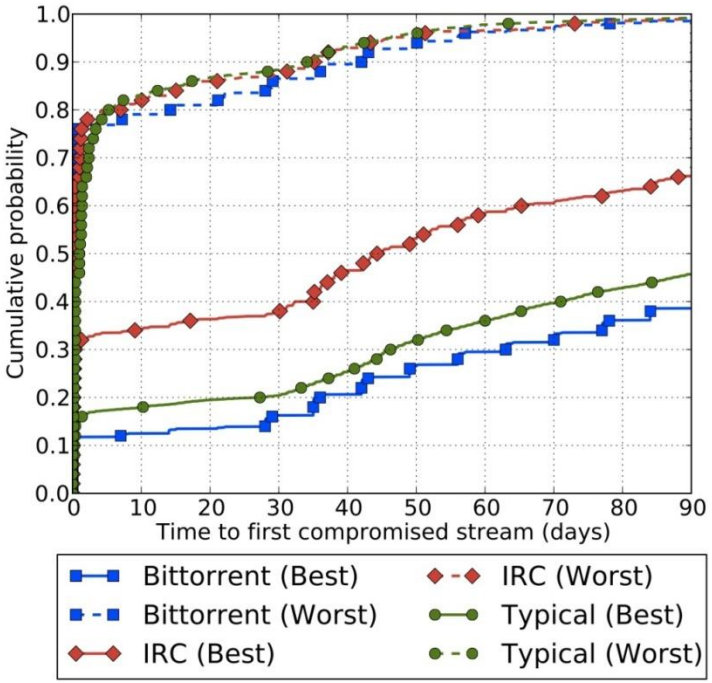
Adversary with total
100 MiB/s bandwidth
(83.3 guard, 16.7 exit)

Time to first
compromised stream

Fraction of
compromised streams



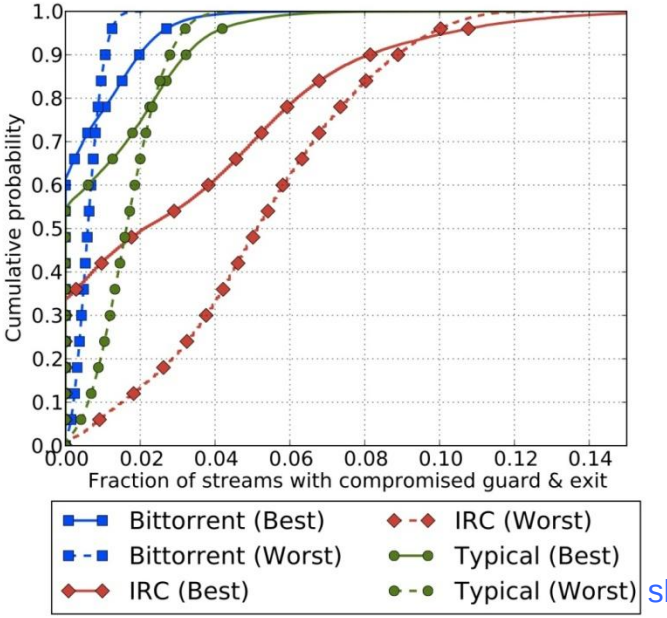
Link Adversary Success



Adversary controls one AS

"best" = most secure client AS,
"worst" = least secure

Time to first compromised stream

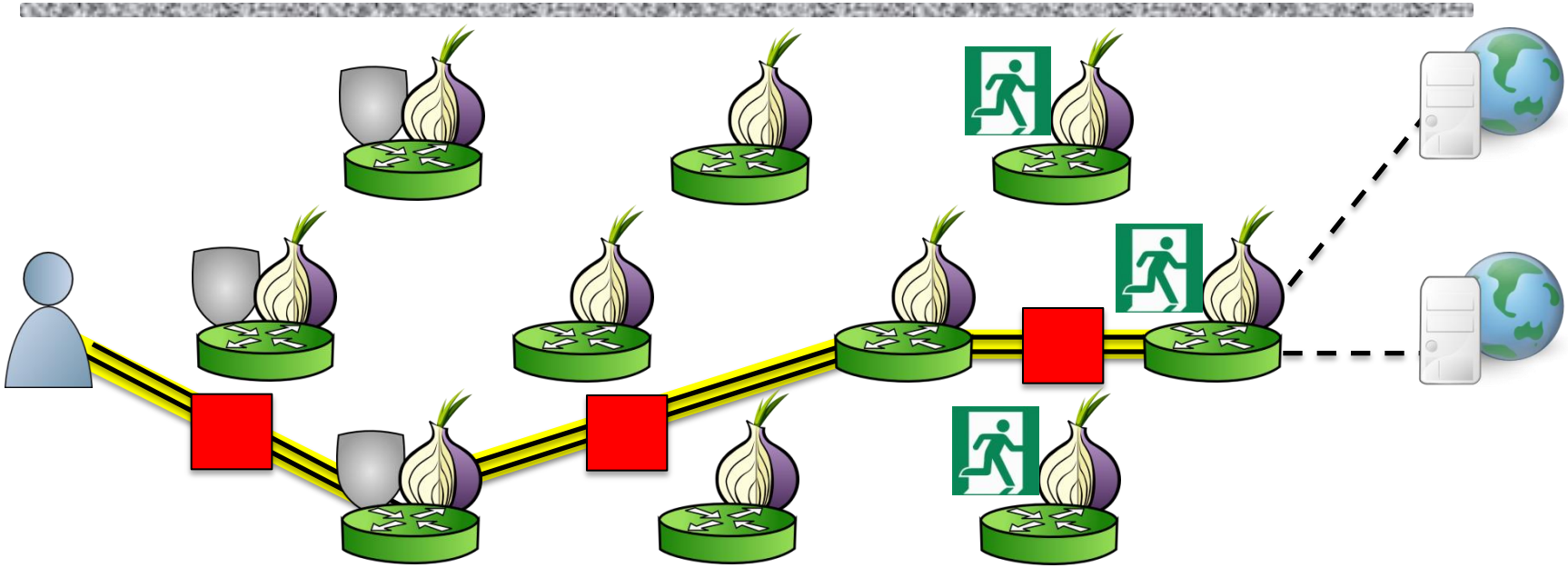


Fraction of compromised streams

Not a Theoretical Threat!

- Sybil attack + traffic confirmation
- In 2014, two CMU CERT "researchers" added 115 fast relays to the Tor network
  - Accounted for about 6.4% of available guards
  - Because of Tor's guard selection algorithm, these relays became entry guards for a significant chunk of users over their five months of operation
- The attackers then used these relays to stage a traffic confirmation attack

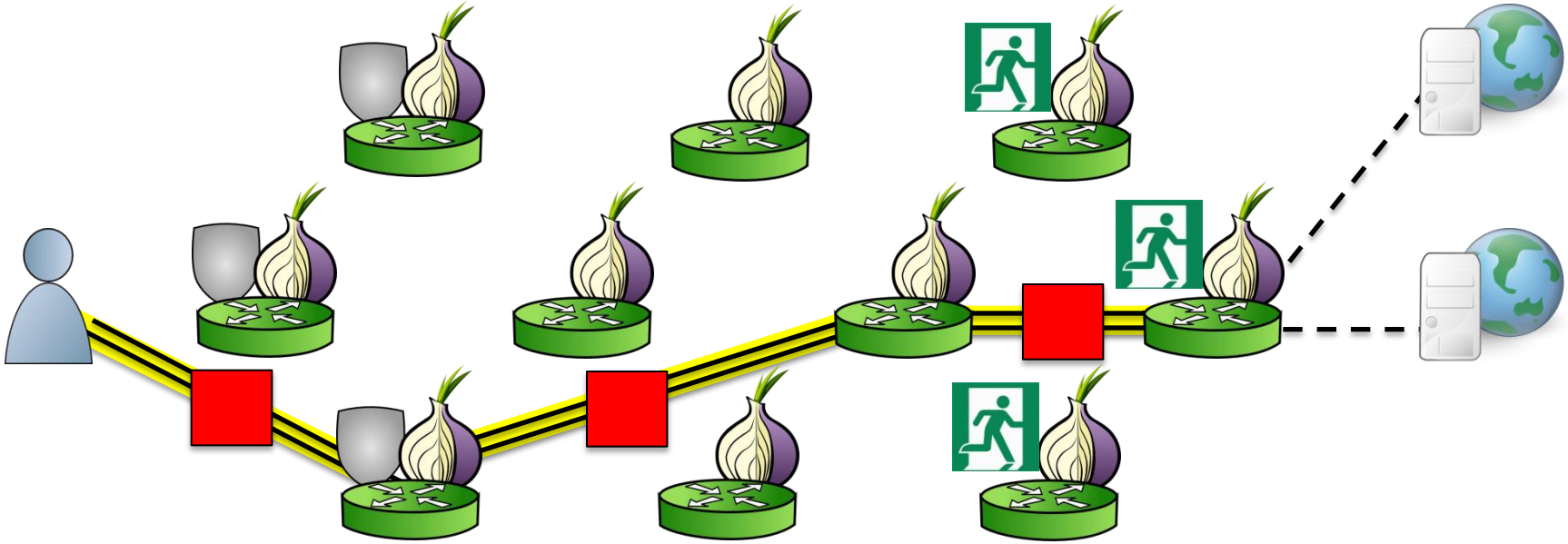
RELAY_EARLY Cell



Special control cell sent to the other end of the circuit (not just the next hop, like normal cell)

Used to prevent building very long Tor paths

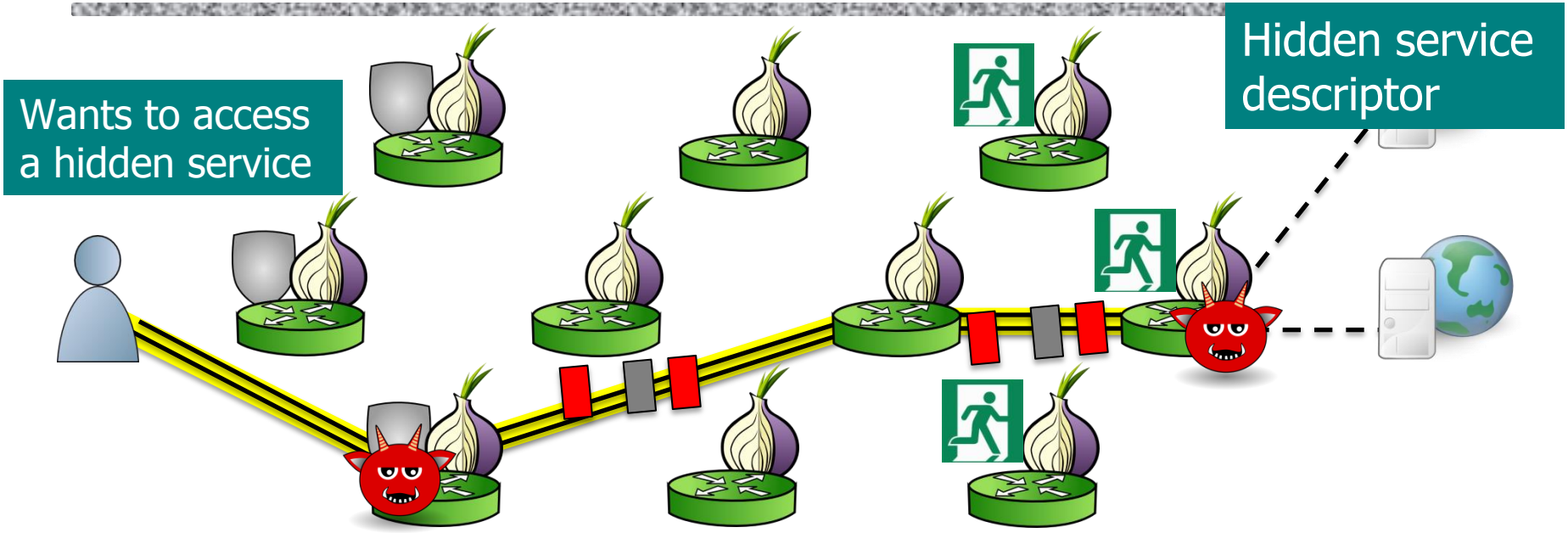
RELAY_EARLY Sent Backward



Any number of RELAY_EARLY cells can be sent backward along the circuit

No legitimate reason for this, just an oversight

Traffic Confirmation

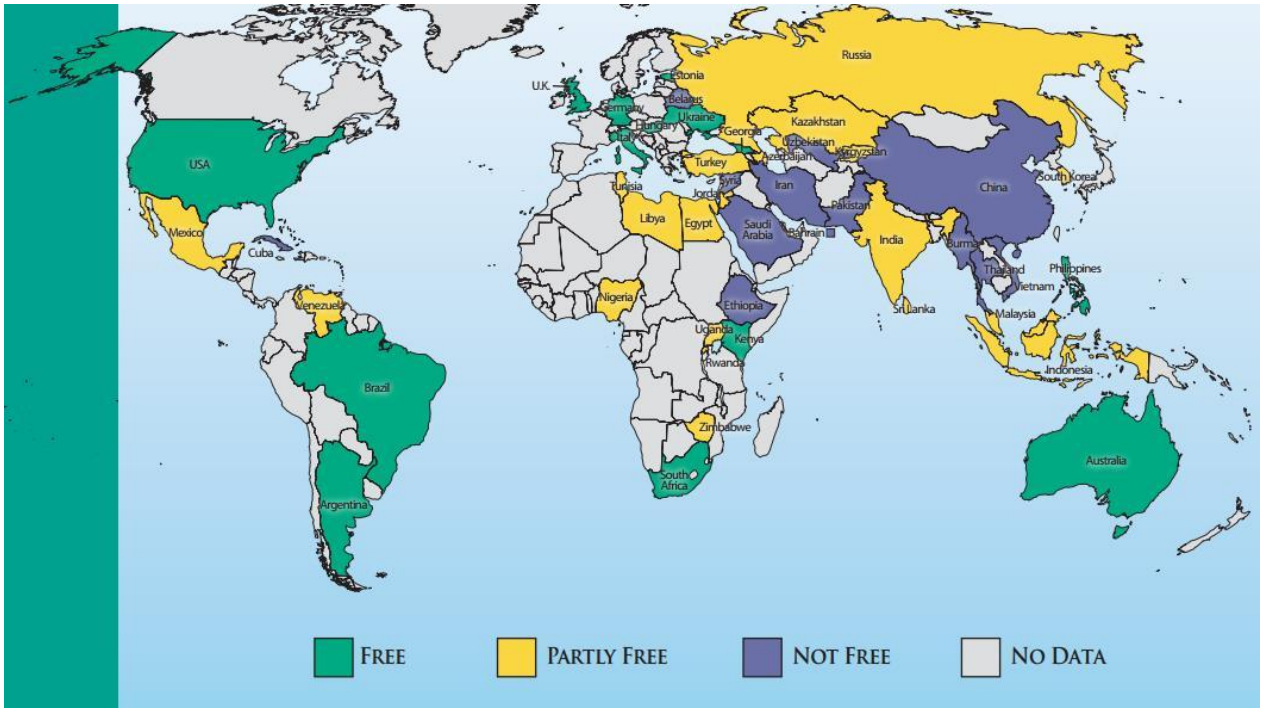


Malicious exit node encodes the name of hidden service in the pattern of relay and padding cells

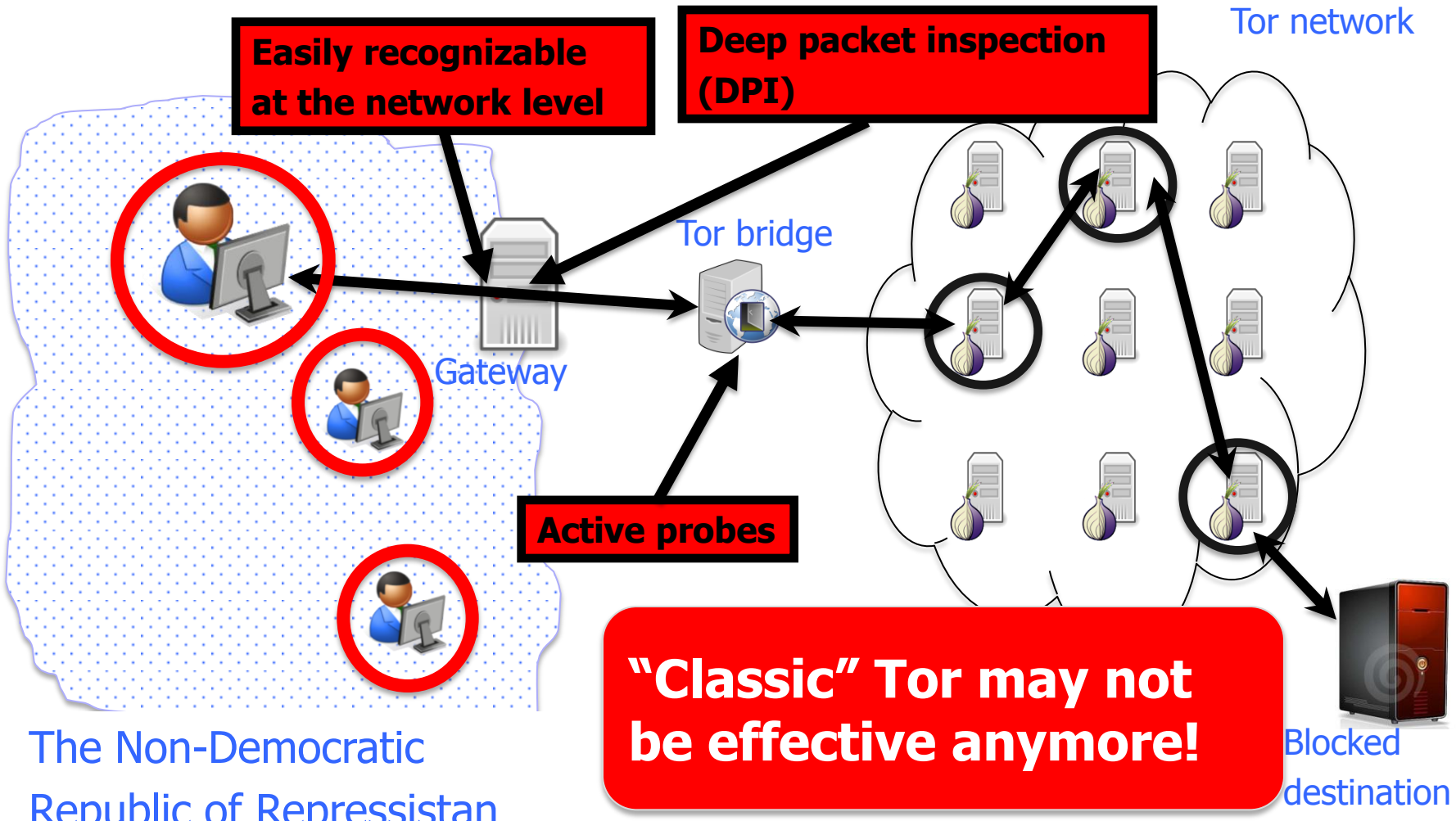
Malicious guard learns which hidden service the client is accessing

Fighting Internet Censorship

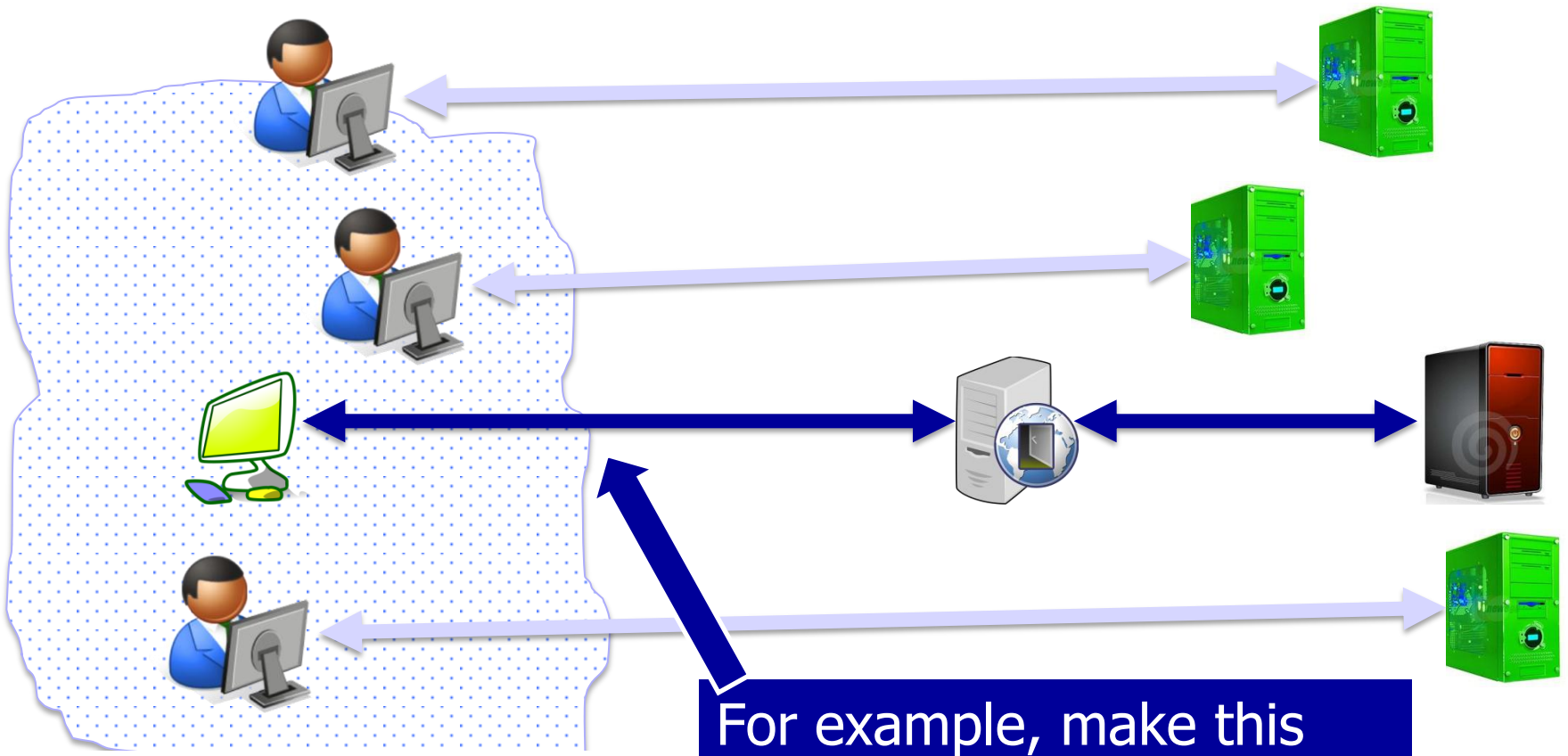
- Key use of anonymity networks – circumventing Internet censorship



Using Tor for Circumvention



Let's Play Hide-and-Seek



The Non-Democratic Republic of Repressistan

For example, make this look like a Skype

Goal: Unobservability

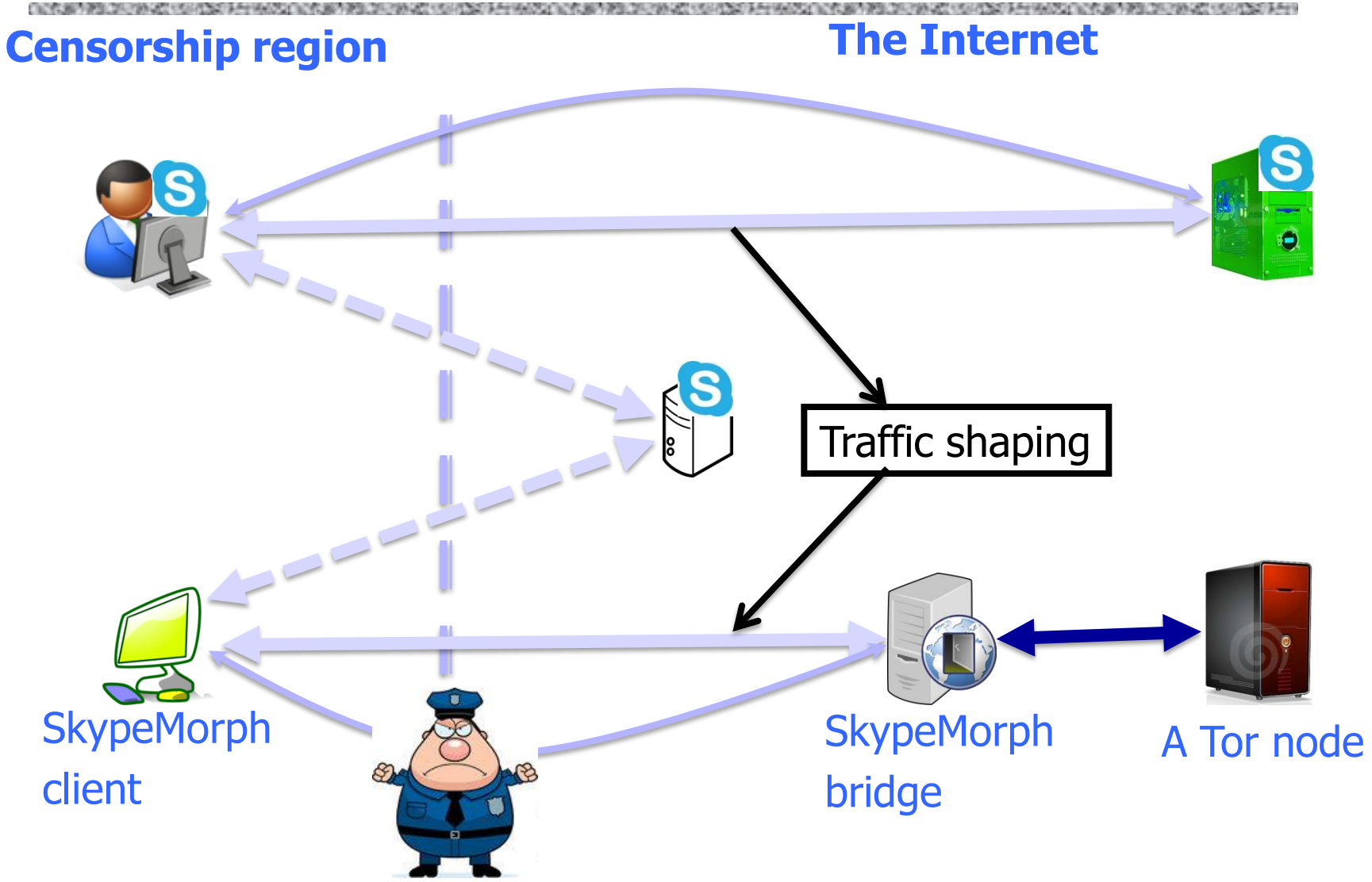
Censors should not be able to identify circumvention traffic, clients, or servers through passive, active, or proactive techniques

Unobservability by Imitation

- "Parrot systems" imitate a popular protocol like Skype or HTTP
  - SkypeMorph (CCS 2012)
  - StegoTorus (CCS 2012)
  - CensorSpoofer (CCS 2012)


What's, uh...
What's wrong with it?
'E's dead, that's what's wrong with it!

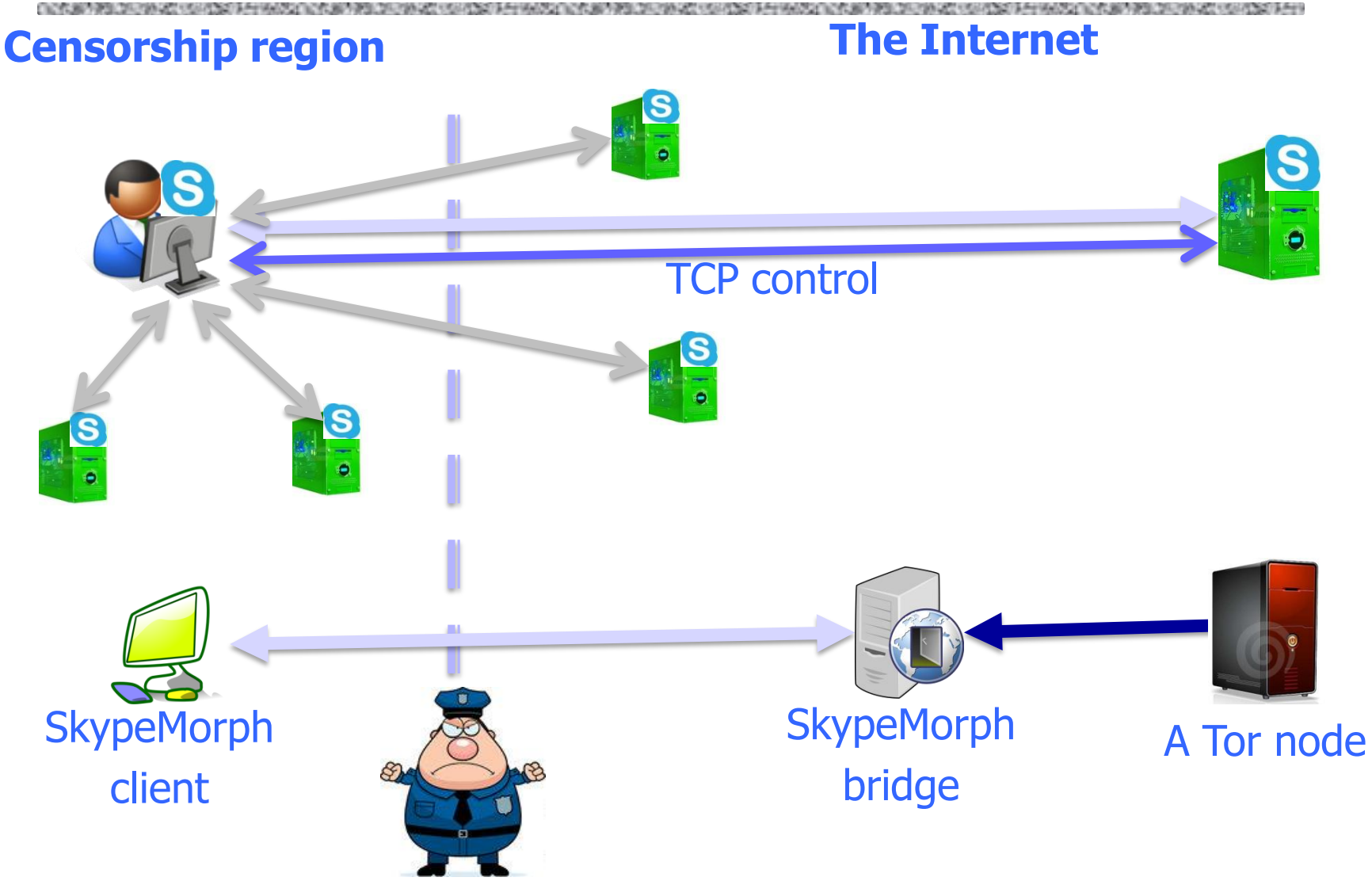
SkypeMorph


Censorship region
The Internet
Traffic shaping
SkypeMorph client
SkypeMorph bridge
A Tor node

Incorrect Packet Headers

- The start of message (SoM) header field is MISSING
- This is a single-packet identifier for SkypeMorph traffic
  - No need for sophisticated statistical traffic analysis

Missing Control Channels



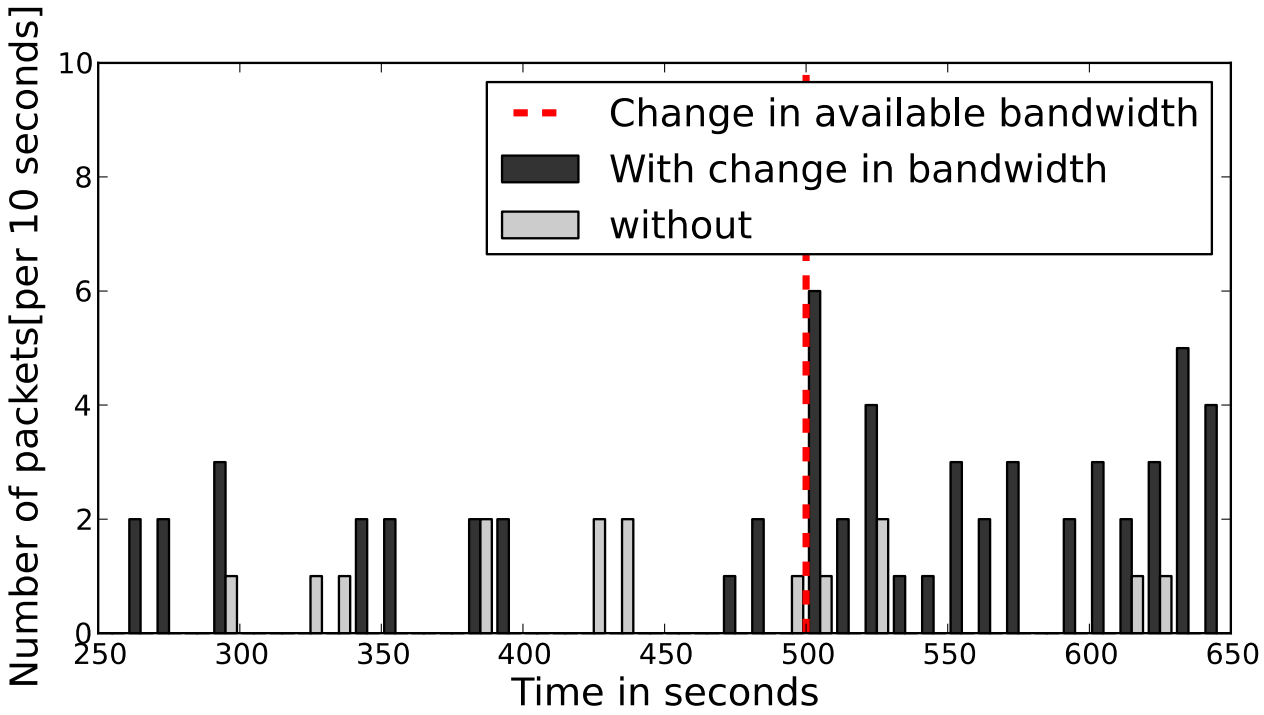

No, no.....No,
'e's stunned!

SkypeMorph+

Let’s imitate the missing parts!

- Problem: hard to mimic dynamic behavior
  - Active and proactive tests

Dropping UDP Packets

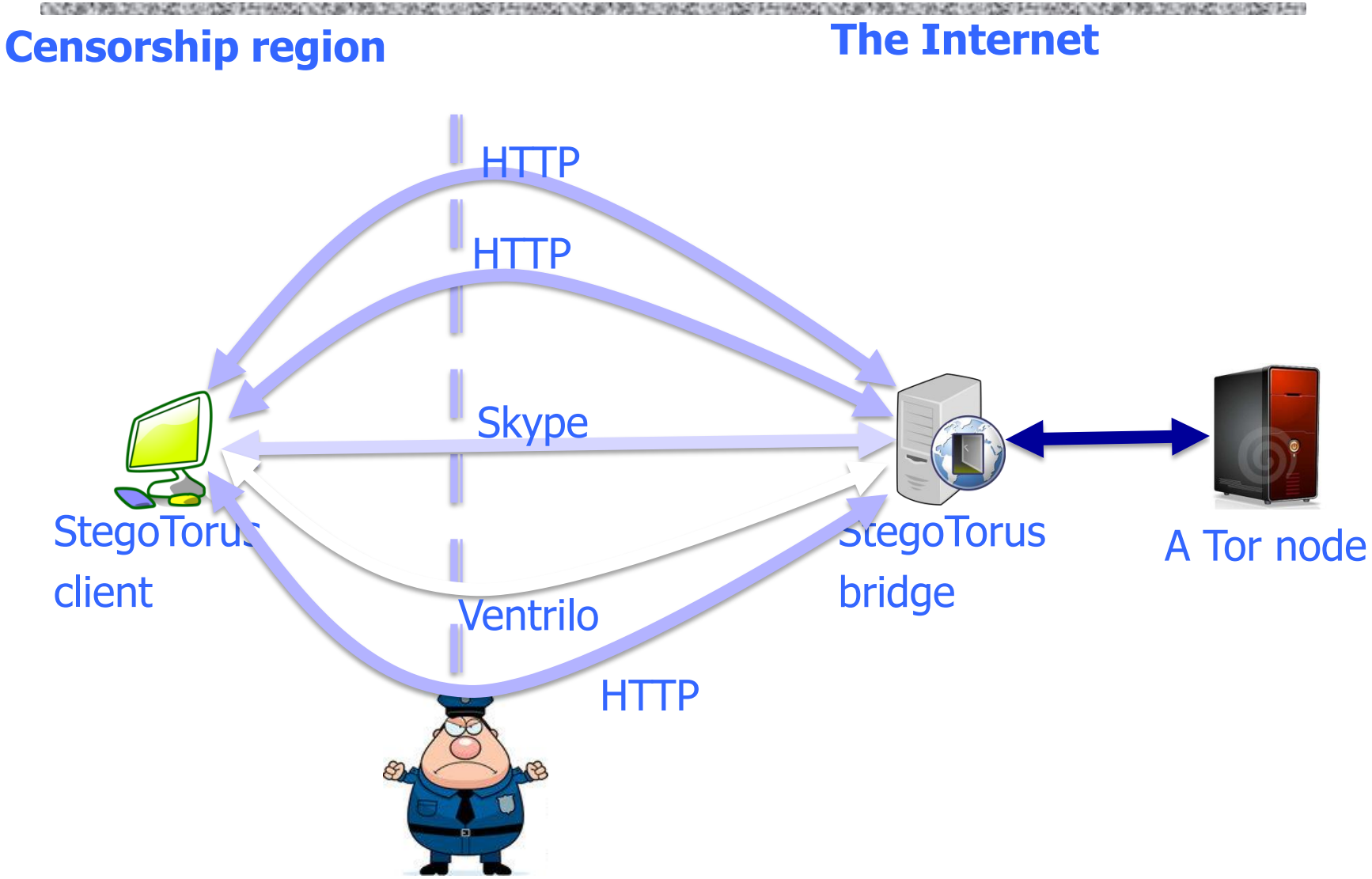


Other Tests

Test	Skype	SkypeMorph+
Flush Supernode cache	Serves as a SN	Rejects all Skype messages
Drop UDP packets	Burst of packets in TCP control	No reaction
Close TCP channel	Ends the UDP stream	No reaction
Delay TCP packets	Reacts depending on the type of message	No reaction
Close TCP connection to a SN	Initiates UDP probes	No reaction
Block the default TCP port	Connects to TCP ports 80 and 443	No reaction

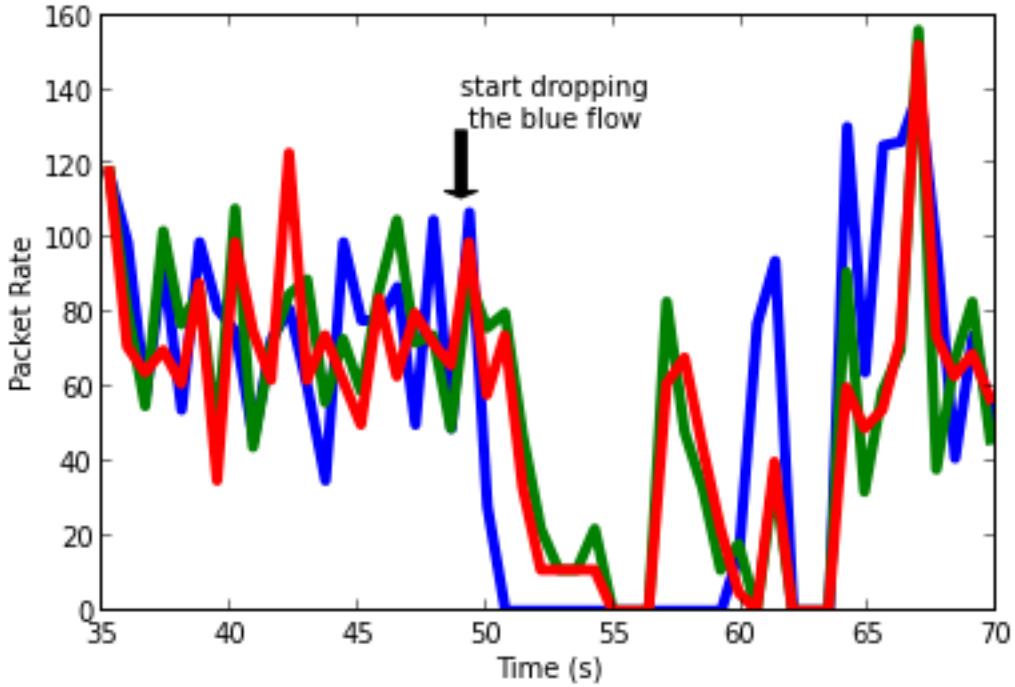

No no!
'E's pining!
'E's not pinin'!
'E's expired and gone to meet 'is maker!

StegoTorus


Censorship region
The Internet
HTTP
HTTP
Skype
Ventrilo
HTTP
StegoTorus client
StegoTorus bridge
A Tor node

StegoTorus Chopper

- Dependencies between links



StegoTorus-HTTP

- Does not look like any HTTP server!
- Most HTTP methods not supported!

HTTP request	Real HTTP server	StegoTorus's HTTP module
GET existing	Returns "200 OK" and sets `Connection` to `keep-alive`	Arbitrarily sets `Connection` to either `keep-alive` or `Close`
GET long request	Returns "404 Not Found" since URI does not exist	No response
GET non-existing	Returns "404 Not Found"	Returns "200 OK"
GET wrong protocol	Most servers produce an error message, e.g., "400 Bad Request"	Returns "200 OK"
HEAD existing	Returns the common HTTP headers	No response
OPTIONS common	Returns the supported methods in the Allow line	No response
DELETE existing	Most servers have this method not activated and produce an error message	No response
TEST method	Returns an error message, e.g., "405 Method Not Allowed" and sets `Connection=Close`	No response
Attack request	Returns an error message, e.g., "404 Not Found"	No response


Now that's what
I call a dead parrot

Lesson #1

Unobservability by imitation is fundamentally flawed!

Imitating a Real System Is Hard

Not enough to mimic a "protocol," need to mimic a specific implementation with all its quirks

- A complex protocol in it entirety
- Inter-dependent sub-protocols with complex, dynamic behavior
- Bugs in specific versions of the software
- User behavior

Lesson #2

Partial imitation is worse than no imitation

Bad imitation of Skype is easier to recognize than Tor

This is an ex-parrot!
This parrot is no more
This is a late parrot
It's stone dead