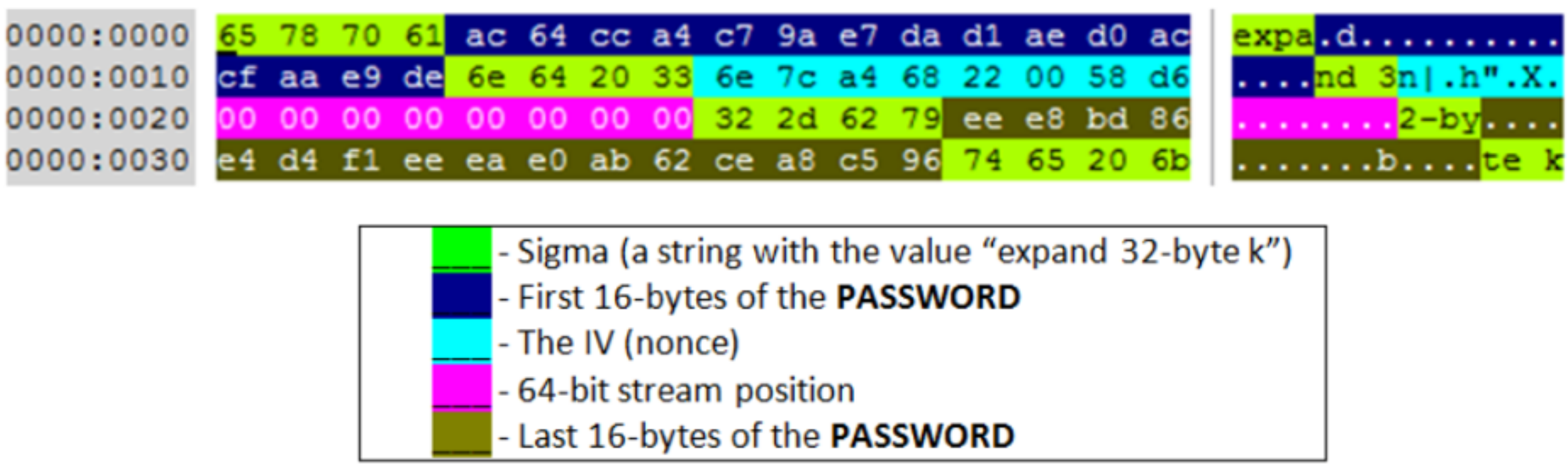
PASSWORD GENERATOR TOOL BREAKS PETYA RANSOMWARE ENCRYPTION

threatpost

by Chris Brook April 11, 2016 , 2:33 pm

Researchers have been combing through code related to the Petya ransomware long enough they've been able to cobble together a decryption tool that should allow most victims to generate keys in less than 10 seconds.

The original SALSA20 implementation uses a 32-byte encryption key and an 8-byte initialization vector to produce the final 512-bit key-stream:



Petya's implementation of this simple encryption key generation is seriously flawed, which allows us to predict 256 bits out of the total 512 used in the key-stream. With this knowledge, we can brute force the encryption in a very reasonable time-frame, breaking the encryption and subverting Petya's malicious actions without paying any ransom at all.

surveillance & anonymity

cs642

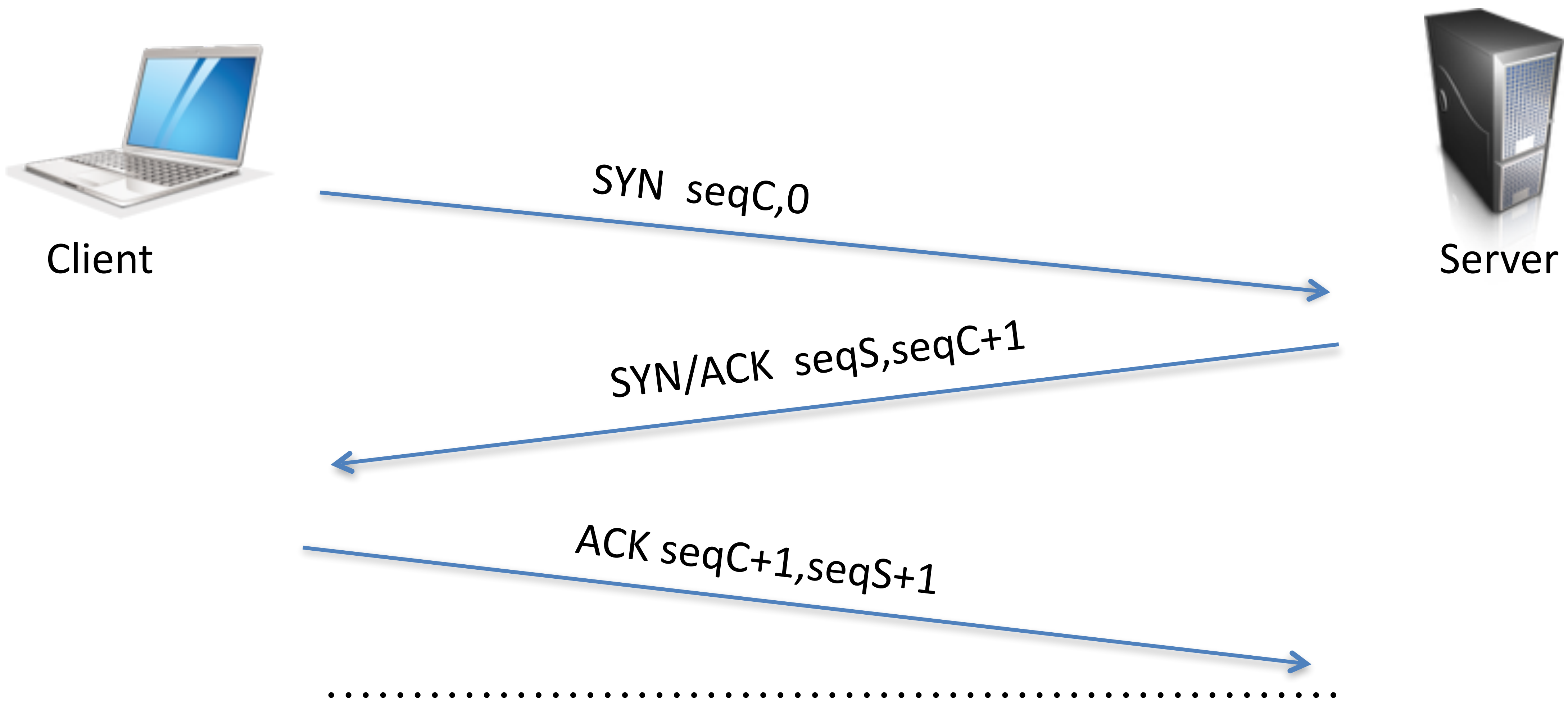
computer security

adam everspaugh
ace@cs.wisc.edu

today

* Internet-wide scanning, zmap
* Massive surveillance, packet inspection
* Anonymous browsing, TOR

TCP handshake



TCP connection established

SYN = syn flag set
ACK = ack flag set
x,y = x is sequence #, y is acknowledge #

mass scanning

- What if we want to scan the "whole internet"?
- Why?
  - Find all the unsecured webcams [shodani.io]
  - Find all the broken webservers
- How would we do this?
  - nmap -p 443 0.0.0.0/32
  - IPv4: 32-bits - 14% IANA reserved addresses
- How long would this take?
  - Assume mean round-trip time = 100ms

SHODAN

think-*pair*-share

zmap

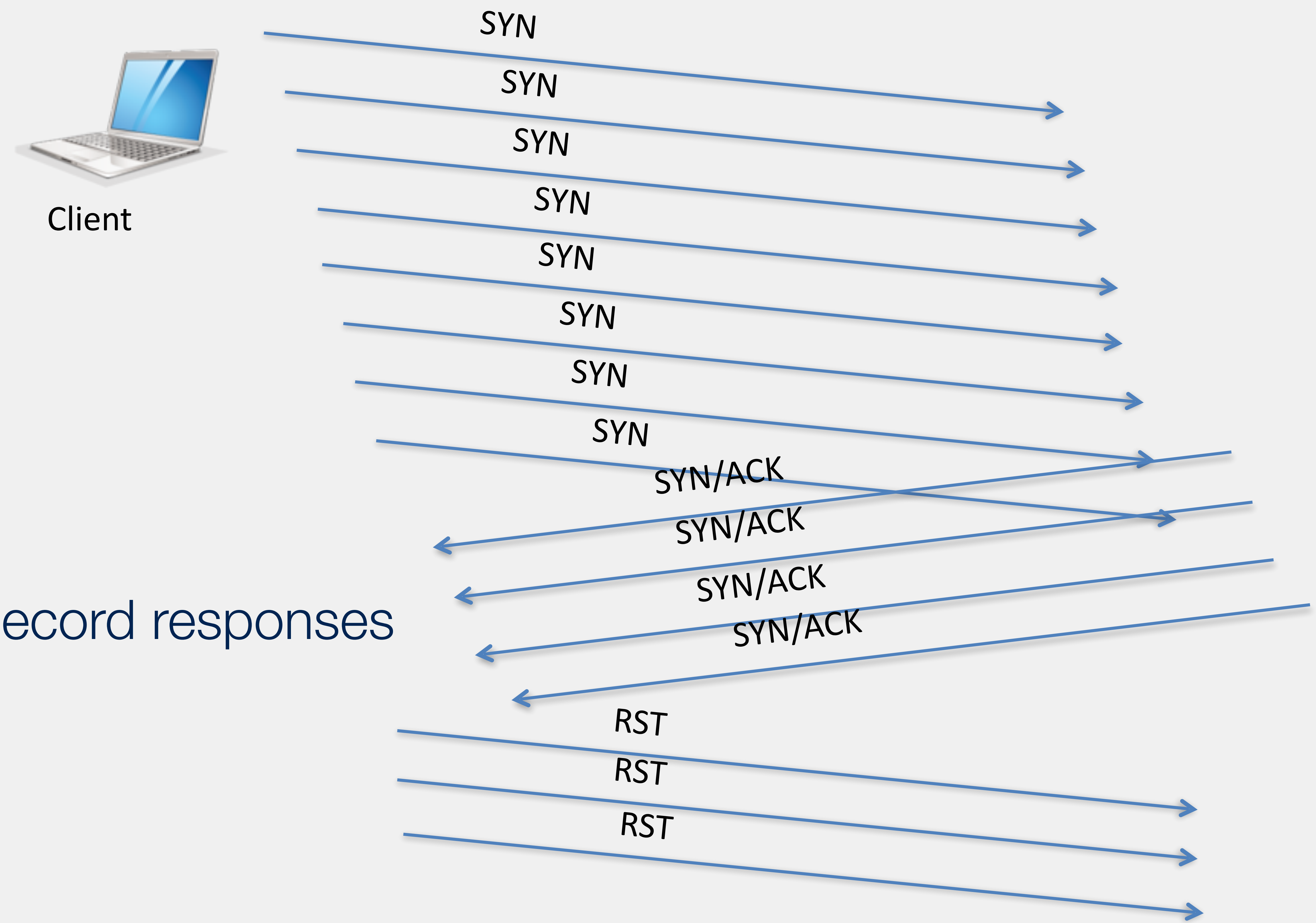
	Normalized Coverage	Duration (mm:ss)	Est. Internet Wide Scan
Nmap (1 probe)	81.4%	24:12	62.5 days
Nmap (2 probes)	97.8%	45:03	116.3 days
ZMap (1 probe)	98.7%	00:10	1:09:35
ZMap (2 probes)	100.0%	00:11	2:12:35

[zmap, Durumeric et al.]

ZMap paper: 1300x faster than nmap

How?

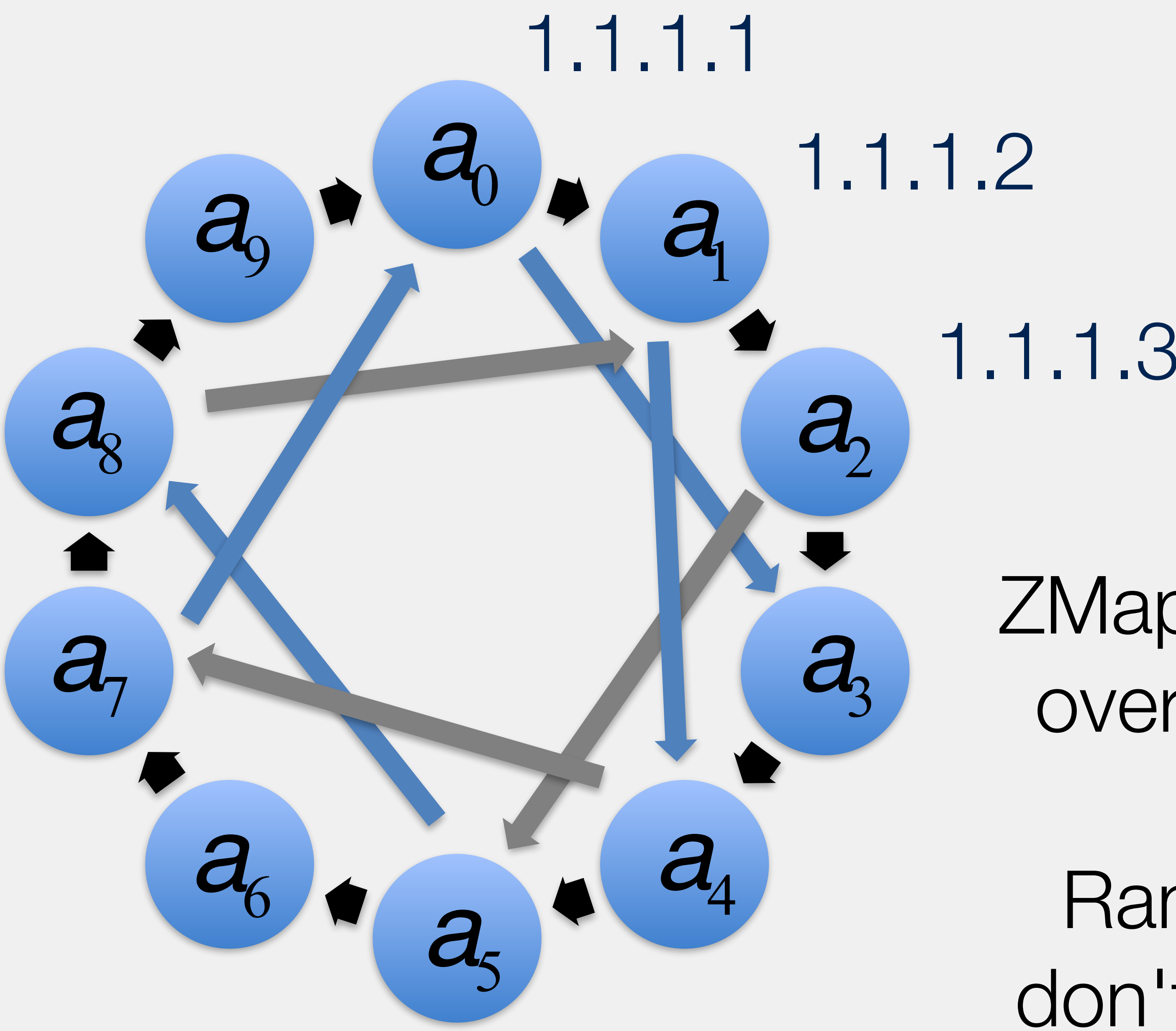
fast scanning



Record responses

zmap

Can't scan at high-speed
in-order
Why?



ZMap uses a permutation over the address space

Random ordering, but don't have to track list of scanned addresses

dual ec

* Investigating "rigged" random number generator (RNG) called "dual elliptic curve" (dual EC) RNG
* … that could be used in setting up TLS connections
* **Q:** How many web servers support this RNG in real life?
* Scanned IPv4 with ZMap
  / 39M servers responding on port 443
  / Took 48 hours from CSL@UW
* Probed each web server with instrumented OpenSSL client (recorded TLS handshake)
  / 22M TLS (half-)handshakes; took 4 weeks

[On the Practical Exploitability of Dual-EC, Checkoway et al.]

AT&T Wiretap case

- Mark Klein discloses potential wiretapping activities by NSA at San Francisco AT&T office
- Fiber optic splitter on major trunk line for Internet communications
  - Electronic voice and data communications copied to “secret room”
  - Narus STA 6400 device

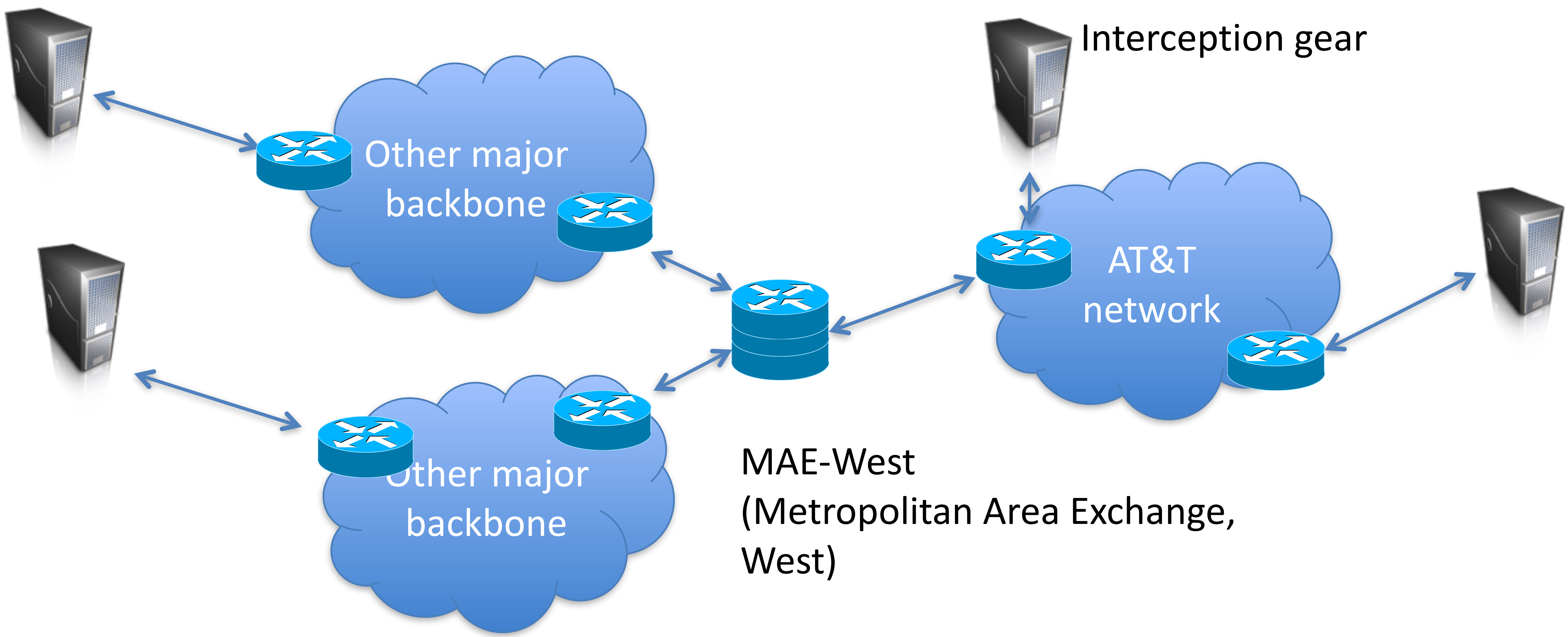
Interception technology

- From Narus website [http://narus.com/index.php/product/narusinsight-intercept]
  - “Target by phone number, URI, email account, user name, keyword, protocol, application and more”, “Service- and network agnostic”, “IPV 6 ready”
  - Collects at wire speeds beyond 10 Gbps

NARUS

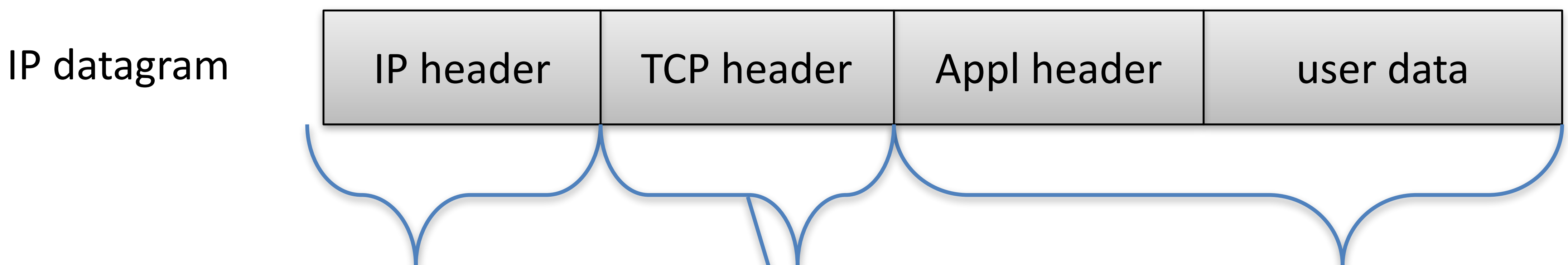
BOEING

Wiretap surveillance



Large amounts of Internet traffic cross relatively few key points

Types of packet inspection



Internet service providers need only look at IP headers to perform **routing**

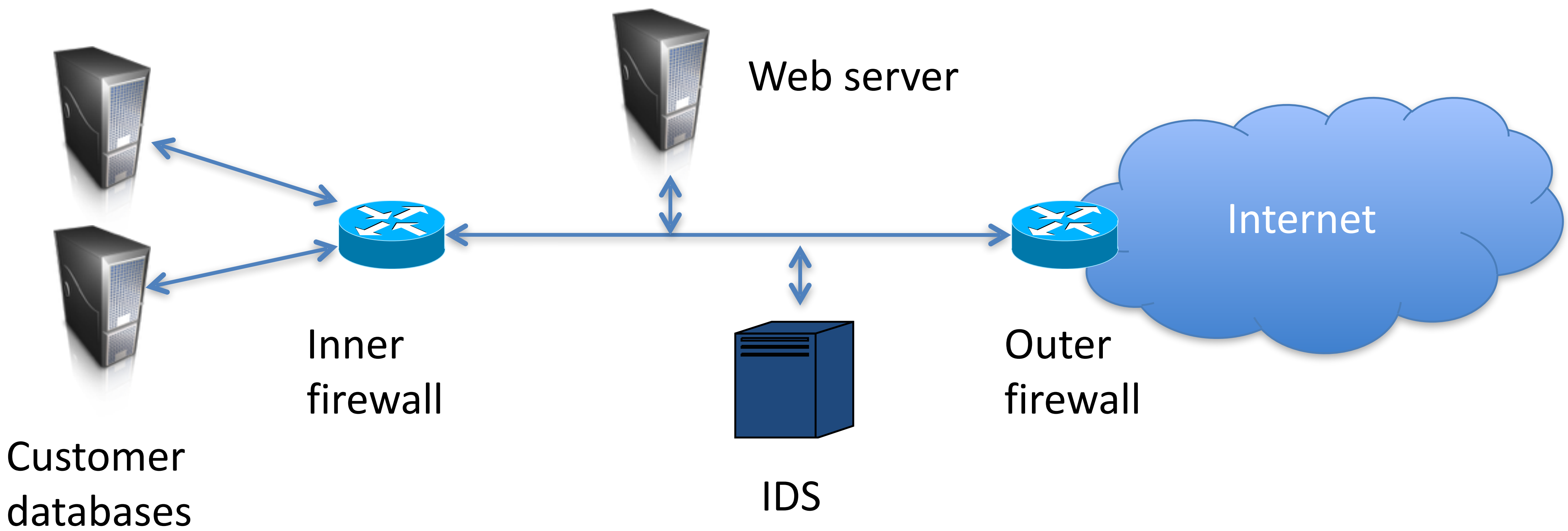
Shallow packet inspection investigates lower level headers such as TCP/UDP

Deep packet inspection (DPI) analyzes application headers and data

Which inspection is most powerful?

What are the technology challenges?

Intrusion Detection Systems (IDS)



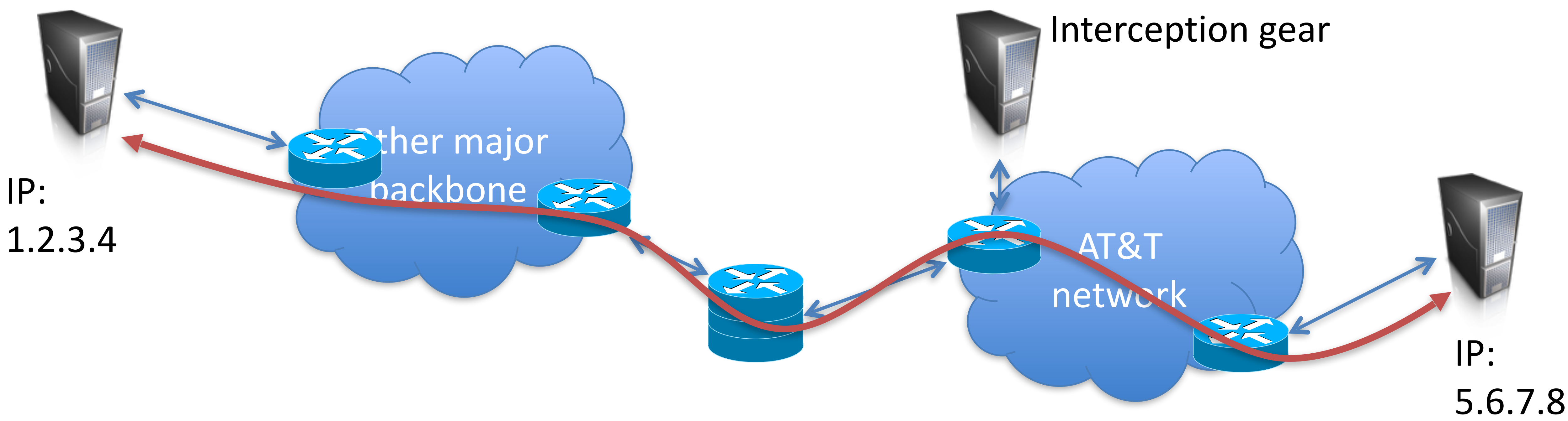
What can an IDS do that a router cannot?

Store information for forensics

Match known attack patterns (malware, XSS, SQL injection)

Preventing intercept

- End-to-end encryption (TLS, SSH)

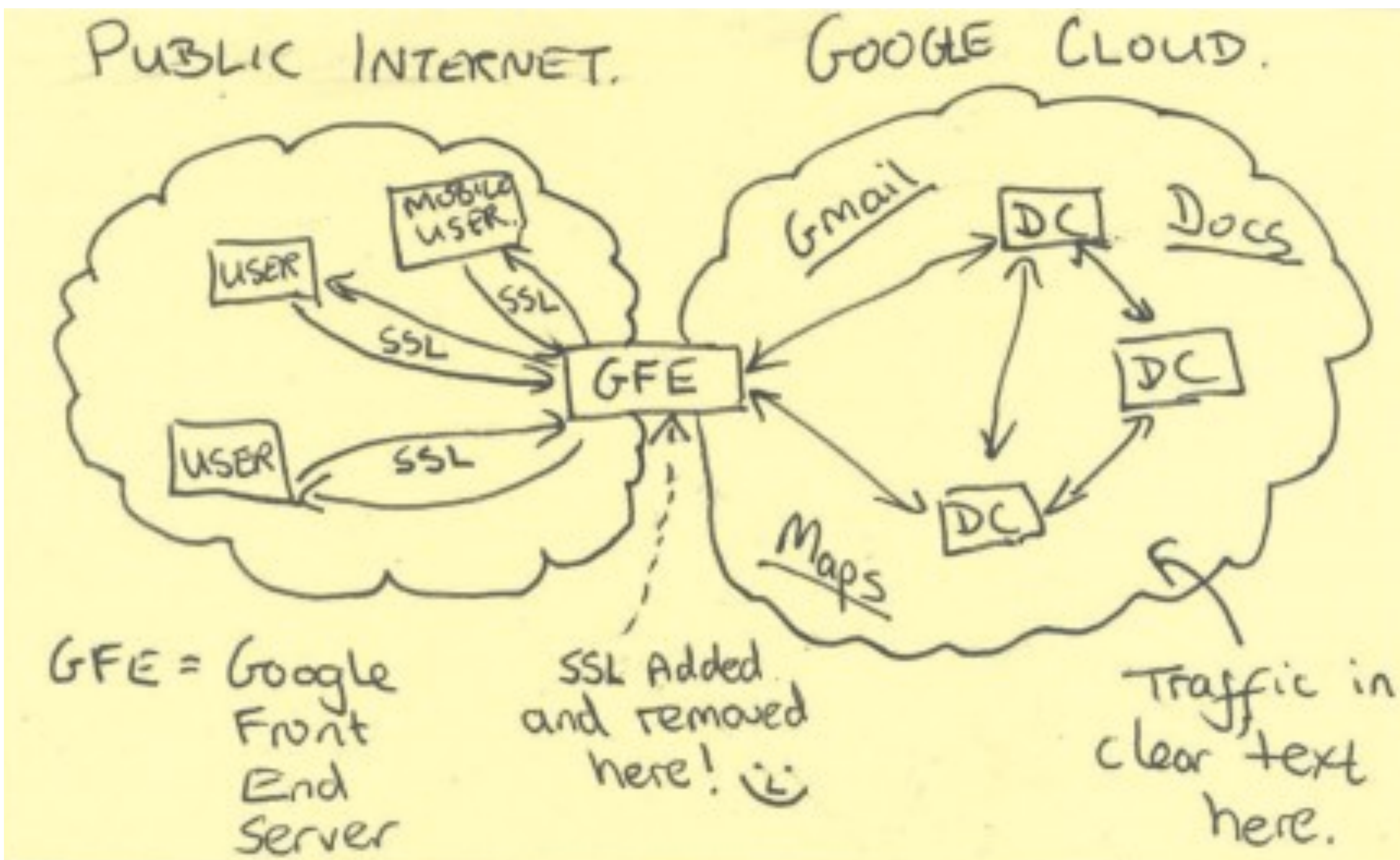


- What does this protect? What does it leak?
- What can go wrong?

think-*pair*-share

End-run around HTTPS

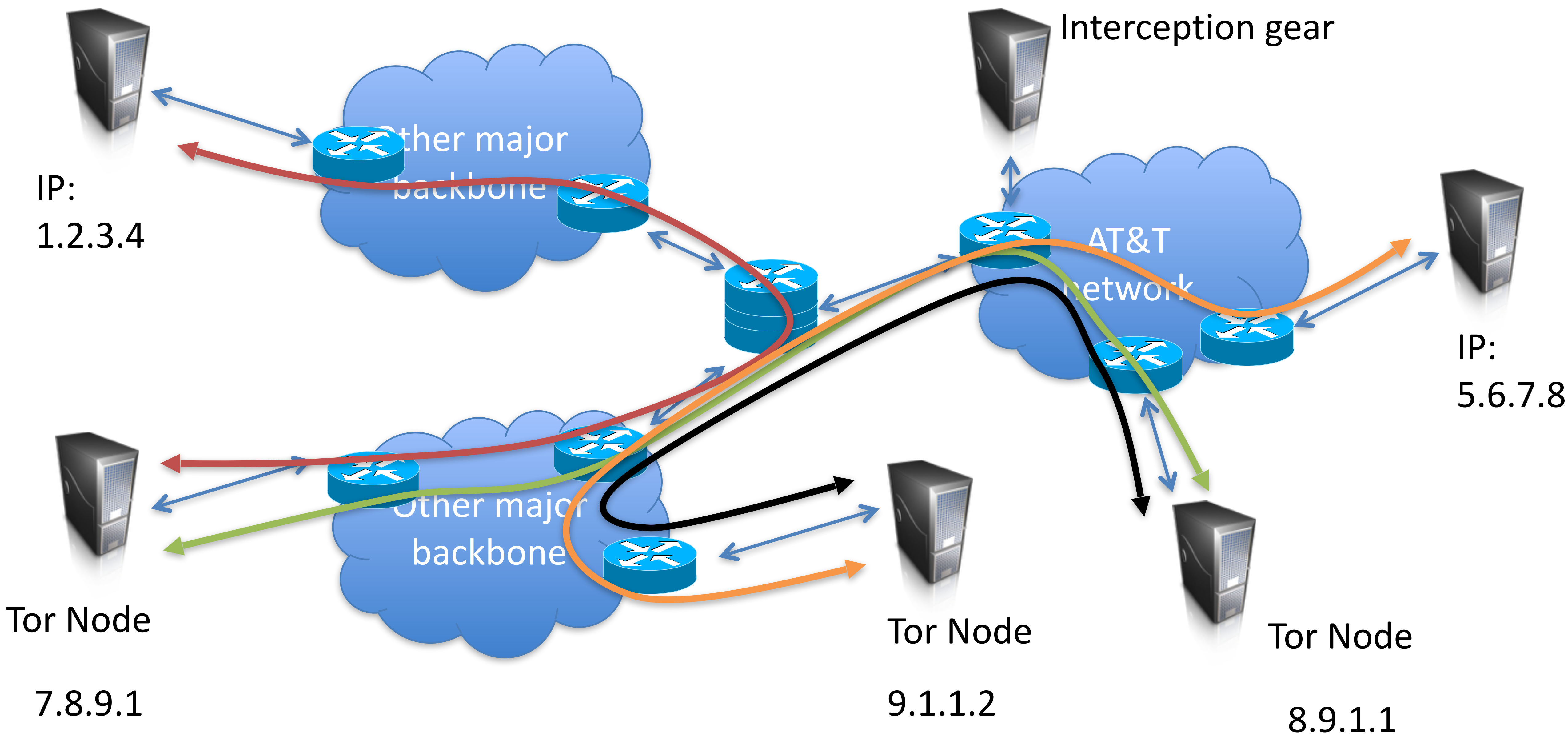
- HTTPS terminated at edge of Google networks
- Internal data center-to-data center communications on privately leased lines



Hiding connectivity is harder

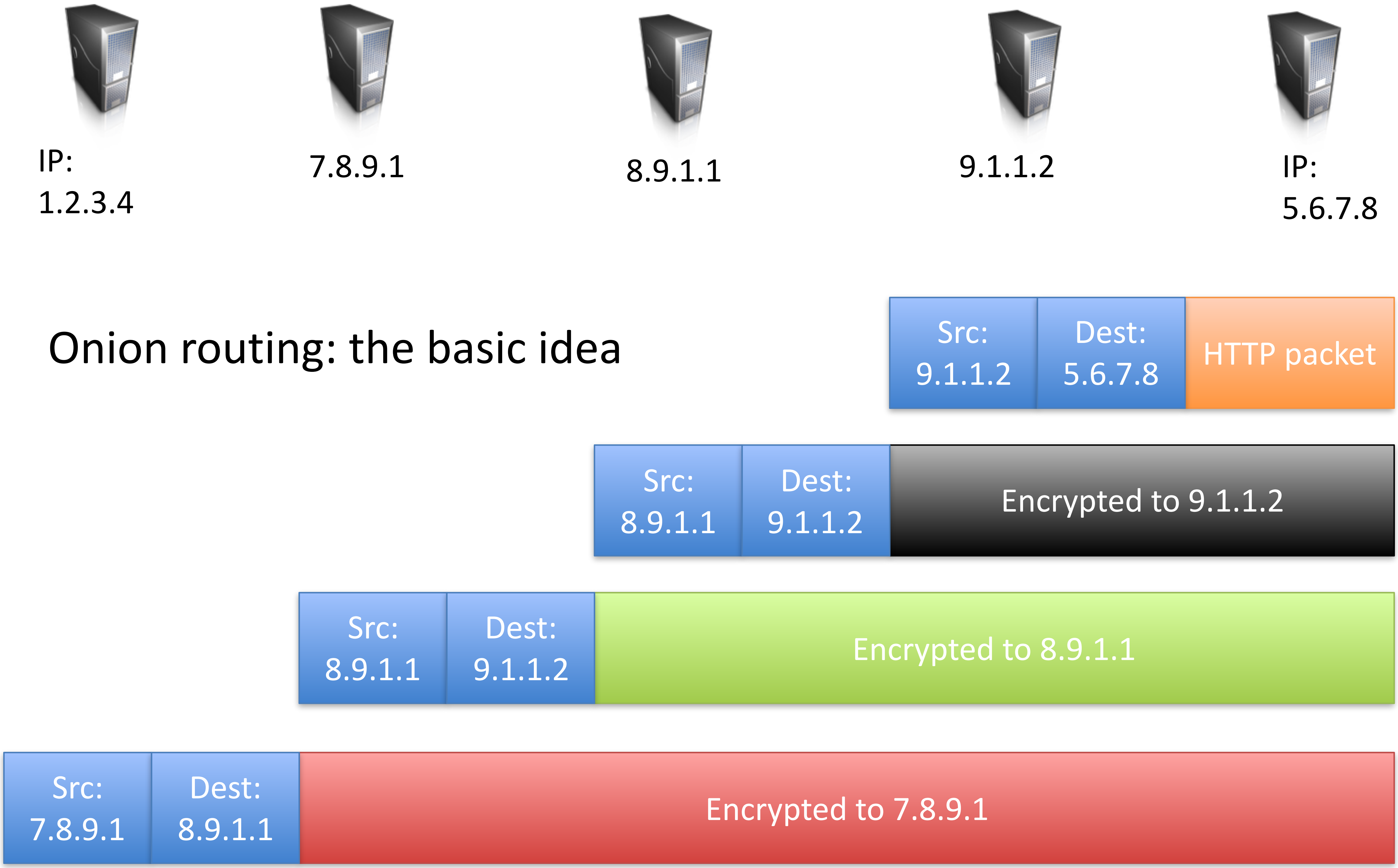
- IP addresses are required to route communication, yet not encrypted by normal end-to-end encryption
  - 1.2.3.4 talked to 5.6.7.8 over HTTPs
- How can we hide connectivity information?

Tor (The Onion Router)



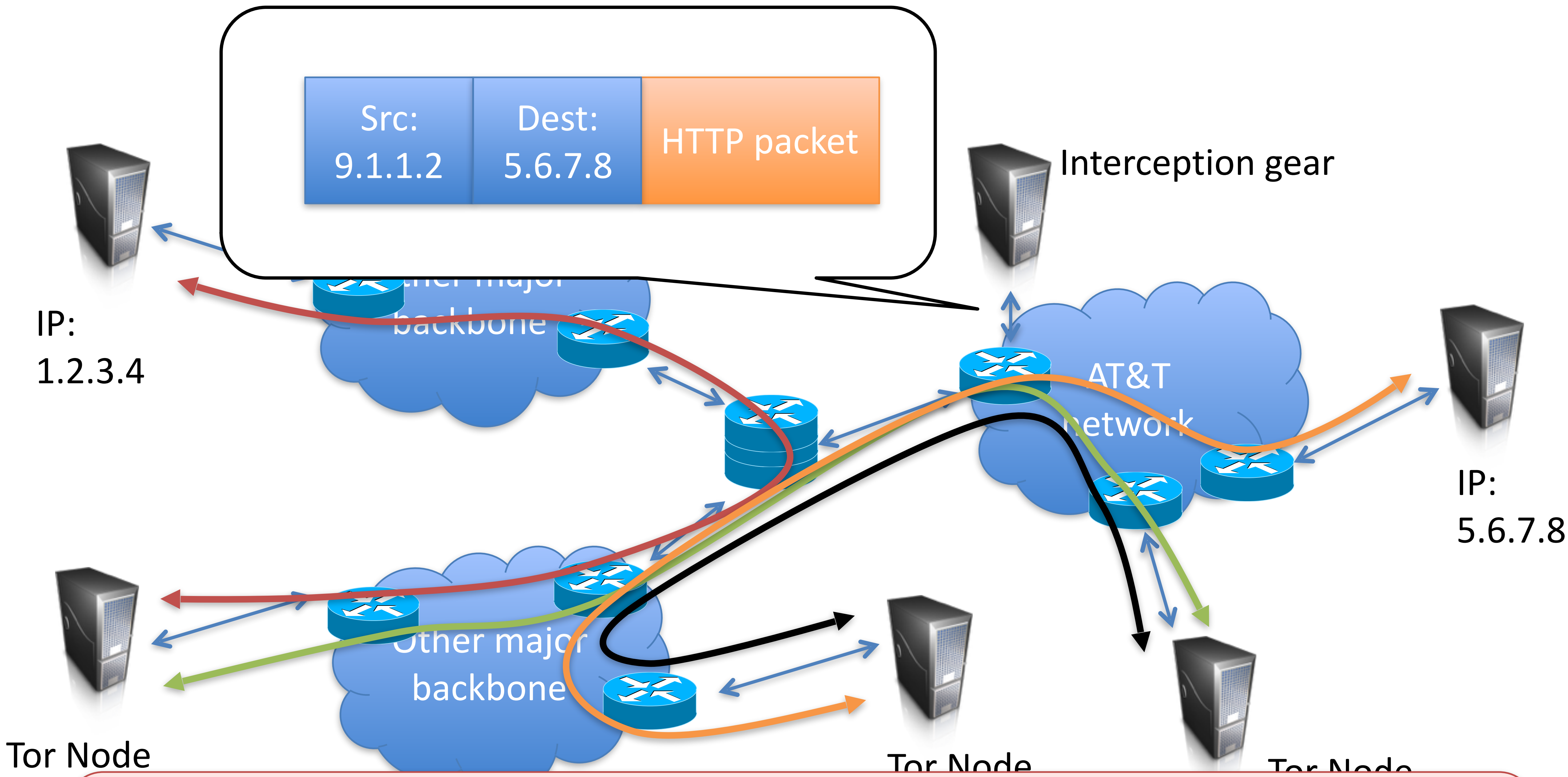
Client -> **7.8.9.1 -> 8.9.1.1 -> 9.1.1.2** -> Destination   Called a *circuit*

Onion routing: the basic idea



Tor implements more complex version of this basic idea

What does adversary see?



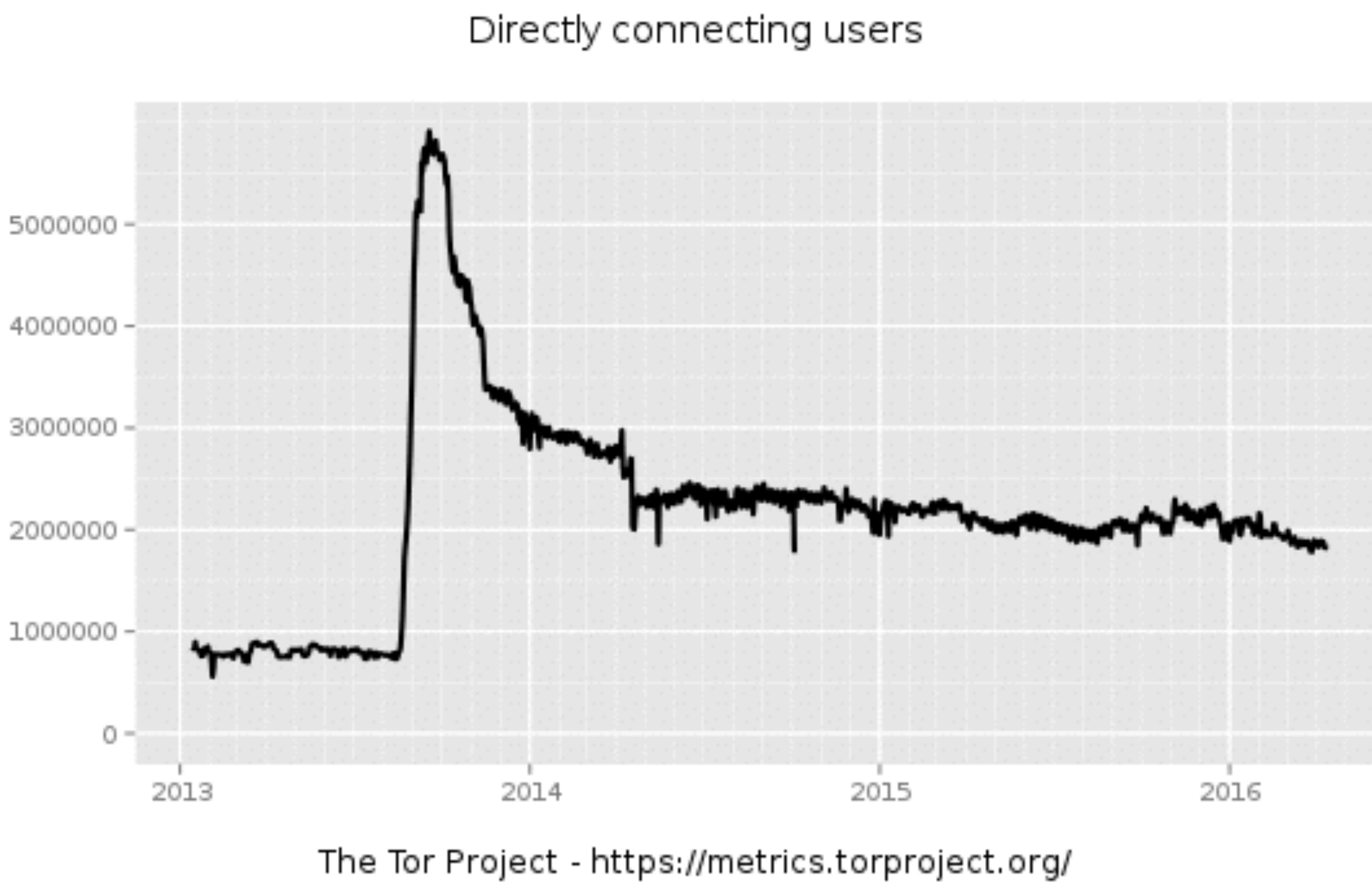
Tor obfuscates who talked to whom, need end-to-end encryption (e.g., HTTPS) to protect payload

FBI agents tracked Harvard bomb threats despite Tor

By Russell Brandom on December 18, 2013 12:55 pm Email @russellbrandom

- Dec 2016: Eldo Kim, Harvard sophomore, sent bomb threats using Guerilla Mail (anonymous email service)
- Used ToR to connect to Guerilla Mail (from his dorm room)
- Caught within 2 days

- How did he get caught?
  - Guerilla Mail indicated user connected via ToR node
  - FBI compared timestamp on email to Harvard network logs,
  - He was the only one using ToR at that time, confessed when confronted



[As of: April 13, 2016]

Other anonymization systems

- Single-hop proxy services

- JonDonym, anonymous remailers (MixMaster, MixMinion), many more…

Thursday, April 26, 2012

FBI seizes server used to anonymize e-mail

Jeffrey Brown 1 comment

recap

* Internet-wide scanning, zmap
* Massive surveillance, packet inspection
* Anonymous browsing, TOR