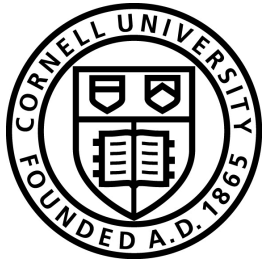
Internet censorship

Tom Ristenpart
CS 6431



Some basic primitives

- Symmetric cryptography (shared key K)
  - encryption & decryption using K
  - message authentication using K
  - pseudorandom functions (PRF)
- Public-key cryptography (public key pk, secret key sk)
  - encrypt with pk and decrypt with sk
  - digitally sign using sk and verify with pk
- Hash functions (no keys)
  - used to “compress” messages in a secure way

TLS handshake for RSA transport

Client

Server

Pick random Nc

ClientHello, MaxVer, Nc, Ciphers/CompMethods →

← ServerHello, Ver, Ns, SessionID, Cipher/CompMethod

Pick random Ns

Check CERT using CA public verification key

← CERT = (pk of bank, signature over it)

Pick random PMS
C <- E(pk,PMS)

C →

PMS <- D(sk,C)

ChangeCipherSpec,
{ Finished, PRF(MS, “Client finished” || H(transcript)) } →

ChangeCipherSpec,
{ Finished, PRF(MS, “Server finished” || H(transcript’)) } ←

Bracket notation means contents encrypted

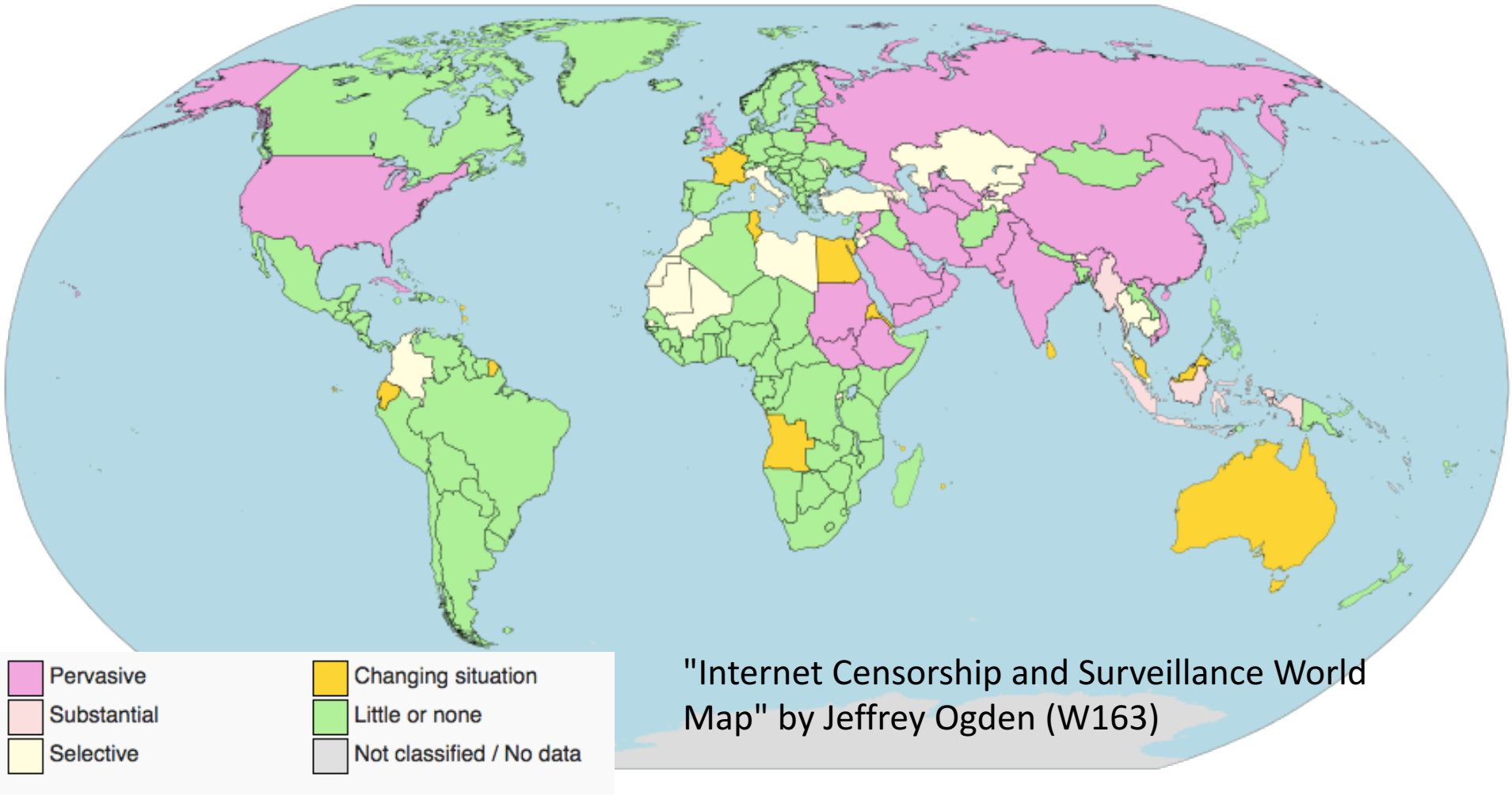
MS <- PRF(PMS, “master secret” || Nc || Ns)

Primitive	Use cases / examples	Security goals	Good schemes	Bad schemes
Block ciphers	Building block for symmetric encryption	Indistinguishable from random permutation	AES, 3DES	DES, Skipjack
Pseudorandom Functions (PRFs) / Message authentication codes (MACs)	Authenticating data with shared secret key, key derivation	Indistinguishable from random function	HMAC w/ good hash function OMAC, CMAC, PMAC	CBC-MAC without prefix-free encoding
Symmetric encryption	Main mechanism for encrypting data; TLS record layer, encrypting data at rest	Message confidentiality and associated data + ciphertext authenticity	Encrypt-then-MAC GCM OCB	Encryption only modes: CTR mode, CBC mode, ECB mode, RC4
Hash functions	Key derivation, PW hashing, digital signatures, HMAC	"Behave" like a public random function (implies coll resist, one-wayness, etc.)	SHA-256 SHA-3	MD4, MD5, SHA-1
Password-based key derivation	Password hashing, PW-based encryption	No shortcut attacks	PBKDF2, scrypt, bcrypt, argon2	Plain hash function

Primitive	Use cases / examples	Security goals	Good schemes	Bad schemes
RSA PKE	Encrypt symmetric key	No partial info on messages leaked to active attacker	RSA-OAEP w/ 2048 bit moduli	RSA-PKCS#1 v1.5, “raw” RSA, < 2048 bit N
ECC PKE	Encrypt symmetric key	No partial info on messages leaked to active attacker		ElGamal by itself
Hybrid encryption	Encrypt data efficiently using recipient public key	No partial information on messages leaked; attacker can’t maul ciphertext	ECIES (ElGamal KEM), RSA-OAEP w/ one-time Encrypt-then-MAC scheme	Raw RSA kem, bad sym encryption (e.g., CBC mode)
Digital signatures	Authenticated key exchange, code signing	Unforgeability under chosen message attacks	ECDSA (careful of randomness), RSA PSS	RSA-PKCS#1 v1.5
Diffie-Hellman key exchange	Establishing secure channel	Attacker can’t recover derived session key	ECC DH, Finite field DH	<< 256 bit ECC groups, << 2048 bit FF groups

Current estimates of Internet censorship

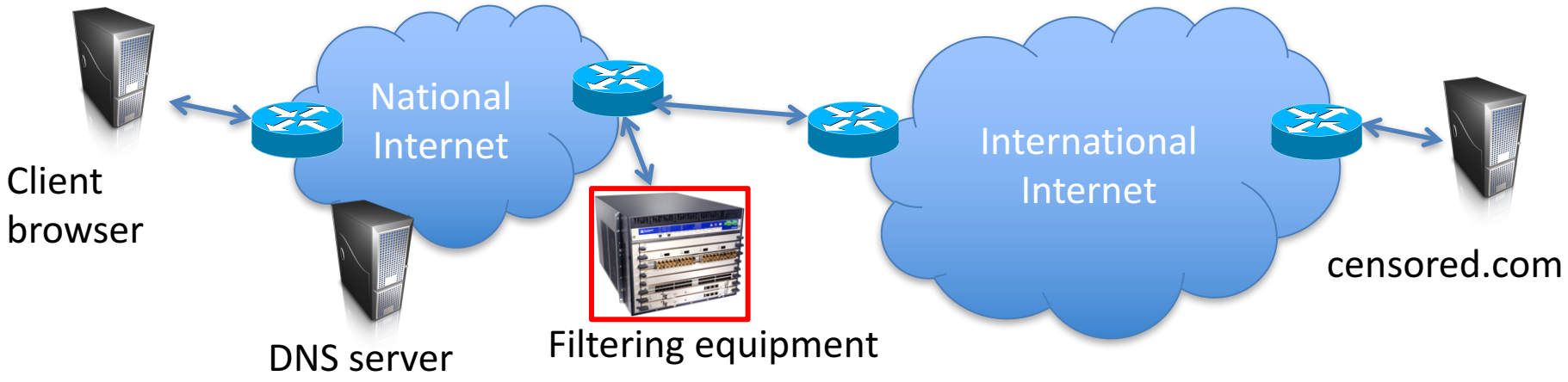
OpenNet Initiative (ONI), Reporters Without Borders



"Internet Censorship and Surveillance World Map" by Jeffrey Ogden (W163)

Magenta-colored countries are countries with pervasive censorship and surveillance of the Internet

How would you censor web requests?



- IP filtering
- DNS filtering / redirection
- URL filtering
- Packet filtering (search keywords in TCP packets)
- Protocol filtering (detect Tor protocol)

Golden Shield Project

"If you open the window for fresh air, you have to expect some flies to blow in" – Deng Xiaoping in 1980s

Great Firewall of China:

- IP filtering
- DNS filtering / redirection
- URL filtering
- Packet filtering (search keywords in TCP packets)
- Protocol filtering
- Active probing of suspect destination IP addresses

Islamic Republic of Iran

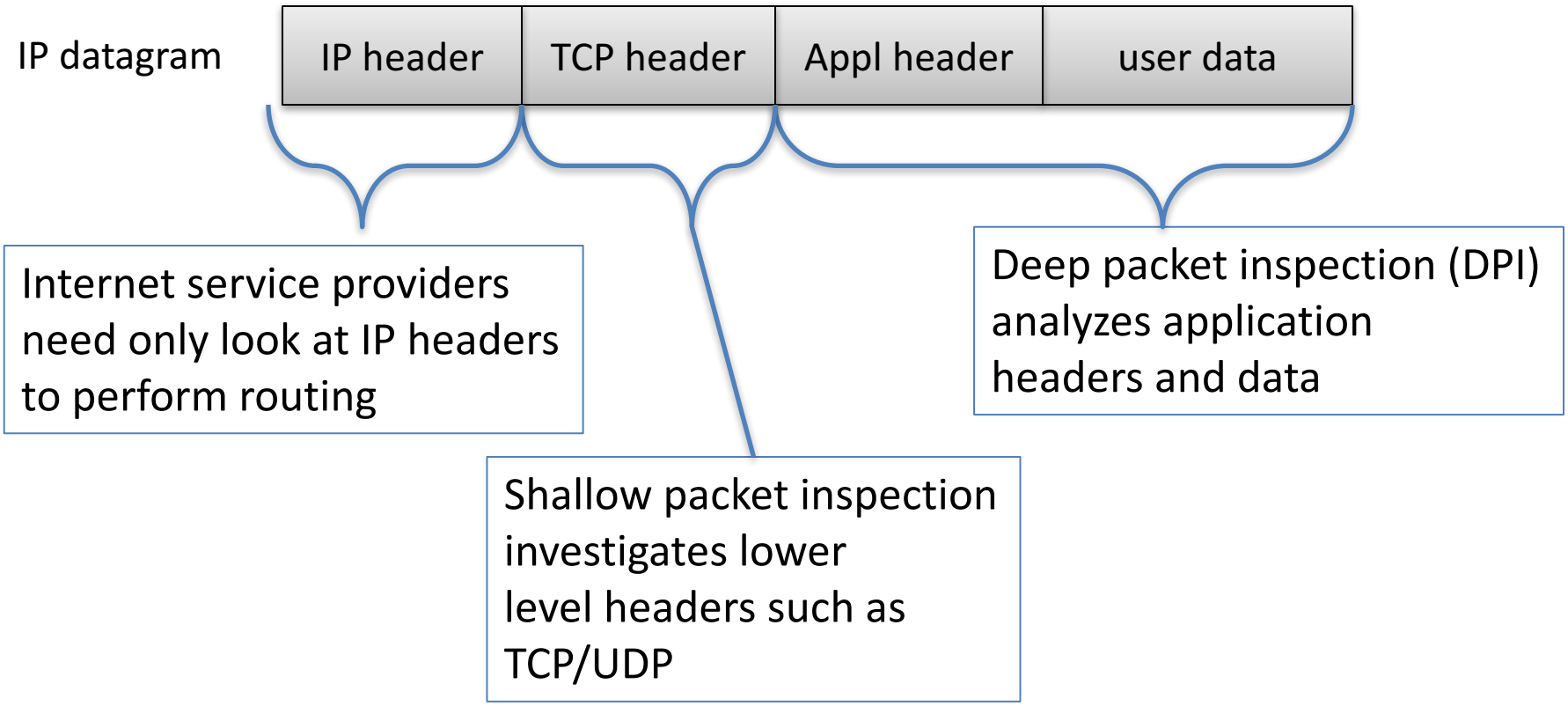
- Every ISP must run “content-control software”
  - SmartFilter (up until 2009) made by USA company
  - Nokia Siemens deep-packet inspection (DPI) systems
- According to wikipedia: 50% of top 500 most popular websites blocked in Iran
- Occassional widespread filtering of Tor, TLS, other encrypted protcols

Censorship as two-step process

1. *Sensitive content identification*
   - DNS and IP blacklists
   - Keyword blacklists with DPI
   - Protocol identification (e.g., TLS)
   - Tool identification (e.g., Tor)

2. *Censoring action*
   - DNS poisoning
   - HTTP man-in-the-middle
   - TCP resets
   - Dropping packets

Types of packet inspection



DPI 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 allegedly used by NSA in San Francisco AT&T office

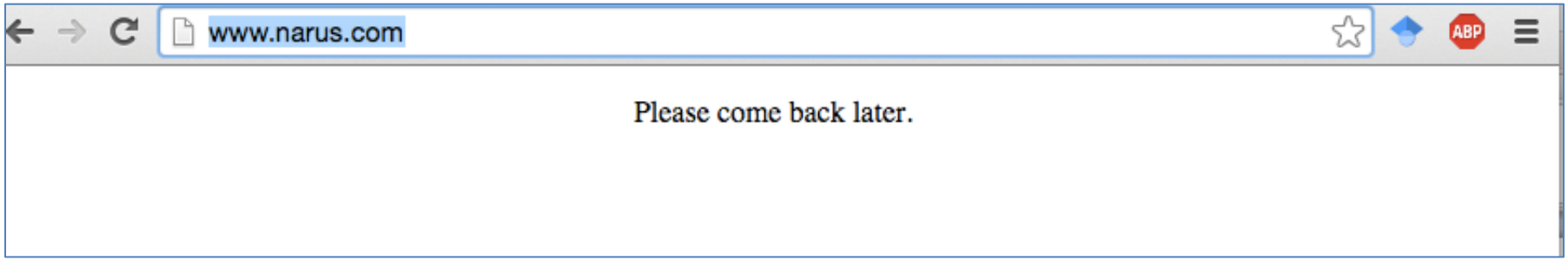
NarusInsight™ Selected To Save Pakistan's Telecommunications Networks Millions Of Dollars Per Year

NarusInsight™ Selected to Save Pakistan's Telecommunications Networks Millions of Dollars Per Year

Narus System Chosen to Detect Rogue VoIP Traffic

MOUNTAIN VIEW, Calif.—September 21, 2007—Narus, Inc., the leader in carrier-class security for the world's largest IP networks, today announced that the company has teamed up with Inbox Business Technologies Pvt. Ltd, a leading total IT solution provider in Pakistan, to keep Pakistan's telecommunication networks clear of illegal, rogue and malicious IP traffic. NarusInsight was chosen by the Pakistan Telecommunication Authority (PTA) (the government administration responsible for regulating the establishment, operation and maintenance of telecommunication systems, and the provision of telecom services) to detect rogue VoIP traffic flowing through the telecommunications network in Pakistan.

http://www.narus.com/index.php/news/279-narusinsight-selected-to-save-pakistans-telecommunications-networks-millions-of-dollars-per-year



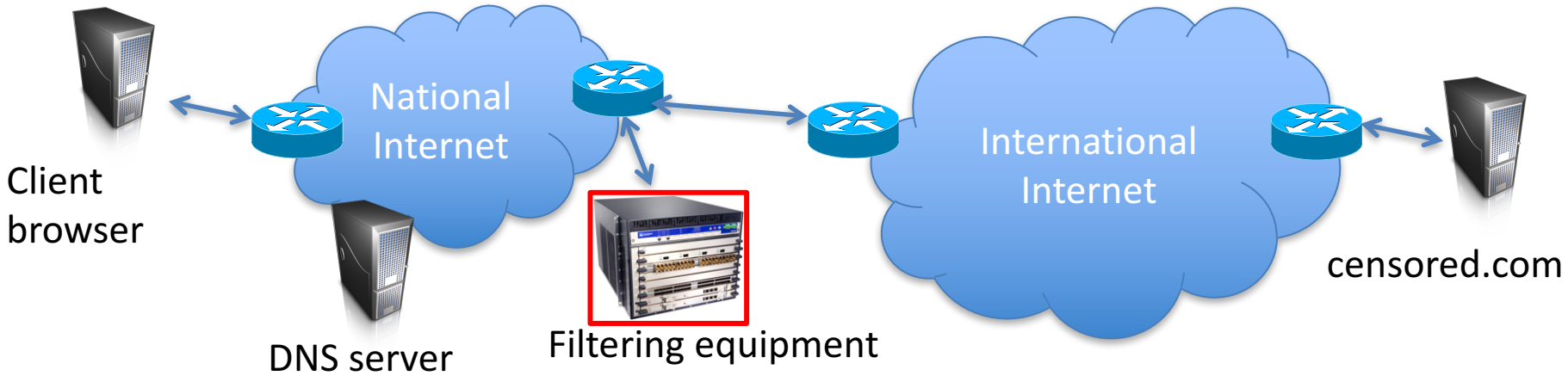
How do we know about censorship?

- Anecdotes from people within censored regions
- More formal surveys
- Network measurements:
  - Web sites aggregating info such as GreatFire
  - Herdict tool (browser plugin to manually report blockage)
  - Open Observatory of Network Interference (opt-in measurements of network connections)
  - Encore paper

Encore web-based measurements

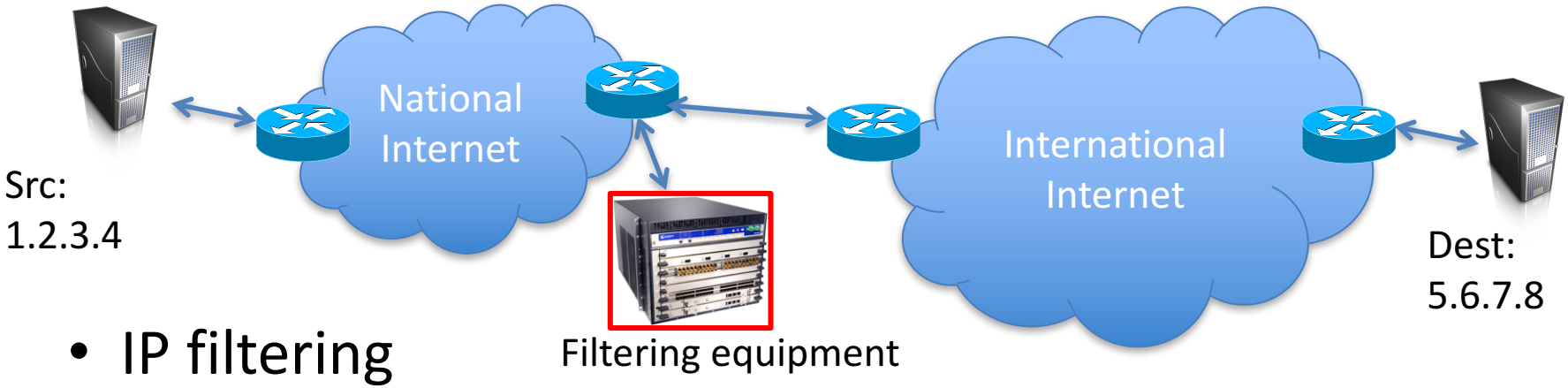
- Burnett-Feamster study
- Embed in other non-blocked web pages measurement functionality of suspect censorship targets
- Use cross-site embeddings and browser side-channels

Censorship circumvention



- IP filtering
- DNS filtering / redirection
- URL filtering
- Packet filtering (search keywords in TCP packets)
- Protocol filtering (detect Tor protocol)

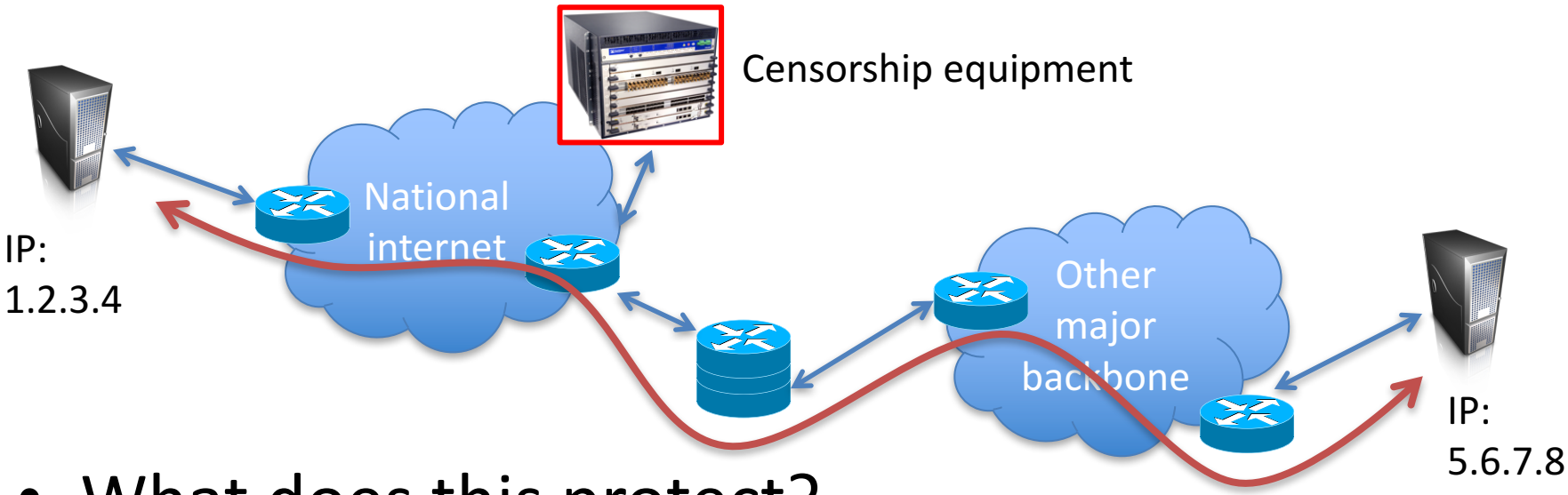
How would you *avoid* censorship?



- IP filtering
  - Proxies
- DNS filtering / redirection
  - DNS proxy
- URL filtering or Packet filtering
  - Encryption / Tunneling / obfuscation
- Protocol filtering
  - Obfuscation techniques

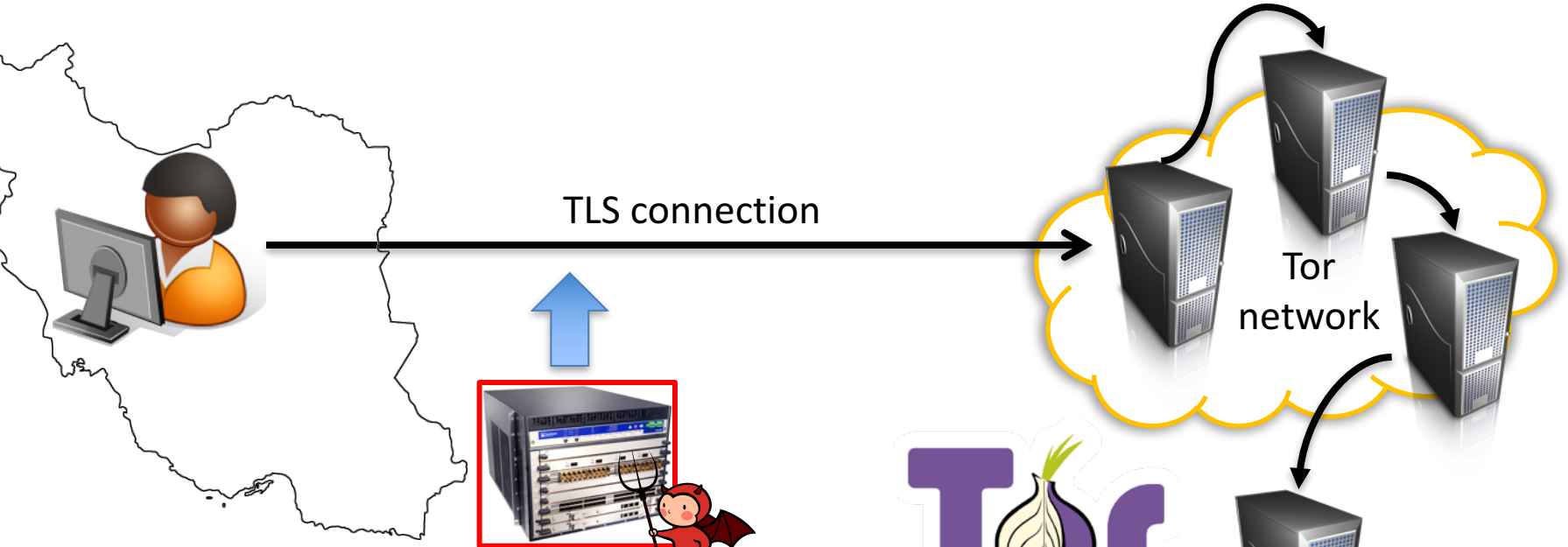
Censorship circumvention

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



- What does this protect?
- What does it leak?

Censorship circumvention



TorProject.org

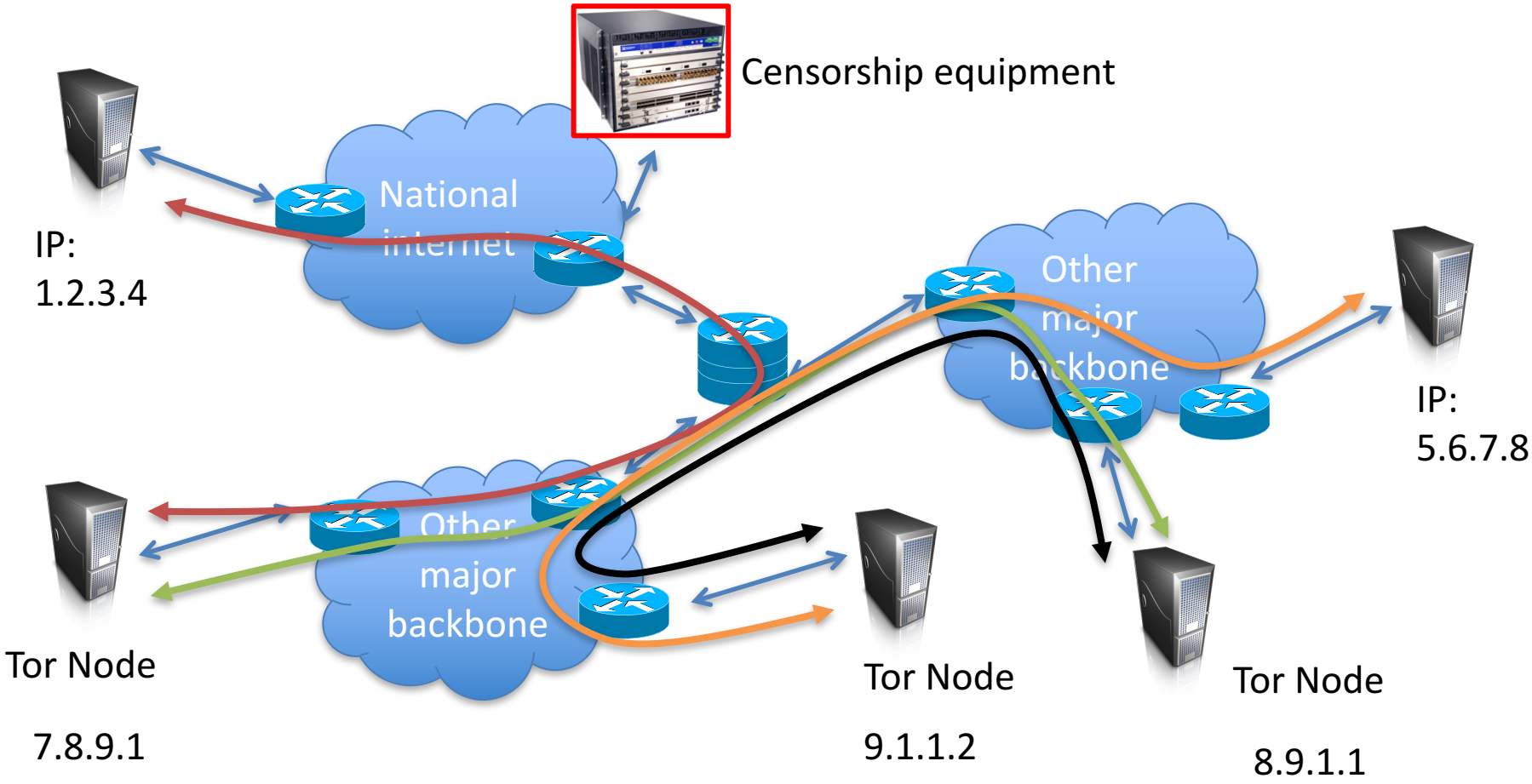
psiphon

uProxy.org

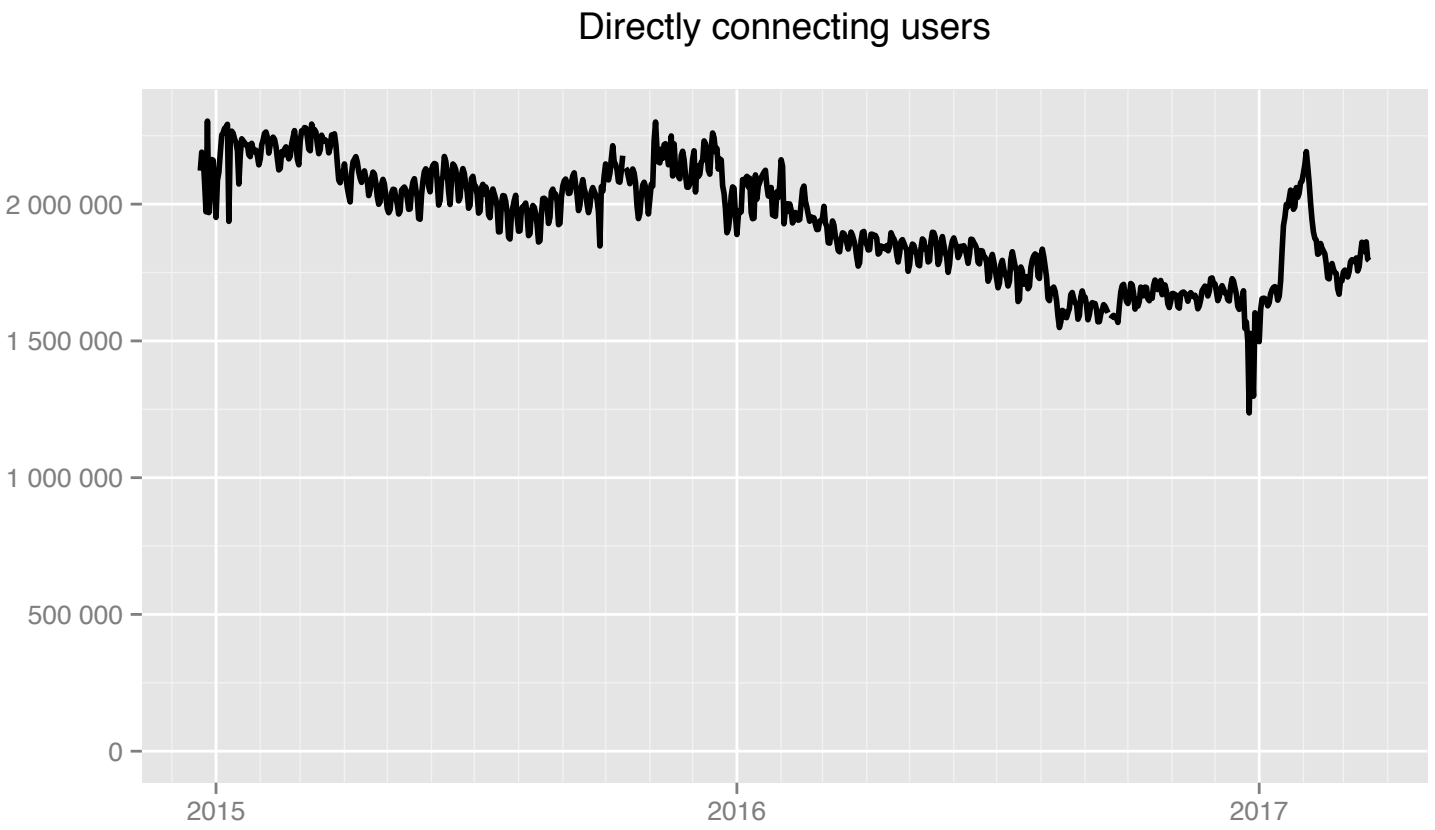
LANTERN

Academic: Telex, Cirripede, Flash proxies, Infranet, Collage, CloudTransport, …

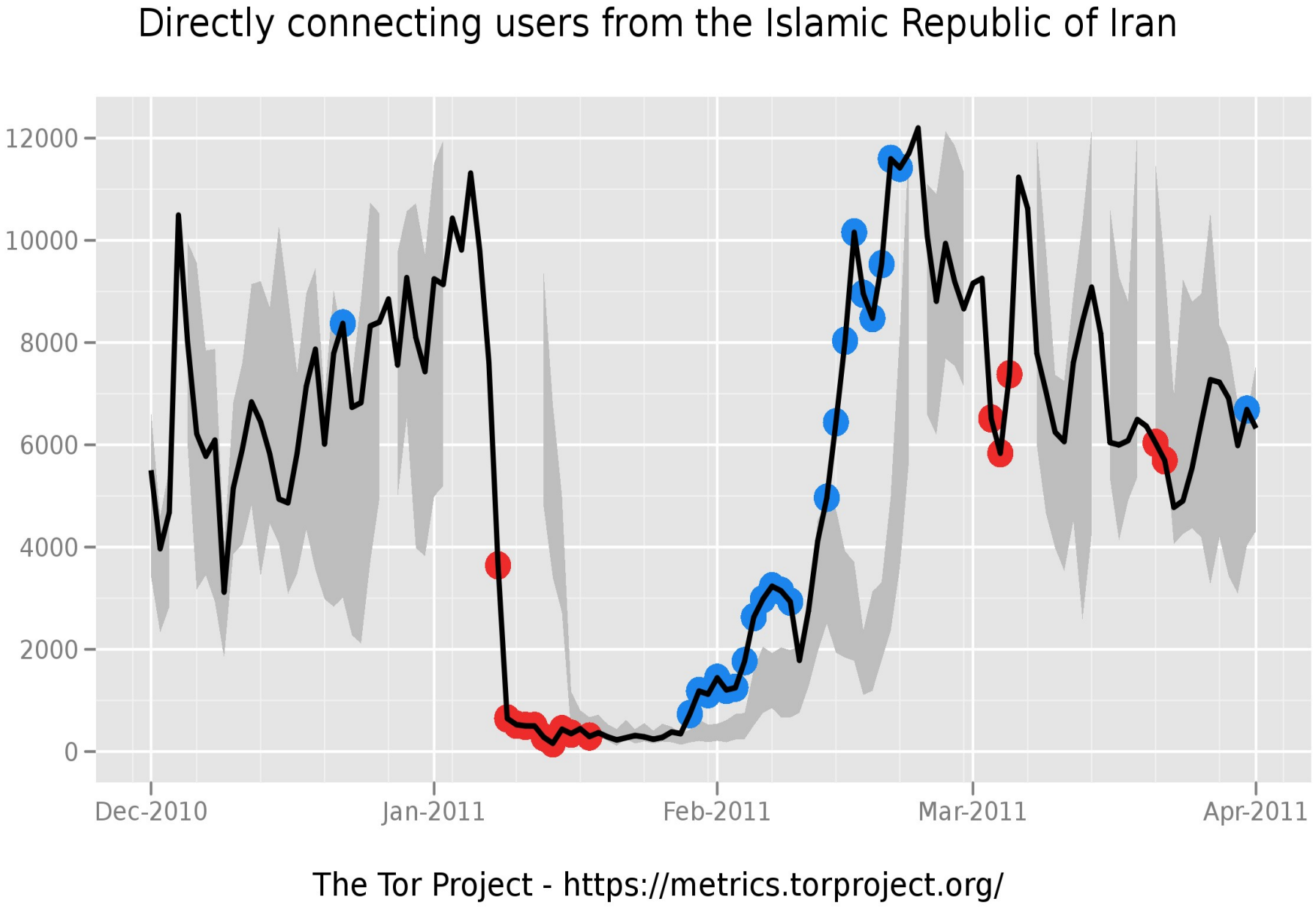
Censorship circumvention



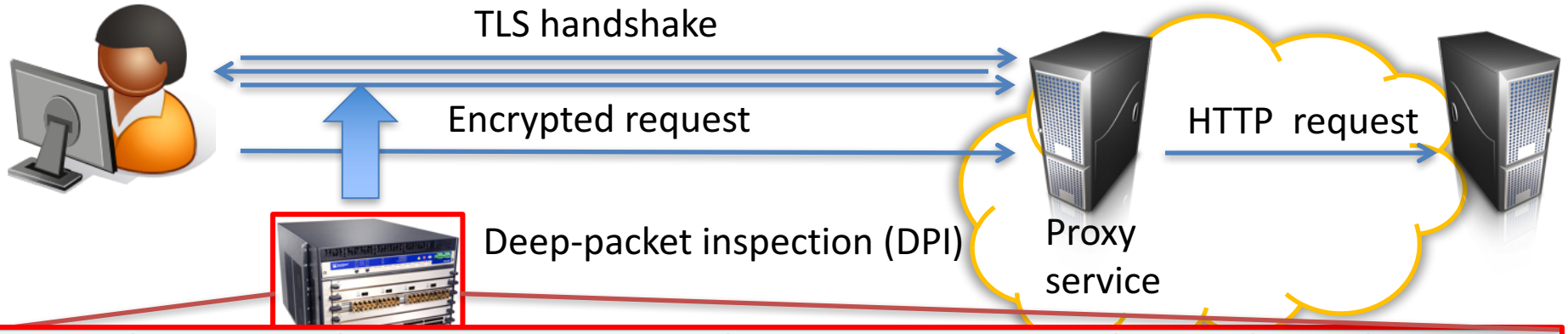
Censorship circumvention



The Tor Project – https://metrics.torproject.org/


Directly connecting users from the Islamic Republic of Iran
12000
10000
8000
6000
4000
2000
0
Dec-2010
Jan-2011
Feb-2011
Mar-2011
Apr-2011
The Tor Project - https://metrics.torproject.org/

Protocol identification

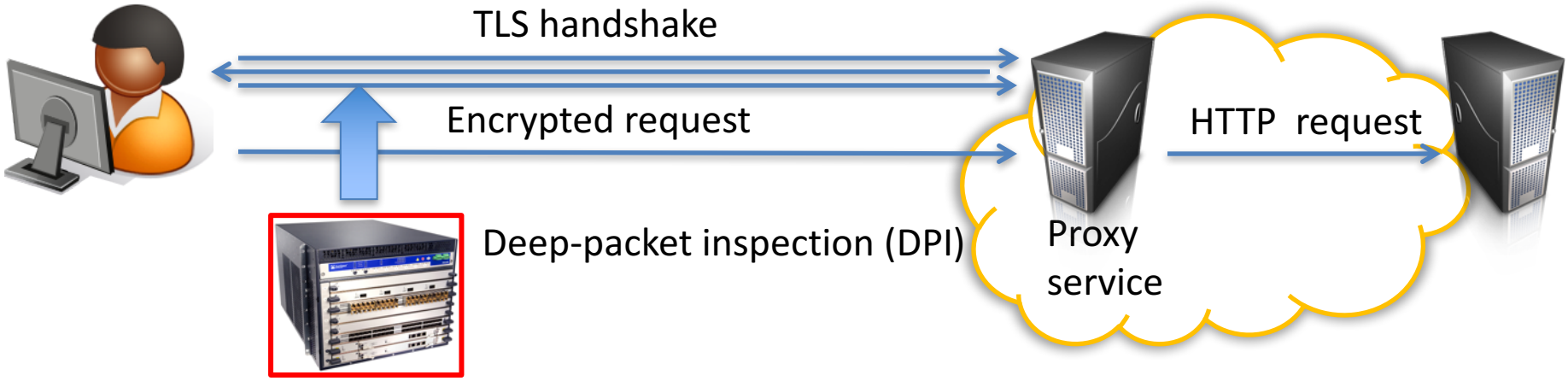


No.	Time	Source	Destination	Protoc	Length	Info
359	2.138821000	128.105.35.160	173.194.46.114	TLSv1.2	583	Client Hello
362	2.140902000	128.105.35.160	173.194.46.122	TLSv1.2	290	Client Hello
369	2.154594000	128.105.35.160	173.194.121.33	TLSv1.2	285	Client Hello
371	2.155001000	128.105.35.160	173.194.121.42	TLSv1.2	291	Client Hello

```
▽ Secure Sockets Layer
  ▽ TLSv1.2 Record Layer: Handshake Protocol: Client Hello
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 512
    ▷ Handshake Protocol: Client Hello
```

```
0040  63 8f 16 03 01 02 00 01  00 01 fc 03 03 1c 96 bf   c....... ........
0050  1a c0 ea b3 bc eb 04 45  d7 89 12 4d a1 6c 32 30   .......E ...M.l20
0060  6a 29 26 38 5e 65 07 a7  0a 72 ed e8 11 20 98 5f   j)&8^e.. .r... ._
0070  19 19 04 6f 36 2d 49 a1  00 a2 89 a9 4d 30 dd cc   ...o6-I. ....M0..
0080  e0 c8 3a 95 ab ed 76 29  ff 8b 0e e9 db 11 00 28   ..:...v) .......(
0090  c0 2b c0 2f 00 9e cc 14  cc 13 c0 0a c0 09 c0 13   .+./.... ........
```

Protocol identification



Scan packet contents for values indicative of protocol

Enterprise-grade DPI use regular expressions to identify protocols
Example:

$$R_{TLS} = /^(\x16\x03[\x00\x01\x02]..\x02...[\x00\x01\x02]|...? .*/$$

Slightly more complex regexes fingerprint various Tor versions

Some censors use multi-stage detection methods

TLS Handshake

Client

Server

Pick random Nc

ClientHello, MaxVer, Nc, Ciphers/CompMethods

ServerHello, Ver, Ns, SessionID, Cipher/CompMethod

Pick random Ns

Check CERT using CA public verification key

CERT = (pk of server, signature over it)

C

Pick random PMS
C <- E(pk,PMS)

PMS <- D(sk,C)

ChangeCipherSpec,
{ Finished, PRF(MS, “Client finished” || H(transcript)) }

ChangeCipherSpec,
{ Finished, PRF(MS, “Server finished” || H(transcript’)) }

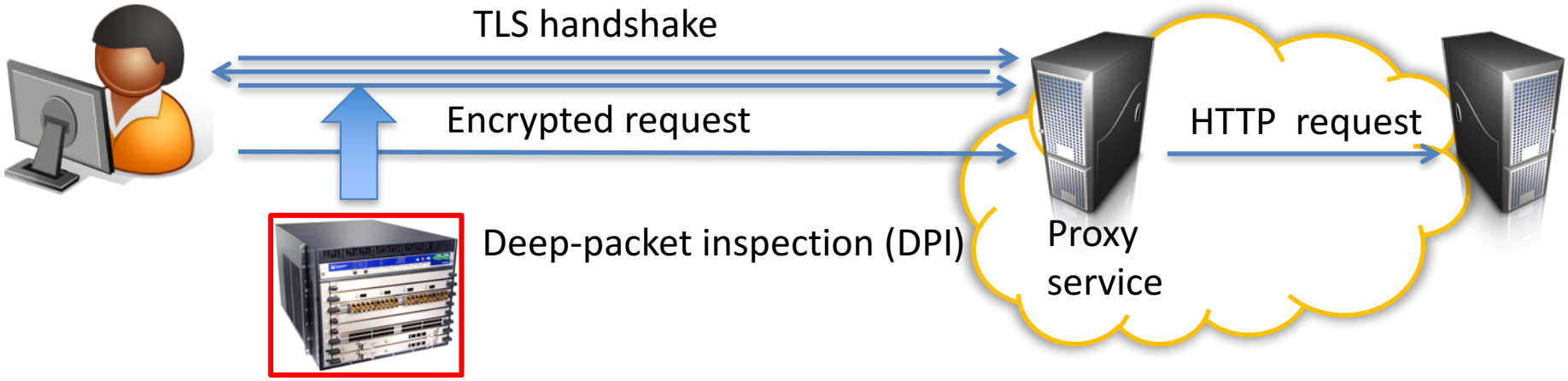
Bracket notation means contents encrypted

MS <- PRF(PMS, “master secret” || Nc || Ns)

Iran DPI blocking of Tor

- Tor point-to-point connections use TLS
- Use DPI to filter Tor connections:
  - Tor certificates have short expiration date
  - Most websites have long expiration date
  - Shut down those connections with short expiration dates
  - https://blog.torproject.org/blog/update-internet-censorship-iran
- Tor fixed via longer expiration dates
- Later in 2012: blocking/degrading all TLS connections

Protocol identification



More generally:

Censors want automated mechanisms to identify circumvention protocols. They desire:

Low false positive rate (FP leads to over-blocking, collateral damage)

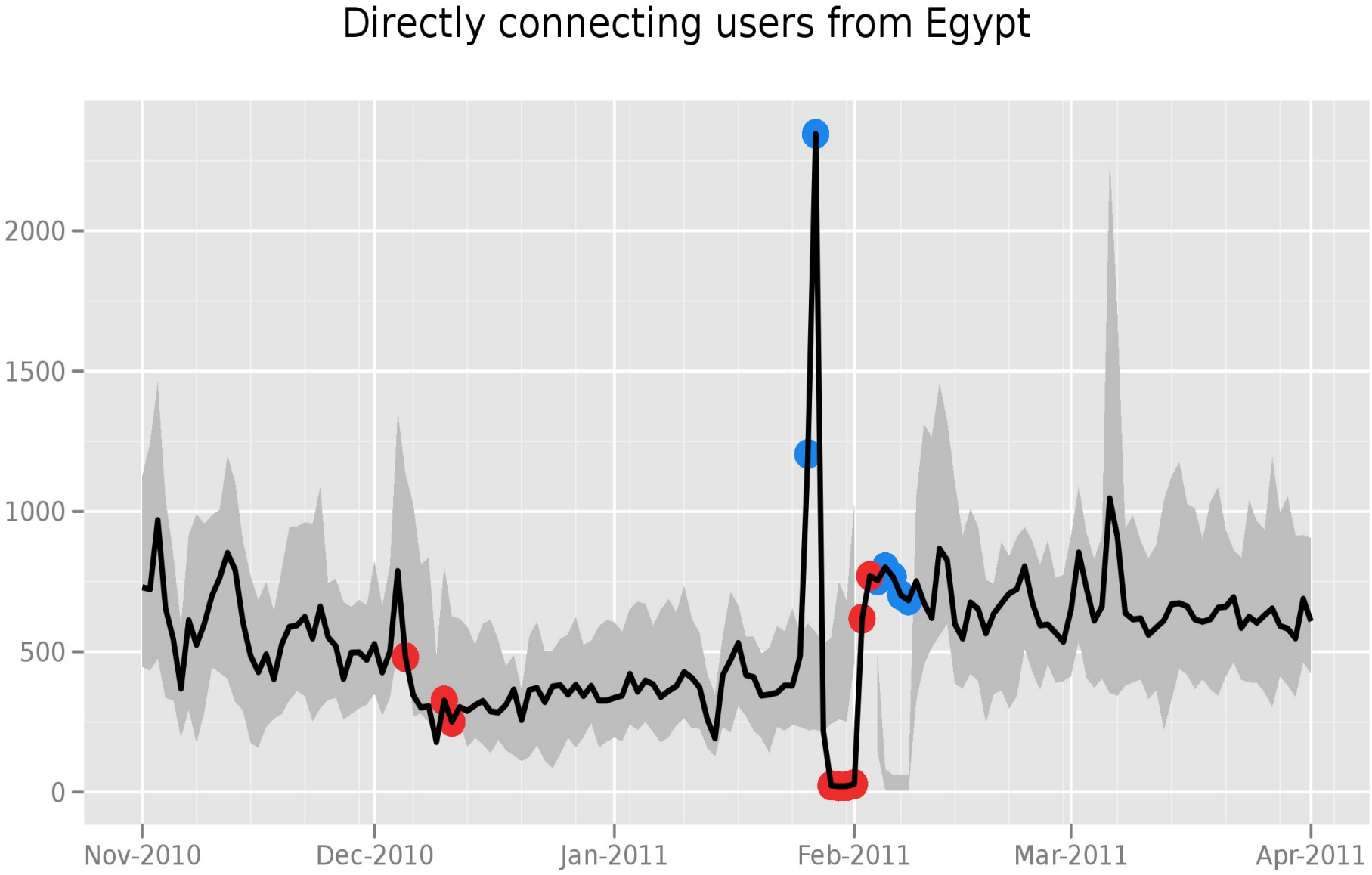
High true positive rate (FN misses tool use)

Protocol identification events

Country	Dates	Event
China	2007-	Tor fingerprinting, Tor active probing
Iran	Sep 2011	Tor fingerprinting (certificate expiration)
Iran	Feb 2012	Keyword flagging, TLS fingerprinting
Ethiopia	2012	Skype detection, Tor fingerprinting
Kazakhstan	2012, 2016	Tor fingerprinting

A lot more countries use DPI to detect keywords,
block non-circumvention network traffic

Some countries seem ok with (temporary) high false positive rates

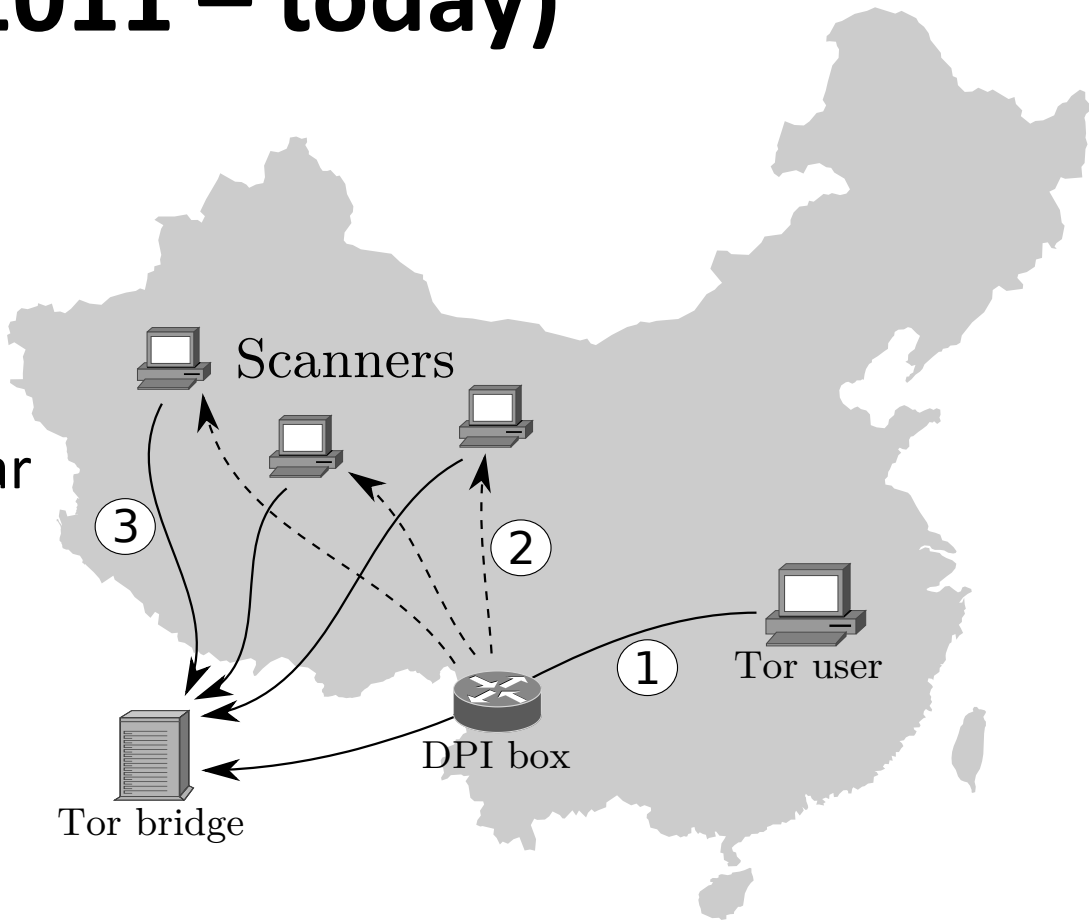

Directly connecting users from Egypt
2000
1500
1000
500
0
Nov-2010
Dec-2010
Jan-2011
Feb-2011
Mar-2011
Apr-2011
The Tor Project - https://metrics.torproject.org/

Great Firewall targeting of Tor (circa 2011 – today)

Admin noticed weird connections from China 2011

TLS connections with particular ciphersuites flagged for active probing

If remote server speaks Tor then add its IP address to blacklist



From [Winter, Lindskog 2012]

https://gist.github.com/da3c7a9af01d74cd7de7

TLS Handshake

Client

Server

Pick random Nc

ClientHello, MaxVer, Nc, Ciphers/CompMethods

ServerHello, Ver, Ns, SessionID, Cipher/CompMethod

Pick random Ns

Check CERT using CA public verification key

CERT = (pk of server, signature over it)

Pick random PMS
C <- E(pk,PMS)

C

PMS <- D(sk,C)

ChangeCipherSpec,
{ Finished, PRF(MS, “Client finished” || H(transcript)) }

ChangeCipherSpec,
{ Finished, PRF(MS, “Server finished” || H(transcript’)) }

Bracket notation means contents encrypted

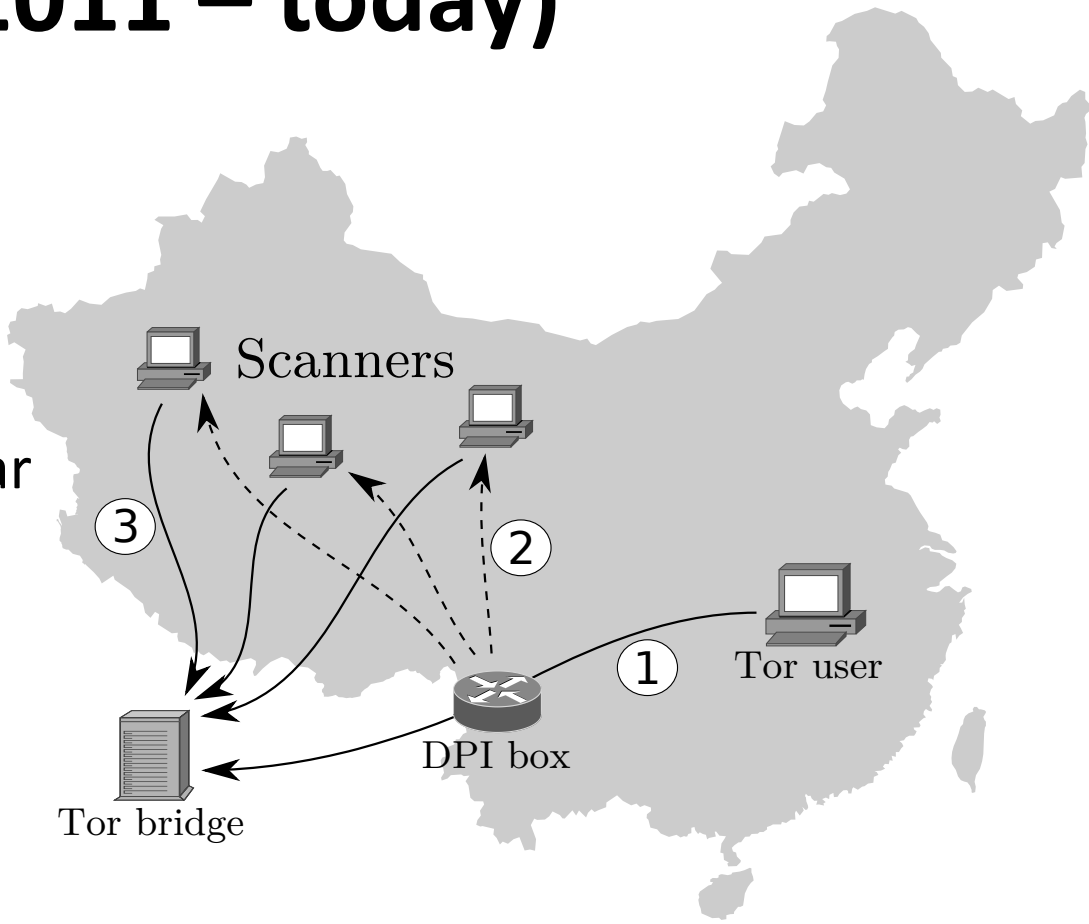
MS <- PRF(PMS, “master secret” || Nc || Ns)

Great Firewall targeting of Tor (circa 2011 – today)

Admin noticed weird connections from China 2011

TLS connections with particular ciphersuites flagged for active probing

If remote server speaks Tor then add its IP address to blacklist



From [Winter, Lindskog 2012]

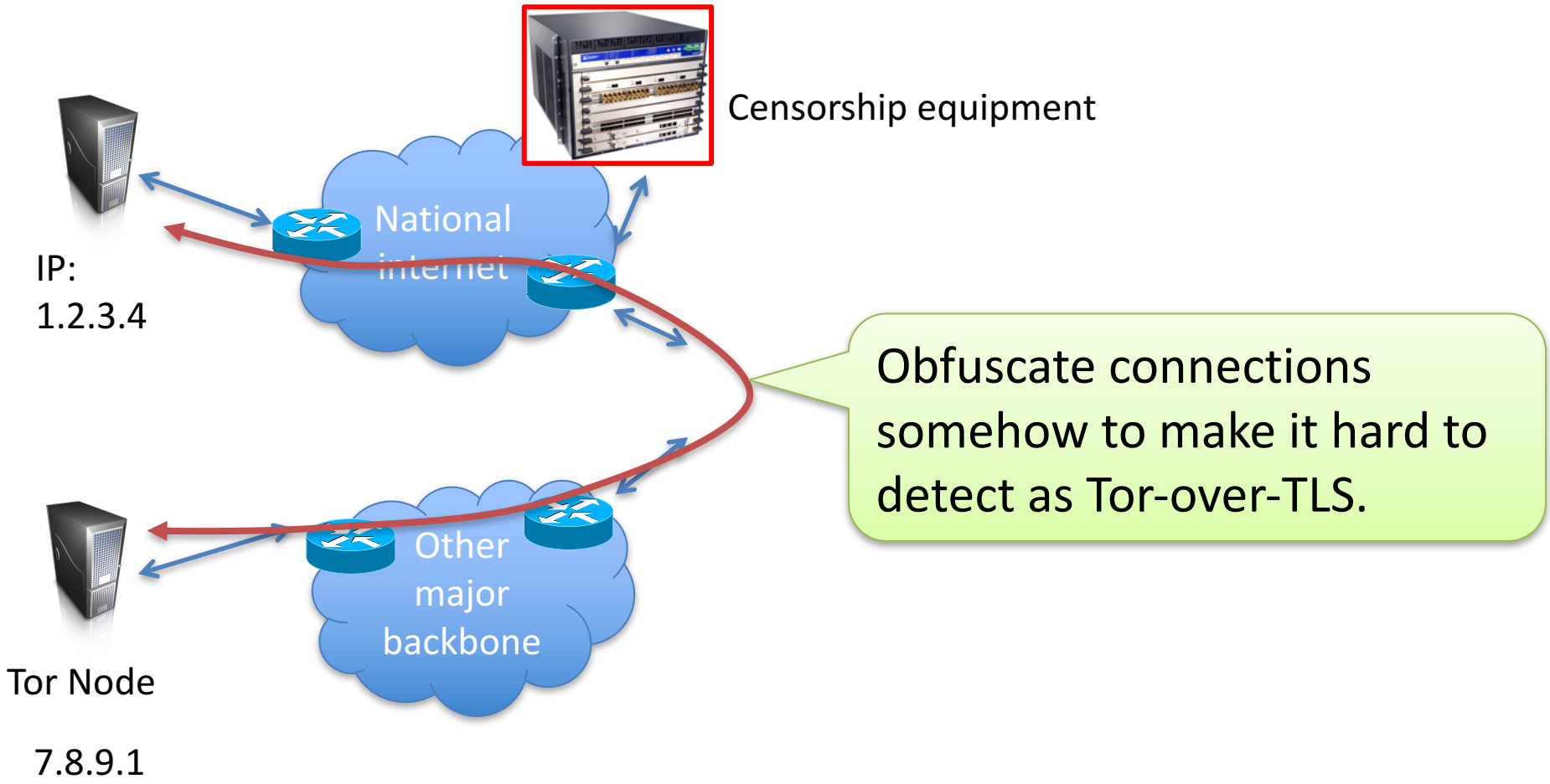
https://gist.github.com/da3c7a9af01d74cd7de7

Great Firewall targeting of Tor (circa 2011 – today)

Ensafi et al. IMC 2015 follow-up study

- Active measurements, log file analysis, etc.
- China is checking obfsproxy3 bridges
- Hijack IP addresses to perform active probing
- DPI is stateful but does not reconstruct TCP streams

How would you defeat DPI-based tool / protocol identification?



Network Protocol Obfuscation

Build cryptographic tools to prevent blocking.
Tool is ***unobservable*** if it forces one or both of:

High false positive rate (FP leads to over-blocking, collateral damage)

Low true positive rate (FN misses tool use)

Tor's pluggable transport initiative first to explore obfuscation approaches.
In response to censorship events in Iran, China

Approaches to Obfuscation

Randomizers

Transcripts should be indistinguishable from random bits

Stream cipher with fixed key, key exchange with specialized ECC

Examples: Dust, obfsproxy3, obfsproxy4

Protocol mimicry

Lightweight stego, look like HTTP or other unencrypted protocol

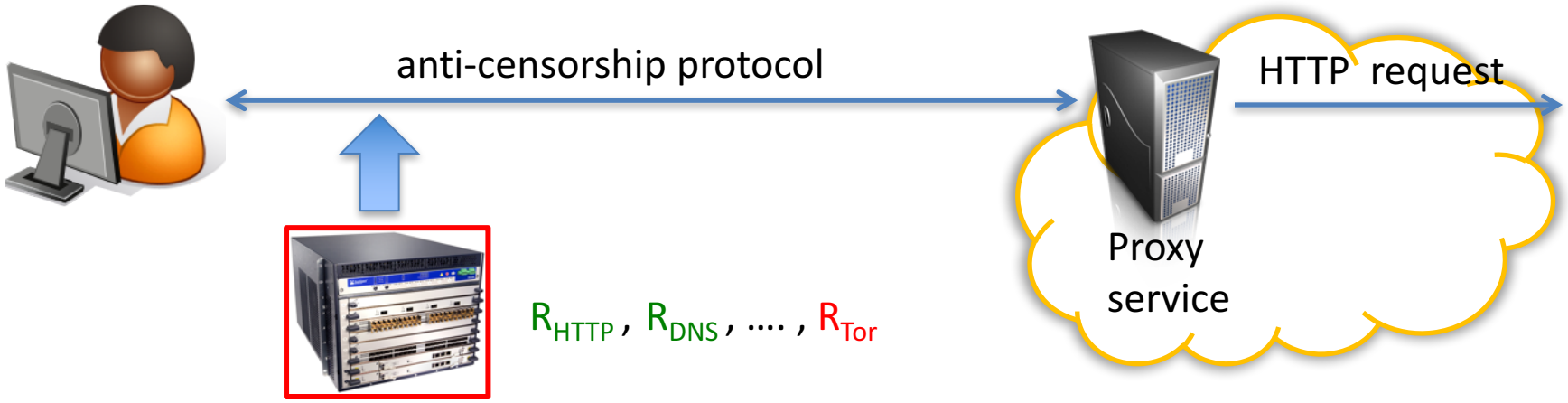
Examples: Stegotorus, SkypeMorph, format-transforming encryption

Tunneling

Tunnel over (unblocked) encrypted protocol implementation

Examples: meek (domain fronting)

Towards format-transforming encryption



Censors program restrictive DPI rules often using regexes

If packet in L(R_{HTTP}) then allow

If packet in L(R_{DNS}) then allow

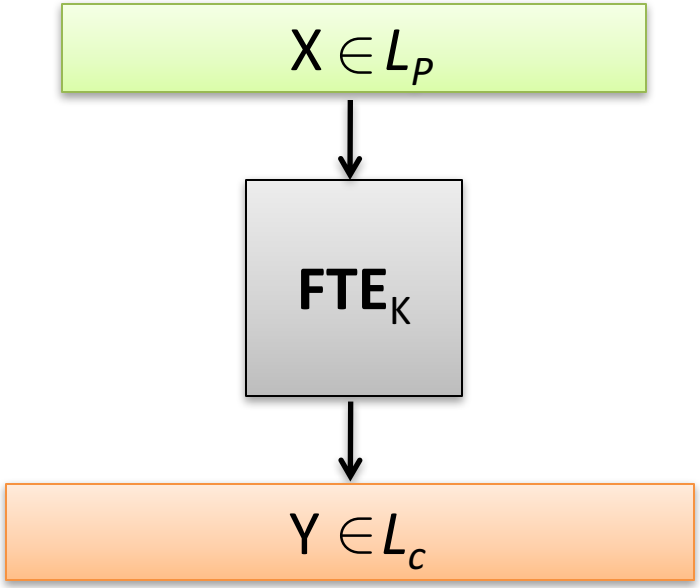
⋮

If packet in L(R_{Tor}) then disable connection

Goal:
trick DPI into *misclassifying* encrypted connections as HTTP, DNS, etc.

May or may not know what regexes being used by DPI

Key insight: DPI's protocol identification = regex-specified format checks on packets

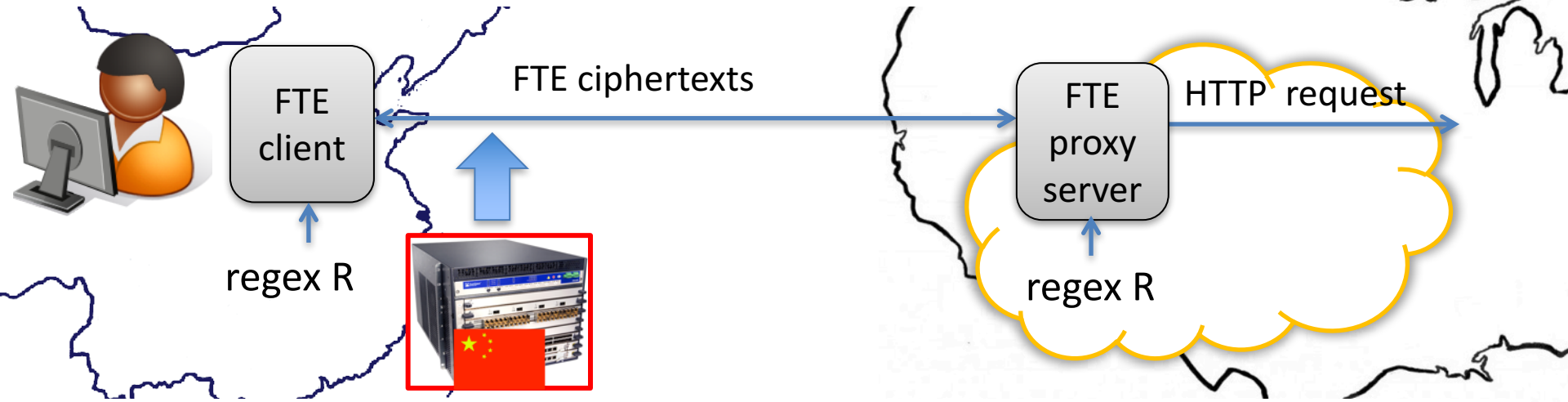


L_P is plaintext language
L_C is ciphertext language

Example in DPI setting:
$L_P = \{0,1\}^*$ $L_C = L(R_{HTTP})$

Format-transforming encryption

FTE proxy for censorship avoidance

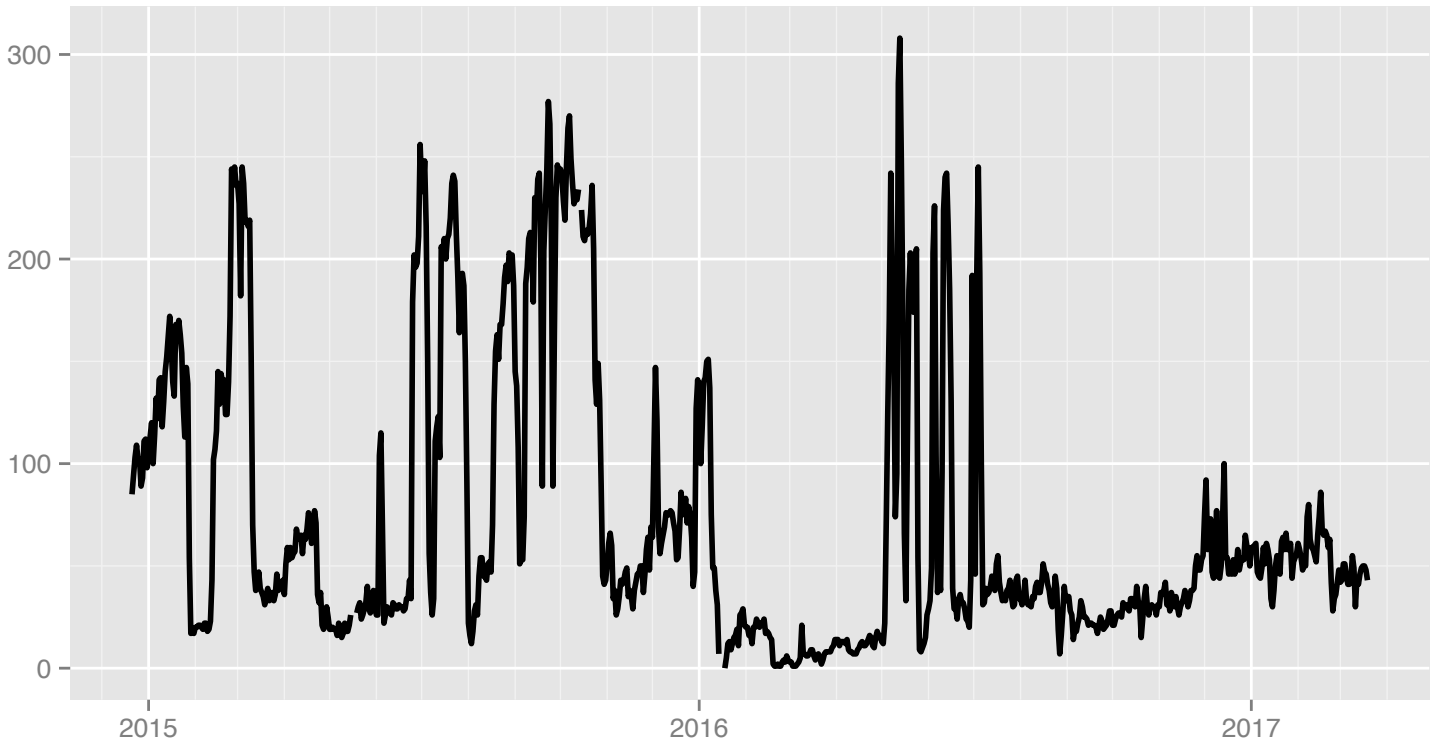


We designed a record layer protocol that wraps arbitrary TCP payloads with FTE. ***Minimalist*** approach to evade regex-based DPI

Tested against a set of 6 DPI systems with a number of target "cover" protocols (HTTP, SMB, SSH, …)

All DPI systems misclassified FTE-wrapped traffic as target protocol

2013: Tested Tor-over-FTE for several months with client in China. Unsurprisingly, not blocked



The Tor Project – https://metrics.torproject.org/

TorProject.org

uProxy.org

LANTERN

Follow-up project: Marionnette [Dyer, Coull, Shrimpton 2014]

Seeing through obfuscation?

In-depth analysis of existing and new censor strategies

[Wang, Dyer, Akella, R., Shrimpton - CCS `15]

Evaluate old attacks

Evaluated some previously suggested obfuscation detection strategies

Conclusion: Some work well, many work very poorly

Present new attacks

1. Entropy-based hypothesis tests for randomizers

   Variant adapted to FTE

2. Traffic analysis for tunneling (meek)

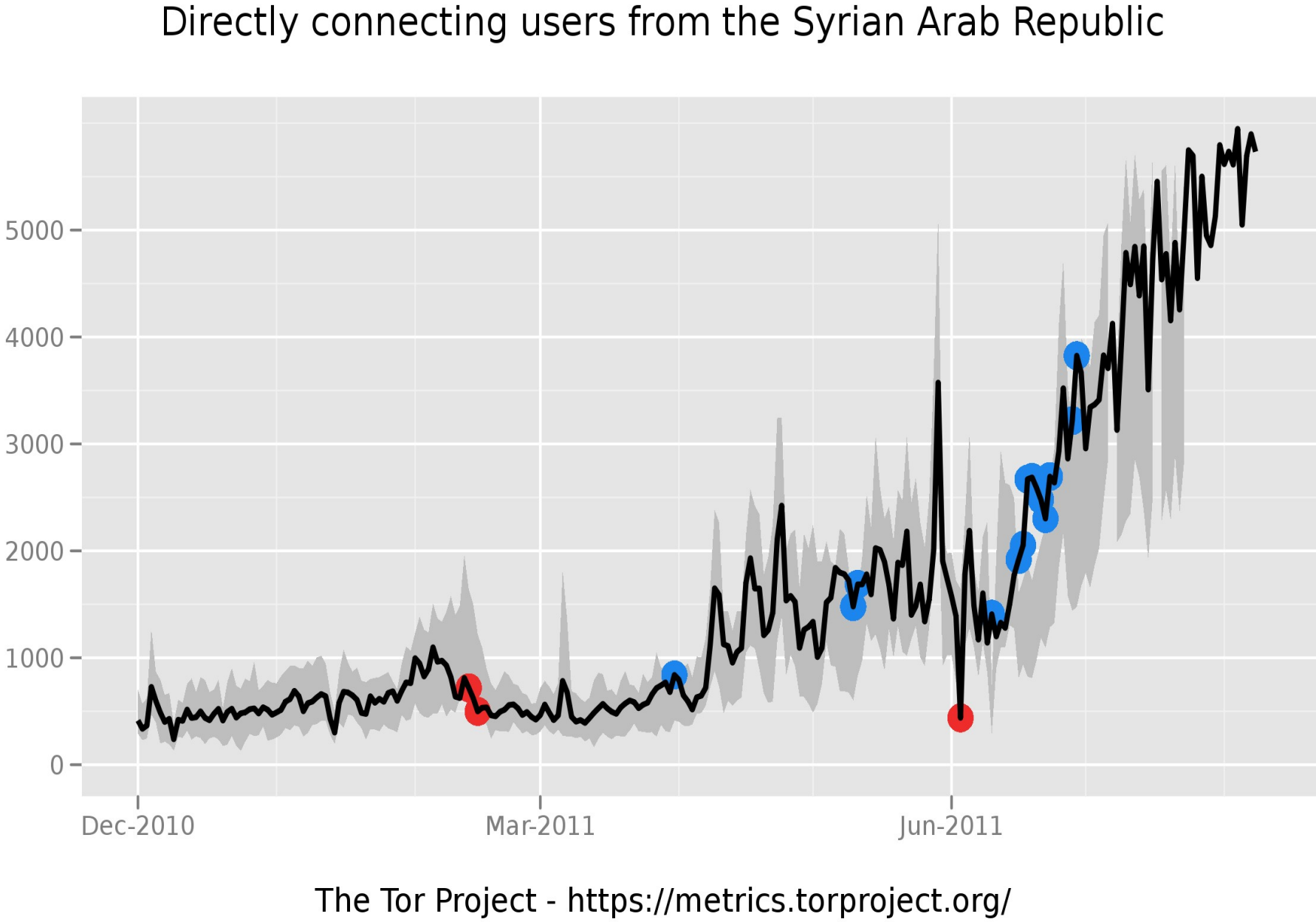
Obfuscator	TPR	FPR
Obfsproxy3	100%	0.2%
Obfsproxy4	100%	0.2%
FTE	100%	0.003%
meek-amazon	98%	0.02%
meek-google	98%	0.006%

Censorship summary

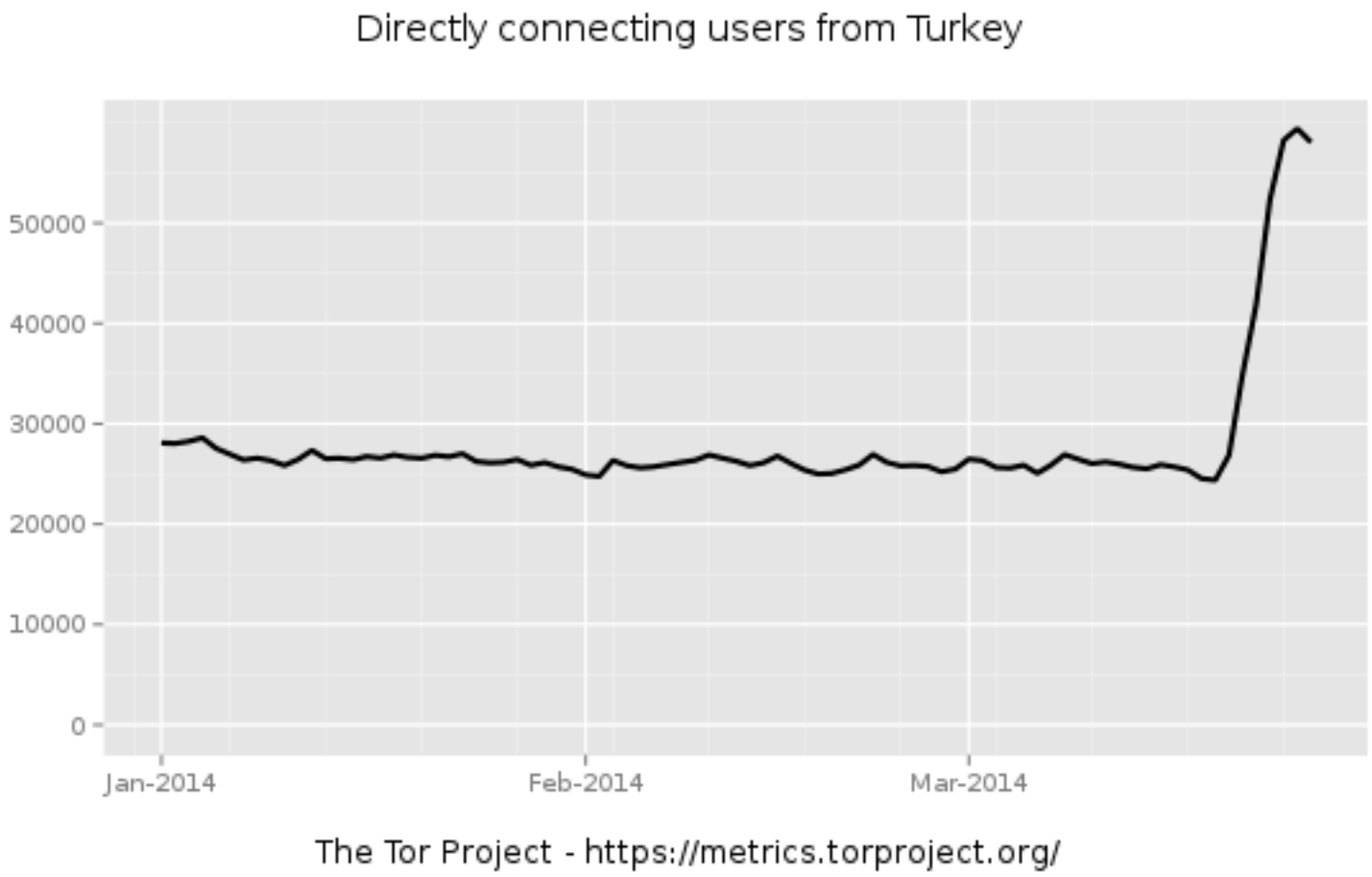
- Nation-state censorship apparatuses are technologically sophisticated
- Arms race between censors and circumvention community (activists, academics, USG)
- Lots of open research questions
  - Better steganography in face of DPI
  - Proxy distribution without enumeration
  - …

From BlueCoat:

- Our awareness of the presence of these ProxySG appliances in Syria came from reviewing online posts made by so-called “hacktivists” that contained logs of internet usage which appear to be generated by ProxySG appliances. We believe that these logs were obtained by hacking into one or more unsecured third-party servers where the log files were exported and stored. We have verified that the logs likely were generated by ProxySG appliances and that these appliances have IP addresses generally assigned to Syria. We do not know who is using the appliances or exactly how they are being used. We currently are conducting an internal review and also are working directly with appropriate government agencies to provide information on this unlawful diversion.


Directly connecting users from the Syrian Arab Republic
5000
4000
3000
2000
1000
0
Dec-2010
Mar-2011
Jun-2011
The Tor Project - https://metrics.torproject.org/

“Twitter, mwitter!”



Protocol identification via deep-packet inspection (DPI)

Check packet contents against ***regular expressions***

/^(\x16\x03[\x00\x01\x02]..\x02...\x03[\x00\x01\x02]|...? .*/

Free translation: Does packet include “I’m TLS 1.1” ?

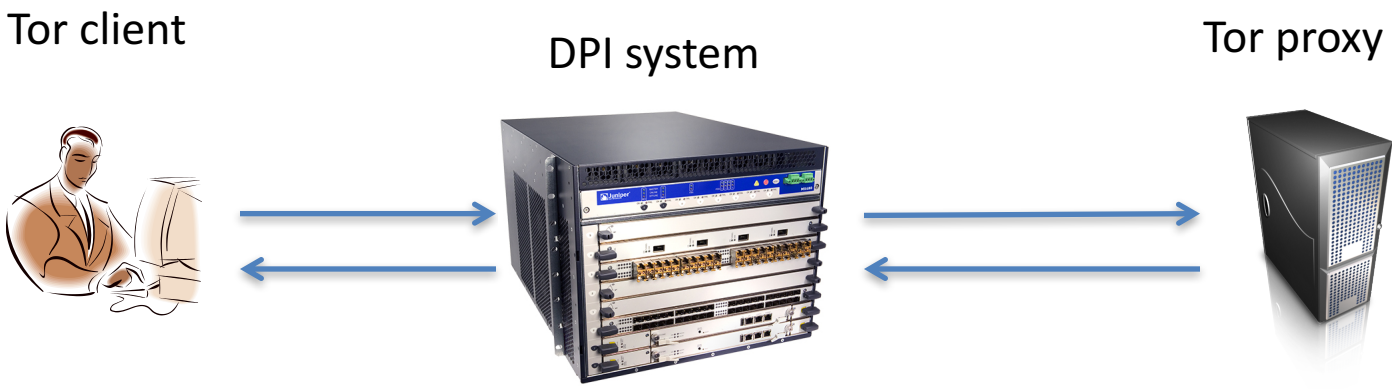
DPI users want to identify protocol X

X = TLS or Tor then throttle connection

X = HTTP then leave it alone

X = ??? then throttle traffic

Scenario:
DPI system only allows HTTP traffic unfettered



Stegonagraphy (e.g., Stegotorus): embed bits into HTTP messages

- Too slow for practical use (56k modem anyone?)

Obsfproxy (built into Tor): encrypt all bits sent over network (no plaintext bits)

- Really fast
- But DPI will flag traffic as ???

Want way to force DPI to classify traffic incorrectly as HTTP
So-called “misclassification attacks” against DPI

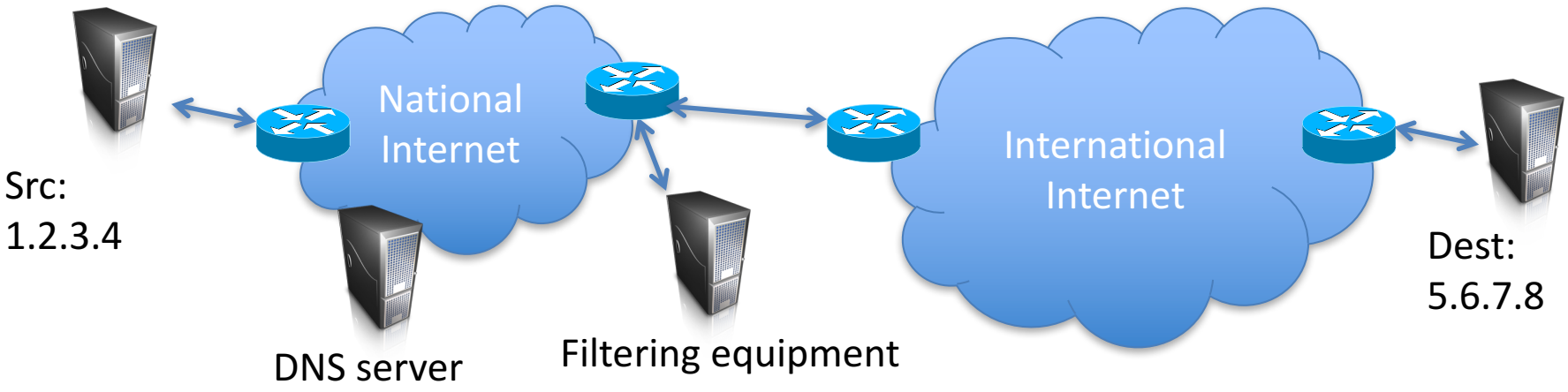
Surveying modern DPI systems

System	Look at ports?	TCP stream reassembly	Uses regex's	Use's C/C++
AppID	Yes	No	Yes	No
L7-filter	Yes	No	Yes	No
Yaf	Yes	Yes	Yes	No
Bro	Yes	Yes	Yes	Yes
nProbe	No	Yes	Not explicitly	Yes
Proprietary*	Yes	Yes	?	?

* Hint: it's a serious product (~$10k) and similar ones seem to be used in Iran.

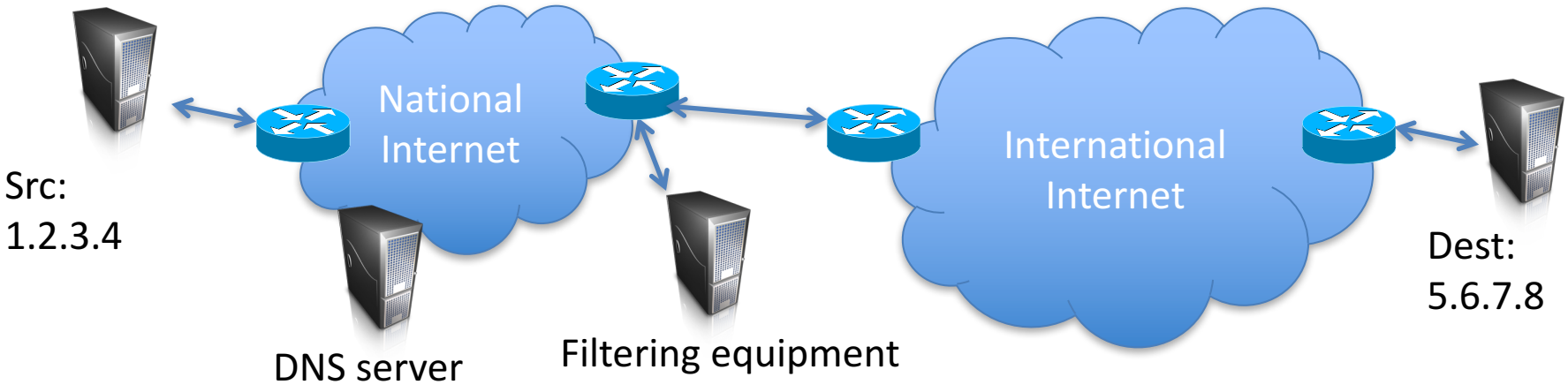
Can we build encryption schemes that fool regex-based systems?

How would you censor web requests?



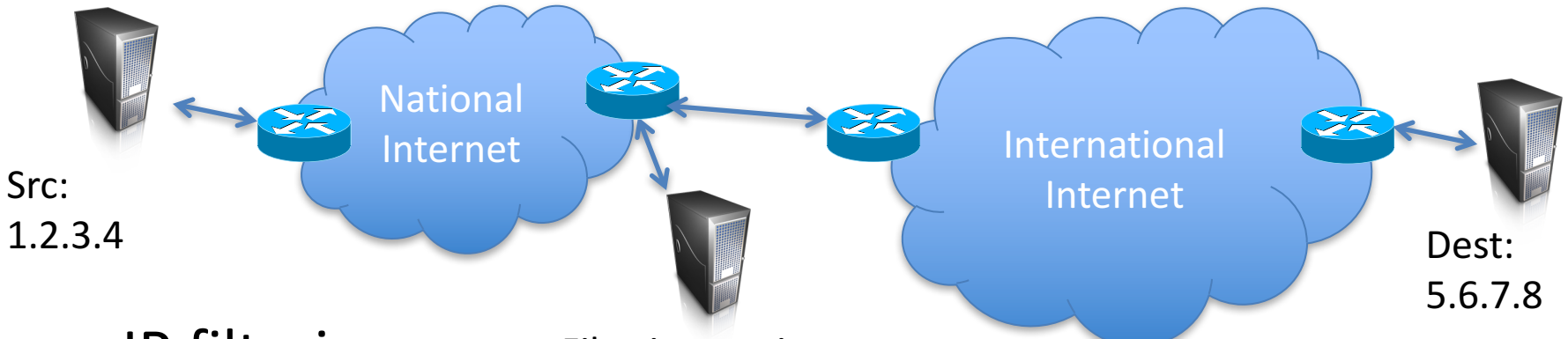
- IP filtering
- DNS filtering / redirection
- URL filtering
- Packet filtering (search keywords in TCP packets)
- Protocol filtering (detect Tor protocol)

How would you censor web requests?

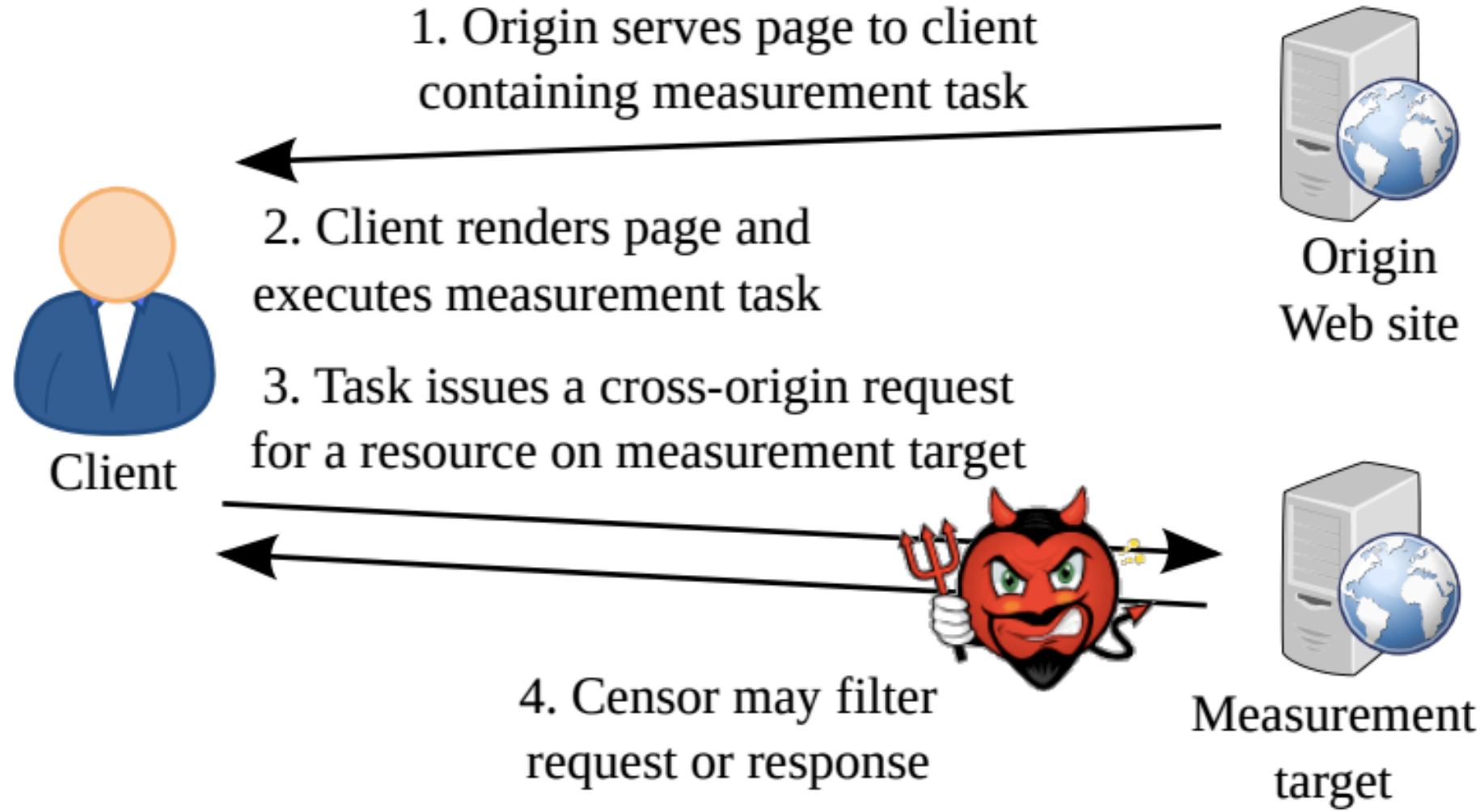


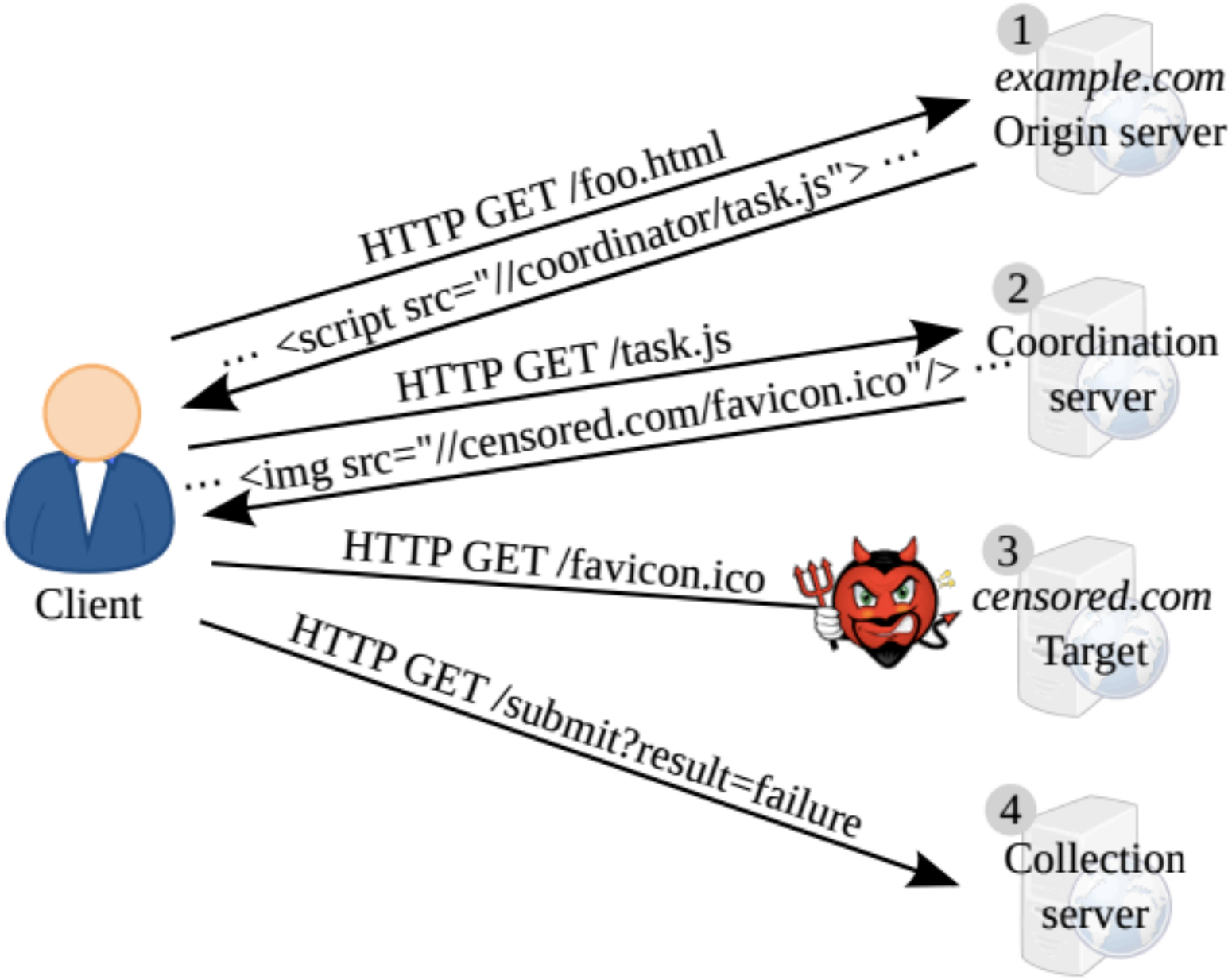
- IP filtering
- DNS filtering / redirection
- URL filtering
- Packet filtering (search keywords in TCP packets)
- Protocol filtering (detect Tor protocol)

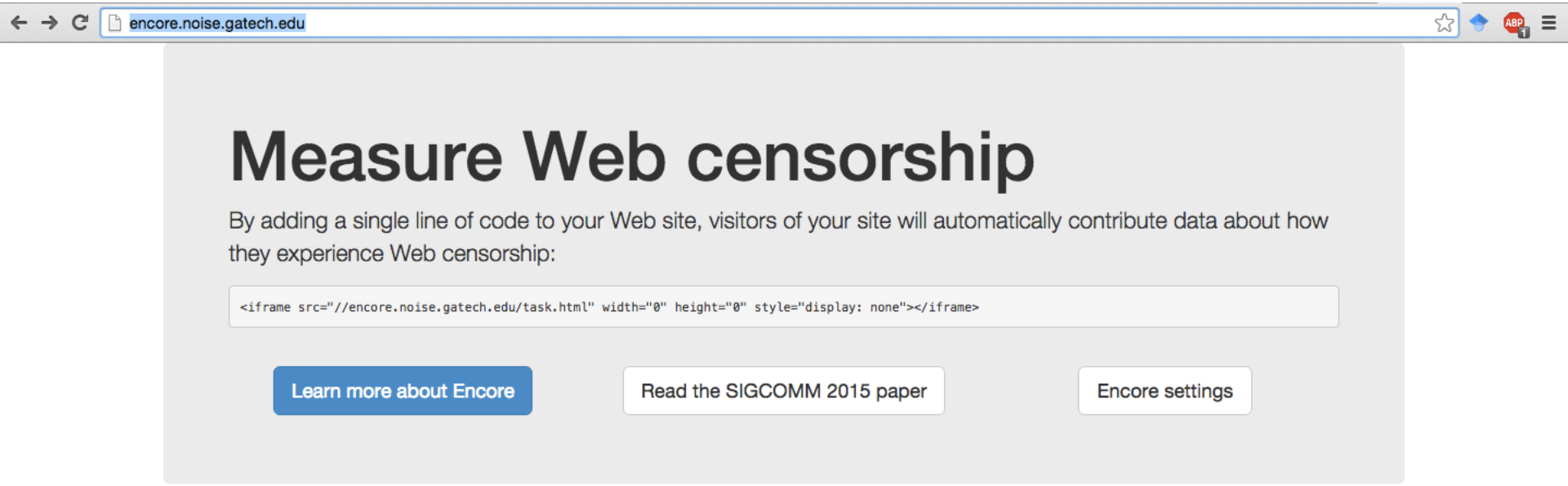
Circumvention of filtering



- IP filtering
  - Proxies
- DNS filtering / redirection
  - DNS proxy
- URL filtering or Packet filtering
  - Encryption / Tunneling / obfuscation
- Protocol filtering
  - Obfuscation techniques


1. Origin serves page to client containing measurement task
2. Client renders page and executes measurement task
3. Task issues a cross-origin request for a resource on measurement target
4. Censor may filter request or response
Client
Origin Web site
Measurement target


1
example.com
Origin server
HTTP GET /foo.html
… <script src="//coordinator/task.js"> …
2
Coordination server
HTTP GET /task.js
… …
Client
HTTP GET /favicon.ico
3
censored.com
Target
HTTP GET /submit?result=failure
4
Collection server


encore.noise.gatech.edu
Measure Web censorship
By adding a single line of code to your Web site, visitors of your site will automatically contribute data about how they experience Web censorship:
<iframe src="//encore.noise.gatech.edu/task.html" width="0" height="0" style="display: none"></iframe>
Learn more about Encore
Read the SIGCOMM 2015 paper
Encore settings

The results:

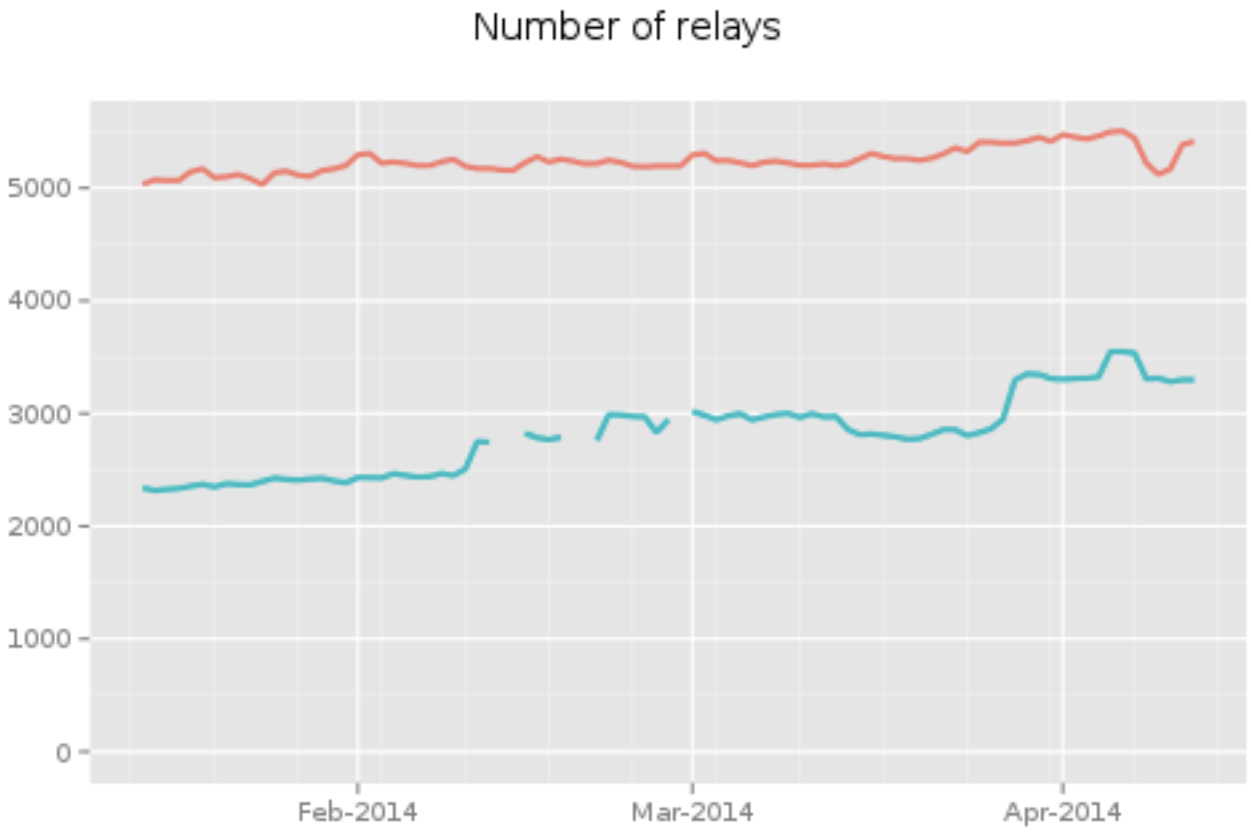
Confirmed blocking of:

- youtube.com in Pakistan, Iran, China
- Twitter.com, facebook.com in China, Iran

Statement from the SIGCOMM 2015 Program Committee: The SIGCOMM 2015 PC appreciated the technical contributions made in this paper, but found the paper controversial because some of the experiments the authors conducted raise ethical concerns. The controversy arose in large part because the networking research community does not yet have widely accepted guidelines or rules for the ethics of experiments that measure online censorship. In accordance with the published submission guidelines for SIGCOMM 2015, had the authors not engaged with their Institutional Review Boards (IRBs) or had their IRBs determined that their research was unethical, the PC would have rejected the paper without review. But the authors did engage with their IRBs, which did not flag the research as unethical. The PC hopes that discussion of the ethical concerns these experiments raise will advance the development of ethical guidelines in this area. It is the PC's view that future guidelines should include as a core principle that researchers should not engage in experiments that subject users to an appreciable risk of substantial harm absent informed consent. The PC endorses neither the use of the experimental techniques this paper describes nor the experiments the authors conducted.

Great Firewall targeting of Tor (circa 2011 and before)

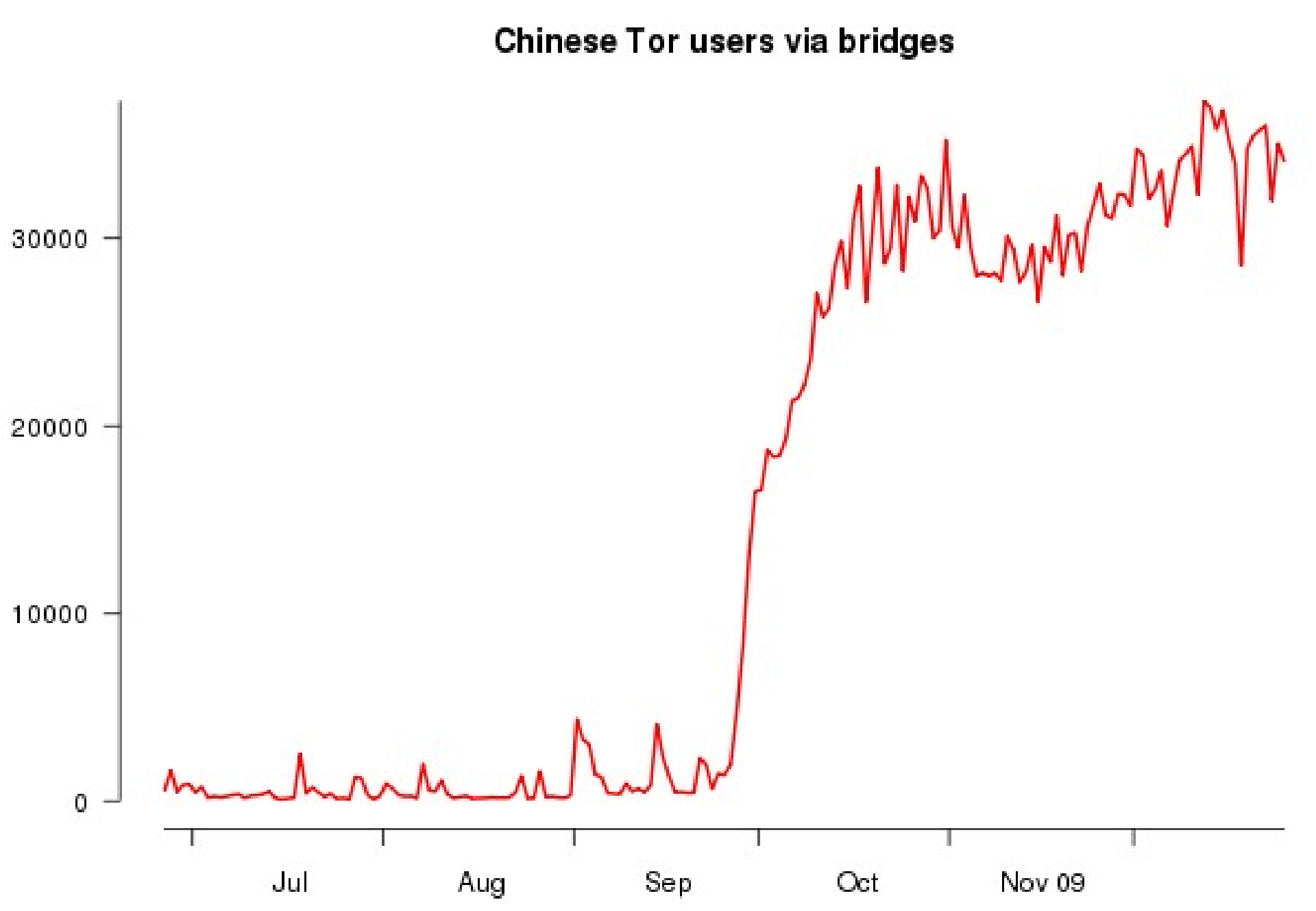
- Enumerate Tor relays and filter them



Relay is publicly listed Tor node

Bridge is Tor node not publicly listed


Number of directory requests to directory mirror trusted
10000
8000
6000
4000
2000
0
Sep 10
Sep 17
Sep 24
Oct 01
China
https://torproject.org



Tor project -- www.torproject.org

Iran DPI blocking of Tor

- Tor point-to-point connections use TLS