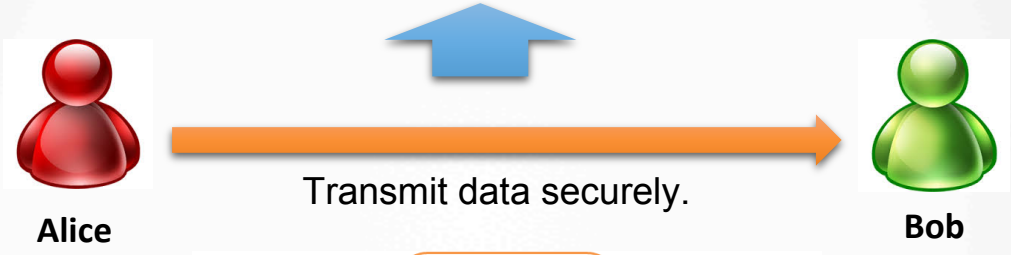
Seeing through Network-Protocol Obfuscation

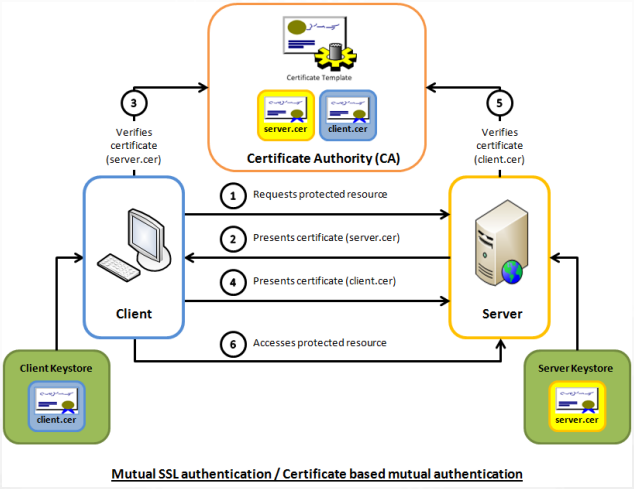
Liang Wang[1], Kevin P. Dyer[2], Aditya Akella[1],
Thomas Ristenpart[3], Thomas Shrimpton[4]

[1] University of Wisconsin-Madison, [2] Portland State University, [3] Cornell Tech, [4] University of Florida

Imagine a representative situation:

SSL/TLS (HTTPS)





Imagine a representative situation:

1、Other Ways?

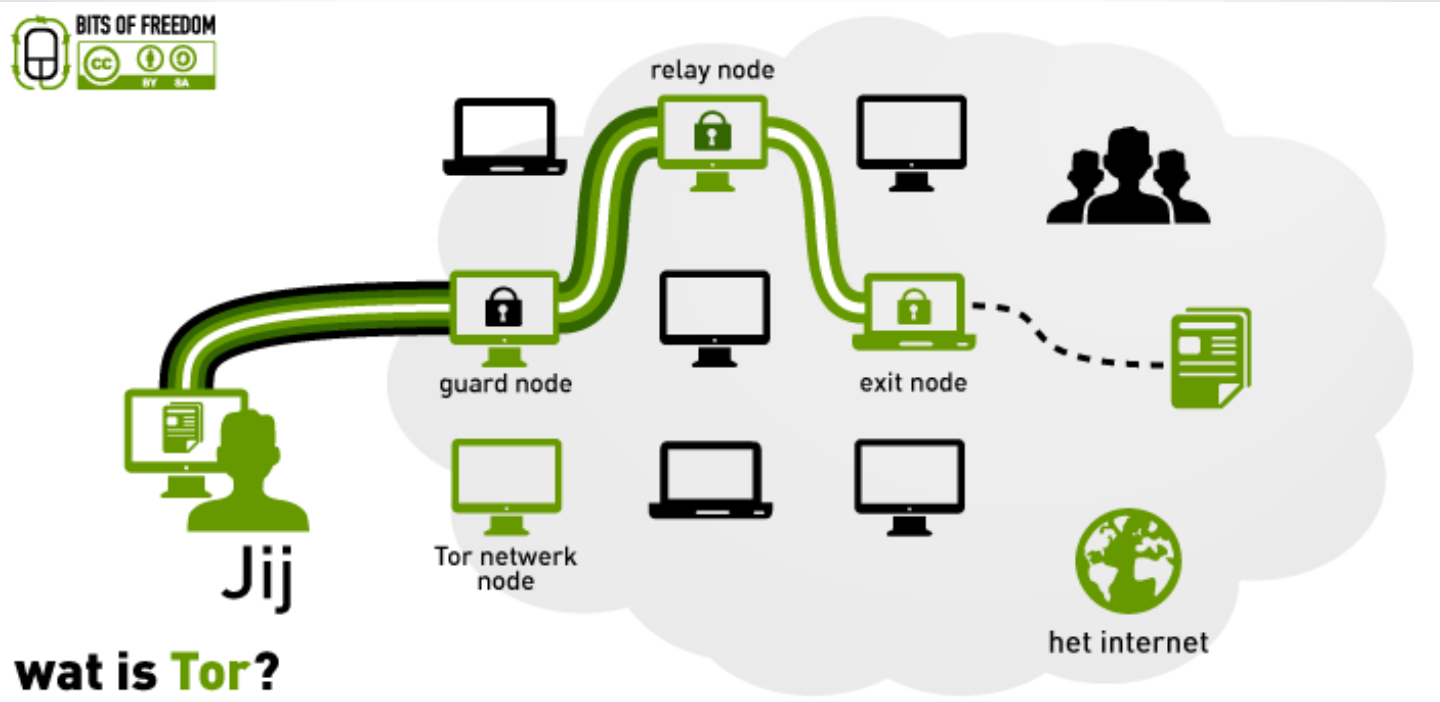
2、More Secure?

Network protocol obfuscation

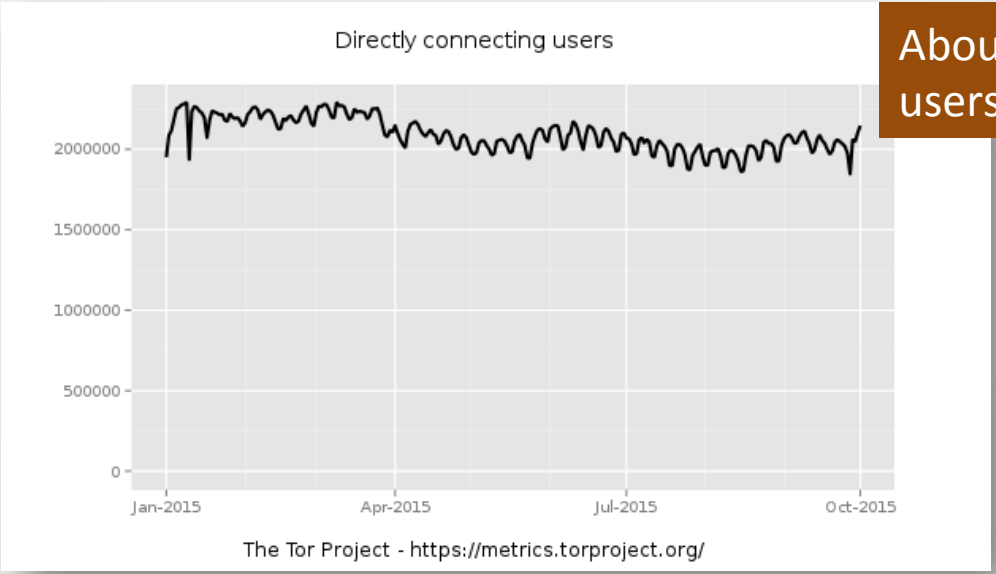
Researchers have proposed a large number of approaches for obfuscating the network protocol being used.

These obfuscation tools can be loosely categorized as either attempting to randomize all bytes sent on the wire, attempting to look like (or mimic) an unblocked protocol such as HTTP, or tunneling traffic over an implementation of an unblocked Protocol.

Anonymous Communication



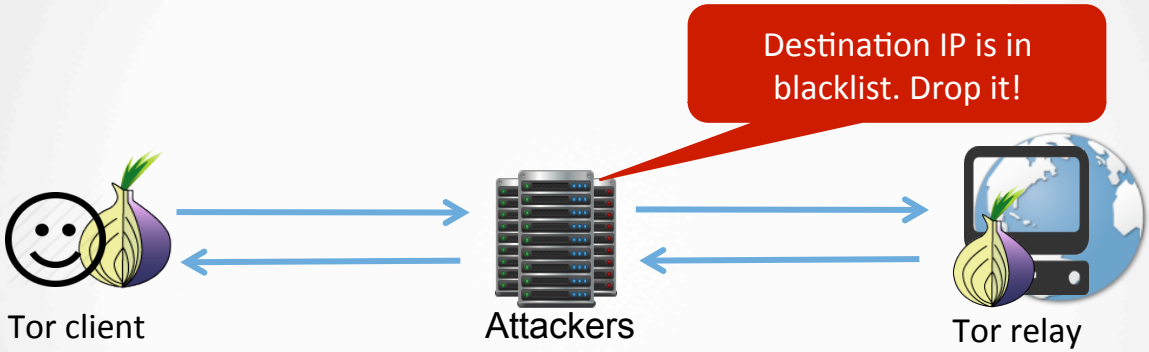
Tor -- the most popular low-latency, anonymous network



About 2M users/day!

However, Tor can be detected, and is blocked in some countries

Approaches to attacked Tor



What attackers can do:

- IP filtering
- Deep packet inspection
- Active probing

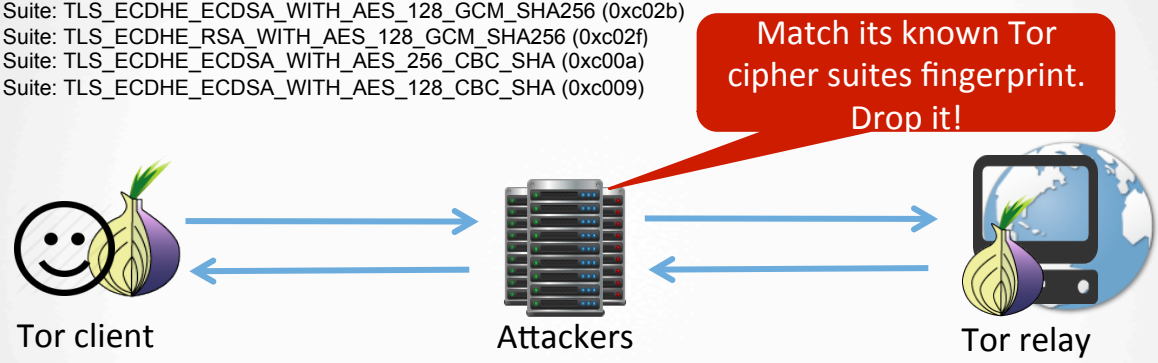
What attackers can see:

- source address
- destination address/port
- application-level protocols
- keywords in payloads
- …

Approaches to attacking Tor

Cipher Suites (24 suites)
Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a)
Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009)

.....



What attackers can do:
- IP filtering
- Deep packet inspection
- Active probing

What attackers can see:
- source address
- destination address/port
- application-level protocols
- keywords in payloads
- ...

Attackers vs. anonymous communication

Attackers

Identity the traffic from anonymous communication tools:

HIGH true-positive rate (TPR)

LOW false-positive rate (FPR)

VS.

Anonymous communication tools

Prevent attackers from identifying its traffic effectively:

LOW true-positive rate

HIGH false-positive rate

protocol / Attack label	Attacked	Not Attacked
Attacked	TP	FP

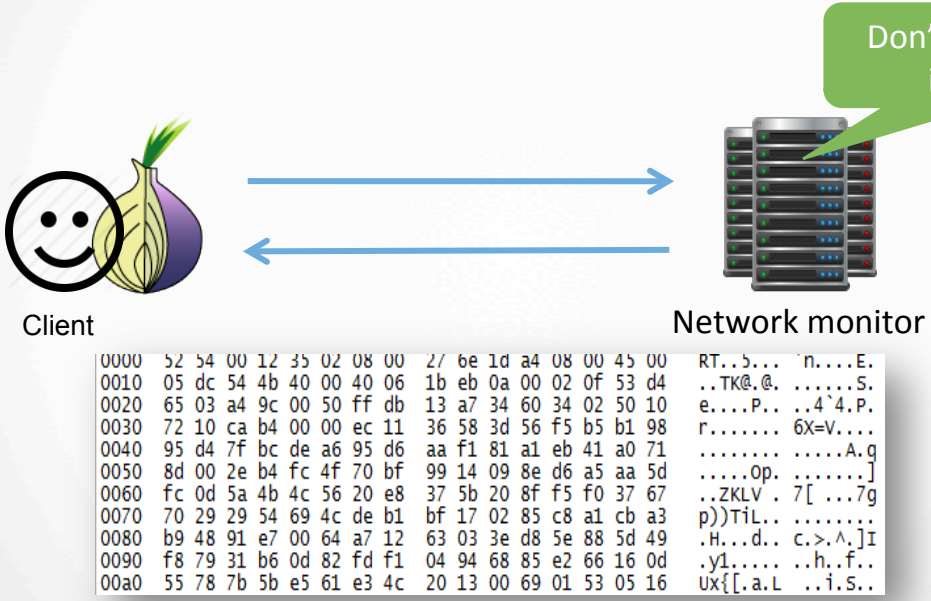
Network Protocol Obfuscation

- Prevent attackers from recognizing the protocol
- Cause attackers to misidentify flows of a attacked protocol as a "benign" protocol

Types: ***Randomizer, Protocol mimicry, Tunneling***

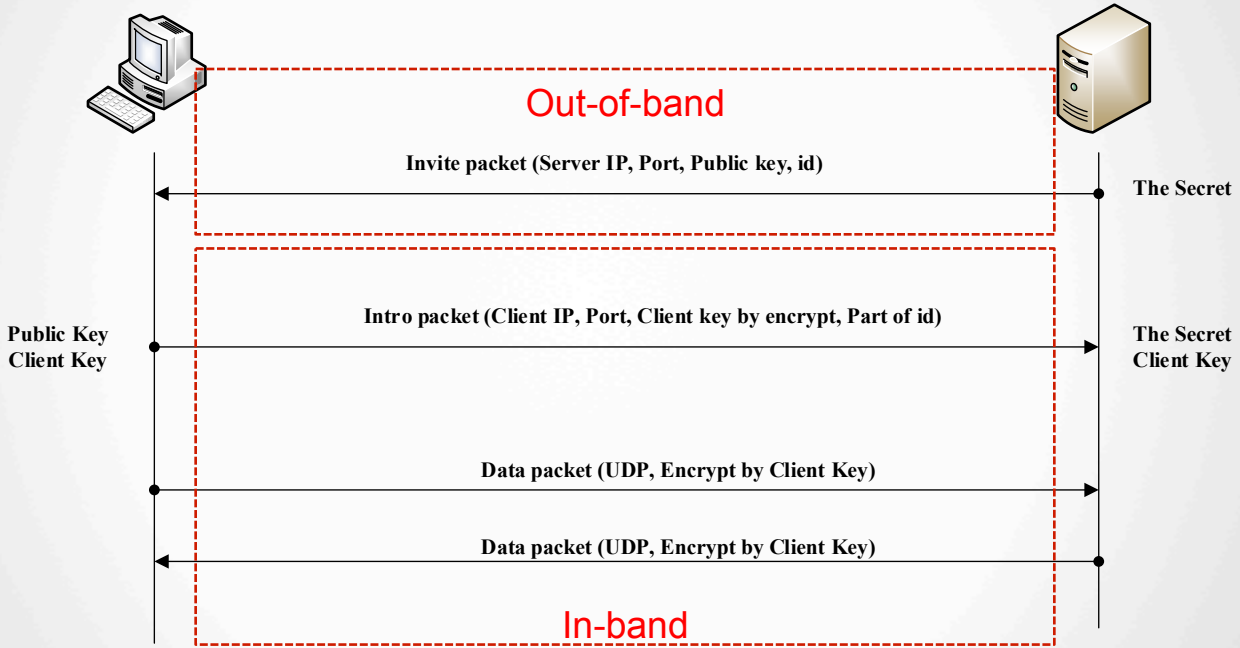
Network protocol obfuscation

Randomizer: “Look like nothing”



Network protocol obfuscation

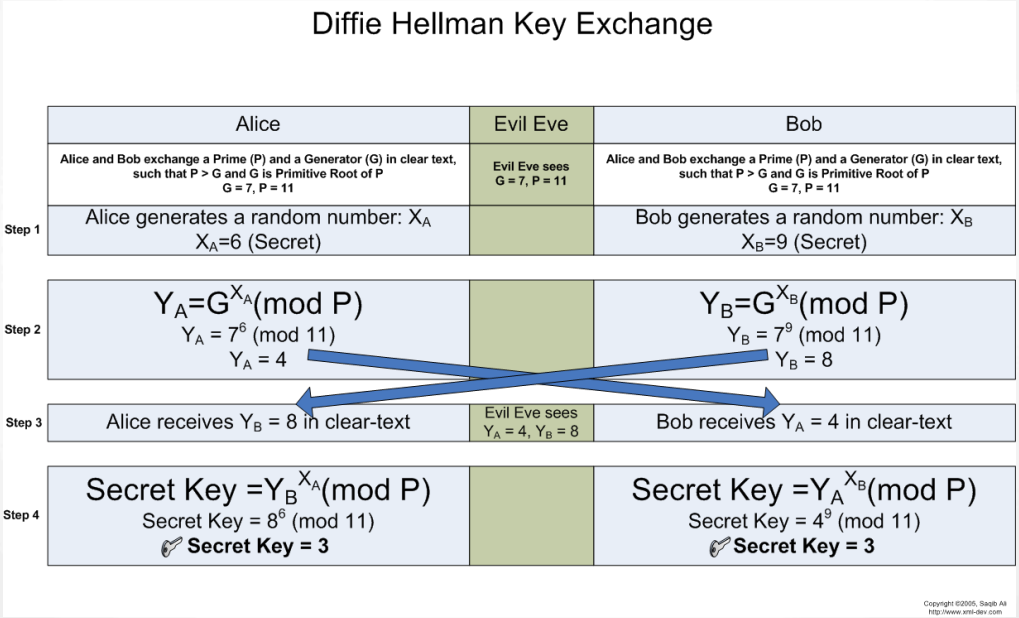
Randomizer: Dust – “Look like nothing”



Wiley, Brandon. "Dust: A blocking-resistant internet transport protocol." 2014-03-26], http://blanu. net/Dust. pdf (2011).

Network protocol obfuscation

Randomizer: Obfsproxy3 – “Look like nothing”



Network protocol obfuscation

Randomizer: Obfsproxy4 – “Look like nothing”

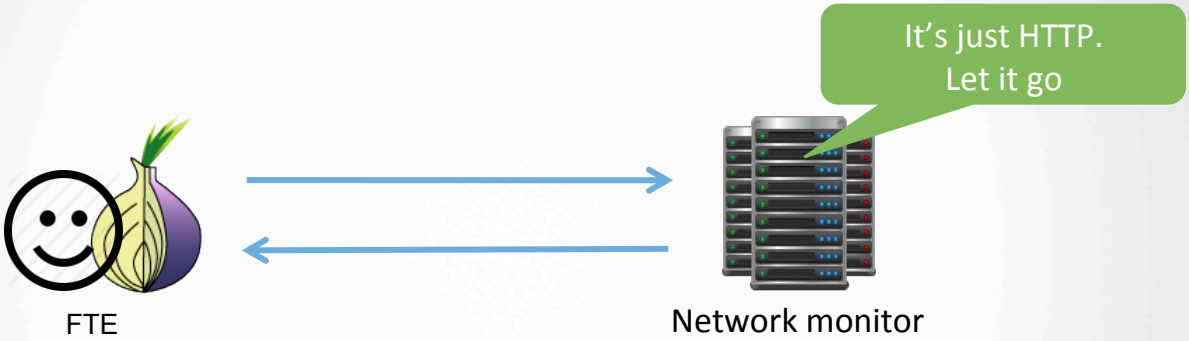
一是对于每个服务器都有属于自己的独特的协议形状（也就是说与不同的服务器建立连接后传输的数据包的长度，传输间隔，头部结构等信息），从而使得**Attackers**找不到明显特征；

二是使用伪随机算法进行填充，从而使得数据包长度和传输间隔变得分散，没有明显规律可循；

三是将数据包的长度与传输间隔向“正常”数据包靠拢（例如普通的**HTTPS**数据包），使得**Attackers**误判率飙升从而不敢进行封锁。

Network protocol obfuscation

Protocol Mimicry: SkypeMorph, Stegotorus, FTE proxy




```
  Request Method: GET
  Request URI [truncated]: /EyyCfNKfjvhizW0
  Request Version: HTTP/1.1
\r\n
```

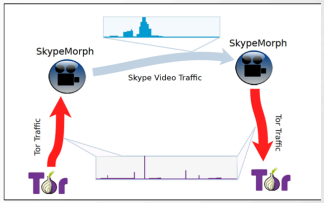
Weinberg, Zachary, et al. "StegoTorus: a camouflage proxy for the Tor anonymity system." CCS 2012
Mohajeri Moghaddam, Hooman, et al. "Skypemorph: Protocol obfuscation for tor bridges." CCS 2012
Dyer, Kevin P., et al. "Protocol misidentification made easy with format-transforming encryption." CCS 2013.

SkypeMorph

Censorship Region

The Internet

Traffic Shaping



SkypeMorph Client

SkypeMorph Bridge

A Tor node

StegoTorus

Censorship Region

The Internet

HTTP

HTTP

Skype

Ventrilo

HTTP

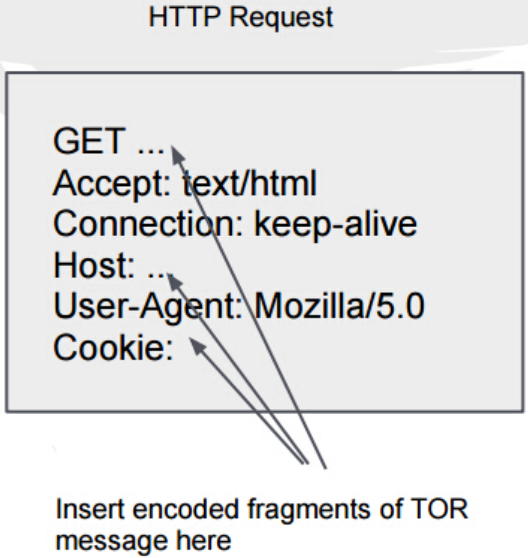
StegoTorus Client

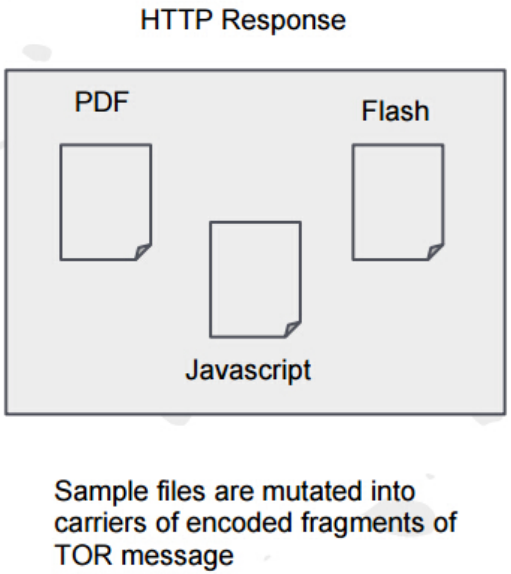
StegoTorus Bridge

A Tor node

Let's Pretend

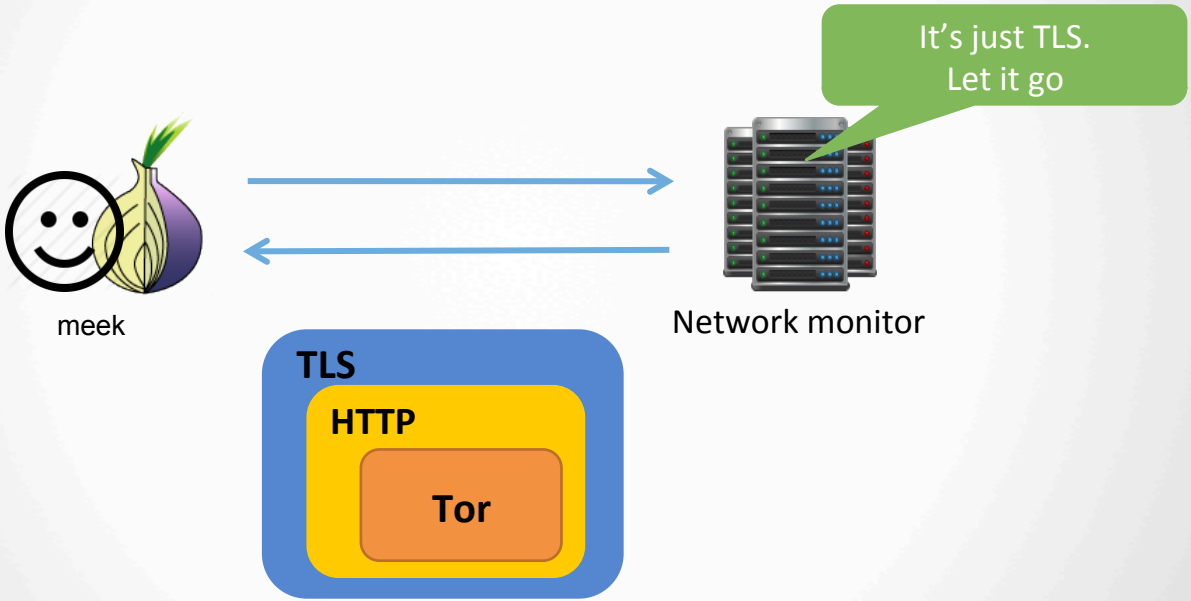
- Make Tor traffic look like HTTP





Network protocol obfuscation

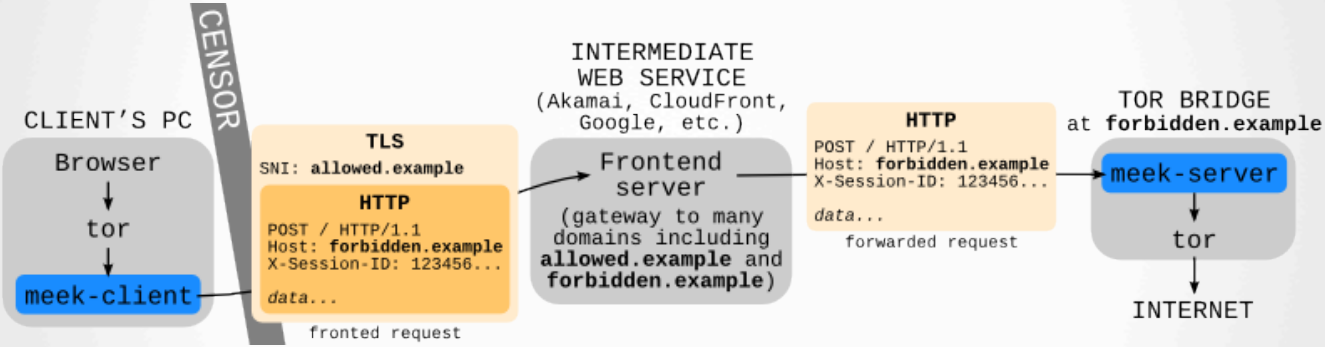
Tunneling: meek – leverage genuine TLS connection and Cloud



Fifield, David, et al. "Blocking-resistant communication through domain fronting." PETS 2015

Network protocol obfuscation

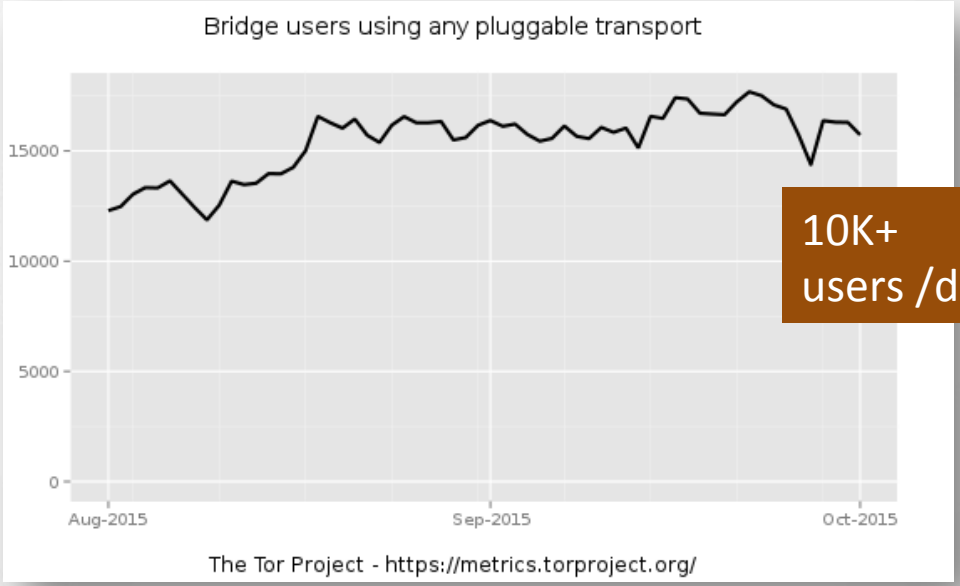
Tunneling: meek – leverage genuine TLS connection and Cloud



Fifield, David, et al. "Blocking-resistant communication through domain fronting." PETS 2015

Network protocol obfuscation

Some obfuscators such as **obfsproxy3**, **obfsproxy4**, **FTE** and **meek** are deployed in the Tor Browser Bundle as Tor Pluggable Transport (PT) and in use



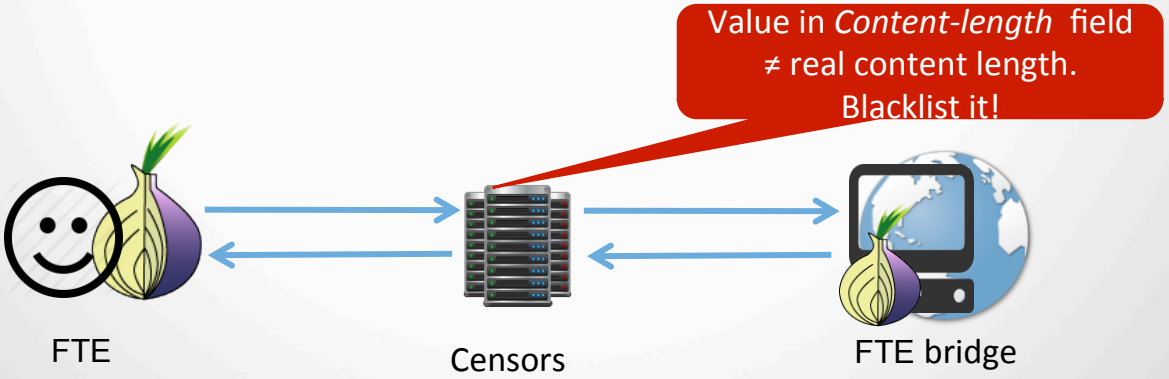
10K+
users /day

Can censors detect the obfuscators reliably?

Proposed Attacks against Tor Pluggable Transports

Name	Type	Used in Tor?	Proposed Attack	Source
Obfsproxy3	Randomizer	**Yes**	**NONE**	
Obfsproxy4	Randomizer	**Yes**	**NONE**	
meek	Tunneling	**Yes**	**NONE**	
FTE	Mimicry	**Yes**	Semantics-based attacks	Dyer et al., 2013
SkypeMorph	Mimicry	No	Semantics-based attacks, traffic analysis	Houmansadr et al., 2013 J. Geddes et al., 2013
Stegotorus	Mimicry	No	Semantics-based attacks	Houmansadr et al., 2013

Semantics-based attacks: looking for the deviations of a target system from expected behaviors



Proposed Attacks against Tor Pluggable Transports

Name	Type	Used in Tor?	Proposed Attack	Source
Obfsproxy3	Randomizer	**Yes**	**NONE**	
Obfsproxy4	Randomizer	**Yes**	**NONE**	
meek	Tunneling	**Yes**	**NONE**	
FTE	Mimicry	**Yes**	Semantics-based attacks	Dyer et al., 2013
SkypeMorph	Mimicry	No	Semantics-based attacks, traffic analysis	Houmansadr et al., 2013 J. Geddes et al., 2013
Stegotorus	Mimicry	No	Semantics-based attacks	Houmansadr et al., 2013

Proposed attacks have not been evaluated in terms of false positives

No attacks against obfsproxy3, obfsproxy4 and meek!

Our contributions

- Analysis of false-positive rates of attacks on terabytes of network traces

-- Old attacks: Semantics-based

-- New attacks: **Entropy-based, ML-based**

- New attacks against deployed Tor PTs that achieve:

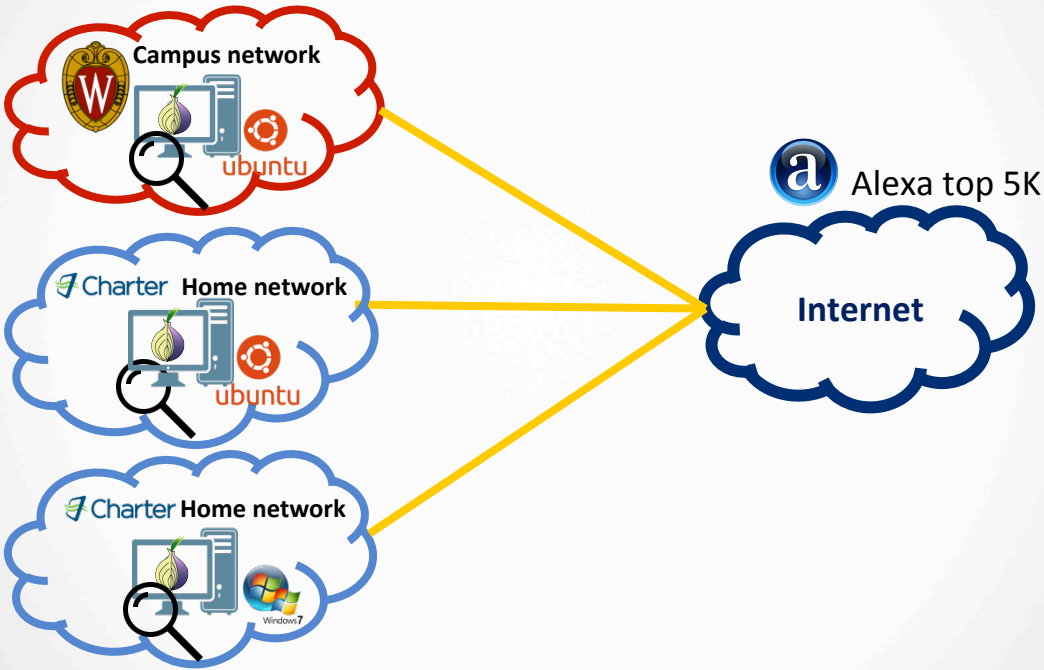
-- High true-positive rates

-- False-positive rates <= 0.2%

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%

Synthetic dataset for TPR

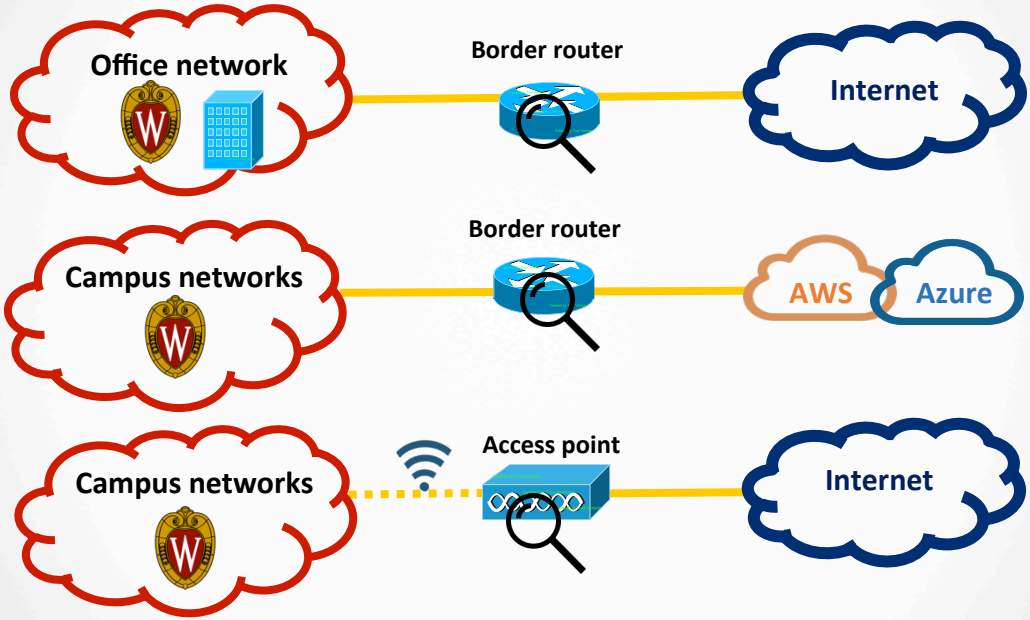
Tor Dataset:



15,000 traces of each of obfsproxy3, obfsproxy4, FTE, meek-google, meek-amazon

Real network traces from UW-Madison for FPR

Campus Dataset:



Networks contains 320K hosts
A total of 14M TCP flows

Proposed Attacks against Tor Pluggable Transports

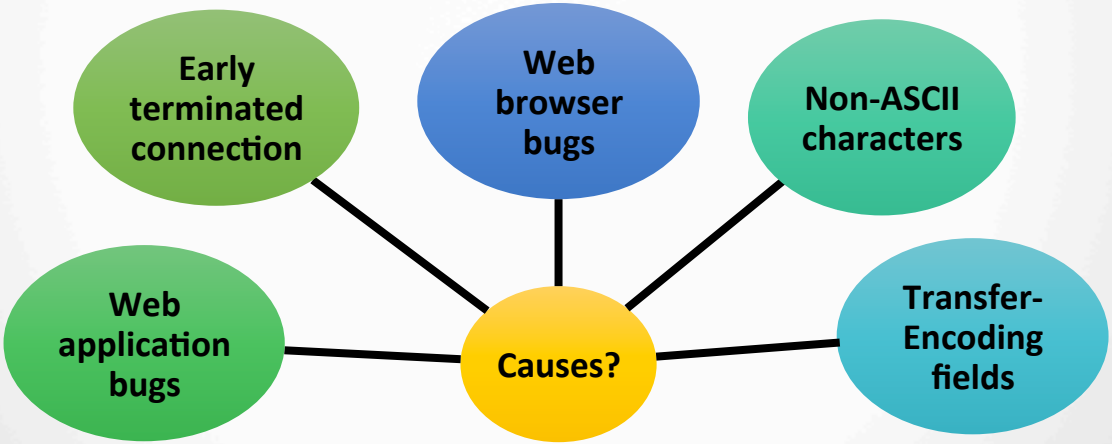
What are the false-positive rates of the proposed attacks?

Name	Type	Used in Tor?	Proposed Attack	Source
Obfsproxy3	Randomizer	**Yes**	**NONE**	
Obfsproxy4	Randomizer	**Yes**	**NONE**	
meek	Tunneling	**Yes**	**NONE**	
FTE	Mimicry	**Yes**	Semantics-based attacks	Dyer et al., 2013
SkypeMorph	Mimicry	No	Semantics-based attacks, traffic analysis	Houmansadr et al., 2013 J. Geddes et al., 2013
Stegotorus	Mimicry	No	Semantics-based attacks	Houmansadr et al., 2013

Semantics-based attack against FTE

DPI test: value in *Content-Length* field ≠ real content length → Block it.

FPR = 1.4%: 1.4% of HTTP messages from campus dataset have incorrect *Content-Length* fields

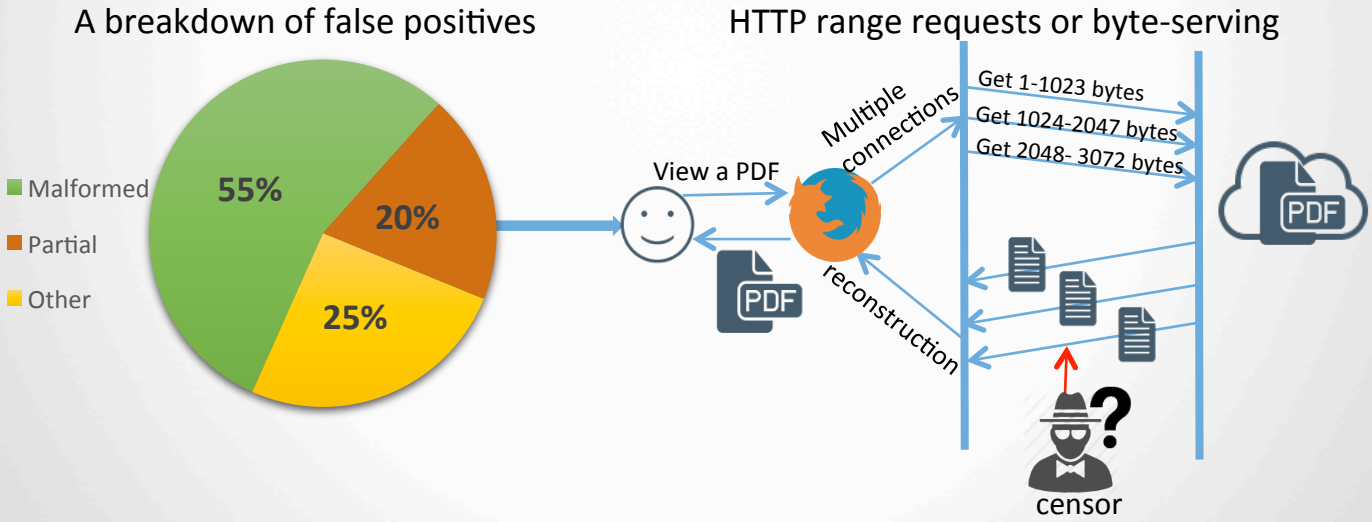


Dyer, Kevin P., et al. "Protocol misidentification made easy with format-transforming encryption." CCS 2013.

Semantics-based attack against Stegotorus

DPI test: a PDF has incorrect xref table → Block it

FPR = 43%: 43% of the 10,847 PDFs from campus datasets have incorrect **xref** tables



Houmansadr, Amir, et al. "The parrot is dead: Observing unobservable network communications." IEEE SP 2013.

Semantics-based attacks may be hard to deploy, so…

We develop new attacks!

- **High true-positive rate**
- **Low false-positive rate**
- **Good performance**

Overview of new attacks

Obfuscator	Attack	TPR	FPR
Obfsproxy3	entropy + length	100%	0.2%
Obfsproxy4	entropy + length	100%	0.2%
FTE	URI entropy + length	100%	0.003%
meek-amazon	decision tree	98%	0.02%
meek-google	decision tree	98%	0.006%

Overview of new attacks

Obfuscator	Attack	TPR	FPR
Obfsproxy3	entropy + length	100%	0.2%
Obfsproxy4	entropy + length	100%	0.2%
FTE	URI entropy + length	100%	0.003%
meek-amazon	decision tree	98%	0.02%
meek-google	decision tree	98%	0.006%

Entropy-based attacks

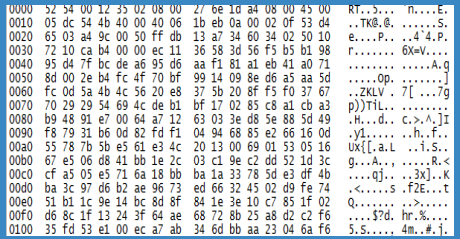
Inspect the handshake message (the first message sent)

Conventional encryption protocols (e.g, TLS): plaintext + random bytes

```
SSLv2 Record Layer: Client Hello
  [Version: SSL 2.0 (0x0002)]
  Length: 103
  Handshake Message Type: Client Hello (1)
  Version: SSL 3.0 (0x0300)
  Cipher Spec Length: 78
  Session ID Length: 0
  Challenge Length: 16
⊞ Cipher Specs (26 specs)
  Challenge
```

VS.

Obfsproxy3/4: encrypted, a random string



Length of handshake message: >= 146 bytes

Entropy-based attack for Obfsproxy3/4

Possible tests

- Shannon entropy
- Kolmogorov-Smirnov two sample test (KS-2) on blocks
- KS-2 test on bytes
- Sequential probability ratio test (SPRT)

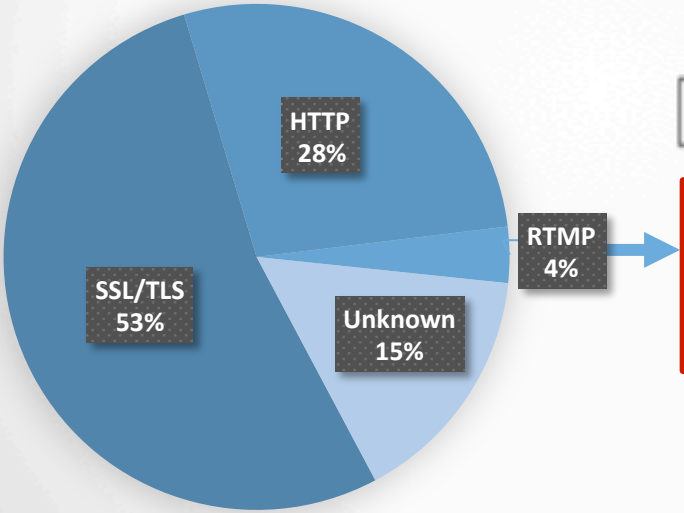
Winner = KS-2 test on 8-byte blocks + payload length check

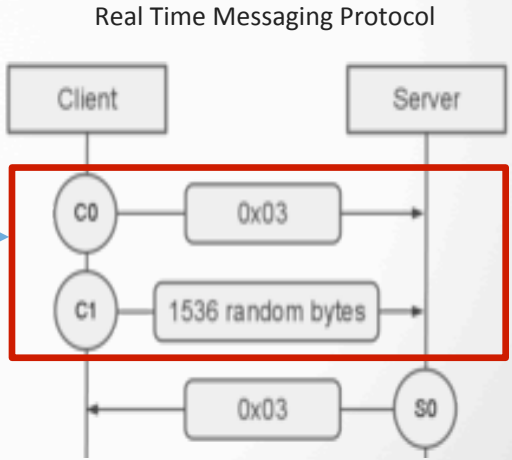
White, Andrew M., et al. "Clear and Present Data: Opaque Traffic and its Security Implications for the Future." NDSS 2013
Everspaugh, Adam, et al. "Not-so-random numbers in virtualized Linux and the Whirlwind RNG." IEEE SP 2014.

Entropy-based attack for Obfsproxy3/4

DPI test: message length >= 146 +
(obfs3 193bytes, obfs4 149bytes)
Pass KS-2 test on 8-byte blocks → Block it

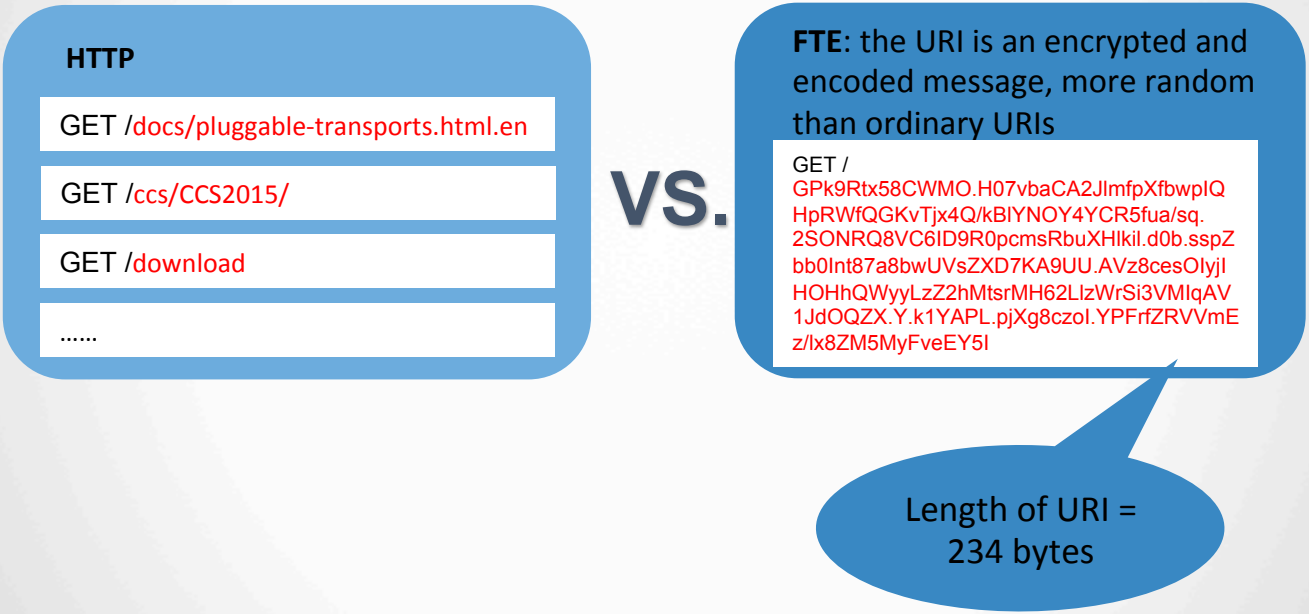
FPR = 0. 2%: 36 K (0.2%) false positives out of **14 M** flows





Entropy-based attacks

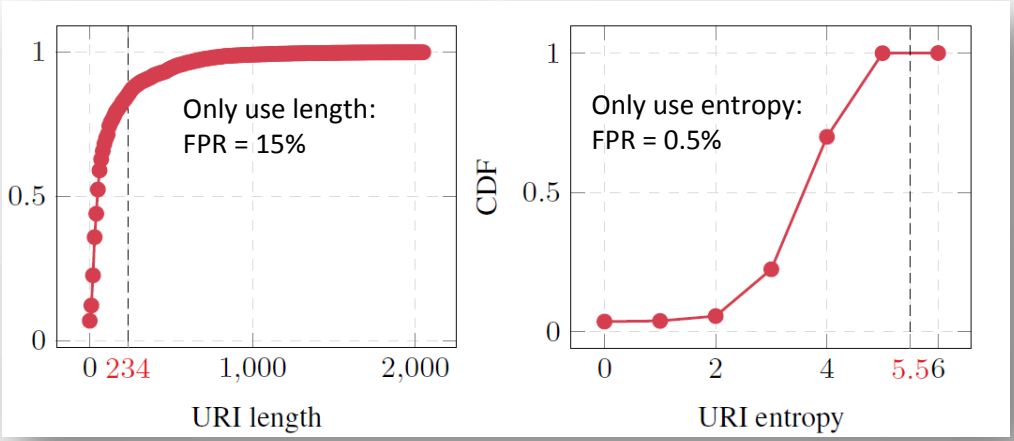
Inspect the handshake message (the first message sent)



Entropy-based attack for FTE

DPI test: URI length = 234 +
URI Shannon entropy > 5.5 → Block it

FPR = 0. 003%: 264 (0.003%) false positives out of **10 M** URIs



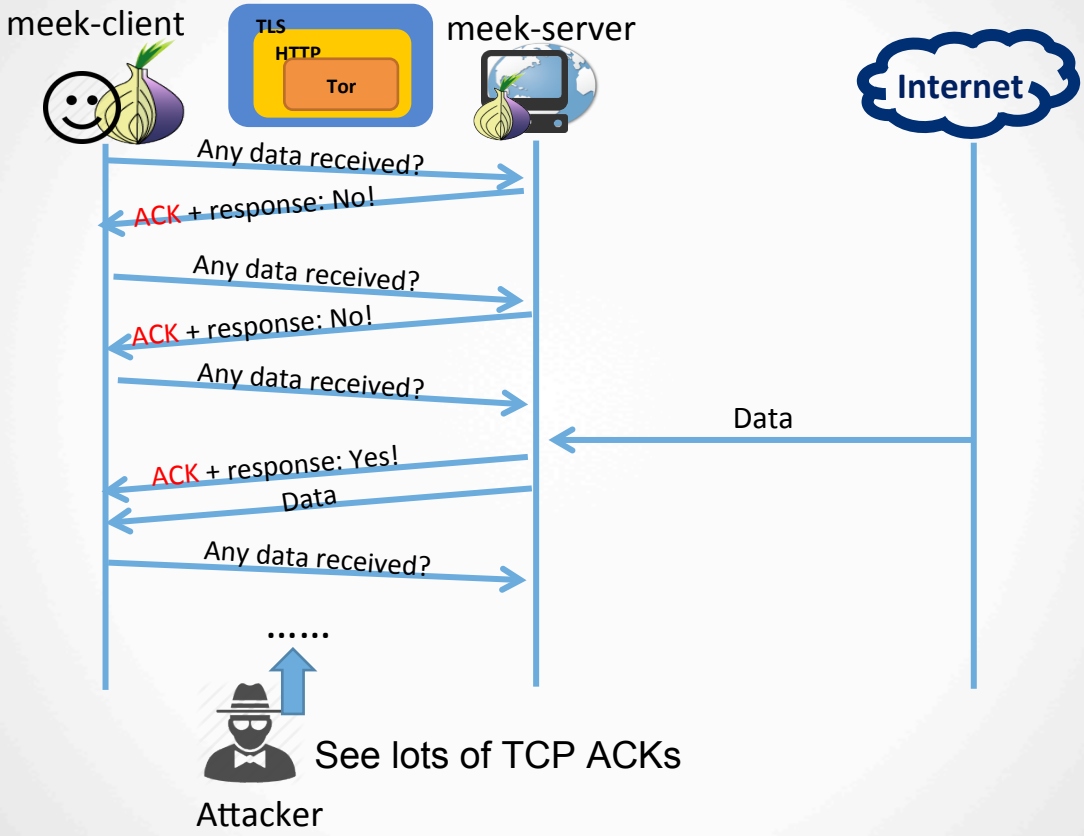
CDF of length and entropy distribution of the 10M URIs from the campus dataset

Overview of new attacks

Obfuscator	Attack	TPR	FPR
Obfsproxy3	entropy + length	100%	0.2%
Obfsproxy4	entropy + length	100%	0.2%
FTE	URI entropy + length	100%	0.003%
meek-amazon	decision tree	98%	0.02%
meek-google	decision tree	98%	0.006%

?

Meek client will poll meek server periodically



Tor project. Tor meek. https://trac.torproject. org/projects/tor/wiki/doc/meek, 2015.

Traffic analysis via ML

How many packets of a flow are used for feature extractions?

- *Time-based or Packet-count*

Timing-based : {2,3,4, … ,10}; Packet-count : {30,35,40,…,300}

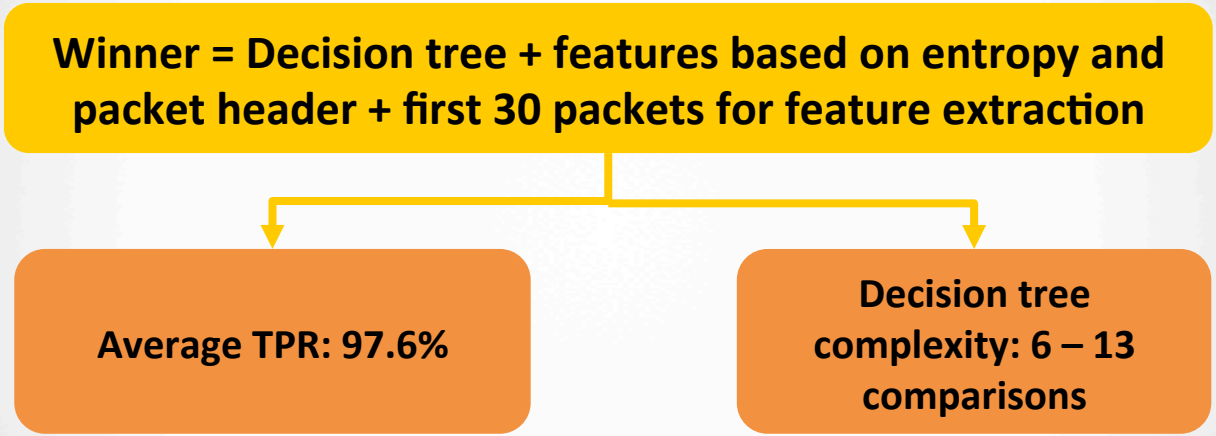
What features are used?

- *Timing-based, Entropy-based, or Packet-header*

(0,1](1,2](2,3](3,4](4,5](5,6](6,7](7,8](8,9](9,10]
(10,20](20,30](30,40](40,50](50,60](60,70](70,80](80,90](90,100]
(100,200](200,300](300,400](400,500](500,600](600,700](700,800](800,900](900,1000]
(1000,∞]

A total of about 1.3K possible combinations

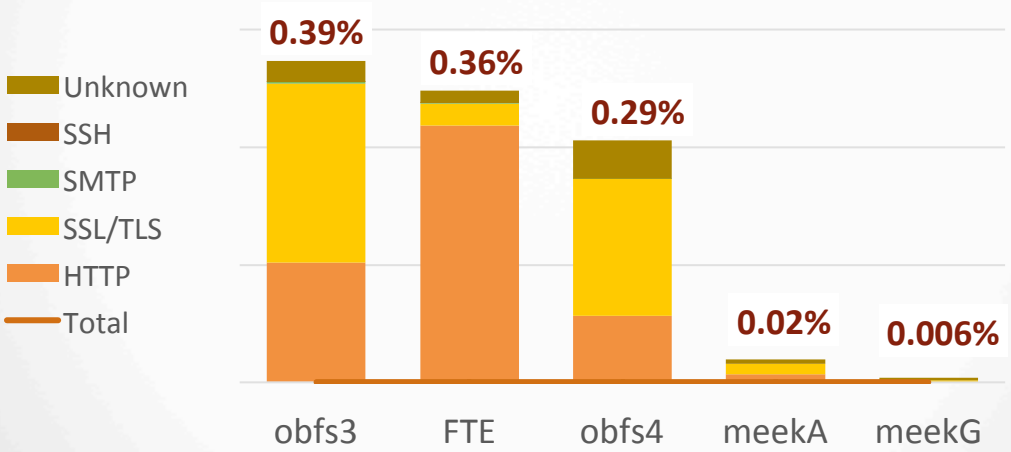
Performance of ML attacks on Tor datasets



False positives of ML attacks on campus datasets

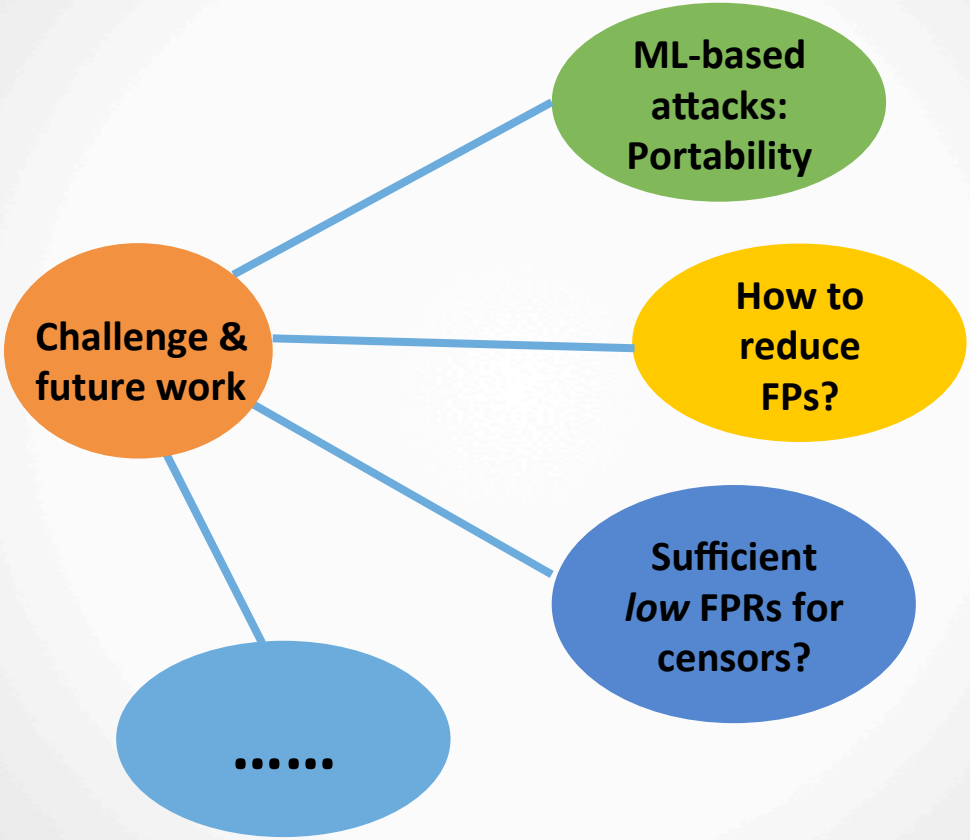
Decision tree + features based on entropy and packet header + first 30 packets for feature extraction

False-positive rate (out of **14 M** TCP flows) for each target:



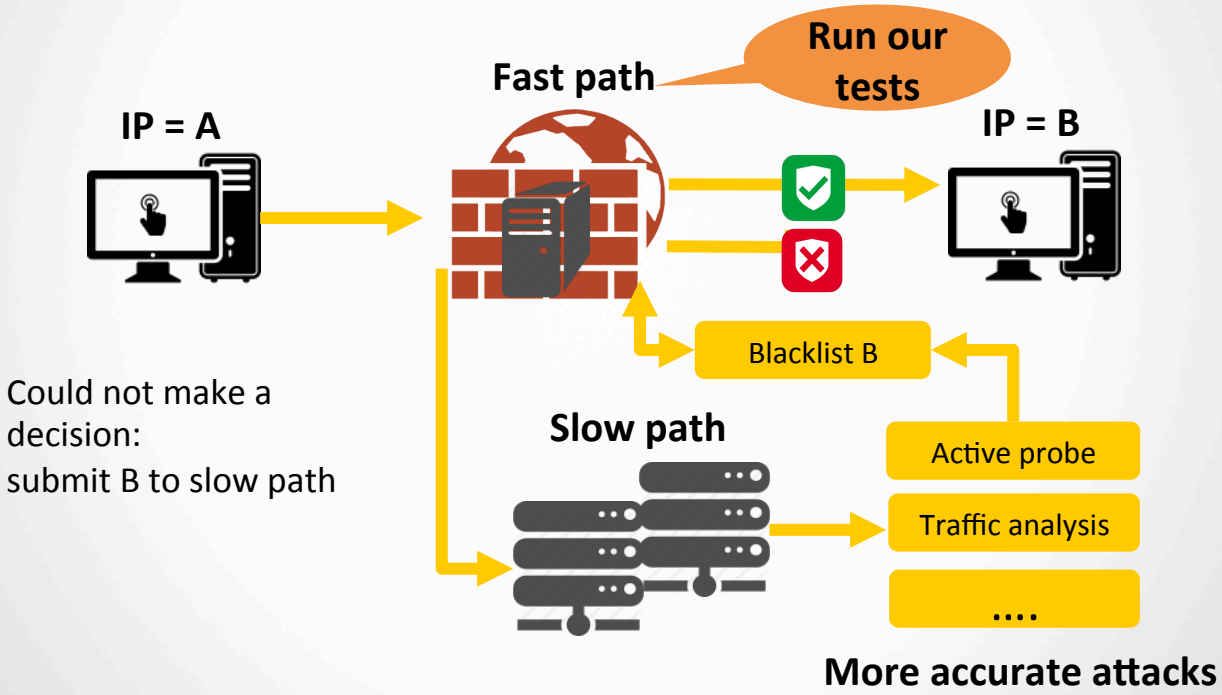
The *aggregated* false-positive rate across all targets is **0.98%**

Challenge and future work



Are the false positives too many for censors?

Censors could use a two-stage detection pipeline (as the *Great Firewall of China*) to reduce false positives



Conclusion

Obfuscators used in Tor can be detected by DPI-based censors reliably with high TPRs and low FPRs

Obfuscator	Attack	TPR	FPR
Obfsproxy3	entropy + length	100%	0.2%
Obfsproxy4	entropy + length	100%	0.2%
FTE	URI entropy + length	100%	0.003%
meek-amazon	decision tree	98%	0.02%
meek-google	decision tree	98%	0.006%

First false-positive rate analysis of DPI-based Tor-obfuscator attacks

Open questions: portability of ML attacks, lower FPs?

Questions?