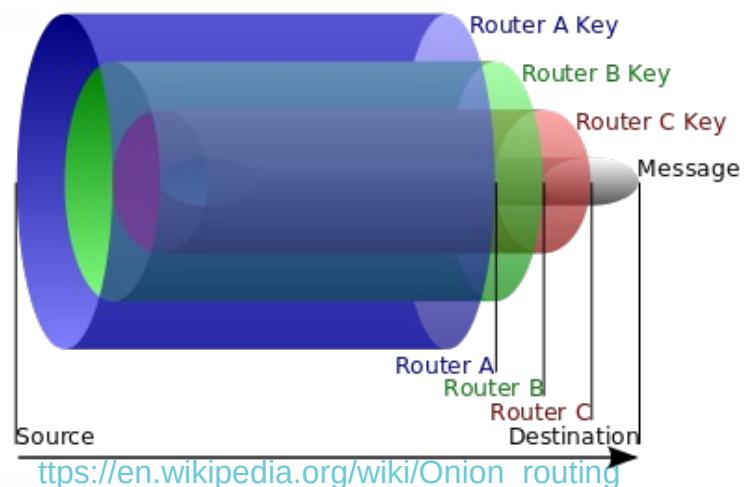
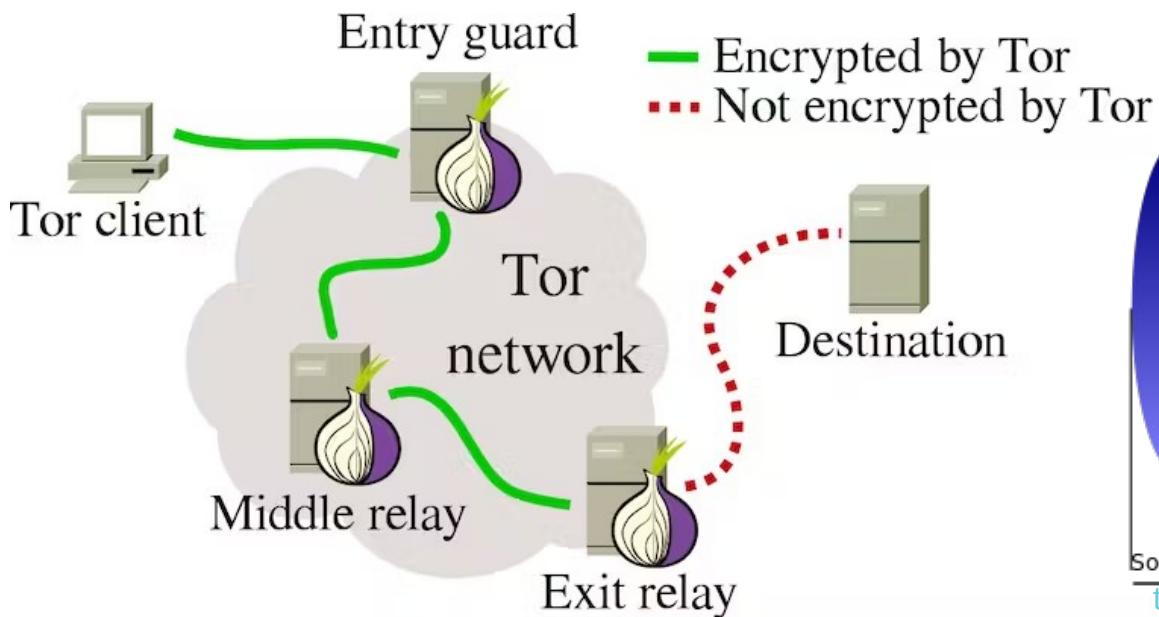


# CSE 468 Fall 2025 Tor lecture

jedimaestro@asu.edu

- Some of these slides are from  
<https://community.torproject.org/training/resources/>
- See also
  - <https://community.torproject.org/onion-services/setup/>

# Tor in a nutshell



<https://theconversation.com/tor-upgrades-to-make-anonymous-publishing-safer-73641>

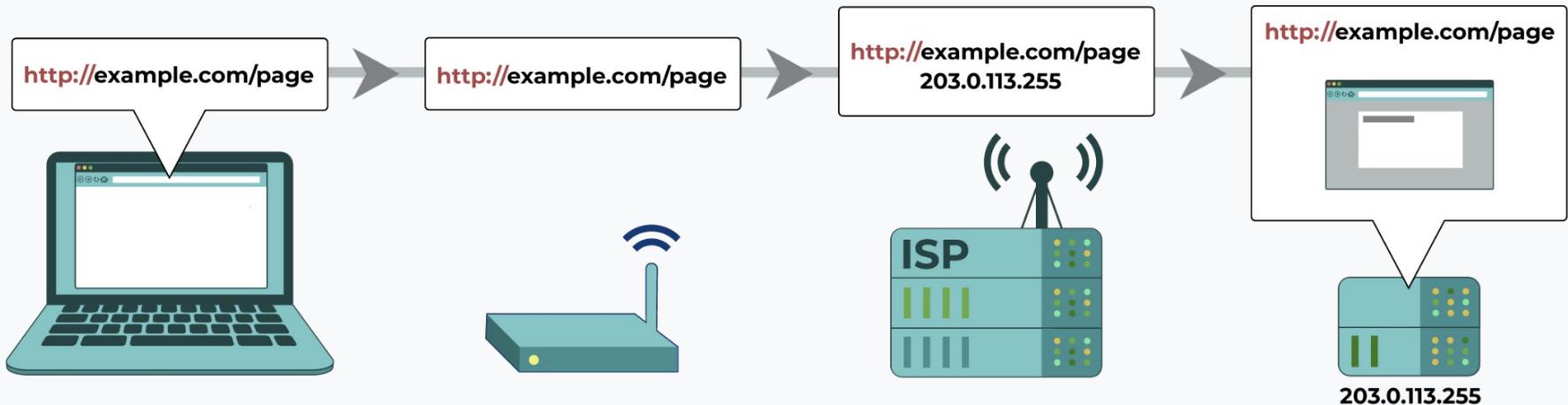
[https://en.wikipedia.org/wiki/Onion\\_routing](https://en.wikipedia.org/wiki/Onion_routing)

# Why are we learning about Tor?

- Brings together many concepts from the course
  - Encryption, anti-censorship and NIDS evasion, privacy, anonymity, etc.
- A basic network security tool that many people use for many different things
  - i.e., a “critical infrastructure”

# Introduction to Tor

# Connecting through HTTP



*Image source: eff.org*

# Connecting through HTTPS

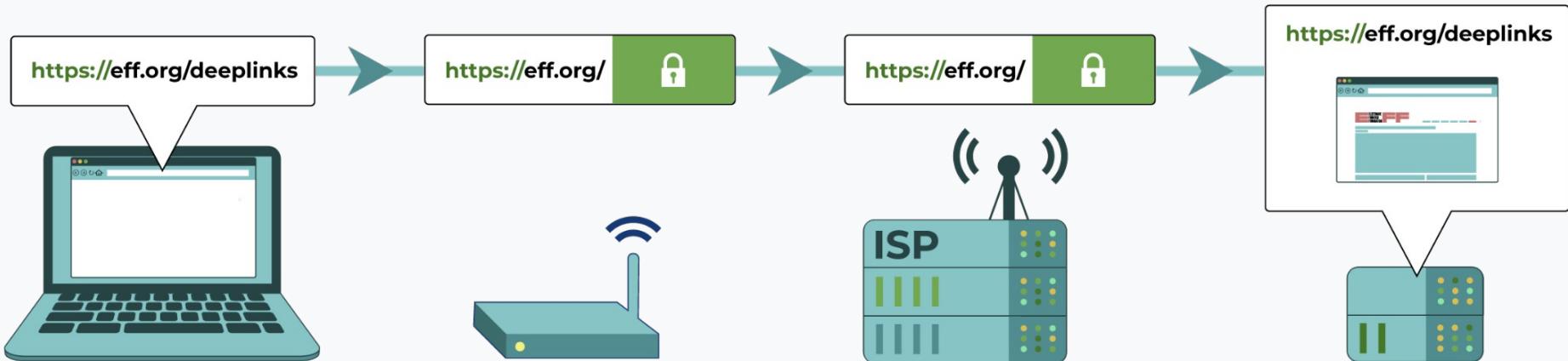


Image source: eff.org

# Connecting through VPN

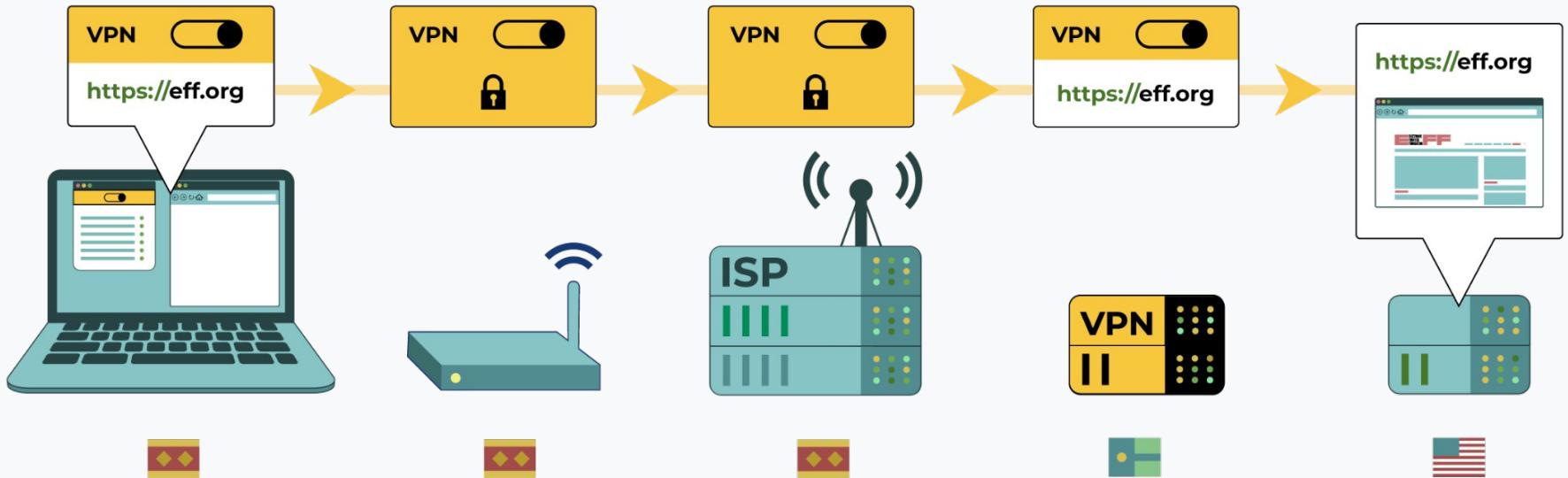


Image source: eff.org

# Connecting through Tor

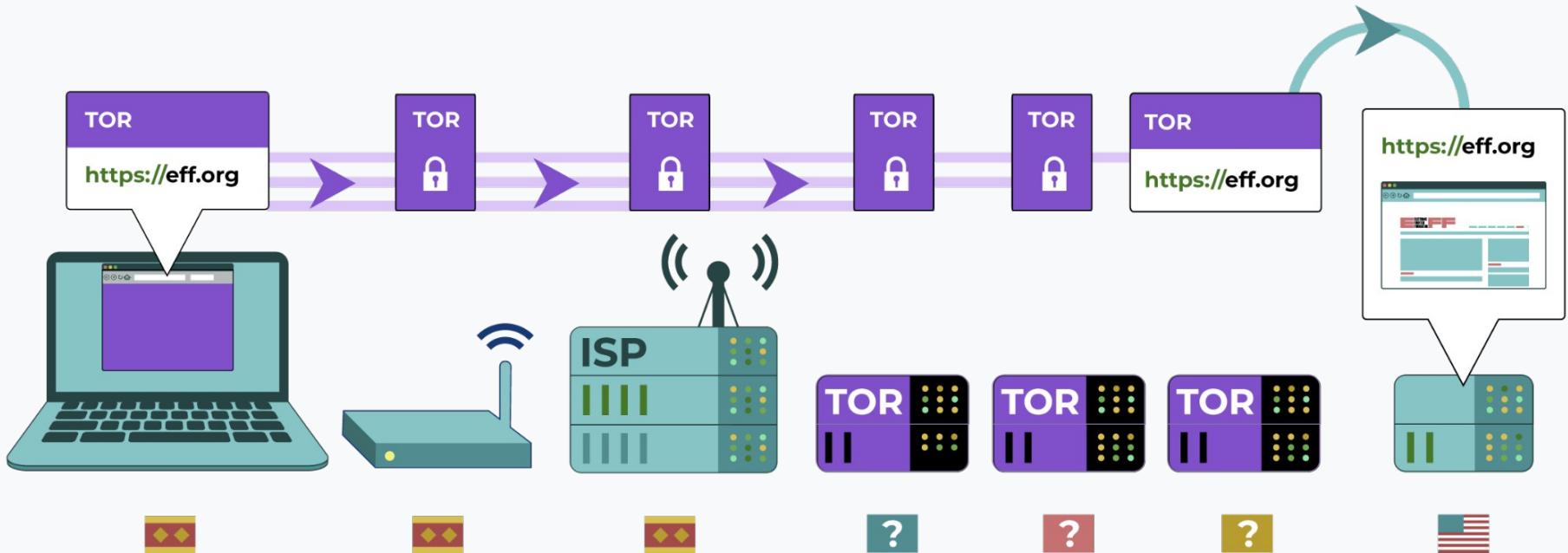


Image source: eff.org

# Who can see your activity through HTTPS and what can they see?

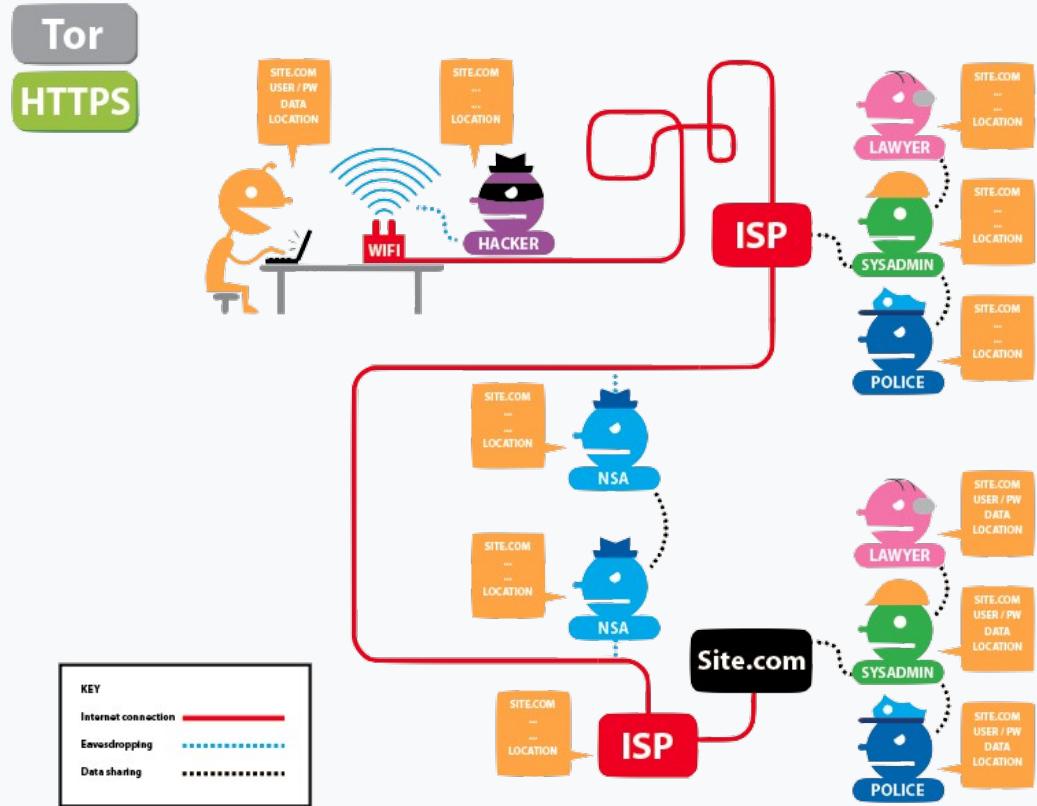


Image source: eff.org

# Who can see your activity through Tor and HTTPS and what can they see?

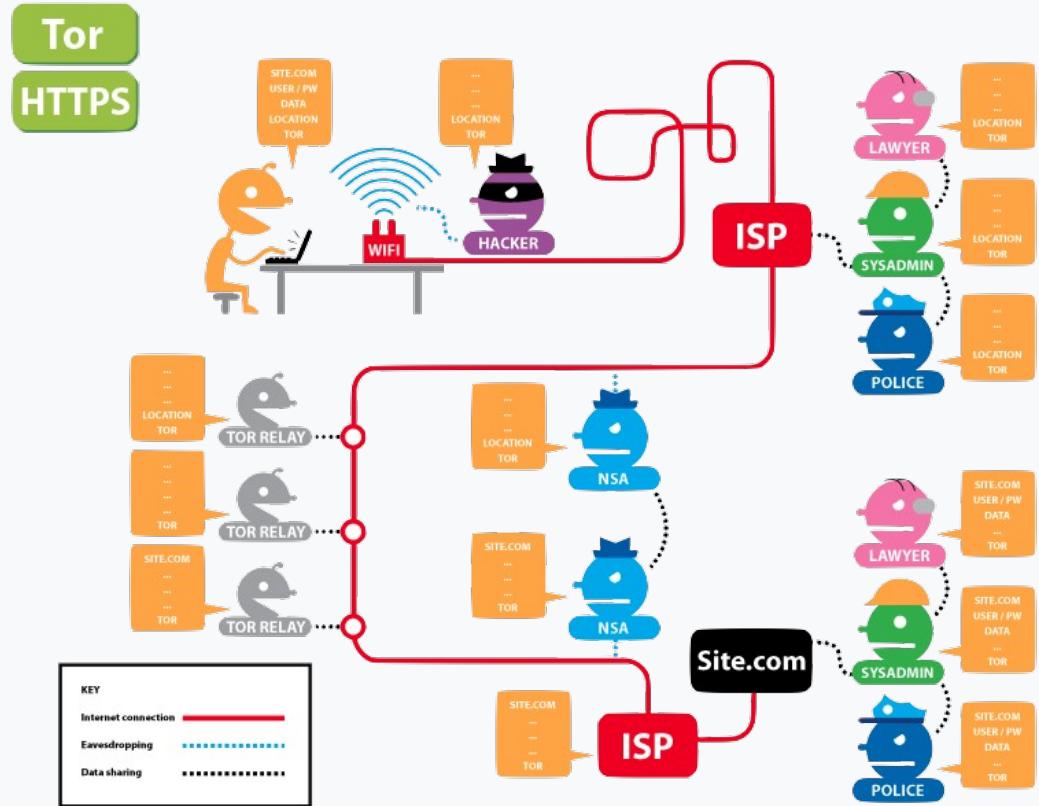
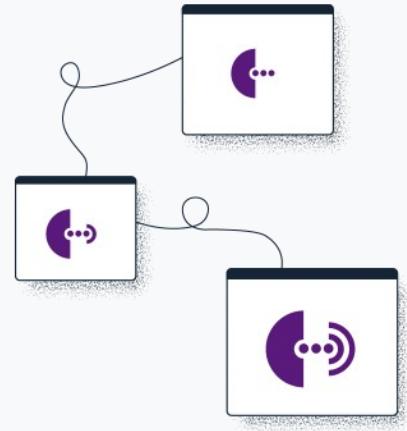


Image source: eff.org

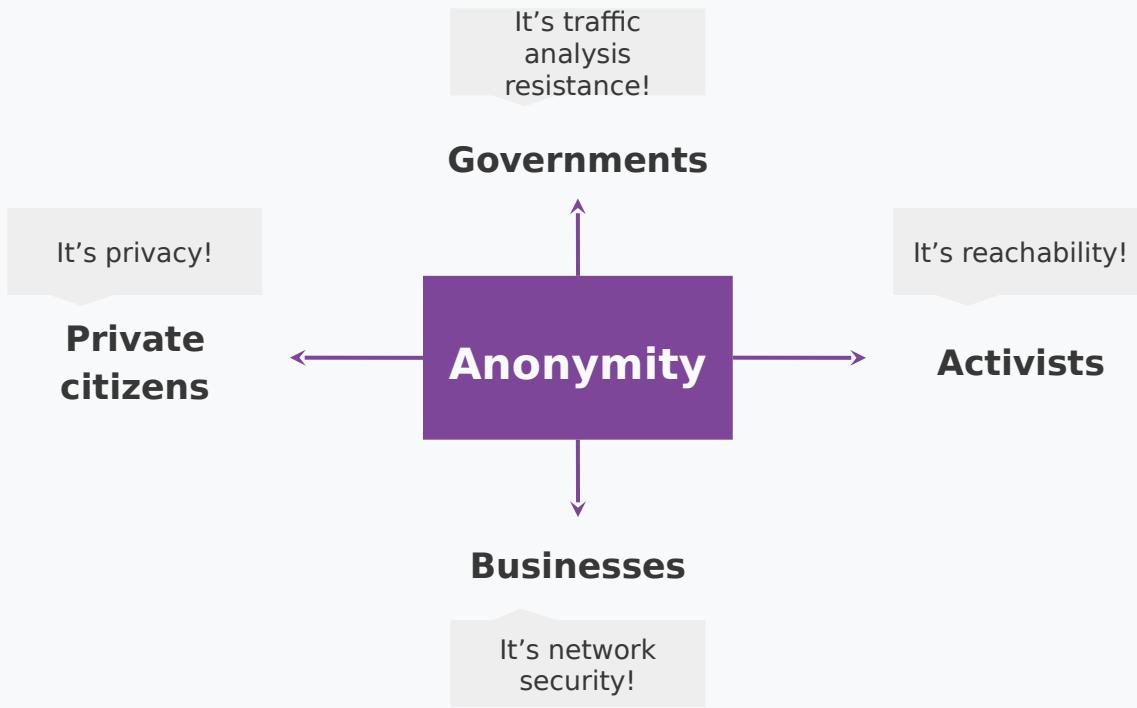
# Different ways of defining Tor

- Tor ⇒ free software created at NRL starting 2001/2.
- Tor ⇒ an open network of ~9,500 nodes – anyone can join!
- Tor ⇒ a browser that connects you to the Tor network.
- Tor ⇒ a US non-profit formed in 2006.
- Tor ⇒ a community of volunteers, researchers, developers, trainers, advocates from all over the world.



# Fighting the Internet's original sins

- It's Tor (not capitalized).
- The goal is to have a way to use the internet with as much privacy as possible:
  - a. by routing traffic through multiple nodes; and
  - b. by encrypting traffic multiple times - hence the term “onion routing”.
- Tor provides **anonymity**, which mitigates against both surveillance and censorship.



..

# We kill people based on metadata

**Director of the NSA and CIA**  
General Michael Hayden



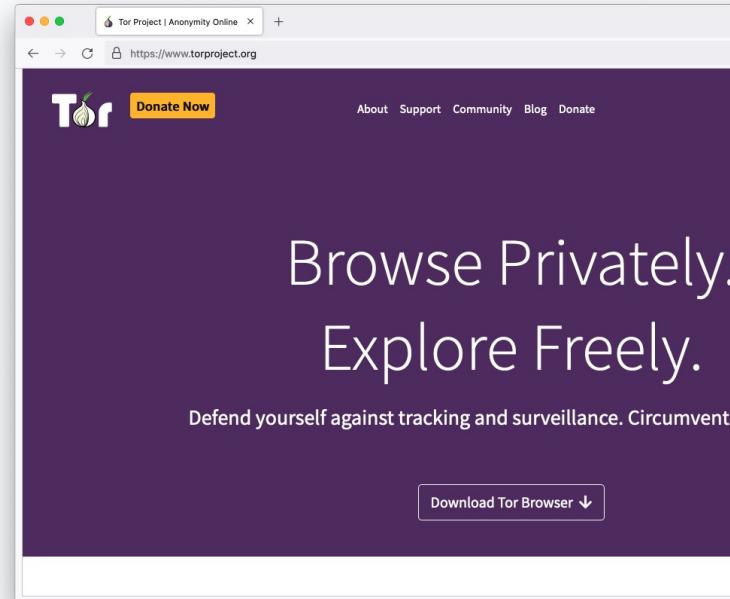
# Two sides of the same coin

- Censorship and surveillance go hand-in-hand.
- In order to **block** access to an online service, censors need to **spot** when users want to access said service.
- Anonymity grants protection from surveillance and censorship.



# What is Tor Browser?

- Just like any other browser (Chrome, Firefox, Safari, Yandex) except it does not expose traffic.
- Traffic is encrypted and bounces through three random volunteer-run nodes called **relays**.
- When using Tor Browser, we don't know who you are or what you're visiting.



## Tor Browser on Android

Developed by the Tor Project

<https://www.torproject.org/download/>



## Onion Browser on iOS

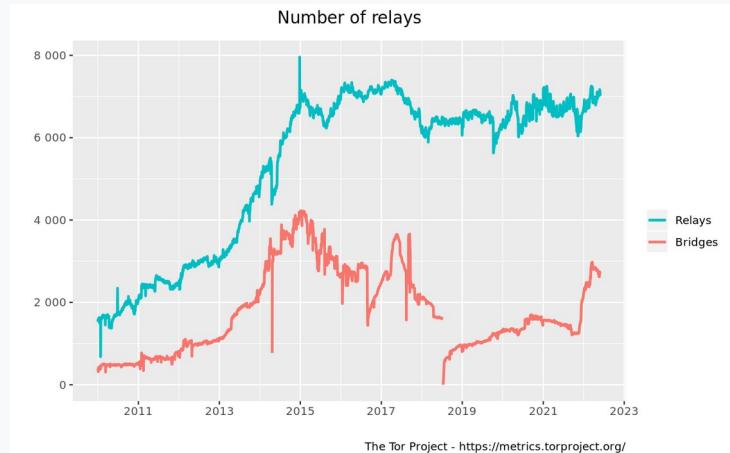
Developed by the Guardian Project

<https://onionbrowser.com/>



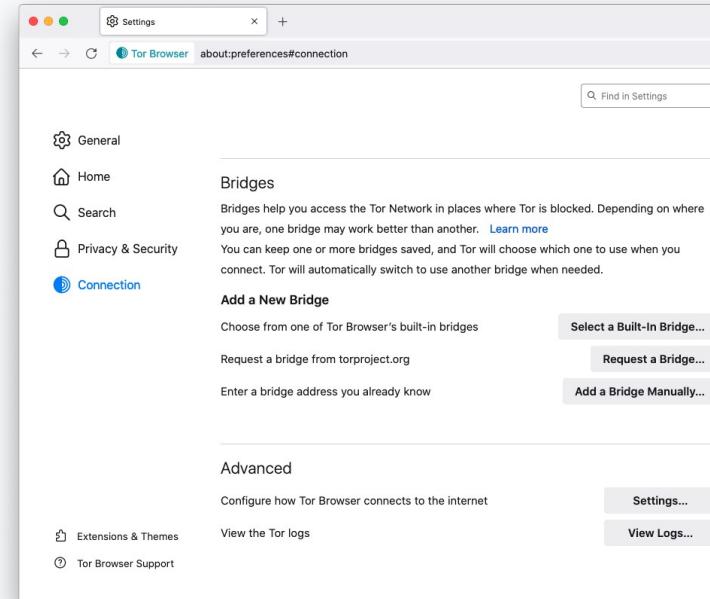
# A growing network of relays

- Tor relays and bridges are run by volunteers from around the world, including individuals, NGOs, and companies.
- They form the backbone of the Tor network.
- Today we count: 7000+ relays and 2660+ bridges.



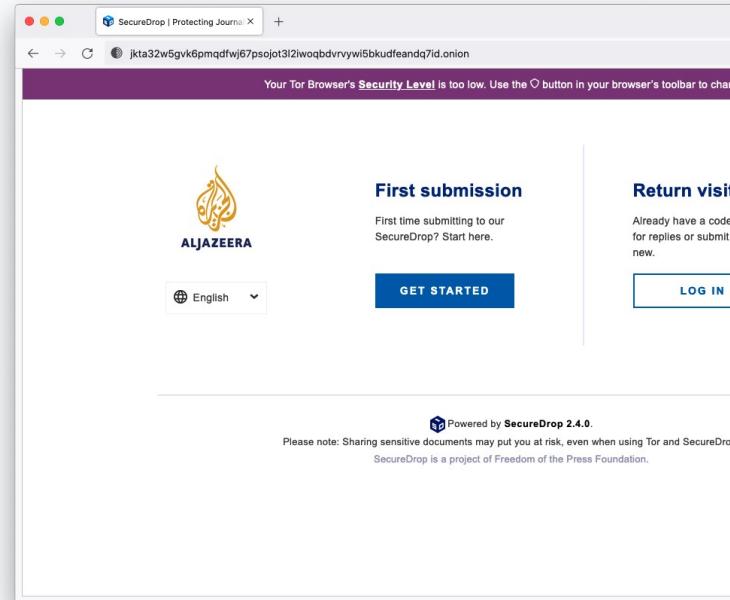
# Bypassing censorship of the Tor network

- Direct access to Tor may be blocked by some Internet Service Providers and governments.
- Tor Browser includes circumvention tools for getting around these blocks called bridges.
- Bridges are relays that are private and harder to block: <https://bridges.torproject.org/>



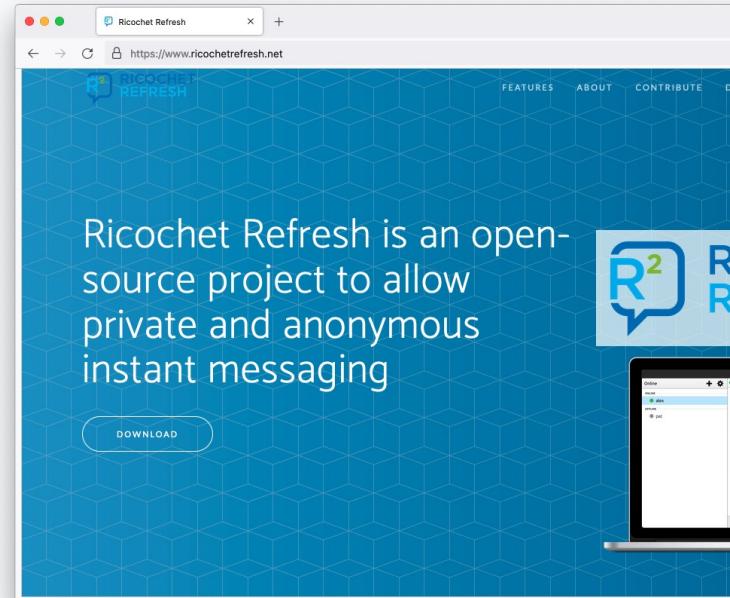
# Secure whistleblowing

- SecureDrop and GlobalLeaks are tools for whistleblowers to communicate securely with journalists.
- Newsrooms around the world have set up their own whistleblowing platforms to receive leaks securely.



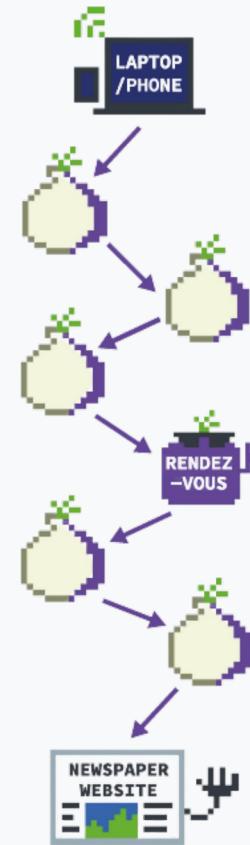
# Anonymous peer-to-peer messaging

- Ricochet Refresh is an instant messenger that routes all messages through Tor.
- Nobody knows who you're talking to, or what you're talking about.
- Supported by Blueprint for Free Speech:  
<https://www.blueprintforfreespeech.net/>



# Introduction to Onion Services .onion)

- Onion Services are online services that are only available through the Tor network.
- An Onion Service connects to a rendez-vous node/relay inside the Tor network; and the user wanting to connect to it does the same.
- As a user, you never leave the Tor network when visiting an Onion Service.
- Onion Services provide end-to-end encryption: both visitor and website use Tor (without HTTPS).



# Visiting the Intercept's site on Tor Browser vs. visiting the Intercept's onion service

Site information for theintercept.com

Connection secure

Tor Circuit

- This browser
- Canada 198.50.238.128 **Guard**
- United Kingdom 54.36.166.86
- Canada 209.209.9.109, 2602:ffd5:1:222::1
- theintercept.com

Your **Guard** node may not change. [Learn more](#)

New Circuit for this Site

Site information for  
27m3p2uv7igmj6kvd4ql3cct5h3sdwrsajovkkndeufumzyfhlfelv4qd.onion

Connection secure

Tor Circuit

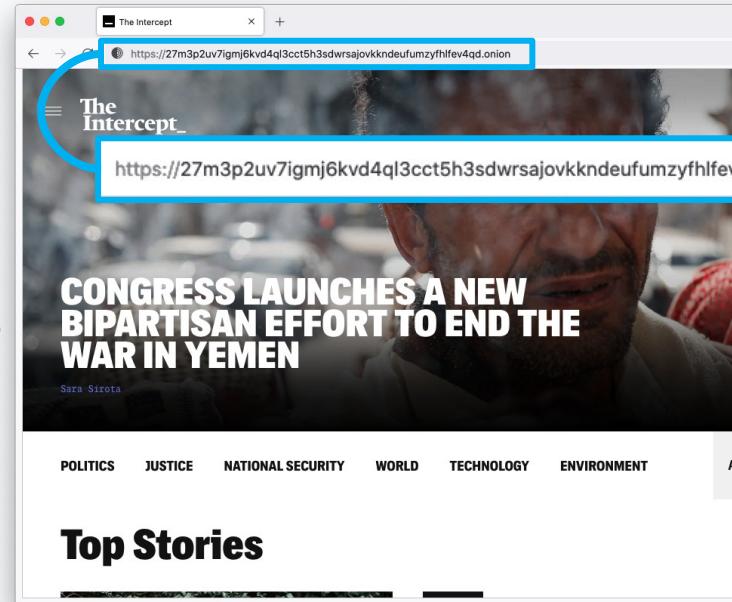
- This browser
- Canada 198.50.238.128 **Guard**
- Germany 89.58.4.238, 2a03:4000:5e:d48:946a:a4ff:fe2a:5f03
- Netherlands 5.255.97.133
- Relay
- Relay
- Relay
- 27m3p2u...fev4qd.onion

Your **Guard** node may not change. [Learn more](#)

New Circuit for this Site

# .onion addresses

- Just like any other website, you need to know the address of an onion service in order to reach it.
- The .onion address is automatically generated, so there is no need to purchase a domain.
- An onion address is a string of 56 random letters and numbers followed by ".onion".



# Censorship resistance

- Both location and IP address of an Onion Service are hidden, making it difficult to censor or identify who runs the service. Used to be called “hidden services”.
- Tor exit nodes can block websites (rare), Onion Services never exit the Tor network.
- It's the **most censorship-resistant technology** available out there as long as the Tor network is reachable.



# Metadata obfuscation and elimination

- When you use the Tor network to browse the web you are not sending any information by default of who you are or where you are connecting from.
- The Onion Services use the Tor network to eliminate information about where they are situated.
- Using them **eliminates all metadata** that may be associated with the service otherwise.

The New York Times - Breaking X +

https://www.nytimes.com

.onion available

U.S. INTERNATIONAL CANADA ESPAÑOL 中文 SUBSCRIBE FOR \$1.50/WEEK LOG IN

Thursday, June 2, 2022 Today's Paper

# The New York Times

16°C 18° 6° S&P 500 -0.75% ↓

World U.S. Politics N.Y. Business Opinion Tech Science Health Sports Arts Books Style Food Travel Magazine T Magazine Real Estate Video

LIVE Queen Elizabeth's Platinum Jubilee 30s ago Russia-Ukraine War 24m ago

Putin's Threats Highlight the Dangers of a New Riskier

The New York Times - Breaking X +

https://www.nytimesn7cgmtshazwhfgzm37qxb44r64ybb2d3x62d2ljljsciyyd.onion

LOG IN

U.S. INTERNATIONAL CANADA ESPAÑOL 中文

Thursday, June 2, 2022 Today's Paper

# The New York Times

70°F 87° 61° Dow -0.54% ↓

World U.S. Politics N.Y. Business Opinion Tech Science Health Sports Arts Books Style Food Travel Magazine T Magazine Real Estate Video

LIVE Russia-Ukraine War 30s ago Queen Elizabeth's Platinum Jubilee 10m ago

Putin's Threats Highlight the Dangers of a New Riskier

# Popular Onion Services

The  
Intercept\_

BBC

PROPUBLICA

The New York Times

The  
Guardian

 RISEUP

facebook



 brave

Proton

 FREEDOM OF THE  
PRESS FOUNDATION

Privacy  
Matters.

 FRONT LINE  
DEFENDERS

 debian

 SECUREDROP

# Benefits of Onion Services

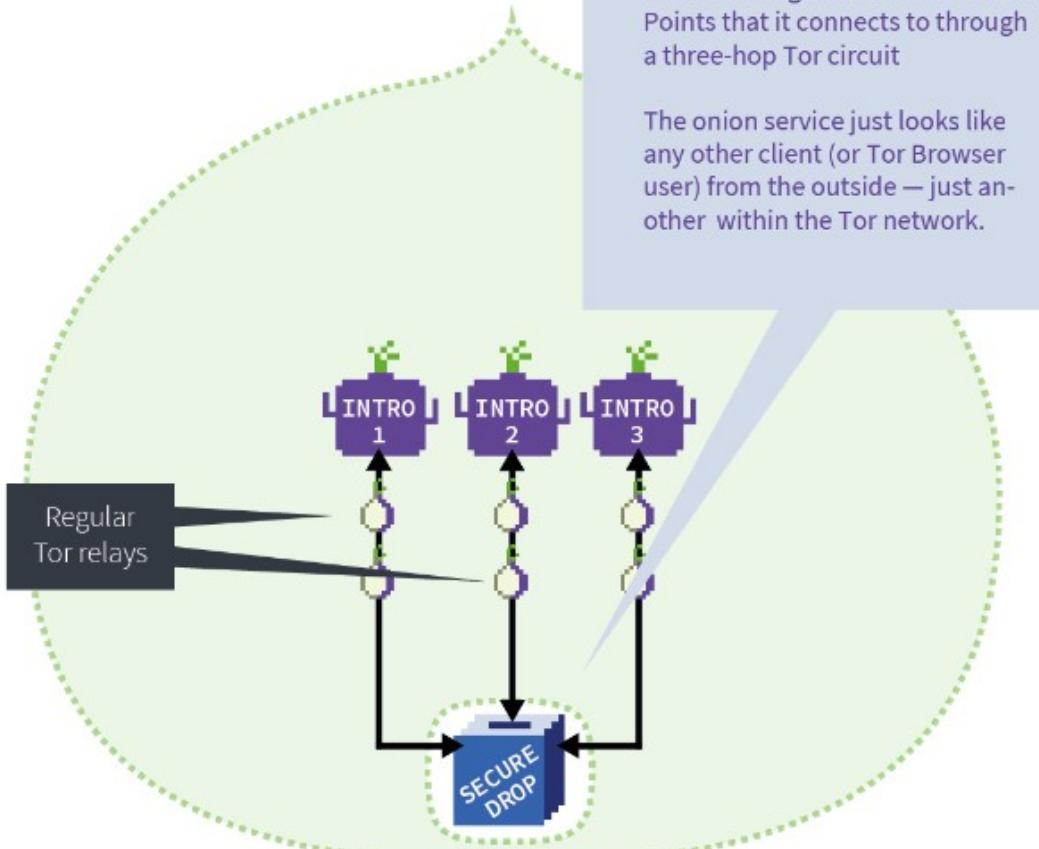
1. Censorship resistance as long as the user has access to Tor.
2. End-to-end encryption between user and website.
3. Contributing to the decentralization of the web.
4. Tor network sustainability.
5. Protection of sources, whistleblowers, and journalists.
6. Opportunity to educate users about privacy by design.
7. Metadata obfuscation and elimination.

# Comparison

	Regular Website	Website Over Tor	Onion Service
Censorship Resistance:	<b>Poor</b> Website can easily be censored	<b>Good</b> Censorship still possible via exit nodes	<b>Very good</b> Accessible as long as Tor is reachable, address not censorable
Privacy Safeguards:	<b>Very poor</b> Minimal safeguards: HTTPS, no tracking, hosting jurisdiction, etc.	<b>Good</b> Data correlation is not an eliminated risk	<b>Very good</b> End-to-end encryption for user and service, anonymity for both
Metadata Elimination:	<b>Poor</b> Data about online activity recorded by websites and entities passing traffic	<b>Good</b> Data about online activity can be recorded by website if user logs in and identifies themselves	<b>Very good</b> Metadata logging eliminated on both ends, but website can record data if user logs in
torproject.org			

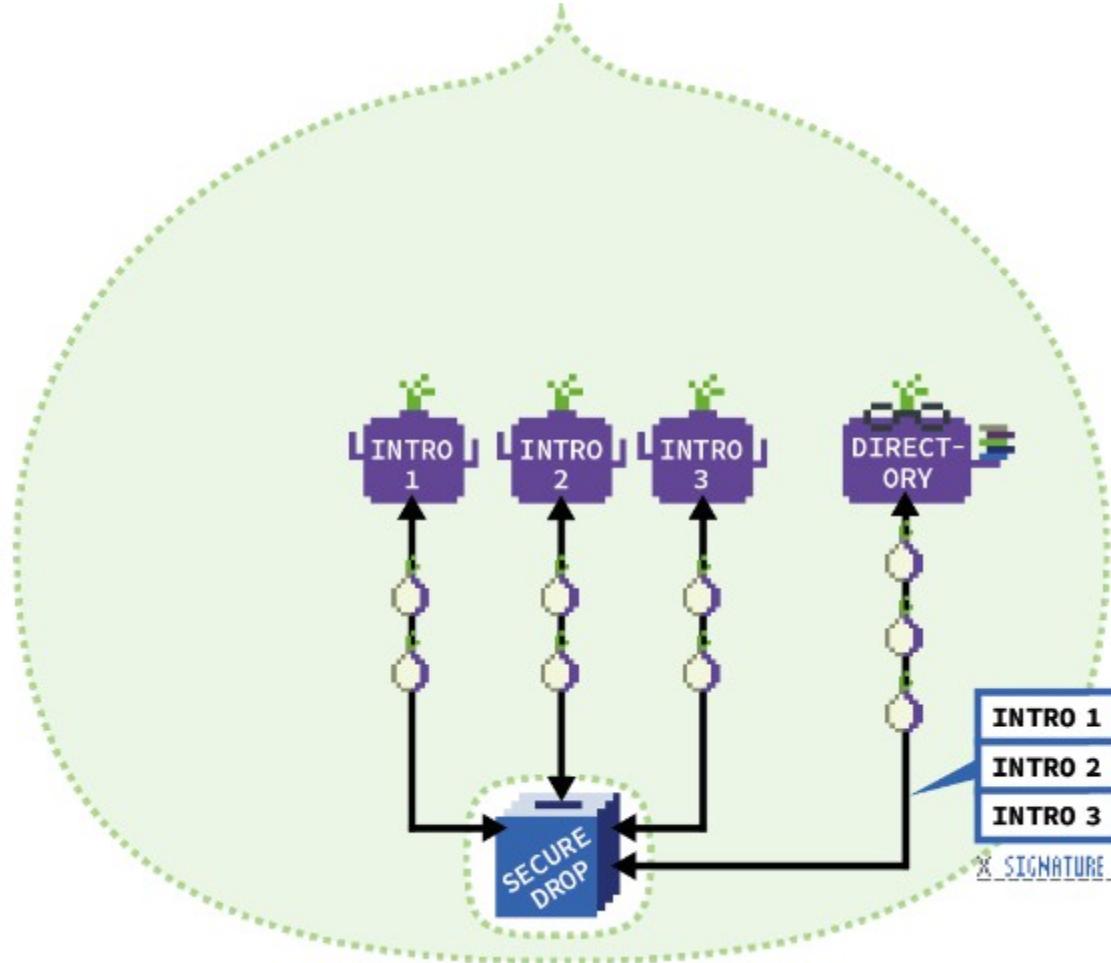
Now some slides from...

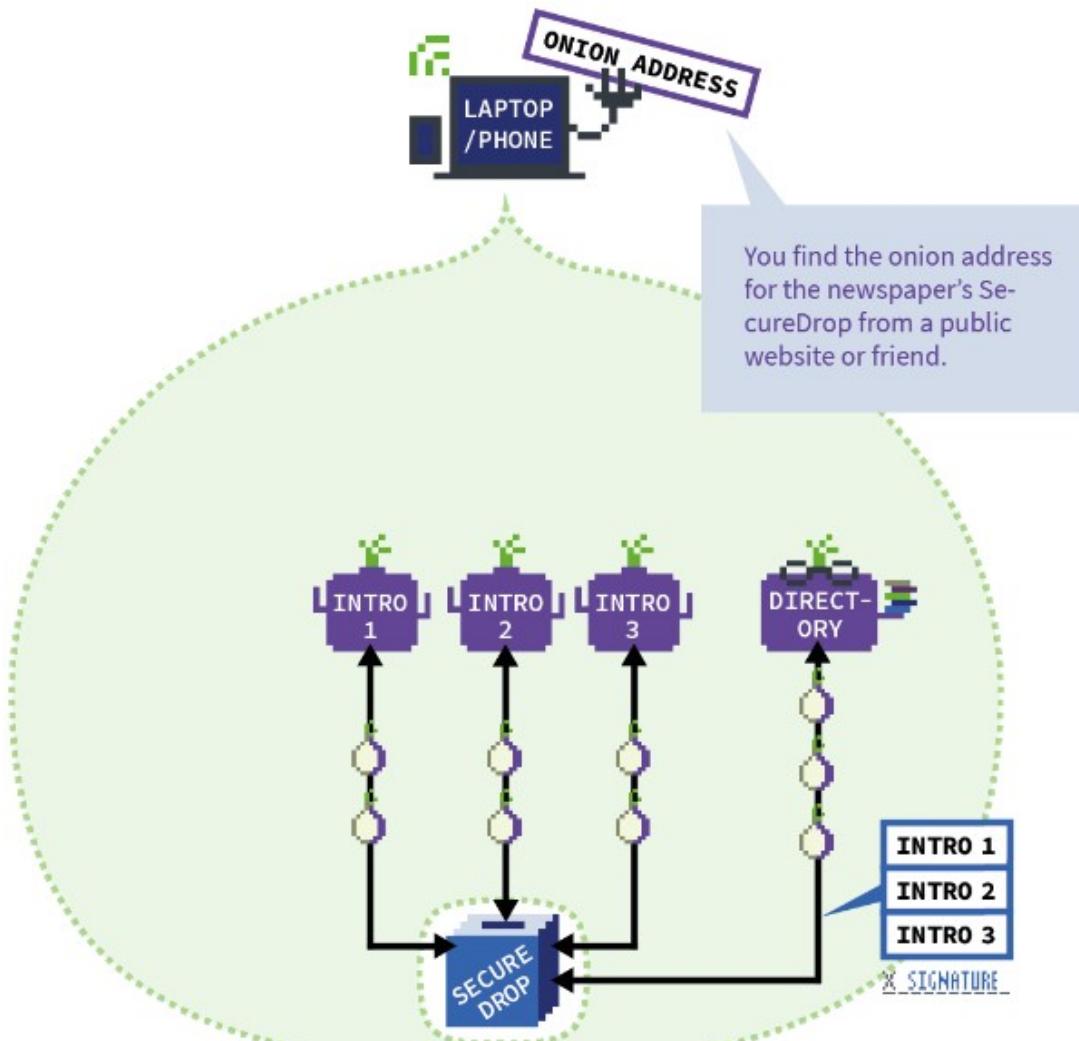
<https://community.torproject.org/onion-service/overview/>

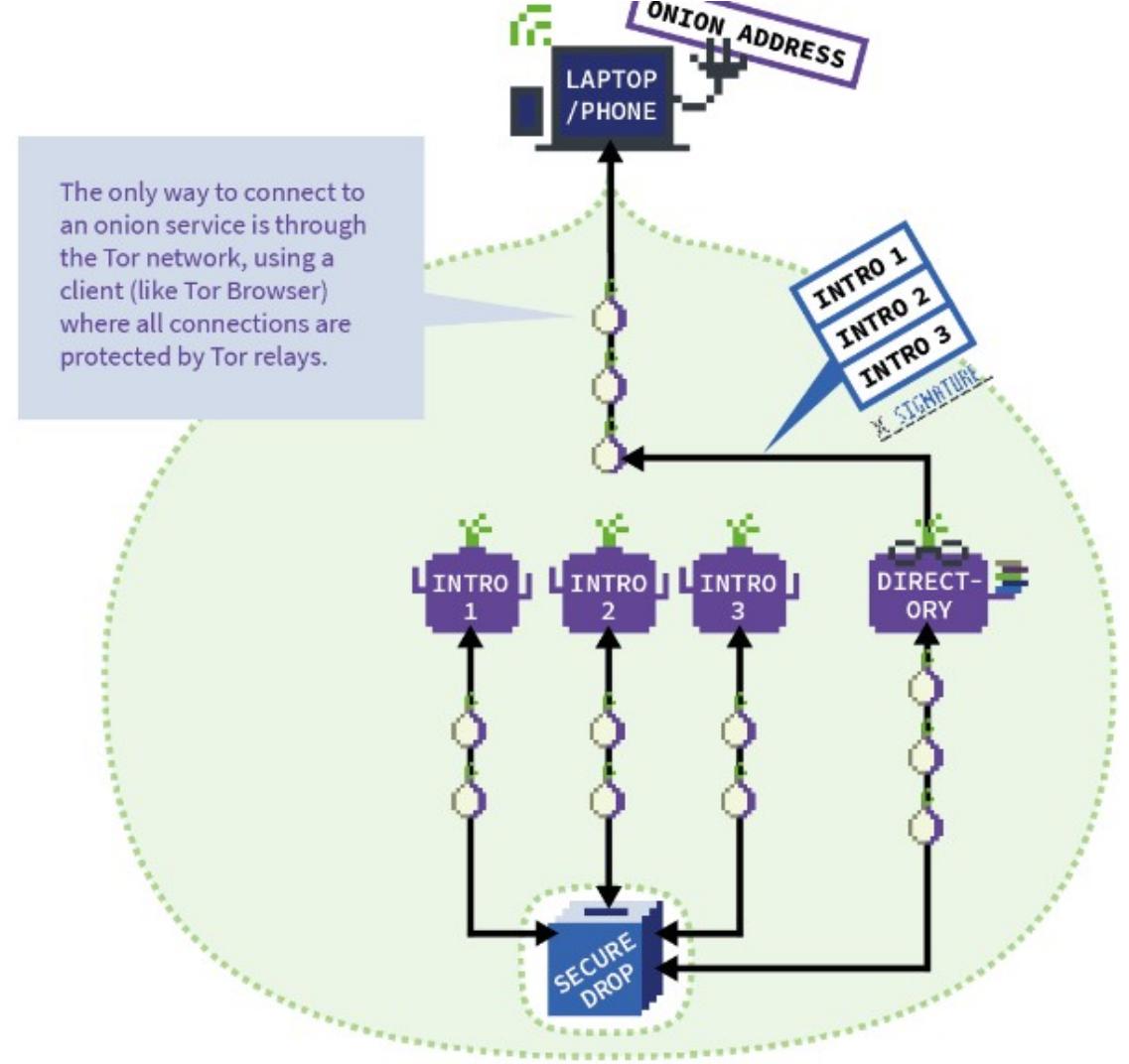


The onion service (SecureDrop) hides and protects itself behind the Tor network by only allowing access through three Introduction Points that it connects to through a three-hop Tor circuit

The onion service just looks like any other client (or Tor Browser user) from the outside — just another within the Tor network.

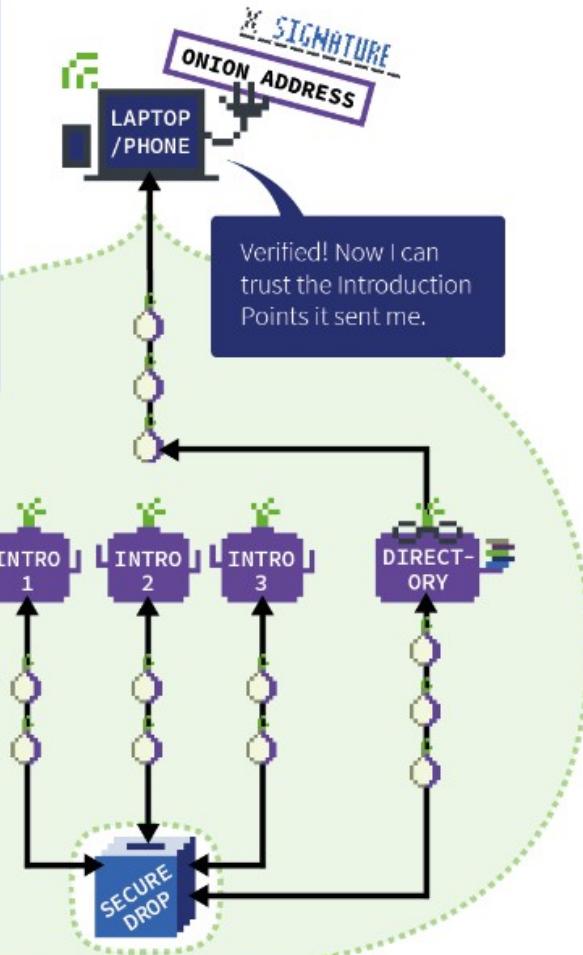


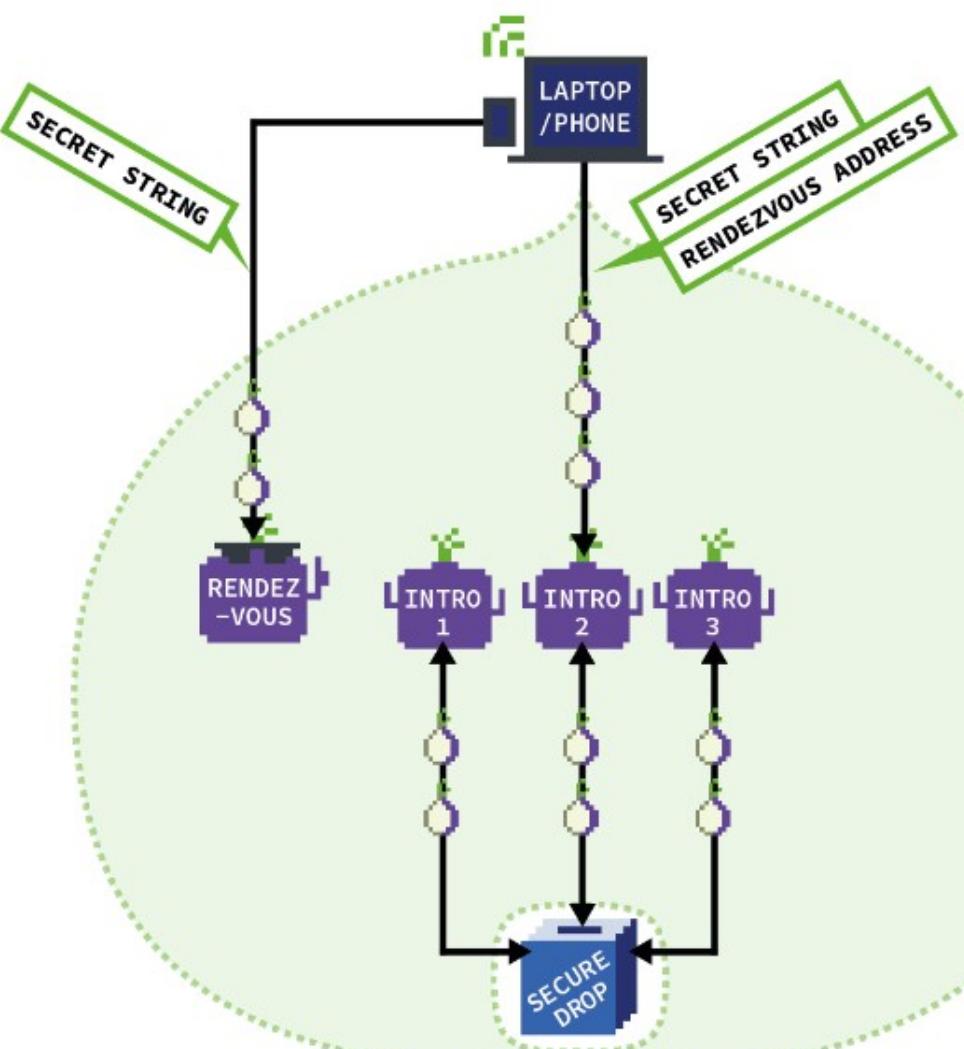


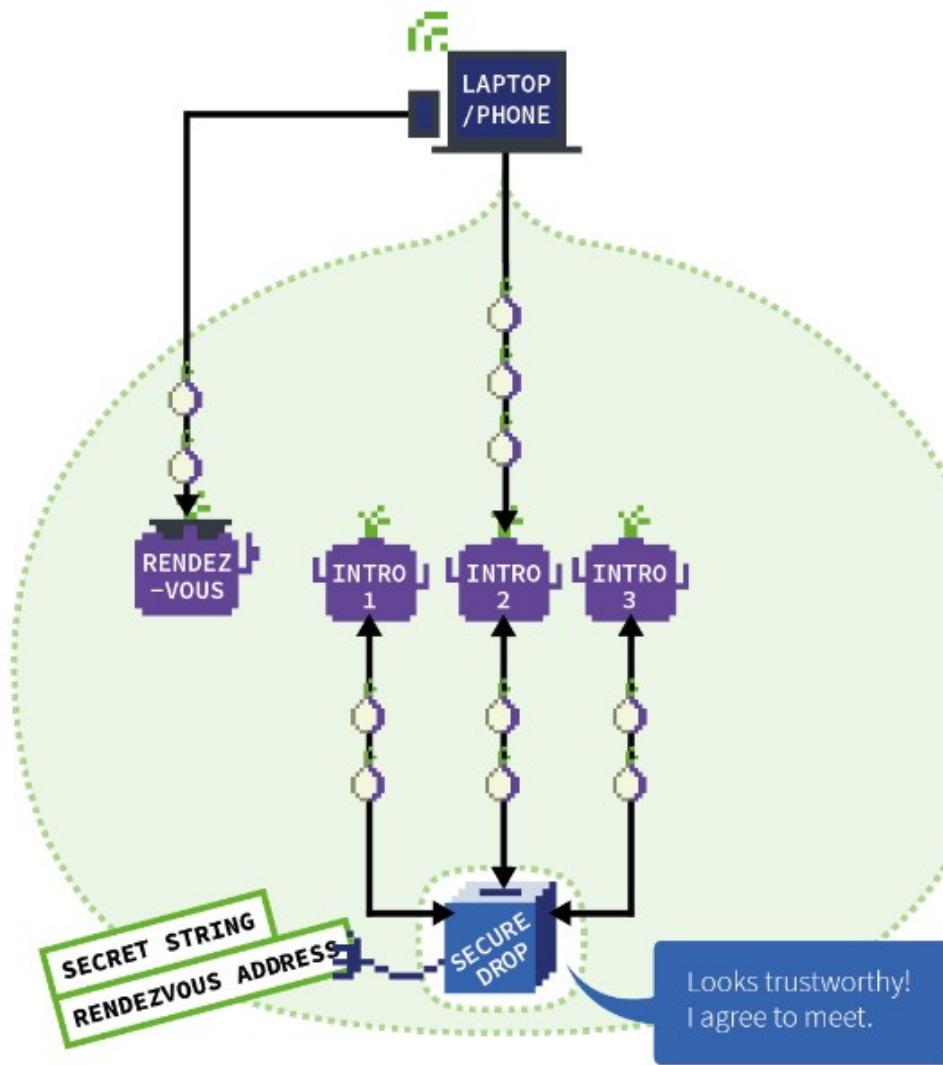


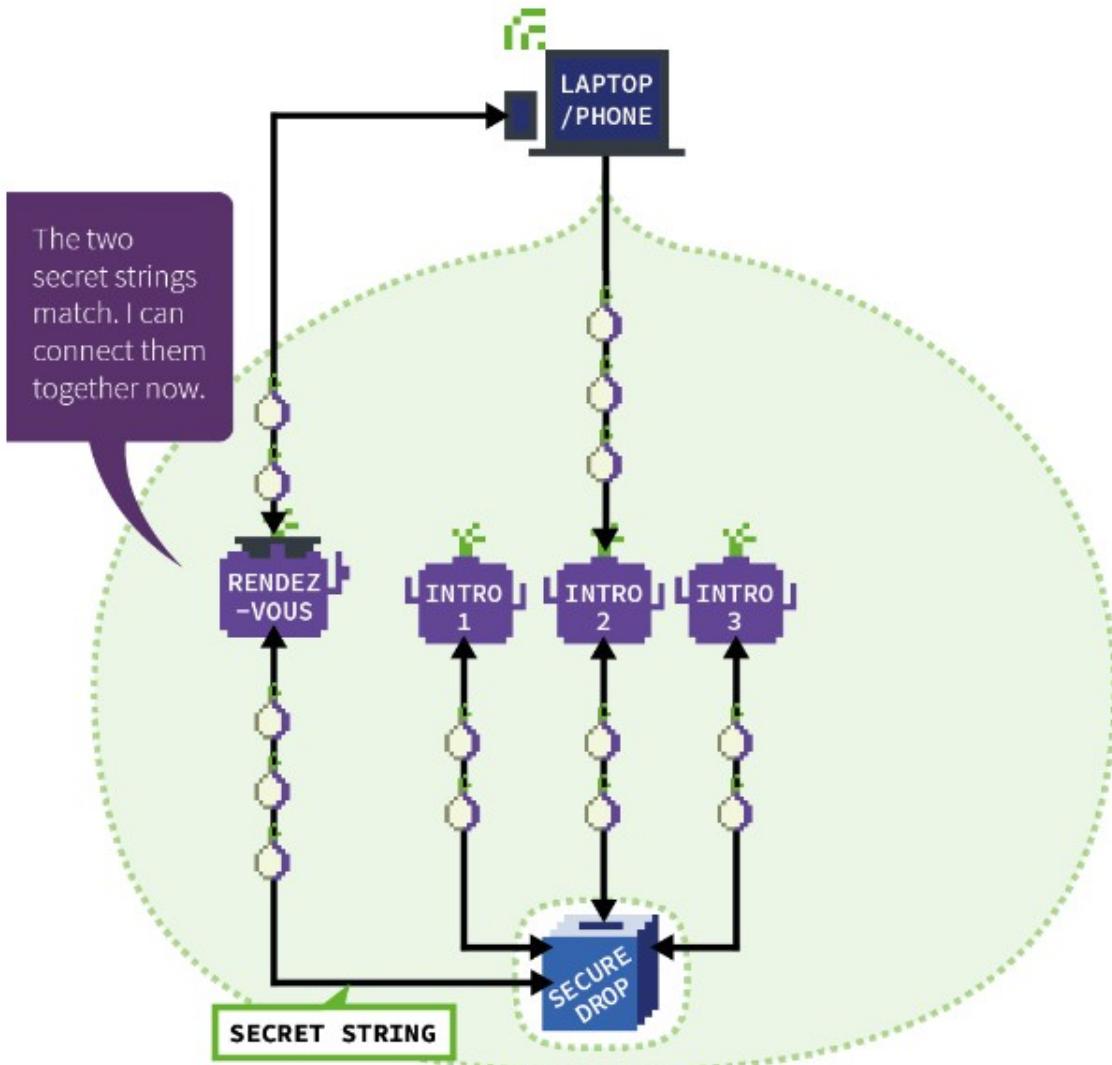
The onion address is also a cryptographic key that can be used to verify the authenticity of the received descriptor. That's why it's so long and random!

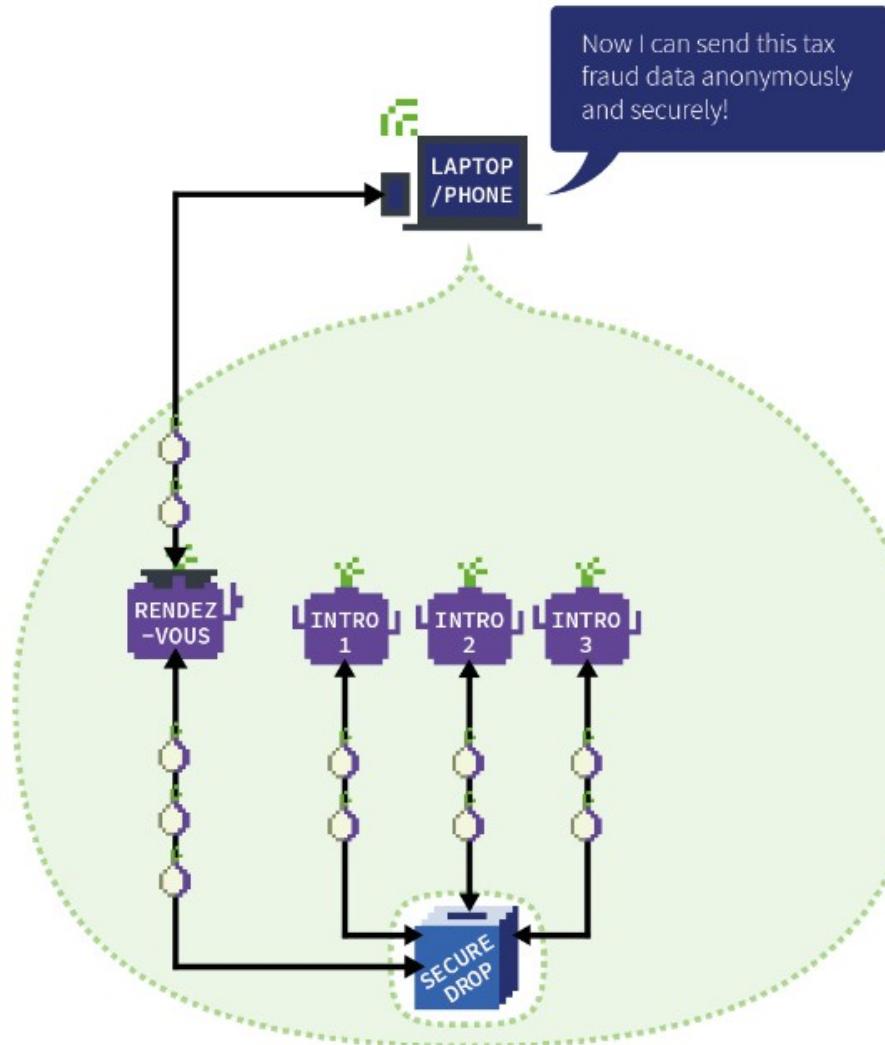
E.g. ProPublica SecureDrop  
lvtu6mh6dd6ynqcxtd2mse-qfkm7g2iuxvjobbyzpgx2-jt427zvd7n3ad.onion



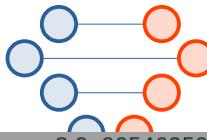








Switching gears... Why do you need something like Tor, a VPN, or ShadowSocks to evade censorship?



3 0.005462504	10.155.30.179	18.205.53.136	TCP	74 40186 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS
8 0.067175913	18.205.53.136	10.155.30.179	TCP	74 443 → 40186 [SYN, ACK] Seq=0 Ack=1 Win=2684
24 0.195979757	10.155.30.179	18.205.53.136	TLSv1.2	104 Application Data
26 0.196261196	18.205.53.136	10.155.30.179	TLSv1.2	104 Application Data
28 0.196849481	10.155.30.179	18.205.53.136	TLSv1.2	108 Application Data
22 0.195647004	18.205.53.136	10.155.30.179	TLSv1.2	144 Application Data
18 0.134217638	10.155.30.179	18.205.53.136	TLSv1.2	192 Client Key Exchange, Change Cipher Spec, Enc
21 0.195646933	18.205.53.136	10.155.30.179	TLSv1.2	270 New Session Ticket, Change Cipher Spec, Enc
25 0.196261086	18.205.53.136	10.155.30.179	TLSv1.2	352 Application Data
11 0.067795926	10.155.30.179	18.205.53.136	TLSv1.2	624 Client Hello
19 0.134536166	10.155.30.179	18.205.53.136	TLSv1.2	656 Application Data, Application Data
10 0.067787461	10.155.30.179	18.205.53.136	TCP	1264 40186 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len
14 0.131083829	18.205.53.136	10.155.30.179	TLSv1.2	1264 Server Hello
16 0.131154339	18.205.53.136	10.155.30.179	TLSv1.2	3202 Certificate, Server Key Exchange, Server He

‐ Extension: server\_name (len=19)

Type: server\_name (0)

Length: 19

‐ Server Name Indication extension

Server Name list length: 17

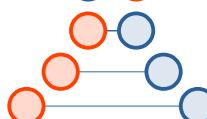
Server Name Type: host\_name (0)

Server Name length: 14

Server Name: boredpanda.com

‐ Extension: Reserved (CPEACE) (len=1)

70	59 7f 51 d5 11 9b ec 81 33 ba 62 5b f0 d5 54 35	Y·Q..... 3·b[··T5
80	80 24 34 4c 3f 3f e7 42 99 35 fd 08 01 f3 77 db	·\$4L??·B ·5·...W·
90	3d 46 d2 b5 96 ce cf c5 45 e9 e2 2b 00 1d 00 20	=F..... E·+....
a0	d2 2c b8 f7 90 4f 37 6b 36 5a 09 32 ab 9a d7 52	·,·..07k 6Z·2···R
b0	f7 30 d3 a5 f7 a0 60 3a e4 0f a1 e2 3d e6 d5 61	·0.....` : ..-=..a
c0	00 00 00 13 00 11 00 00 0e 62 6f 72 65 64 70 61	..... ·boredpa
d0	6e 64 61 2e 63 6f 6d 1a 1a 00 01 00	nda.com ·....



...

zz.....#...3....zz.....A.. t....l...

...\.?n..Sy.nTo.z.2..L..#t...>.P+....d%..2Ar....q..Q.Tx?...\.g...\$e..A...O.k.RE..%

.....wg..r./1....T.kq.....0x\$..x....."e .).s(F.

...dw\.\*....hH.+v....FY.I.2.....[..J.2.3.I.|.@s.le..8..76]...\$.U...?.Cr.H..o..&.(...{.t...

..... ...c..k1.,qM.....Y.Ch.6..l.".....\$.q.....M5..ZWM.\.bKEjo.4^.....+..8.....AQ....Av.p..@.....&....d=..Y.E..c.Y..y..y.wNbd:.....:+1.+7a`

{.....+.....P2.k..LOc...(f.8.QR/X.A.U;c.),'....\_T

.1.Fv.M ..e..X .'8X.....y..v9.....zWF.....L..1cy"....%.HY>...#.iC..+\$#0.\$..+....@.[.....>...4,X.r.....Sq.f....\%"x

....7Z.w;...o..q.....u.z/...F..WY8.....+.^%"G#k...{..U".j..M.....49..2x cng1.P.v.."J..V\$6.l:Aq...v.x

c.l...&...` ..t..@A..M..?!m;C..!r.EF....)W....j..2..V.-....&....M.,K..F.4R.5..{z....G.tQ..z...M#:%.e...B....K+A"B.....&....W]"...)....ru....A{.1syn..1.

\i..P.`j2T'.w....8.\.V0..[C..t..l..C)..QF....&..+..Th+.....

.

.+uIG....7Pk

F,,@.IEL<.....7..Q..=?Q..D(.x.=...w..s..@n+t..H.P

./ ...D.....Y..0a..... ...j..:l.V|.q.Z='..|D.U.A\$R..yye..!.c.3[...3..,j..p..v.....H6.Q..(%..

....8.1...+@..#.6.x..x[....z..r..j!!..) ..`V..Y.Q.....3.b[..T5.\$4L??B.5....w.=F.....E..+....O7k6Z 2...R.

0....`:=..a.....boredpanda.com.....l..h....x.h.i@..:5\$..VA..}....&.....z...q.e..

}9x%....gng...../.....h2.#.....0...0.....>.tK...%..e....0

....\*H..

....0<1.0 ..U....US1.0

..U.

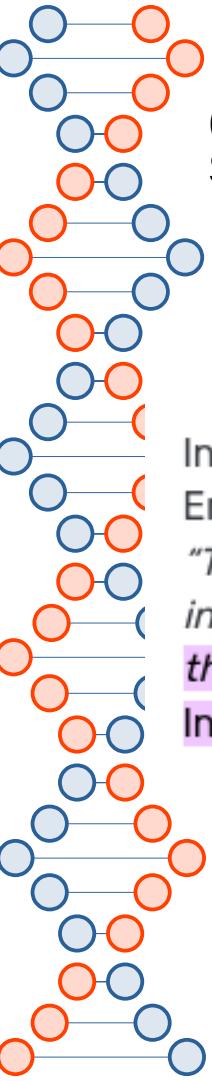
..Amazon1.0...U....Amazon RSA 2048 M030..

250301000000Z.

260330235959Z0.1.0...U....www.boredpanda.com0.."0

\* H

44



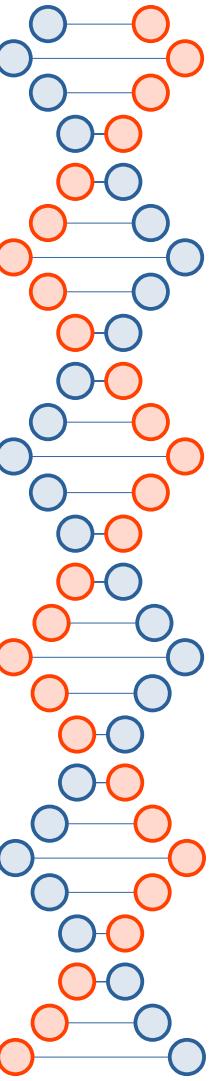
(Encrypted Client Hello, or ECH, is a way to hide the Server Name Indicator, or SNI)

In November 2024, Russia began blocking Cloudflare's implementation of Encrypted Client Hello (ECH), a privacy-focused extension of the TLS protocol. *"This technology is a means of circumventing restrictions on access to information banned in Russia. Its use violates Russian law and is restricted by the Technical Measure to Combat Threats (TSPU),"* the statement by the Russian Internet regulator read.

<https://adguard-dns.io/en/blog/encrypted-client-hello-misconceptions-future.html>

# OONI

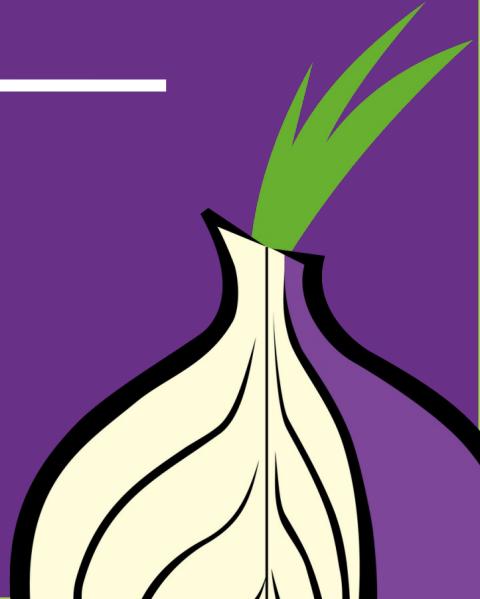
- Open Observatory of Network Interference:  
<https://ooni.torproject.org>
- Country-level reports of specific censorship tools  
in use on certain websites
- View their reports: <https://explorer.ooni.org/>
- Or use your own OONI Probe to test websites:  
available in App Store and Google Play.



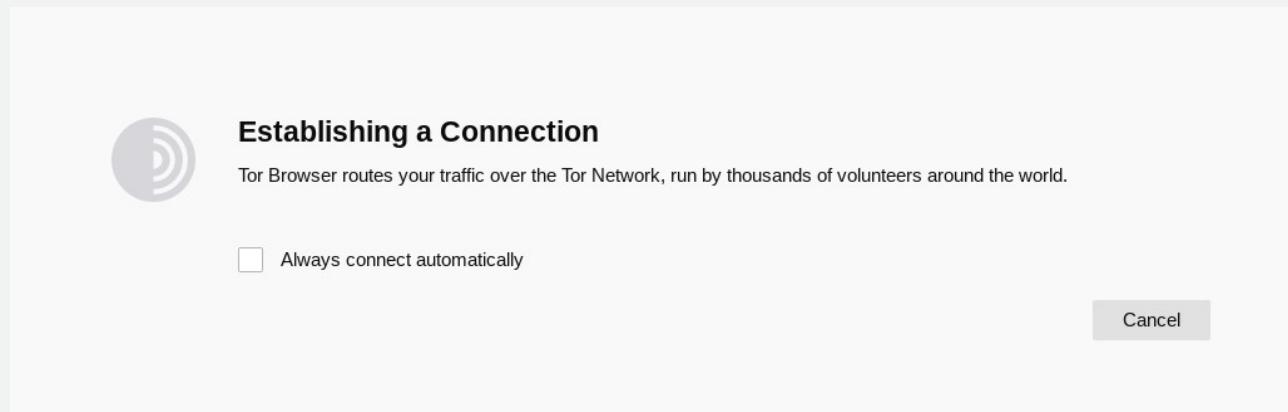
If you get past SNI filtering (*e.g.*, with a VPN or Tor), can encryption hide which apps and platforms you're using, or hide the fact that you're using a VPN or Tor?

# What do you do when Tor is blocked?

---



# I downloaded Tor Browser, but it won't connect

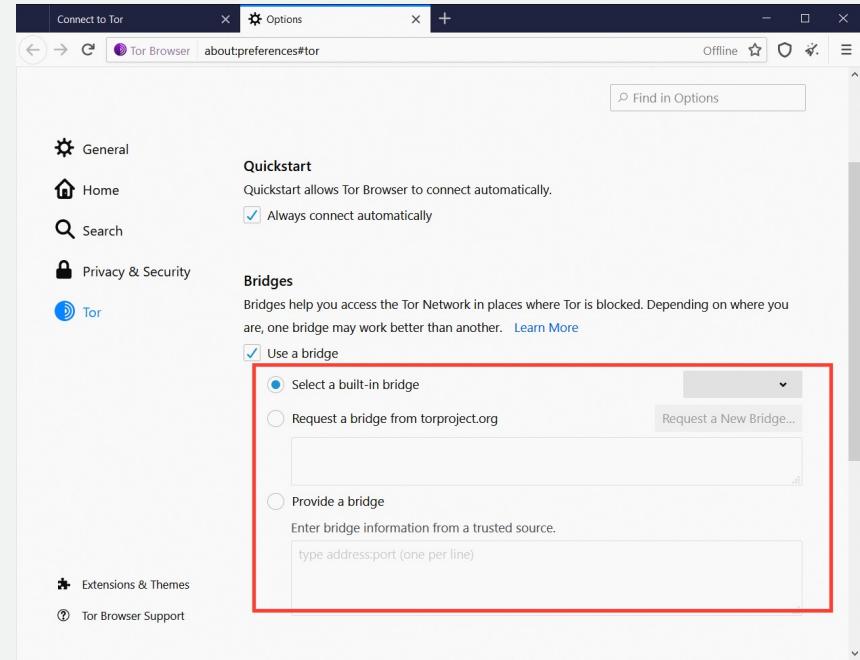
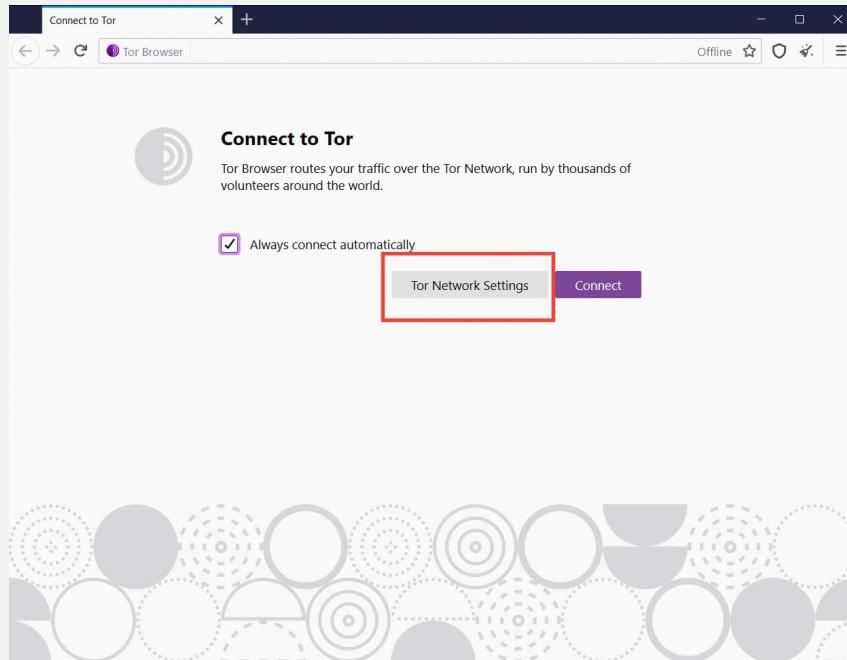


If this screen takes a long time and does not connect, you may need a bridge or pluggable transport

# Bridges and pluggable transports

- Bridges are relays that are not listed publicly
- Get bridges directly from Tor Browser (moat)
- Or from the website <https://bridges.torproject.org> or send an email to bridges@torproject.org from a Gmail, or Riseup.net account
- Or get a bridge address from a trusted person
- Pluggable transports can be used like bridges to disguise Tor traffic (also called “built-in bridges”)

# Bridges and pluggable transports



# Request a bridge

The screenshot shows the 'Options' screen of the Tor Browser. The left sidebar lists categories: General, Home, Search, Privacy & Security, and Tor. The 'Tor' category is currently selected. The main content area is titled 'Bridges' and contains the following text: 'Bridges help you access the Tor Network in places where Tor is blocked. Depending on where you are, one bridge may work better than another.' Below this is a section titled 'Use a bridge' with a checked checkbox. Underneath are three radio button options: 'Select a built-in bridge', 'Request a bridge from torproject.org' (which is selected and highlighted with a red box), and 'Provide a bridge'. A link 'Request a New Bridge...' is also visible. At the bottom of the bridge section is a text input field labeled 'Enter bridge information from a trusted source.' and a placeholder 'type address:port (one per line)'. The 'Advanced' section at the bottom is partially visible.

# Or select a built-in bridge

**Bridges**

Bridges help you access the Tor Network in places where Tor is blocked. Depending on where you are, one bridge may work better than another. [Learn More](#)

Use a bridge

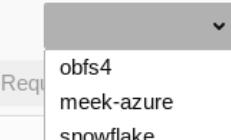
Select a built-in bridge

Request a bridge from torproject.org

Provide a bridge

Enter bridge information from a trusted source.

type address:port (one per line)



The dropdown menu contains the following items:

- obfs4
- meek-azure
- snowflake

# Pluggable transports

- **obfs4**: makes Tor traffic look random; works in many situations including China (if not, try meek).
- **meek-azure**: makes it look like Microsoft traffic; works in China.
- **snowflake**: proxies traffic through temporary proxies using WebRTC (under development).  
<https://snowflake.torproject.org>

# obfs4 = ScrambleSuit, basically

- <https://arxiv.org/pdf/1305.3199.pdf>

## **ScrambleSuit: A Polymorph Network Protocol to Circumvent Censorship**

Philipp Winter

Karlstad University

Tobias Pulls

Karlstad University

Juergen Fuss

Upper Austria University of  
Applied Sciences

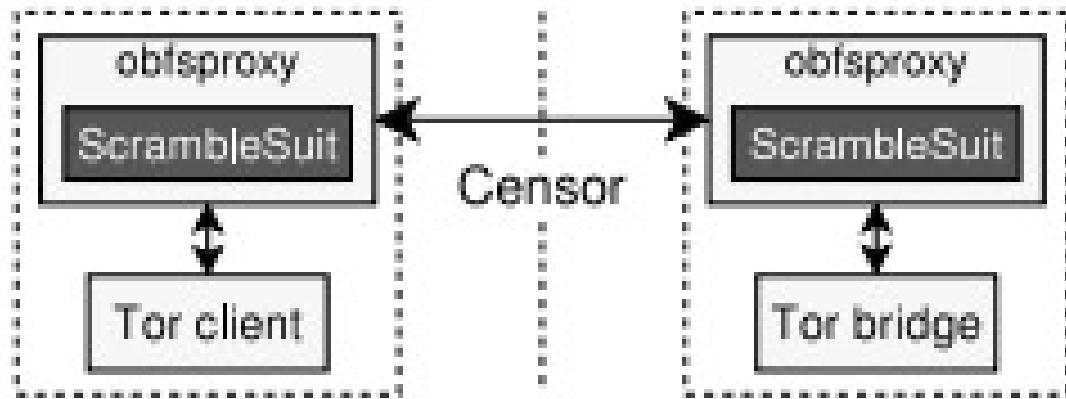
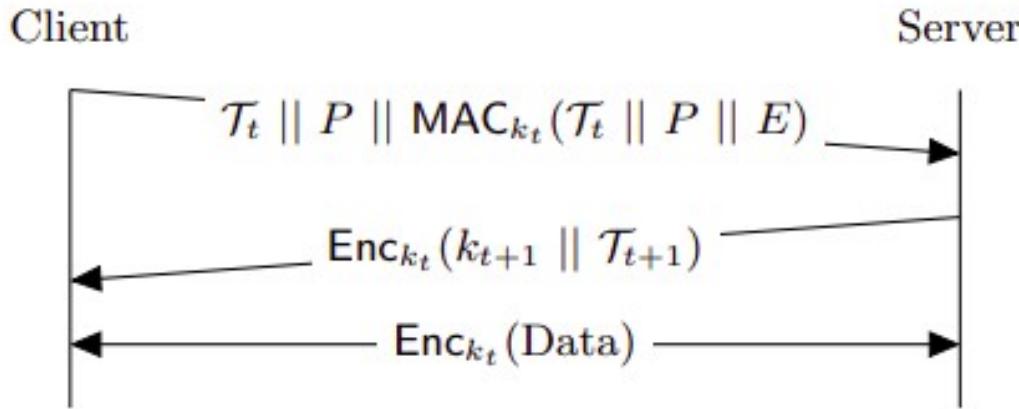


Figure 2: **ScrambleSuit** is a module for **obfsproxy** which provides a SOCKS interface for local applications. The traffic between two **obfsproxy** instances is disguised by **ScrambleSuit**.

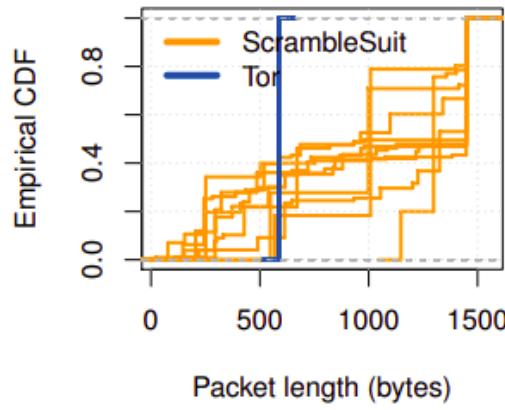


**Figure 4:** The client redeems a valid session ticket  $\mathcal{T}_t$  containing the master key  $k_t$ . The server responds by issuing a new ticket  $\mathcal{T}_{t+1}$  for future use. Both parties then exchange application data.

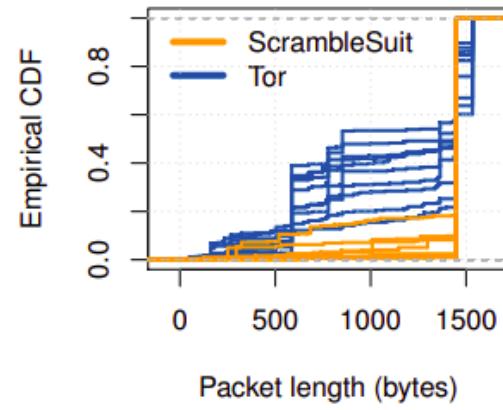
**Payload** By encrypting all ScrambleSuit traffic, we eliminate all payload fingerprints such as Tor’s TLS cipher list [12].

**Packet length distribution** Among other things, we seek to get rid of Tor’s characteristic 586-byte packets [16, 36]. We do so by morphing Tor’s packet length distribution to a randomly chosen distribution.

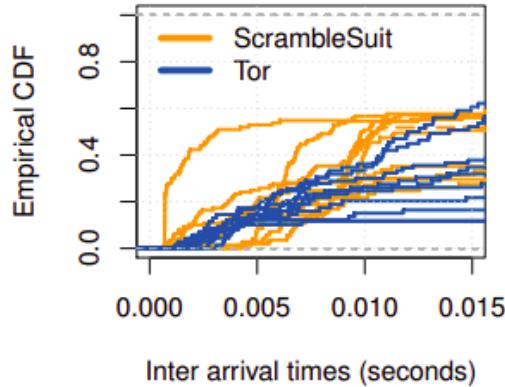
**Inter arrival times** Similar to the packet length obfuscation, we camouflage the inter arrival times by employing small and random sleep intervals before writing data on the wire.



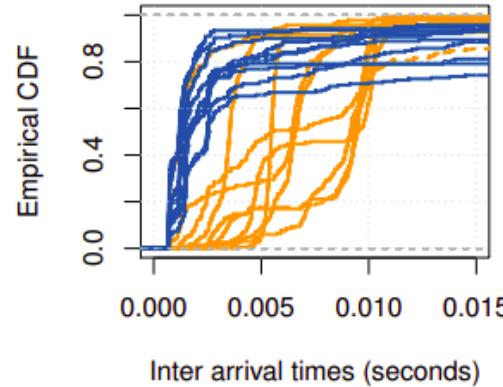
(a) Client-to-server.



(b) Server-to-client.



(c) Client-to-server.

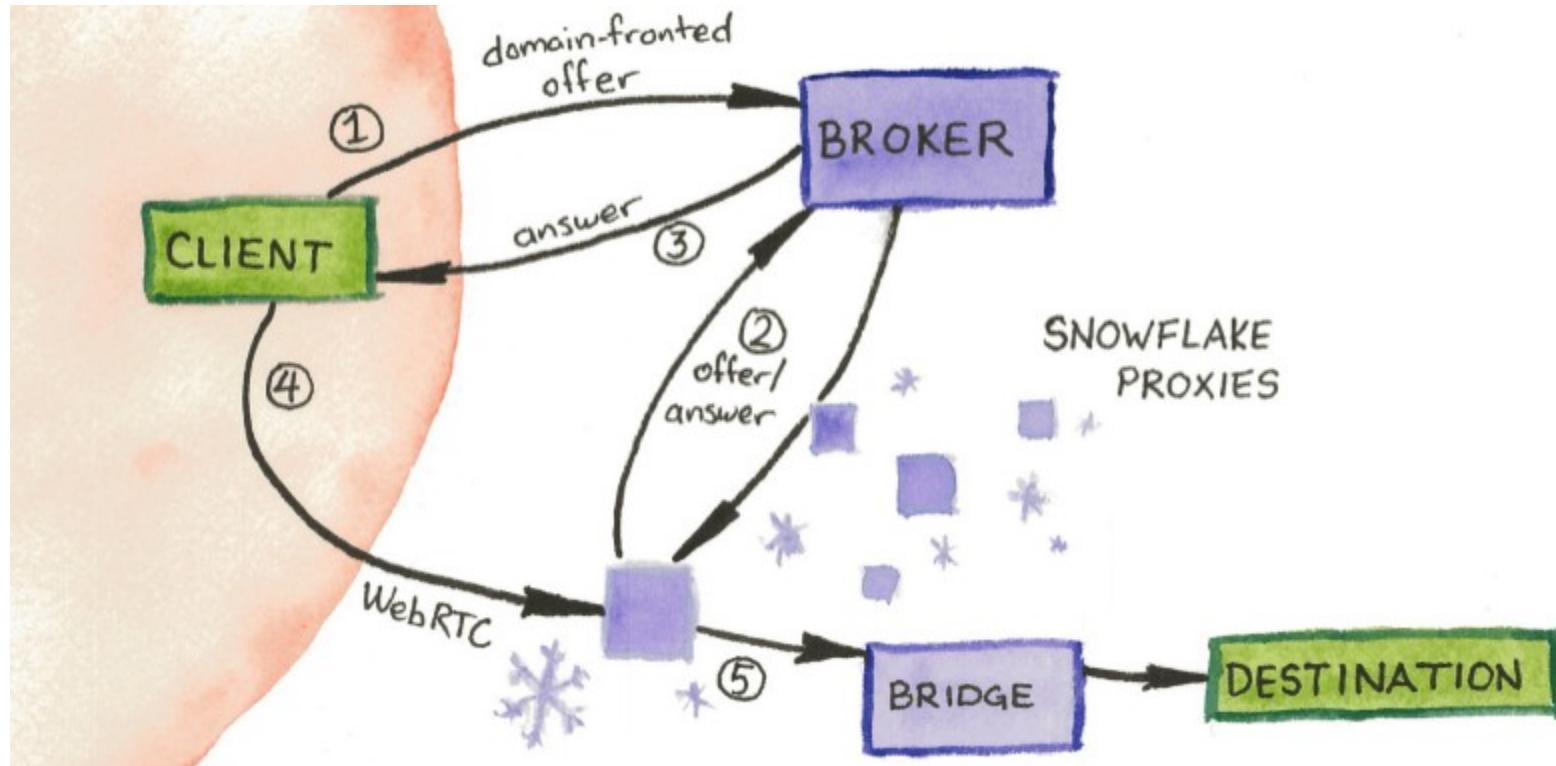


(d) Server-to-client.

# meek

- “Domain fronting”
- Use a Content Distribution Network the censor won’t block
  - Costs money
  - Censors have a business relationship with CDNs

<https://snowflake.torproject.org/>



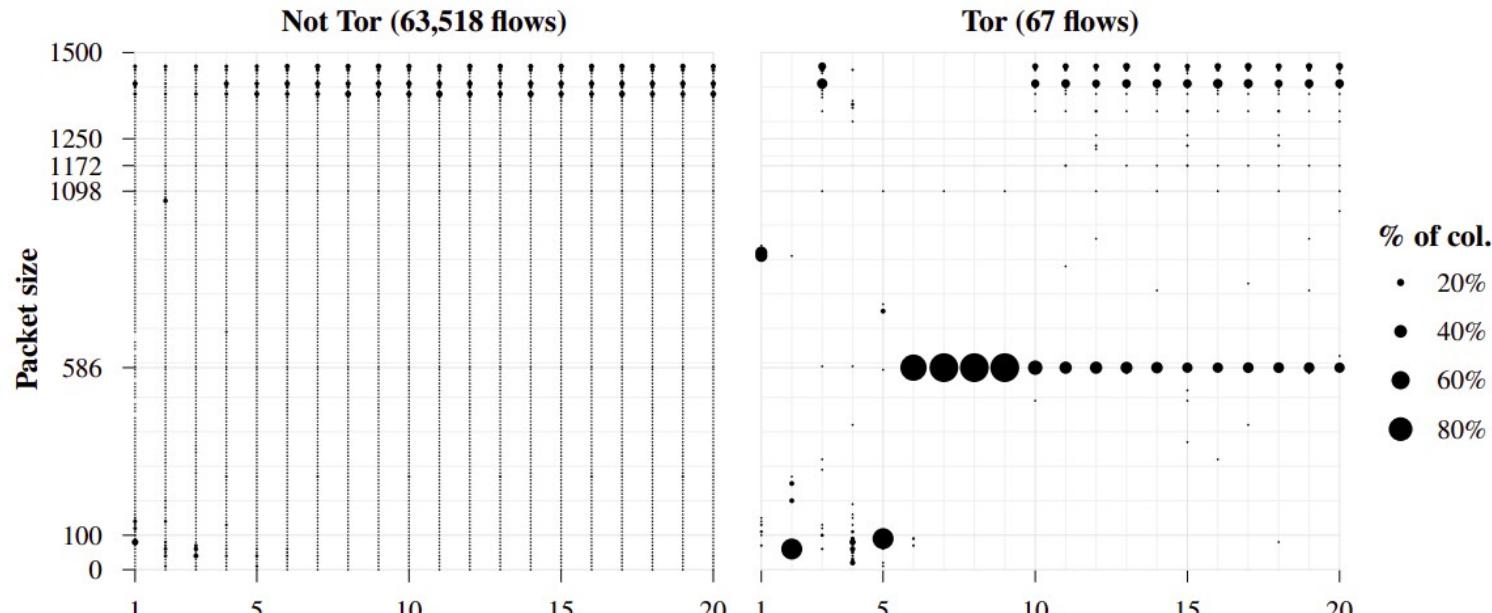
# StegoTorus: A Camouflage Proxy for the Tor Anonymity System

Zachary Weinberg,<sup>1,2</sup> Jeffrey Wang,<sup>3</sup> Vinod Yegneswaran,<sup>2</sup> Linda Briesemeister,<sup>2</sup> Steven Cheung,<sup>2</sup> Frank Wang,<sup>3</sup> and Dan Boneh<sup>3</sup>

<sup>1</sup>Carnegie Mellon University

<sup>2</sup>SRI International

<sup>3</sup>Stanford University



“apps and platforms” → “protocols”

## Two approaches

- Try to look nothing like the banned protocol that you are
- Try to look like a protocol that everyone is using that is not banned

# Fingerprinting Obfuscated Proxy Traffic with Encapsulated TLS Handshakes

Diwen Xue\*

Michalis Kallitsis<sup>†</sup>

Amir Houmansadr<sup>‡</sup>

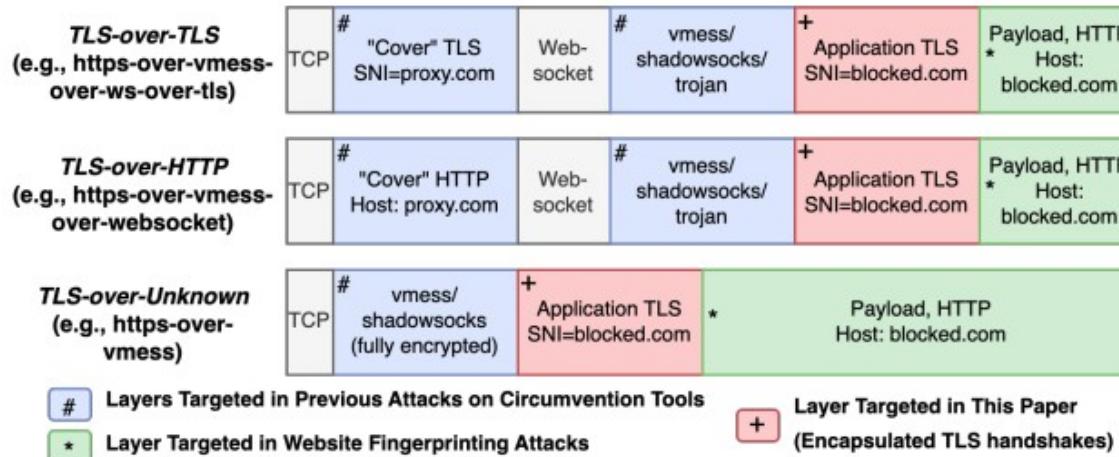
Roya Ensafi\*

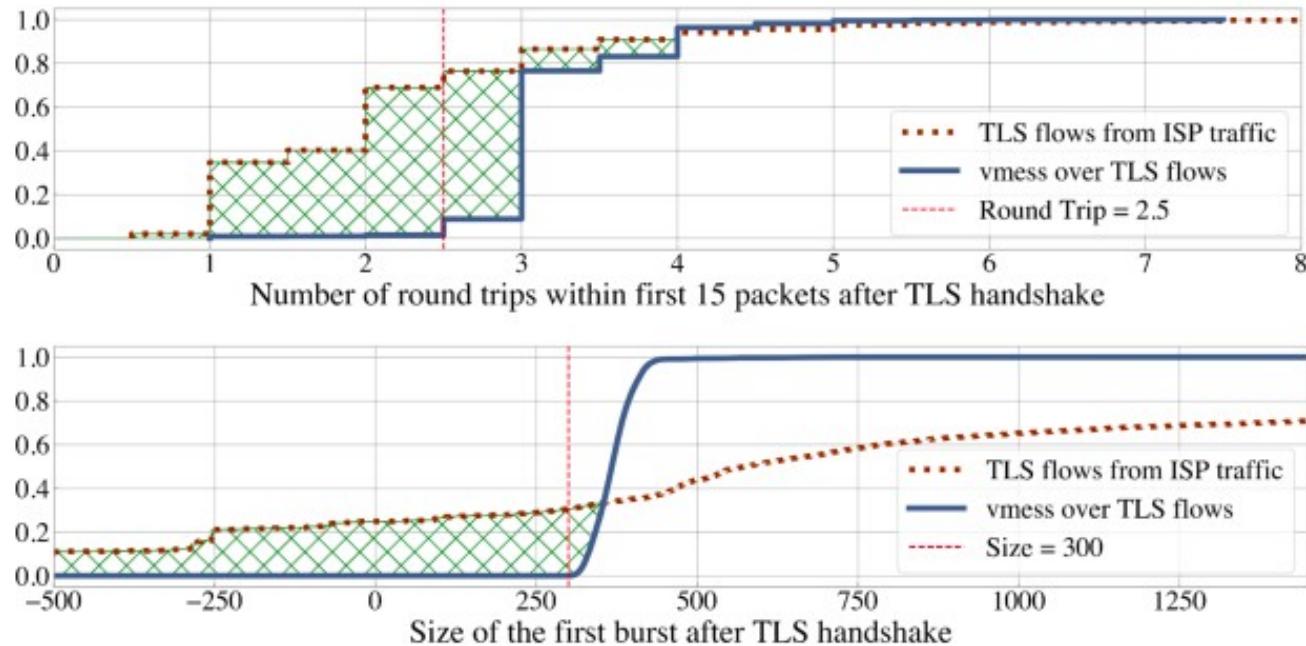
\*University of Michigan

<sup>†</sup>Merit Network, Inc.

<sup>‡</sup> University of Massachusetts Amherst

USENIX Security 2024





**Figure 8: Round trip count and size of first burst after TLS handshakes.** *TLS-over-TLS* requires more round trips and a larger initial burst due to a second (encapsulated) handshake. Shaded areas highlight dissimilarities that padding and multiplexing cannot obfuscate.

# Detecting VPN Traffic through Encapsulated TCP Behavior

Michelina Hanlon  
Stanford University

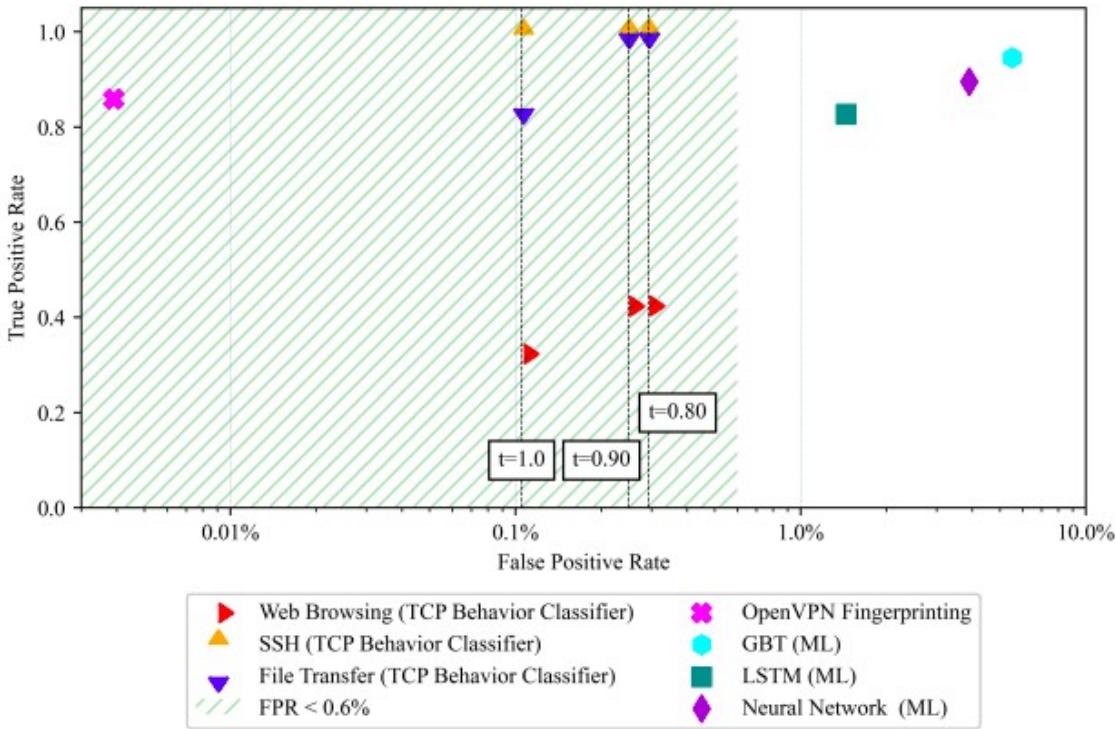
Gerry Wan  
Stanford University

Anna Ascheman  
Stanford University

Zakir Durumeric  
Stanford University

FOCI 2024

- **3WHS:** The presence of a three-way SYN, SYN-ACK, ACK handshake to open the connection (RFC 9293, Section 3.5).
- **500msACK:** The presence of an ACK packet generated within 500 ms of the arrival of a data segment (RFC 9293, Section 3.8.6.3).
- **2RMSS:** The presence of an ACK packet generated after the receipt of  $2 \times \text{RMSS}$  bytes of data, where RMSS is the maximum segment size (MSS) specified by the TCP endpoint receiving the segments (RFC 9293, Section 3.8.6.3).

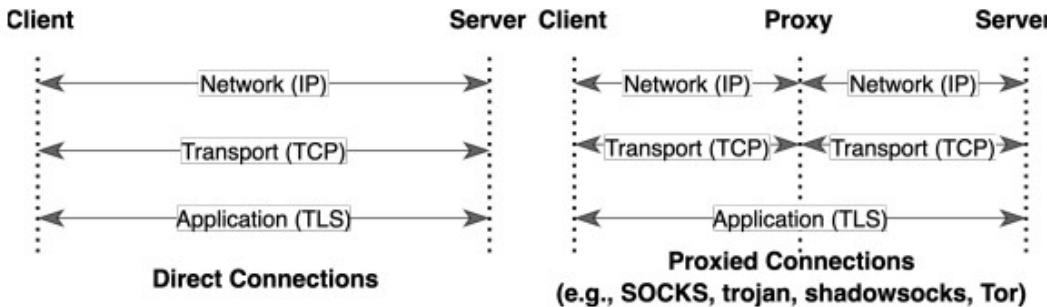


**Figure 1: Classifier Results ( $t = 1.00, 0.90, 0.80; W = 100$ ).** The TPR against the FPR of our classifier (broken down by traffic category) and alternative VPN detection methods. Our protocol-agnostic classifier achieves a FPR an order of magnitude lower than ML detection techniques.

# The Discriminative Power of Cross-layer RTTs in Fingerprinting Proxy Traffic

NDSS 2025

Diwen Xue Robert Stanley Piyush Kumar Roya Ensafi  
University of Michigan  
`{diwenx, rsta, piyushks, ensafi}@umich.edu`



**Fig. 2: Protocol layerings in Direct vs. Proxied Connections.**  
In proxied connections, transport sessions terminate at the proxy, whereas the application layer connection remains end-to-end. ◇

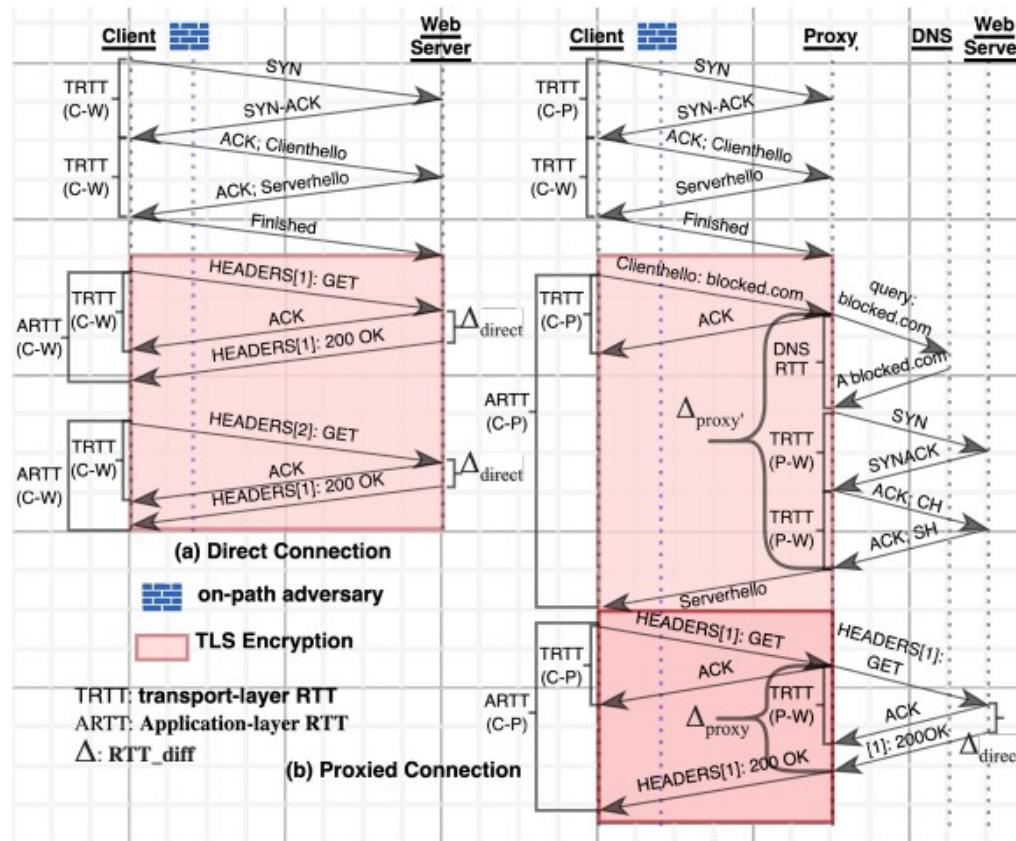


Fig. 3: Sequence timing diagram for (a) direct HTTPS session vs. (b) TLS-based proxy (vless-over-tls) session. For the proxied connection, we observe a much higher discrepancy in transport vs. application-layer delay as the sessions at two layers terminate at different endpoints (proxy and the web server, respectively). ◇

# Is Custom Congestion Control a Bad Idea for Circumvention Tools?

Wayne Wang

University of Michigan  
Ann Arbor, Michigan, USA  
wswang@umich.edu

Ayush Mishra

ETH Zürich  
Zürich, Switzerland  
aymishra@ethz.ch

Diwen Xue

University of Michigan  
Ann Arbor, Michigan, USA  
diwenx@umich.edu

Anonymous

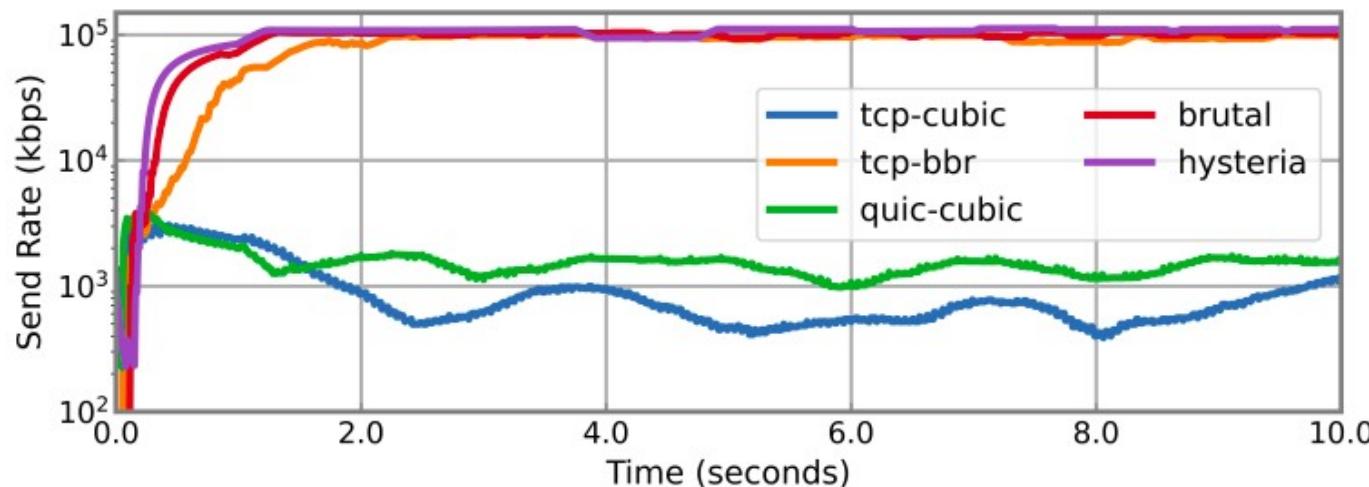
Piyush Kumar

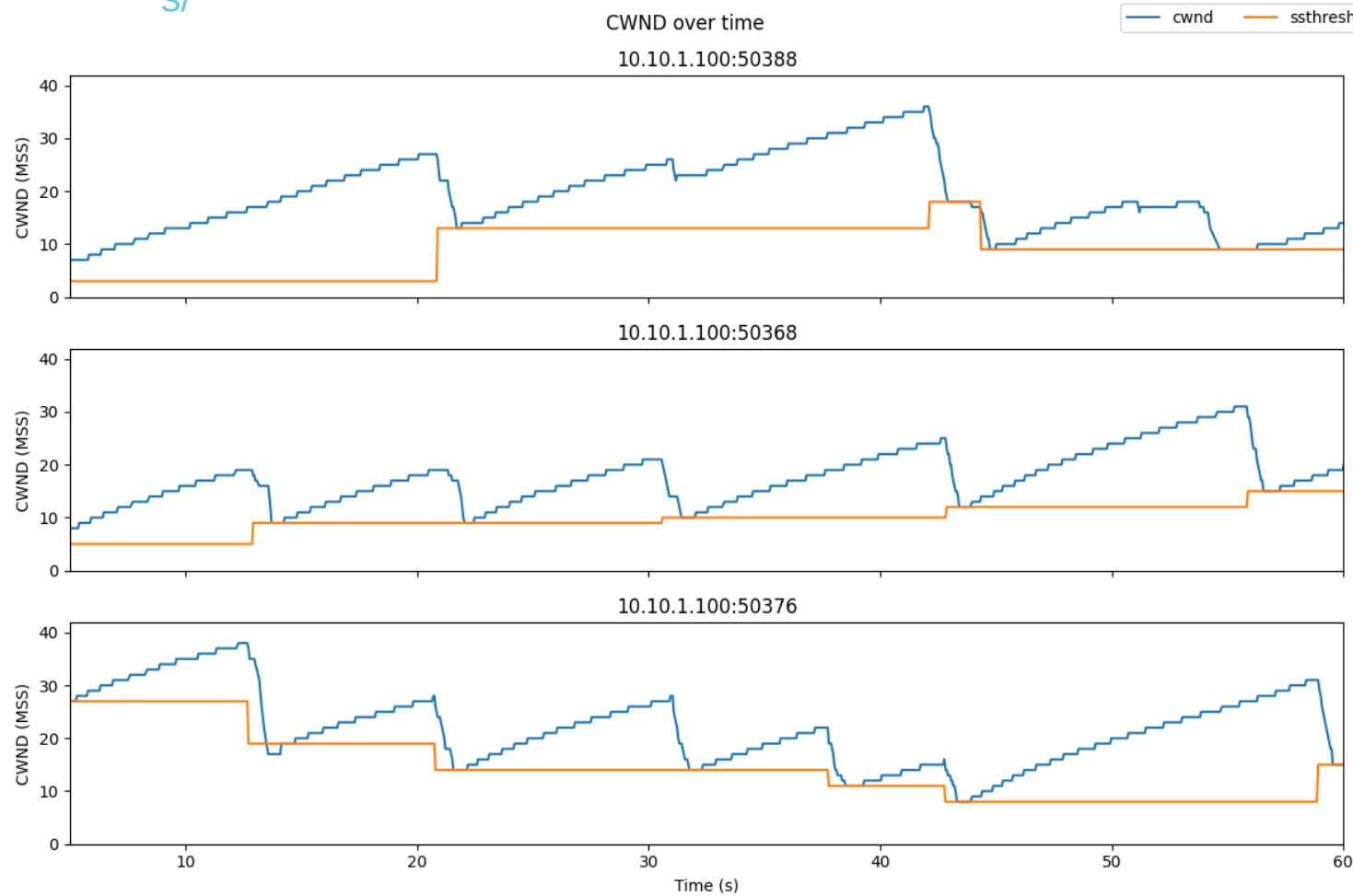
University of Michigan  
Ann Arbor, Michigan, USA  
piyushks@umich.edu

Roya Ensafi

University of Michigan  
Ann Arbor, Michigan, USA  
ensafi@umich.edu

(FOCI 2025)





# Does the DPI need to be nearly perfect w.r.t. false positives and false negatives?

- Can shut down VPNs at critical times
- Can collect data about connections over a long period of time
  - <https://censorbib.nymity.ch/pdf/Wails2024a.pdf>
  - correlate with other data (like financial or social graph information)?
- Can use laws as a deterrent
- Can perform active probing, forensic analysis of confiscated devices, etc.
  - *E.g.*, in Brazil: daily fine of fifty thousand reals (US\$9,104) for users who bypass the ban with a VPN

# OpenVPN is Open to VPN Fingerprinting

Diwen Xue, Reethika Ramesh, and Arham Jain, *University of Michigan*;  
Michalis Kallitsis, *Merit Network, Inc.*; J. Alex Halderman, *University of Michigan*;  
Jedidiah R. Crandall, *Arizona State University/Breakpointing Bad*; Roya Ensafi,  
*University of Michigan*

<https://www.usenix.org/conference/usenixsecurity22/presentation/xue-diwen>

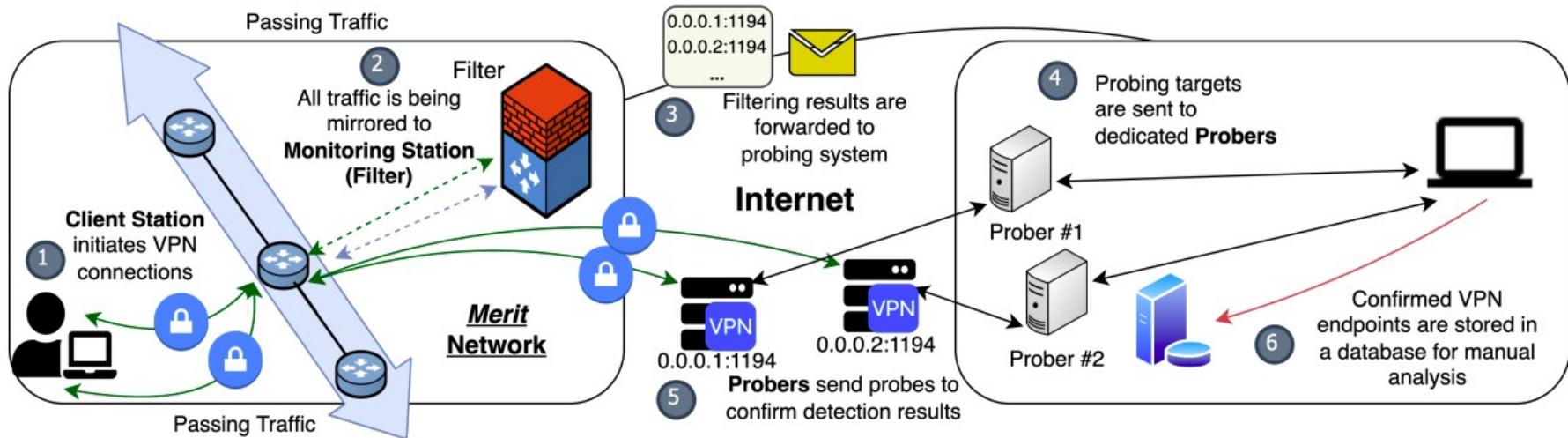
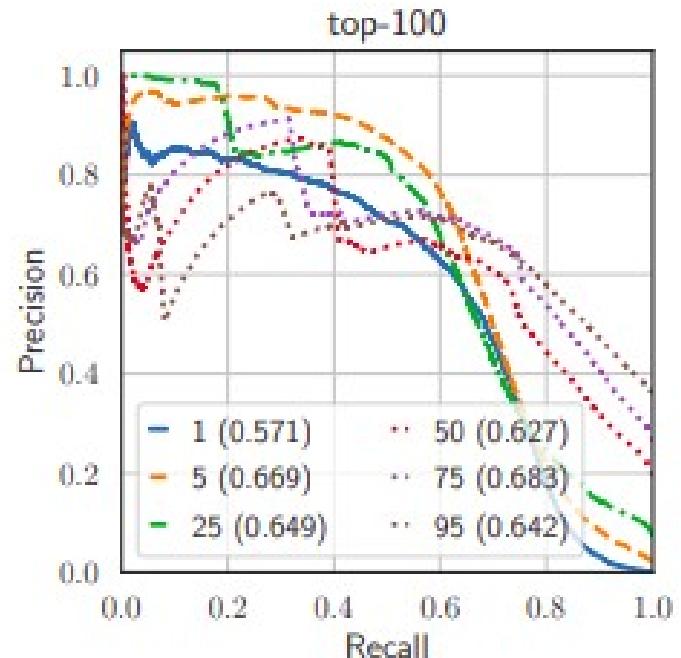


Figure 2: **Framework Deployment on Merit** Steps: (1) Client connects to VPN servers. (2) VPN connections, along with passing traffic, are being mirrored to the *Filter*. (3) *Filter* forwards server IP of suspected connections to the probing system. (4) Targets are sent to each dedicated *Probers*. (5) *Probers* send probes asynchronously. (6) Connections confirmed by probing are logged.

# **Online Website Fingerprinting: Evaluating Website Fingerprinting Attacks on Tor in the Real World**

*Giovanni Cherubin, Alan Turing Institute; Rob Jansen, U.S. Naval  
Research Laboratory; Carmela Troncoso, EPFL SPRING Lab*

<https://www.usenix.org/conference/usenixsecurity22/presentation/cherubin>



# Websites vs. webpages

- Website fingerprinting is knowing that you're going to <https://en.wikipedia.org>, webpage fingerprinting is knowing that you're going to [https://en.wikipedia.org/wiki/Operation\\_Sundevil](https://en.wikipedia.org/wiki/Operation_Sundevil)
- Device fingerprinting, browser fingerprinting, etc. are also an issue for VPNs and Tor.

# Takeaways

- You should check out Tor
  - Also check out OONI
- Tor hidden services are cool
- Designing privacy, anonymity, and anti-censorship tools is a fascinating research problem