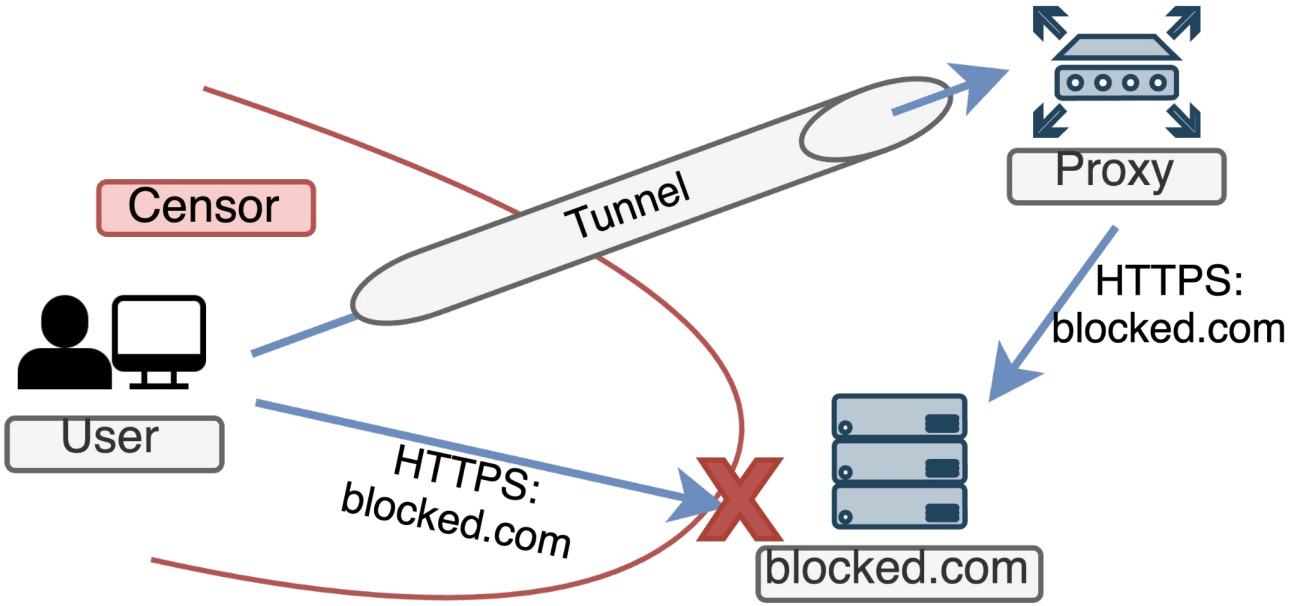
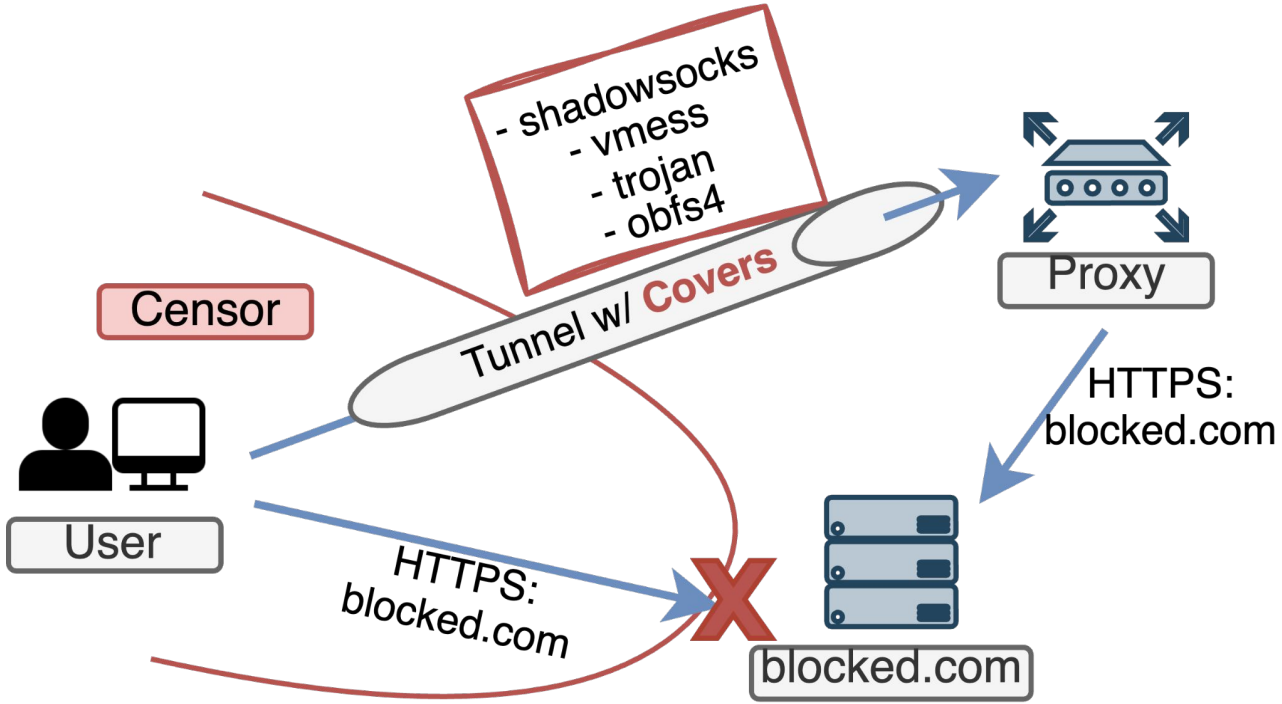
Fingerprinting Obfuscated Proxy Traffic with Encapsulated TLS Handshakes

Diwen Xue, Michalis Kallitsis, Amir Houmansadr, Roya Ensafi
University of Michigan
Merit Network
University of Massachusetts Amherst


Censor
Tunnel
Proxy
HTTPS:
blocked.com
User
HTTPS:
blocked.com
blocked.com


- shadowsocks
- vmess
- trojan
- obfs4
Censor
Tunnel w/ Covers
Proxy
HTTPS:
blocked.com
User
HTTPS:
blocked.com
blocked.com

A Cat-and-mouse Game

Obfuscation in Circumvention Tunnels vs. Evolving Censor **Detection** Methods.

Shadowsocks

➔ Cipher implementation leads to *specific* reactions to probes with different lengths.

* How China Detects and Blocks Shadowsocks. IMC'20

Snowflake

➔ Use DTLS as "cover"; but implementation nuances differ from mainstream browsers, creating exploitable fingerprints.

* Snowflake, a censorship circumvention system using temporary WebRTC proxies. USENIX'24

A Cat-and-mouse Game

Obfuscation in Circumvention Tunnels vs. Evolving Censor **Detection** Methods.

Shadowsocks

➔ Cipher implementation leads to *specific* reactions to probes with different lengths.

* How China Detects and Blocks Shadowsocks. IMC'20

Snowflake

➔ Use DTLS as "cover"; but implementation nuances differ from mainstream browsers, creating exploitable fingerprints.

* Snowflake, a censorship circumvention system using temporary WebRTC proxies. USENIX'24

Attacks exploit design/implementation flaws specific to individual cover protocols.

“Letting a thousand flowers bloom”

Protocol-specific fingerprinting attacks

- Highly specialized fingerprints
- Common thinking:
  each new cover protocol requires its own **unique set of features** and **separate analysis**.

Strategy:
Increase the diversity of protocols to overwhelm censors.

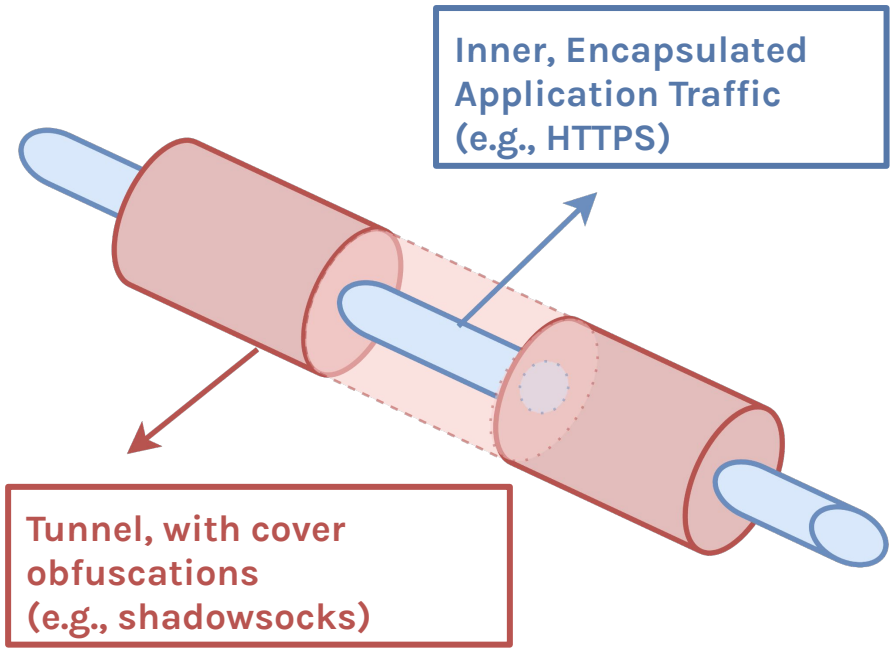
“ How to exploit censor weaknesses?
Increase the diversity of censorship circumvention solutions by letting a thousand flowers bloom…
The more anti-censorship solutions the community can create, the less likely these limited resource censor teams will be able to block all tools in one go. ”

“Letting a thousand flowers bloom”

Must censors fingerprint each cover protocol separately?

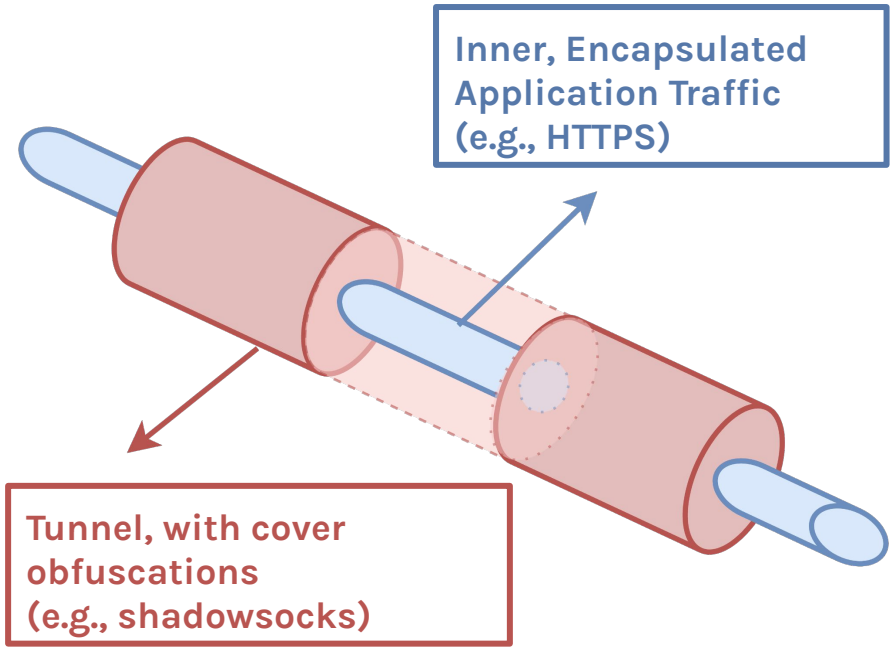
- **The main goal of censors is <u>not</u> to identify protocol, but to identify circumvention.**

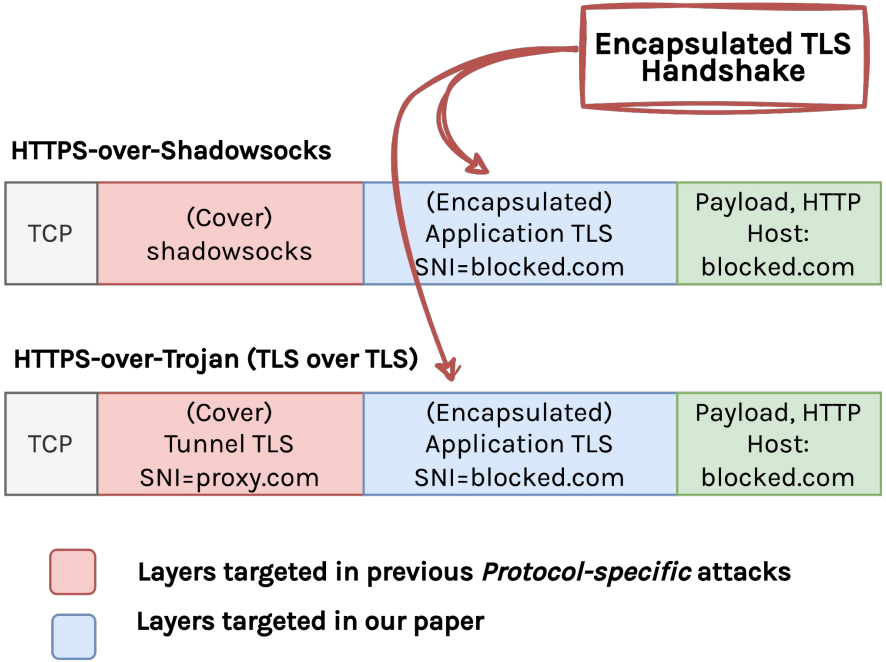
Encapsulated TLS Handshake as a *Protocol-agnostic* Fingerprint



Censored Planet

Encapsulated TLS Handshake as a *Protocol-agnostic* Fingerprint





Censored Planet

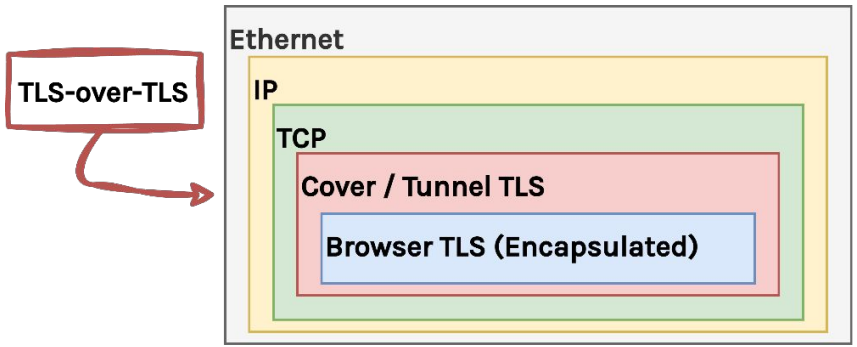
Encapsulated TLS Handshake as a *Protocol-agnostic* Fingerprint

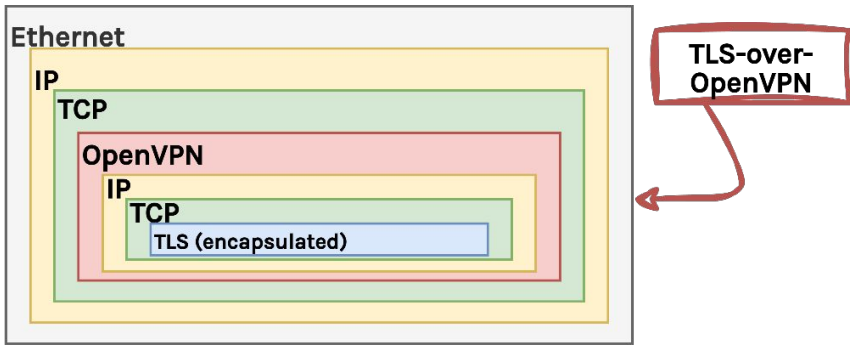
"Nested Protocol Stacking" underpins all forms of proxying and tunneling.

- One protocol stack is encapsulated within the payload of another.

Focus on Encapsulated TLS:

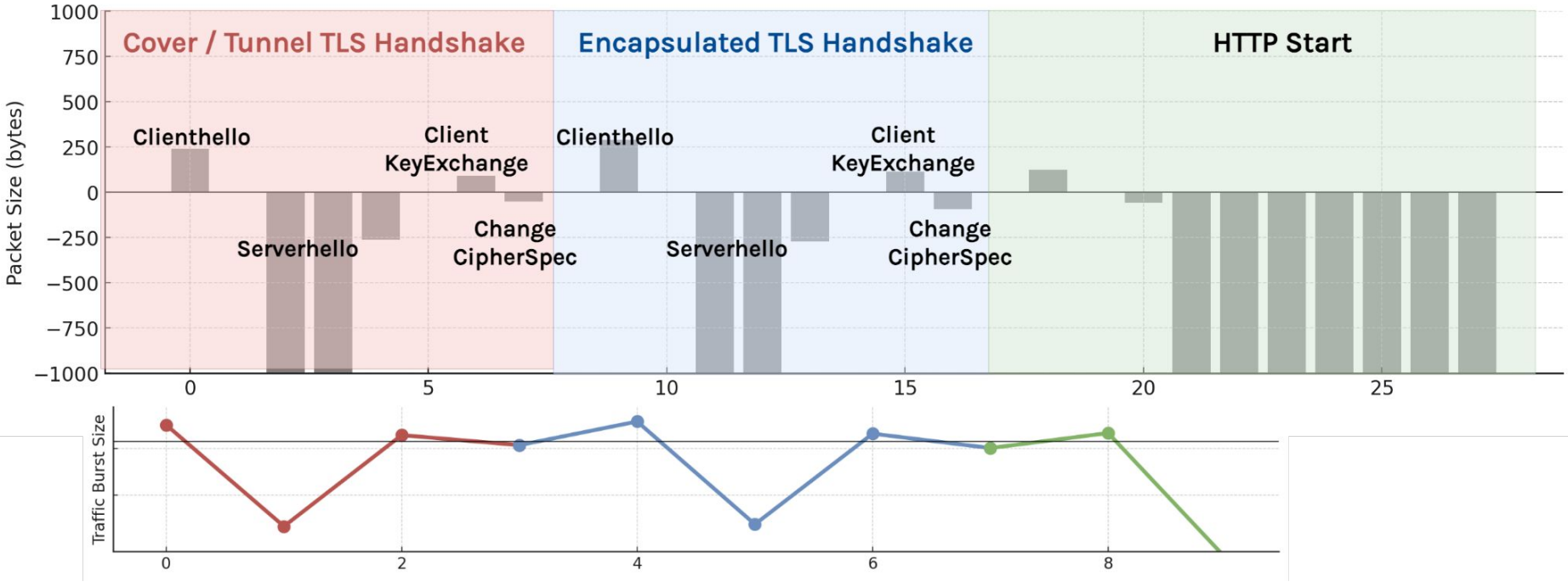
- Prevalence of TLS makes it a **reliable target** for fingerprinting: users cannot "not use TLS".
- Encapsulating TLS within another secure layer is unusual.





Censored Planet

Encapsulated TLS Handshake: What does it look like?



Implementation of the Fingerprint

- **Approach as a binary classification problem:**
  Determining the presence of absence of encapsulated TLS handshakes.

- **Flow representation:**
  - **Tri-grams** extracted from sequence of packet sizes. E.g., (+517,–1400,–1400)
  - **Traffic burst**: sequences of consecutive packets traveling in the same direction.

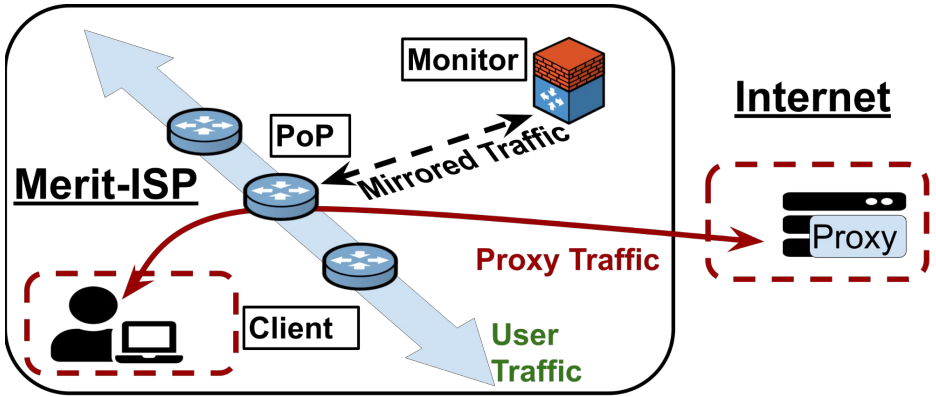
- **Similarity-based classification** (χ^2 test, Mahalanobis distance)

- **Learn from cleartext TLS, apply to payload of encrypted traffic.**
  - Labeled circumvention traffic is hard to obtain

Evaluation

Deployed the fingerprint as a Zeek cluster inside Merit ISP for 30 days.

- Tested the fingerprint on both (mirrored) **real-user traffic** & **circumvention traffic**.



Tested 23 different obfuscated proxy configurations:

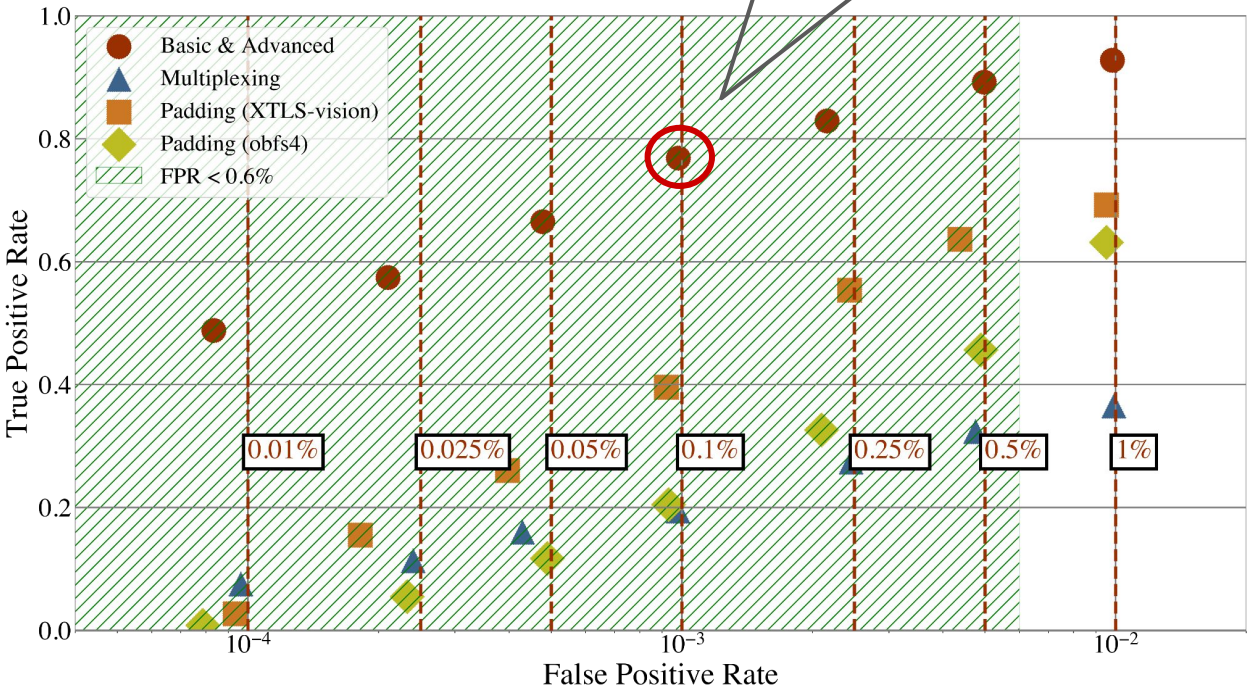
- Shadowsocks, vmess, Trojan, obfs4, etc

Efficacy of the fingerprint is largely independent of the specific proxy protocol tested.

Censored Planet

Evaluation

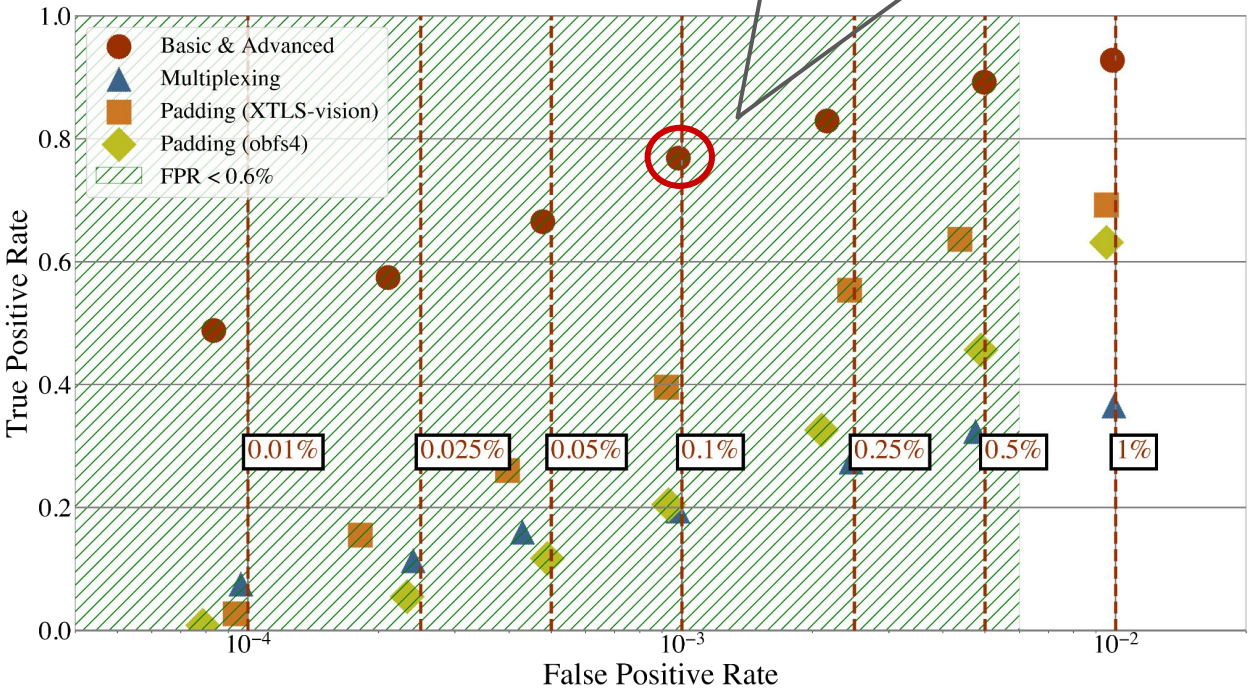
vmess: 77.14%
shadowsocks: 85.38%
vless: 74.83%
trojan: 73.71%
FPR < 0.1%



Censored Planet

Evaluation

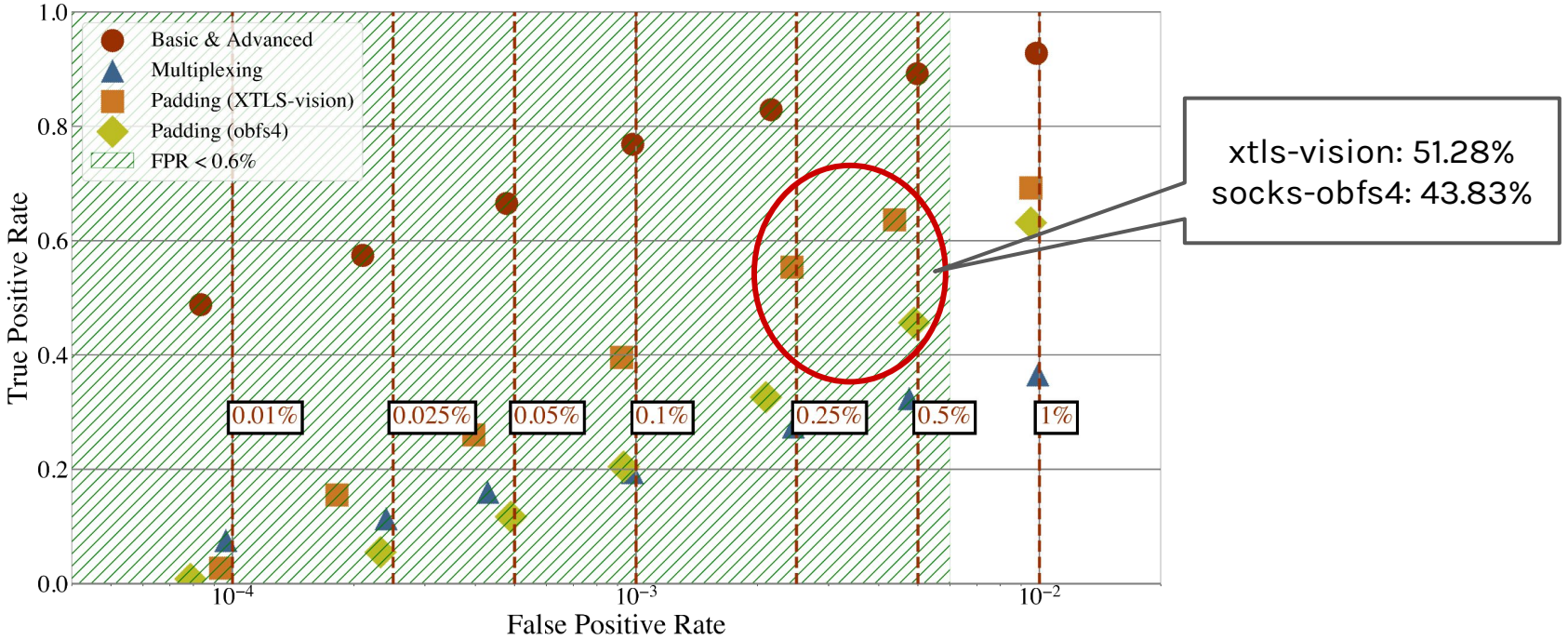
vmess: 77.14%
vmess-tls: 74.46%
vmess-ws-tls: 68.78%
shadowsocks: 85.38%
shadowsocks-tls: 78.75%
shadowsocks-ws-tls: 69.68%



Censored Planet

Evaluation

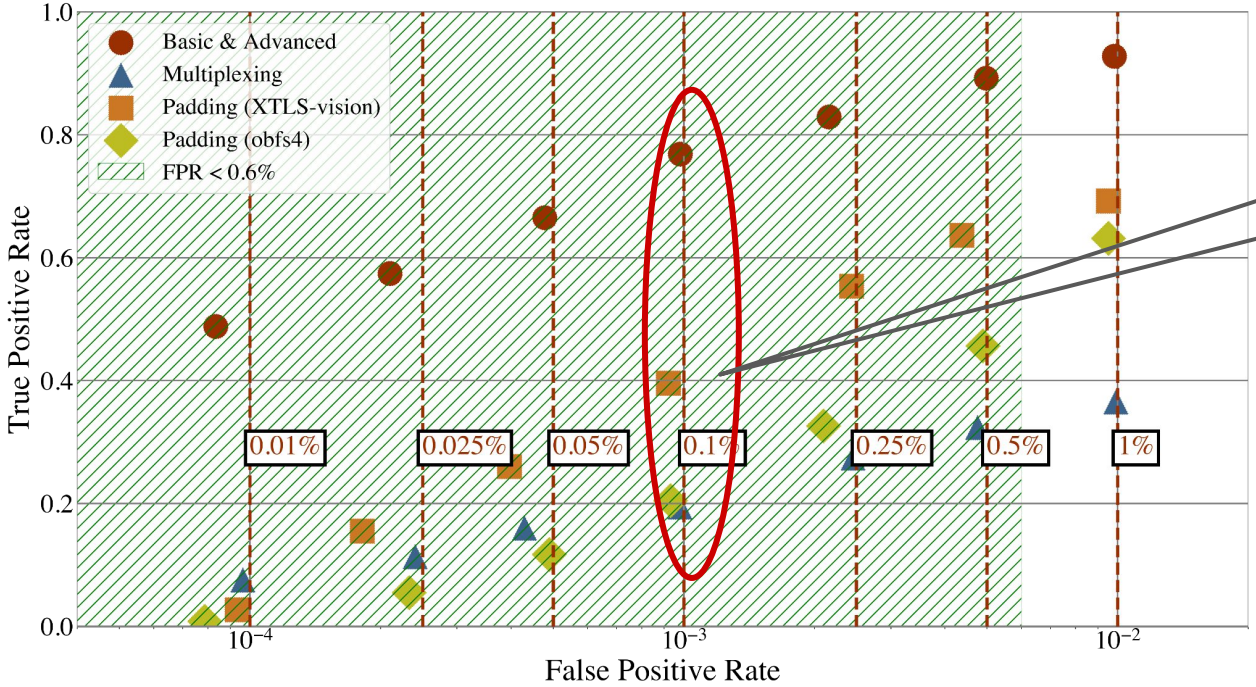
Random padding is NOT the final word when it comes to obfuscating traffic patterns



Censored Planet

Evaluation

Connection multiplexing could be a mitigation for the short term



vmess: 77.14%
vmess-tls: 74.46%
vmess-tls w/ MUX: 14.84%

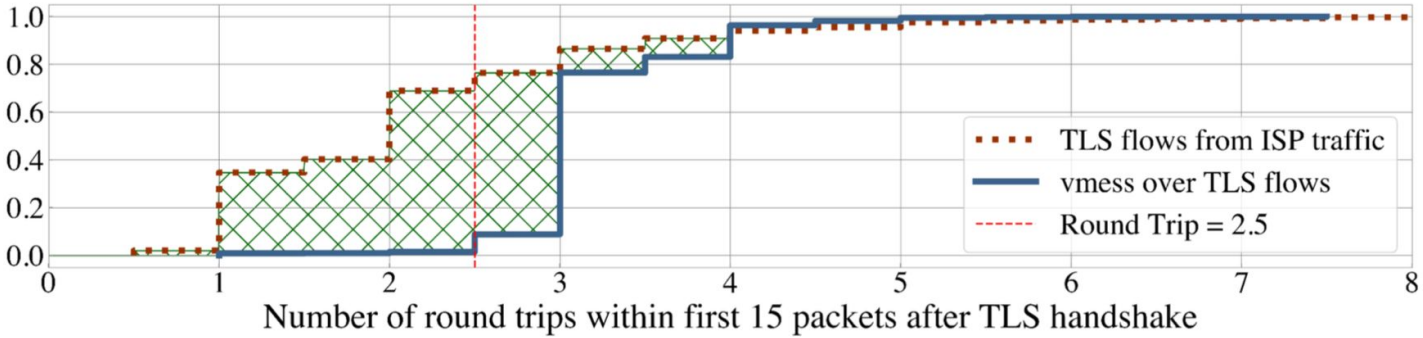
Recent developments in Russia (2024.4-)

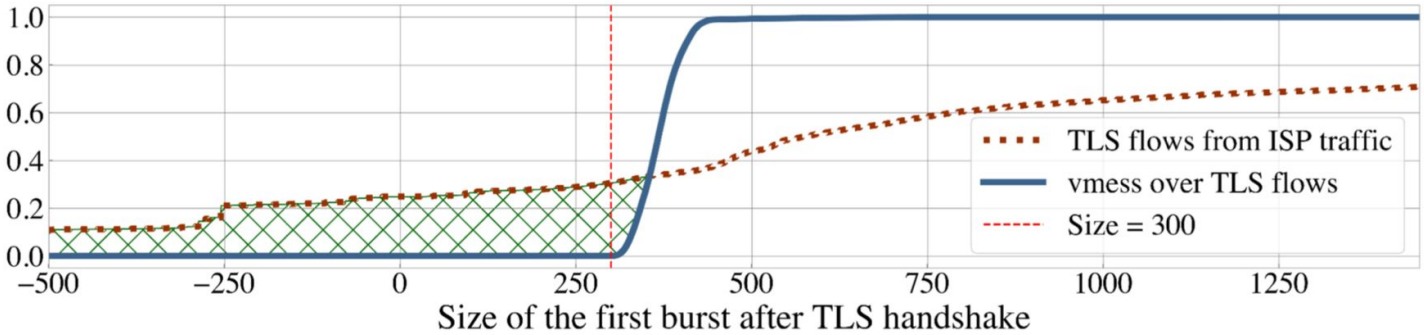
Russia tested a new approach to blocking circumvention traffic starting from 2024-04-25

- Targets **encapsulated HTTPS/TLS exchanges**, instead of specific cover protocols.
- Highlights the urgency for developing principled countermeasures

Fingerprinting Obfuscated Proxy Traffic with Encapsulated TLS Handshakes

Diwen Xue, Michalis Kallitsis, Amir Houmansadr, Roya Ensafi
University of Michigan
Merit Network
University of Massachusetts Amherst


1.0
0.8
0.6
0.4
0.2
0.0
0
1
2
3
4
5
6
7
8
TLS flows from ISP traffic
vmess over TLS flows
Round Trip = 2.5
Number of round trips within first 15 packets after TLS handshake


1.0
0.8
0.6
0.4
0.2
0.0
−500
−250
0
250
500
750
1000
1250
TLS flows from ISP traffic
vmess over TLS flows
Size = 300
Size of the first burst after TLS handshake