De-anonymizing Tor Onion Services with Flow Correlation Attacks

Diogo Barradas

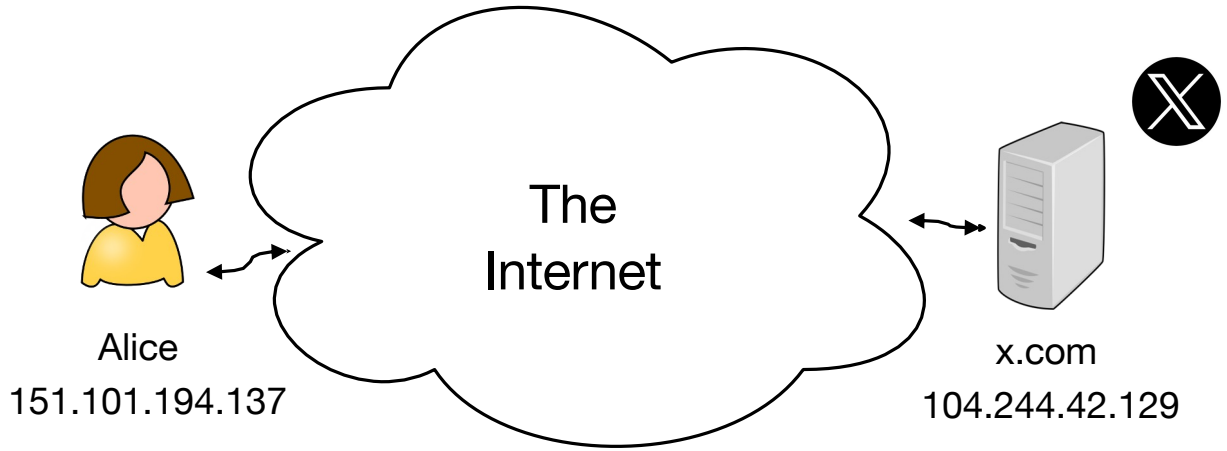
INRIA-UW-UB Workshop

Bordeaux, France

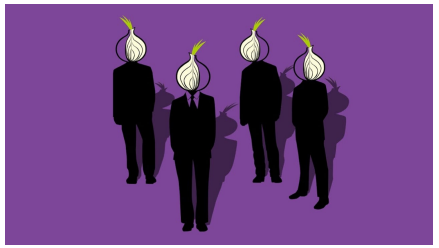
February 22nd, 2024

The need for online anonymity

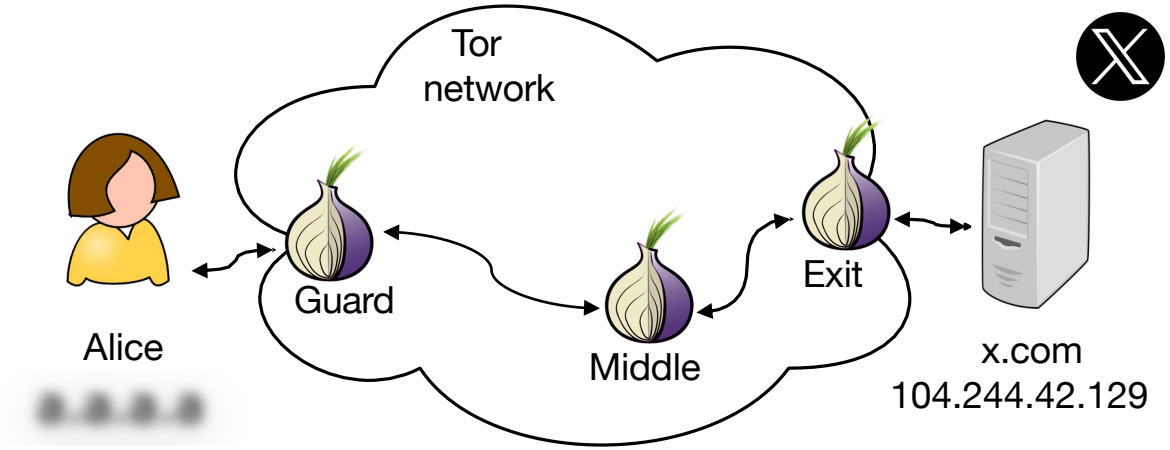
- Internet users are at odds with pervasive tracking and online surveillance



- Onion routing aims to provide online anonymity by sending users' network traffic through multiple relays

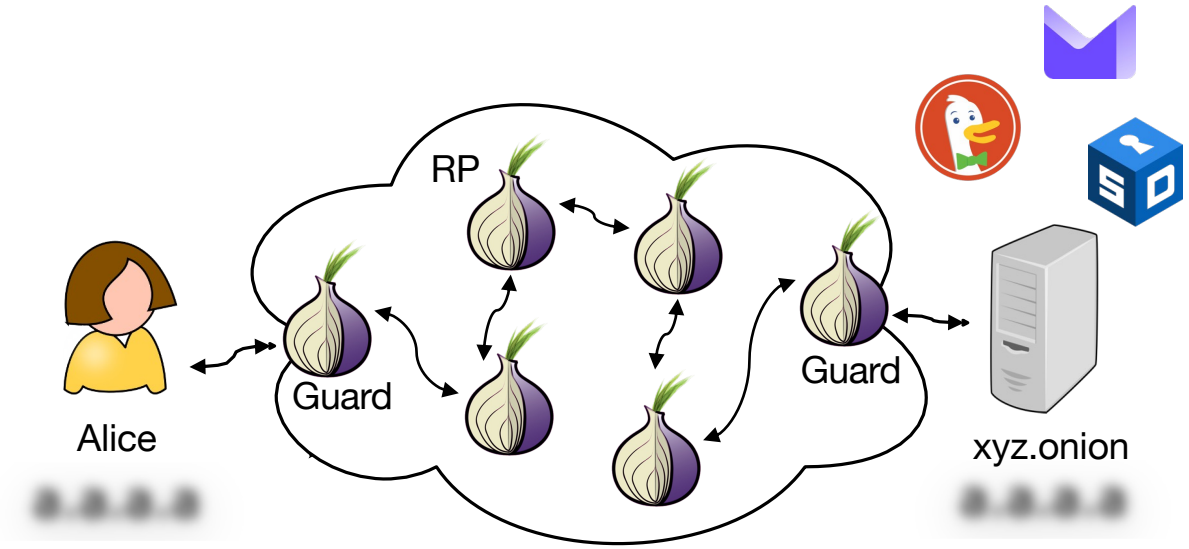


Preserving online anonymity with Tor



Client-side anonymity

Circuits to the “clearweb”



Client- and server-side anonymity

Circuits to onion services

Can we deanonymize Tor traffic? Should we?

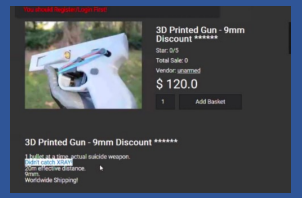
SECUREDROP

The Guardian

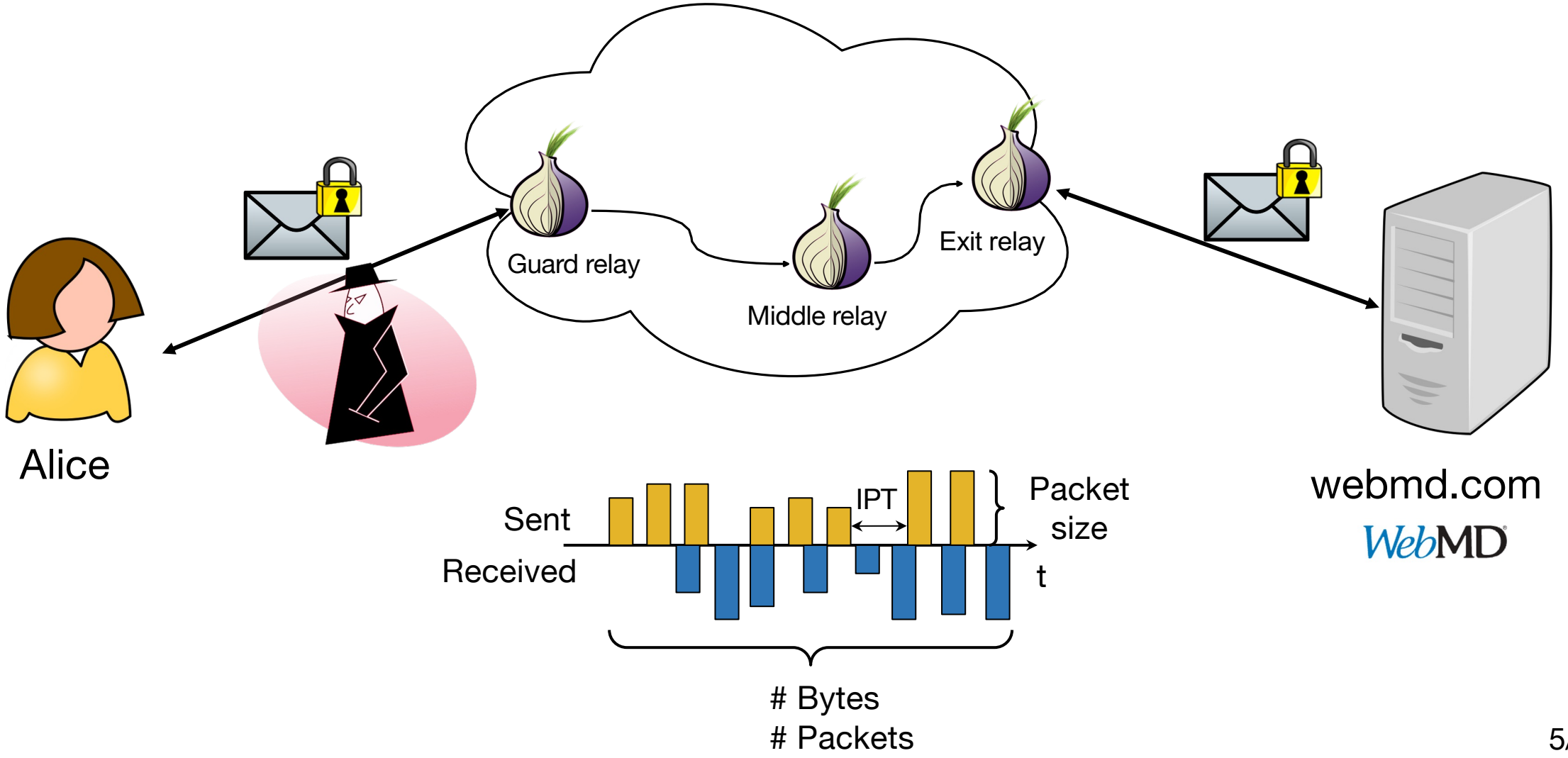
+ Protect activists, whistleblowers, etc.

+ Catch (cyber)criminals

AlphaBay Market



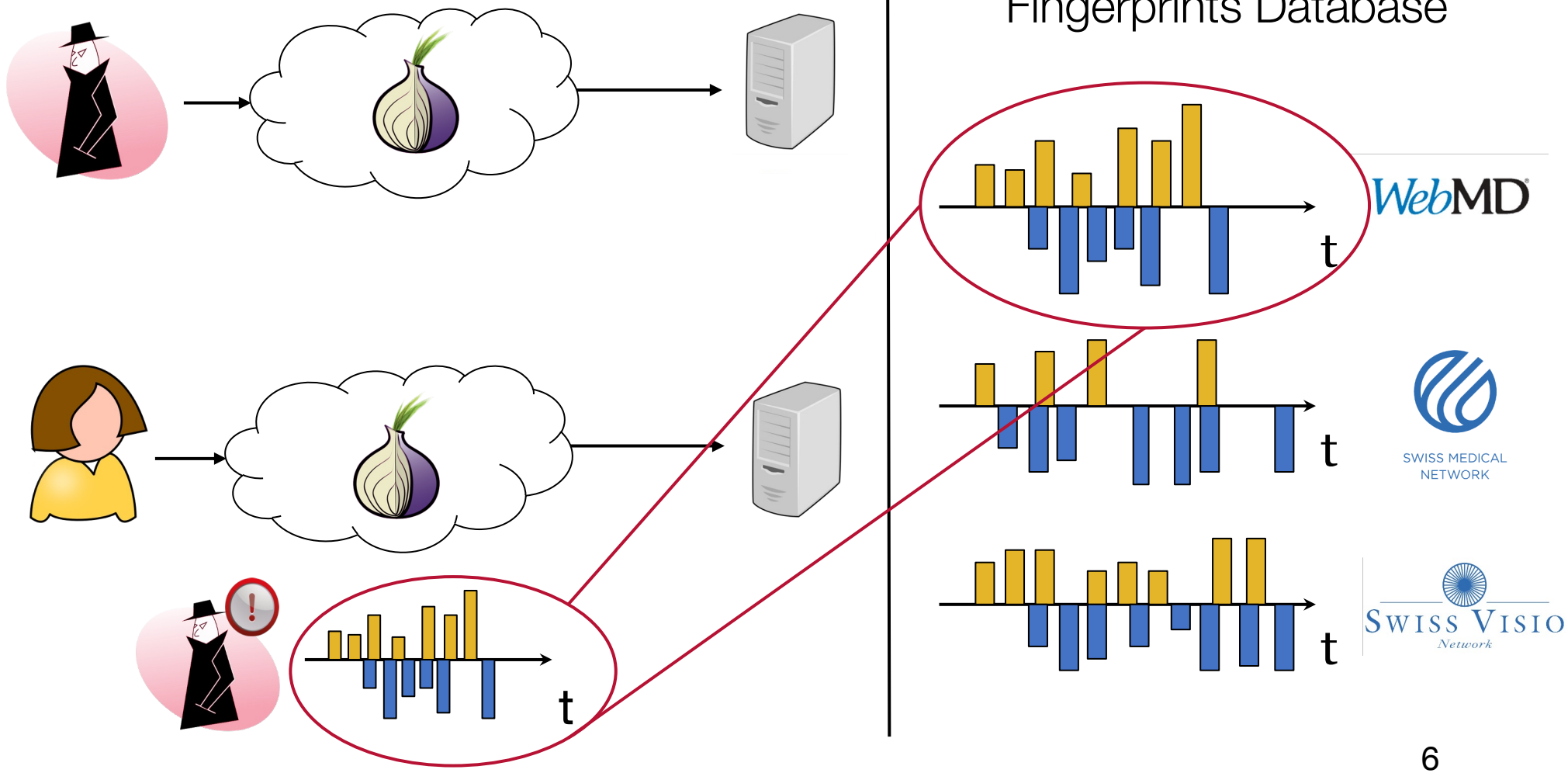
Encrypted Tor connections leak metadata



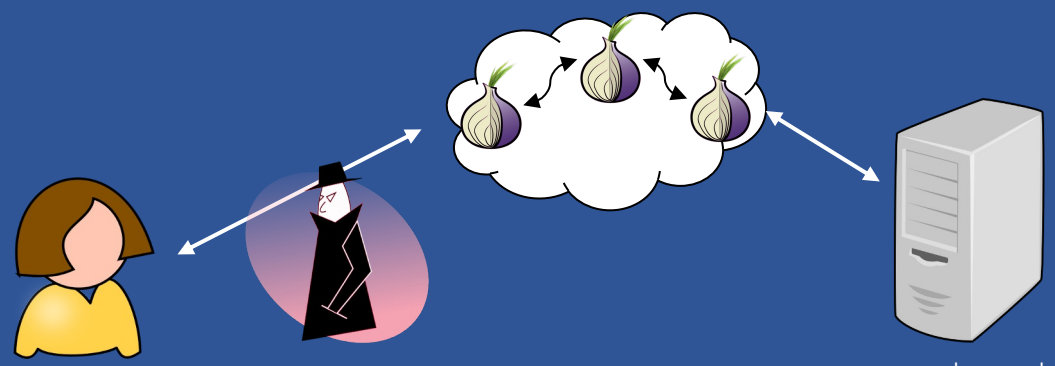
Deanonymizing clients' accesses via Tor (I)

Step 1:
Build fingerprint database

Step 2:
Match Alice's traffic



How can one deanonymize Tor traffic?



webmd.com

…

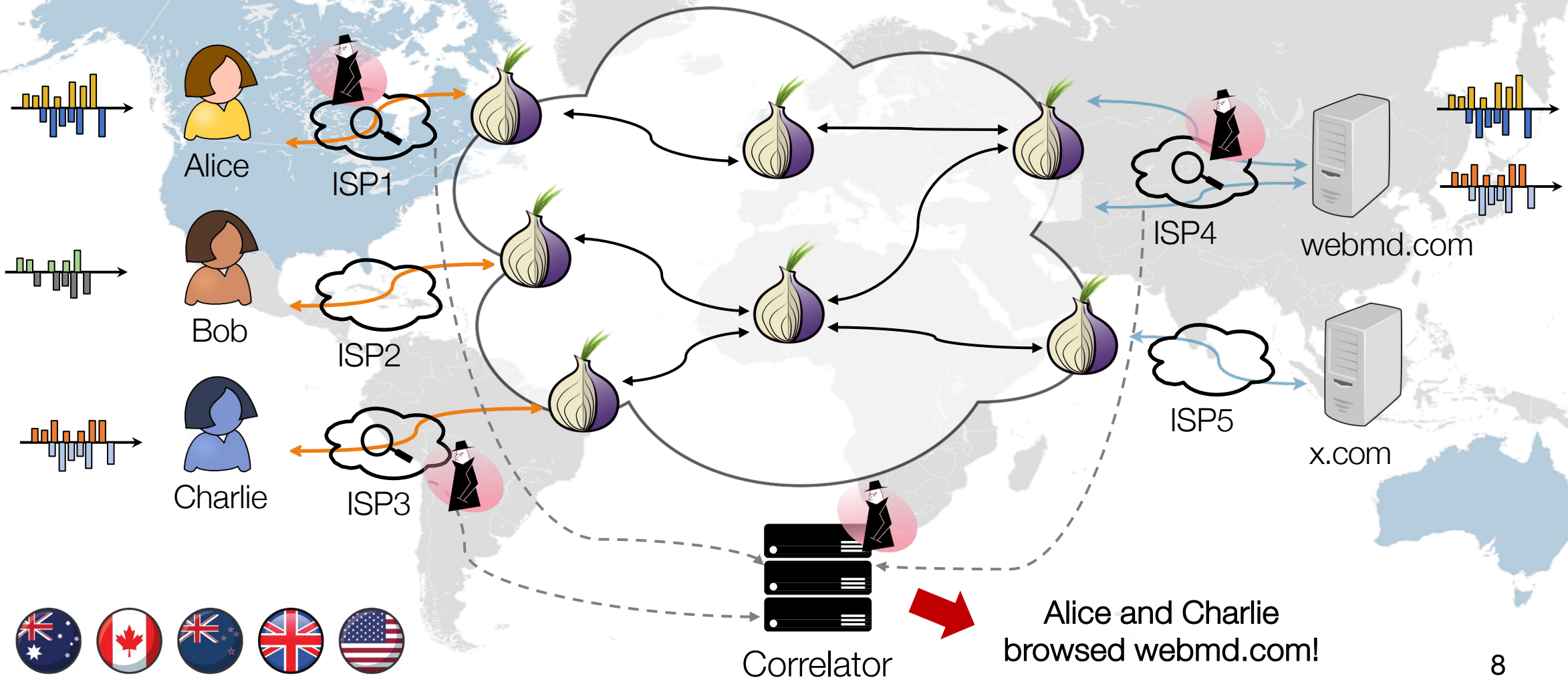
xyz.onion

Website Fingerprinting
(local adversary)

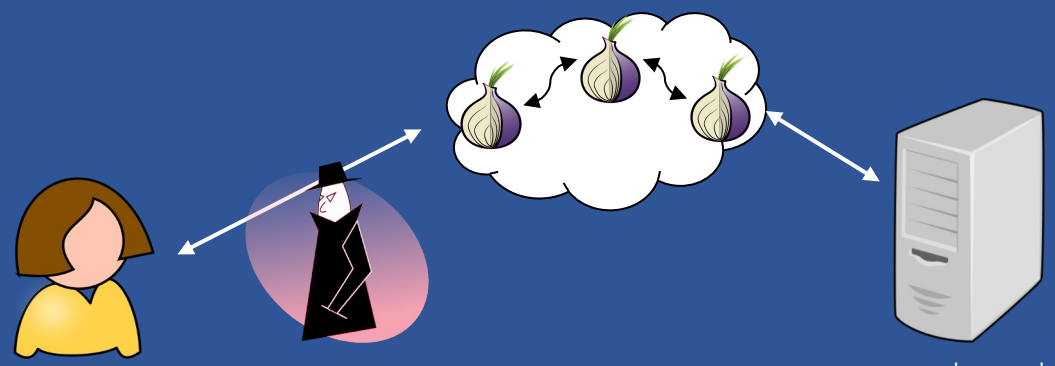
+ Can tell the website/.onion a client connects to
  - Requires a pre-built database
  - Cannot find a .onion's IP address

Is there another way?

Deanonymizing clients' accesses via Tor (II)

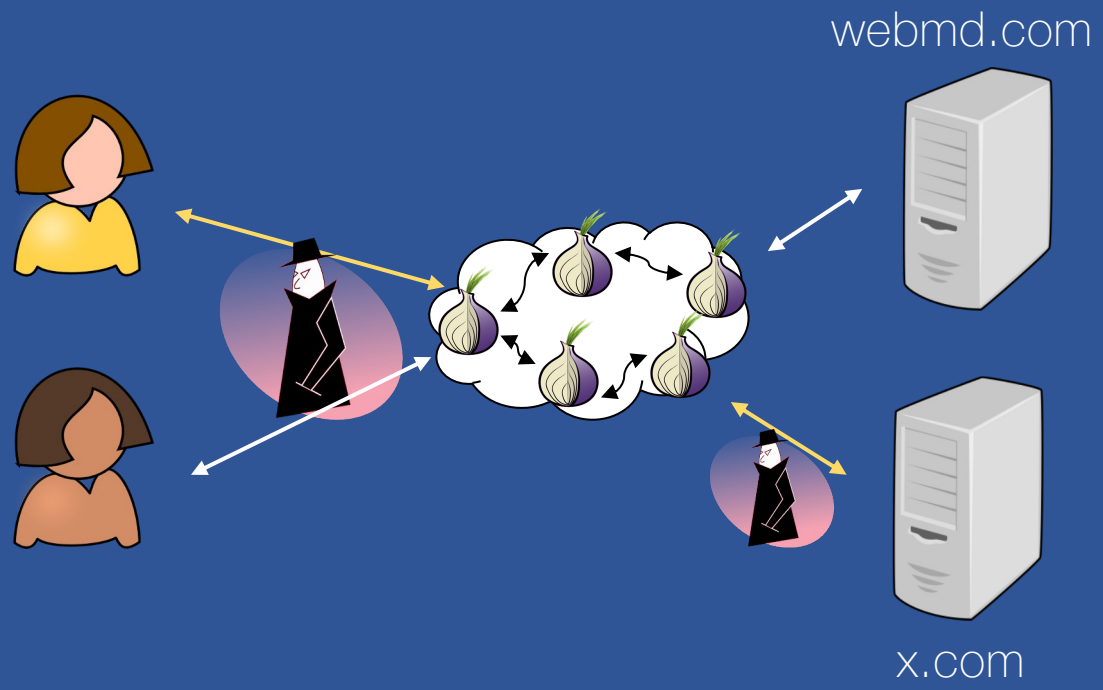


How can one deanonymize Tor traffic?



Website Fingerprinting
(local adversary)

+ Can tell the website/.onion a client connects to
- Requires a pre-built database
- Cannot find a .onion's IP address

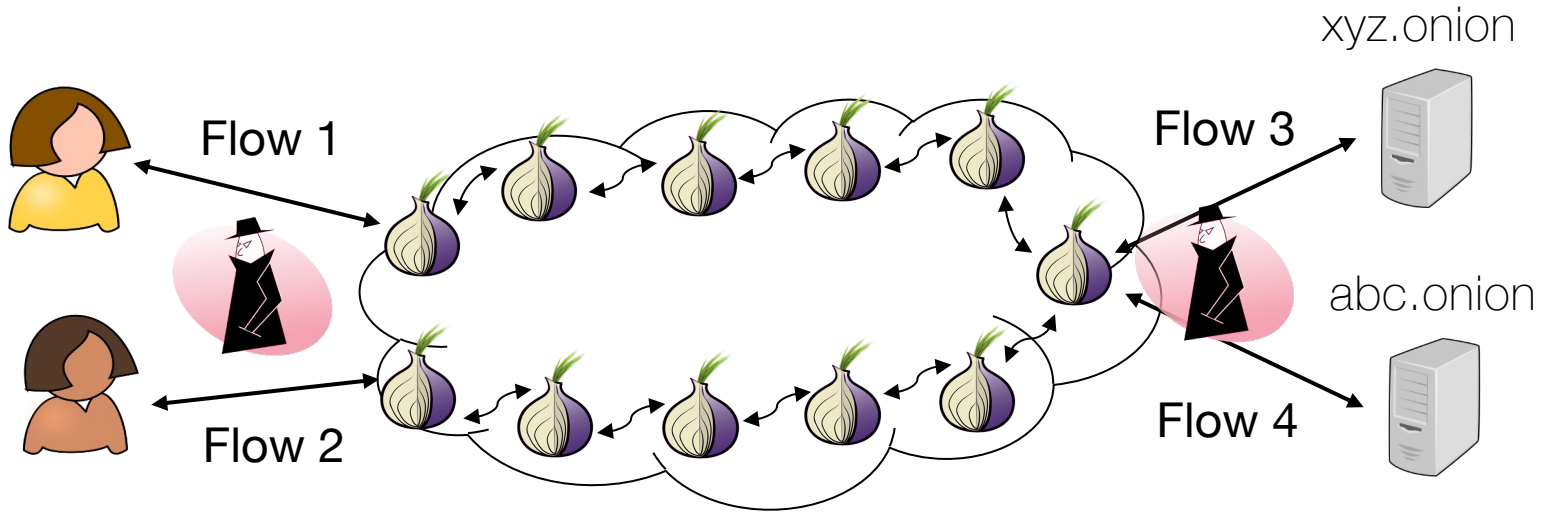


Flow Correlation
(global adversary)

+ Can tell what website a client connects to
+ No need for a database
- What about onion services?

Challenges for onion service flow correlation (I)

- Onion services connect to the Tor network very much like a client
  - A naïve correlation method may try to **match** flow pairs that are **surely uncorrelated**

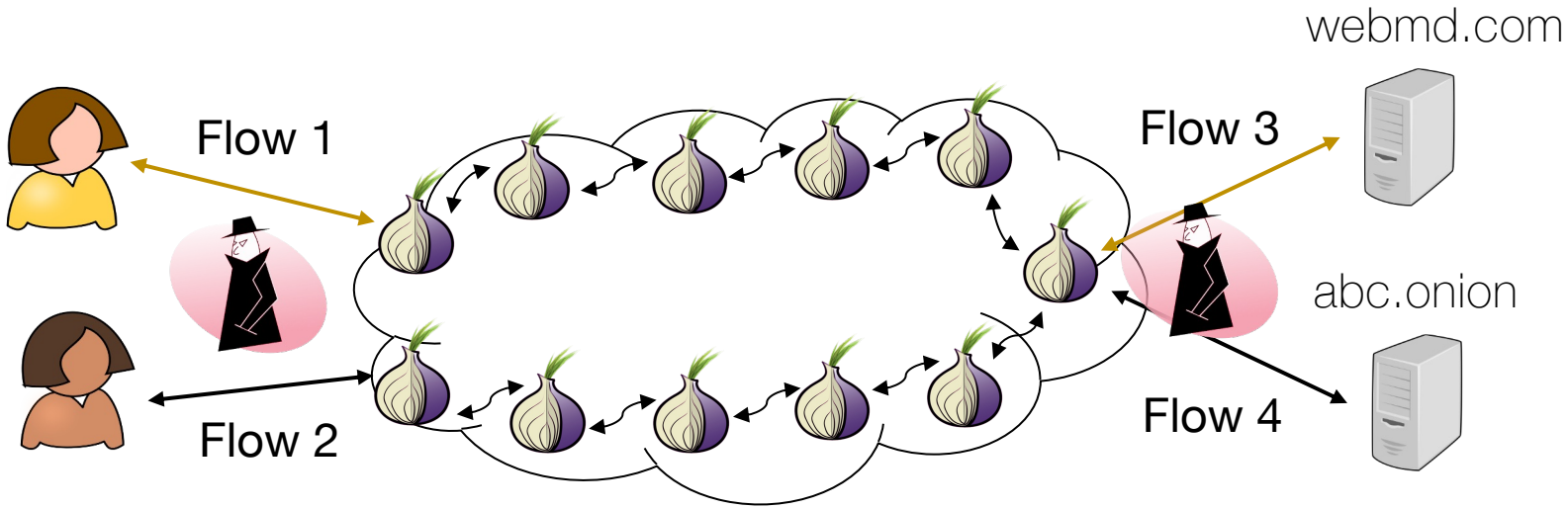


Feasible pairs:
1 – 3
2 – 3
1 – 4
2 – 4

Unfeasible pairs:
1 – 2
3 – 4

Challenges for onion service flow correlation (II)

- Onion services connect to the Tor network very much like a client
  - A naïve correlation method may try to **match** flow pairs that are **surely uncorrelated**
- Identify and **discard client requests** towards the **clearweb**

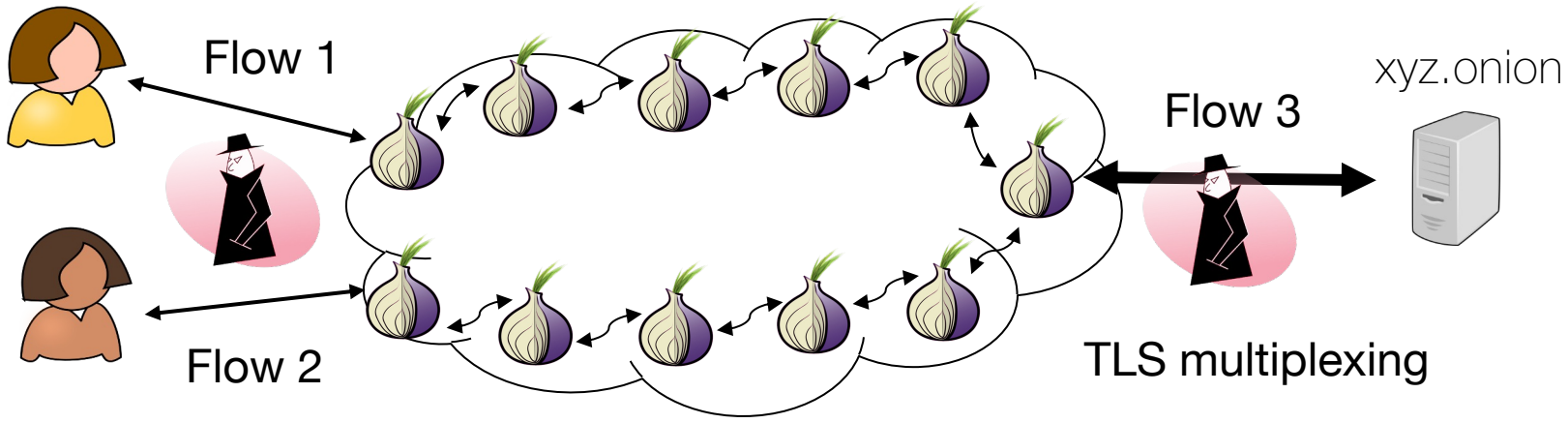


Feasible pairs:
1 – 3
2 – 3
1 – 4
2 – 4

Unfeasible pairs:
1 – 2
3 – 4

Challenges for onion service flow correlation (III)

- Onion services connect to the Tor network very much like a client
  - A naïve correlation method may try to **match** flow pairs that are **surely uncorrelated**
- Identify and **discard client requests** towards the **clearweb**
- **Untangle** concurrent client requests/responses at the onion service guard

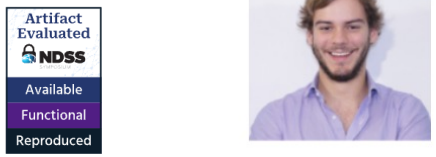


Our recent efforts

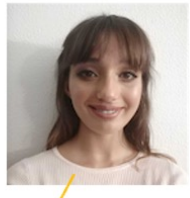
- Introduce **SUMo**, a novel classification pipeline that enables efficient and accurate flow correlation for Tor onion service sessions
- Collect a large dataset for evaluating flow correlation on Tor, encompassing accesses both to clearnet and onion service websites
- Provide an implementation and evaluation of SUMo

[shameless advertising spot]

Flow Correlation Attacks on Tor Onion Service Sessions with Sliding Subset Sum



Daniela Lopes*, Jin-Dong Dong†, Pedro Medeiros*, Daniel Castro*, Diogo Barradas‡, Bernardo Portela§, João Vinagre§, Bernardo Ferreira¶, Nicolas Christin†, Nuno Santos*
*INESC-ID / IST, Universidade de Lisboa, {daniela.lopes,pedro.de.medeiros,daniel.castro,nuno.m.santos}@tecnico.ulisboa.pt
†Carnegie Mellon University, jd0@cmu.edu, nicolasc@andrew.cmu.edu
‡University of Waterloo, diogo.barradas@uwaterloo.ca
§INESC TEC / Universidade do Porto, bernardo.portela@fc.up.pt, jnsilva@inesctec.pt
¶LASIGE, Faculdade de Ciências, Universidade de Lisboa, blferreira@fc.ul.pt


Pedro Medeiros
INESC-ID/IST

Daniel Castro
INESC-ID/IST

Daniela Lopes
INESC-ID/IST

Jin-Dong Dong
CMU

Nicolas Christin
CMU

Nuno Santos
INESC-ID/IST

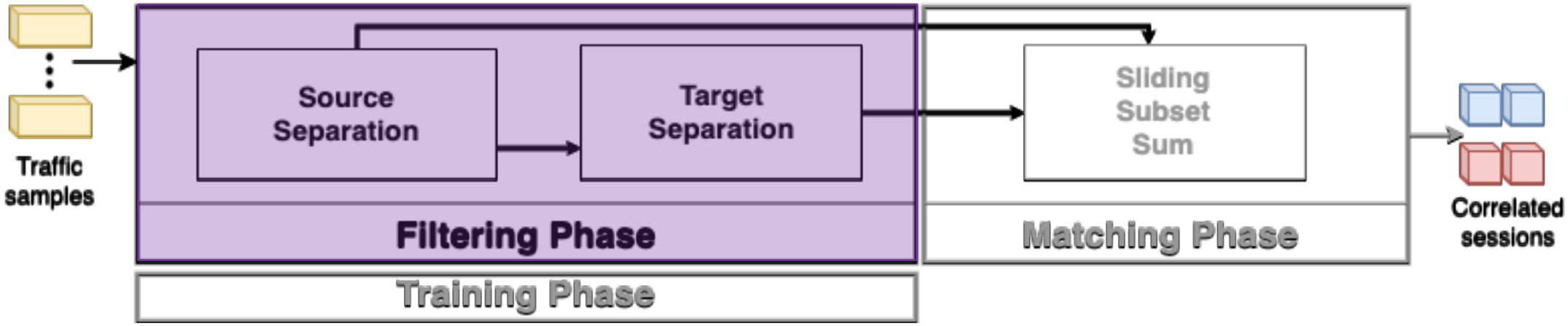
Diogo Barradas
University of Waterloo

Bernardo Ferreira
F. Ciências, U. Lisboa

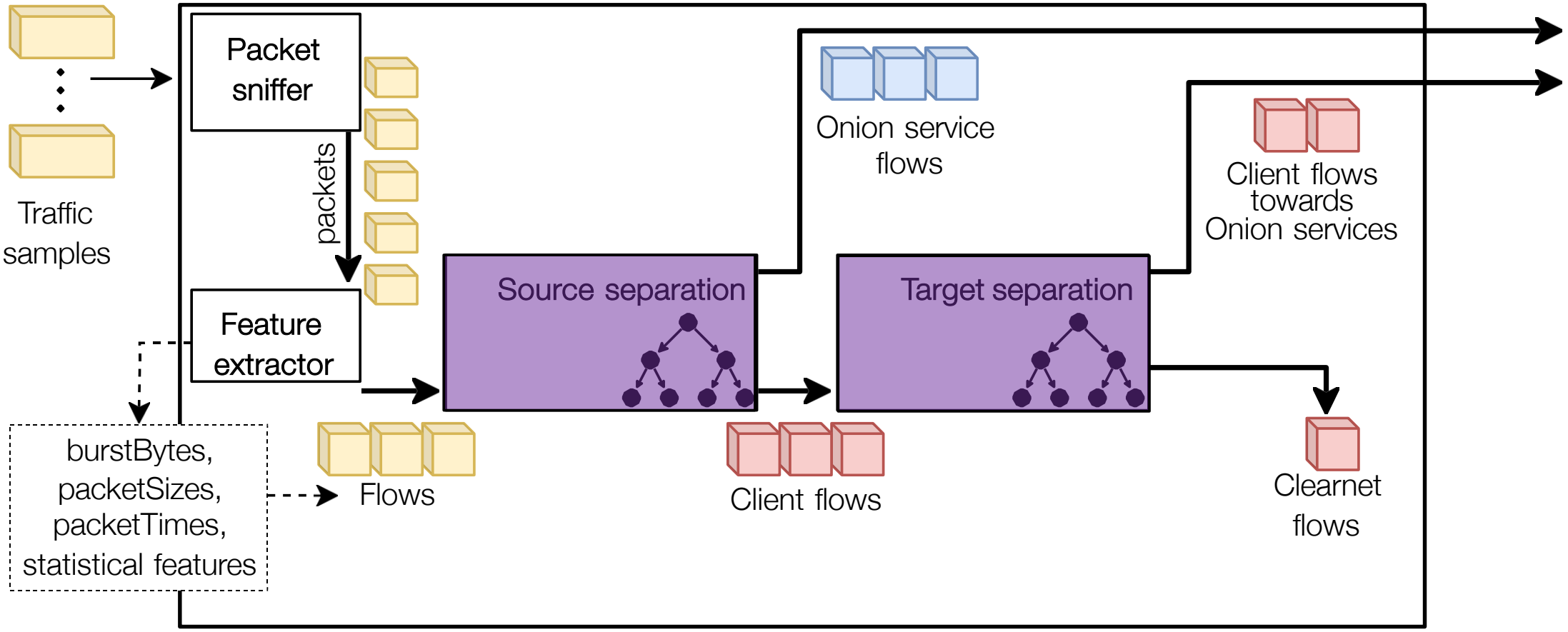
João Vinagre
INESC TEC

Bernardo Portela
FCUP / INESC TEC

The SUMo Pipeline

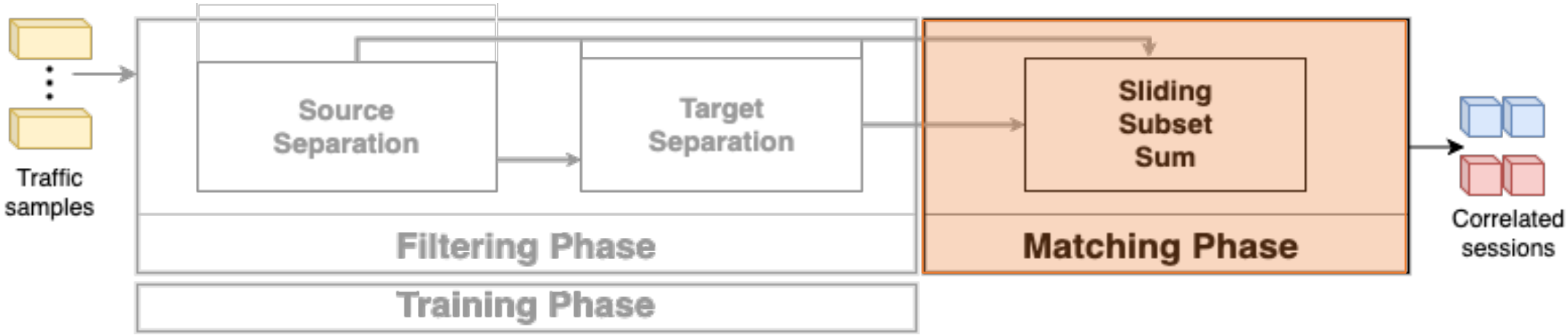


Distinguishing flows (by destination)

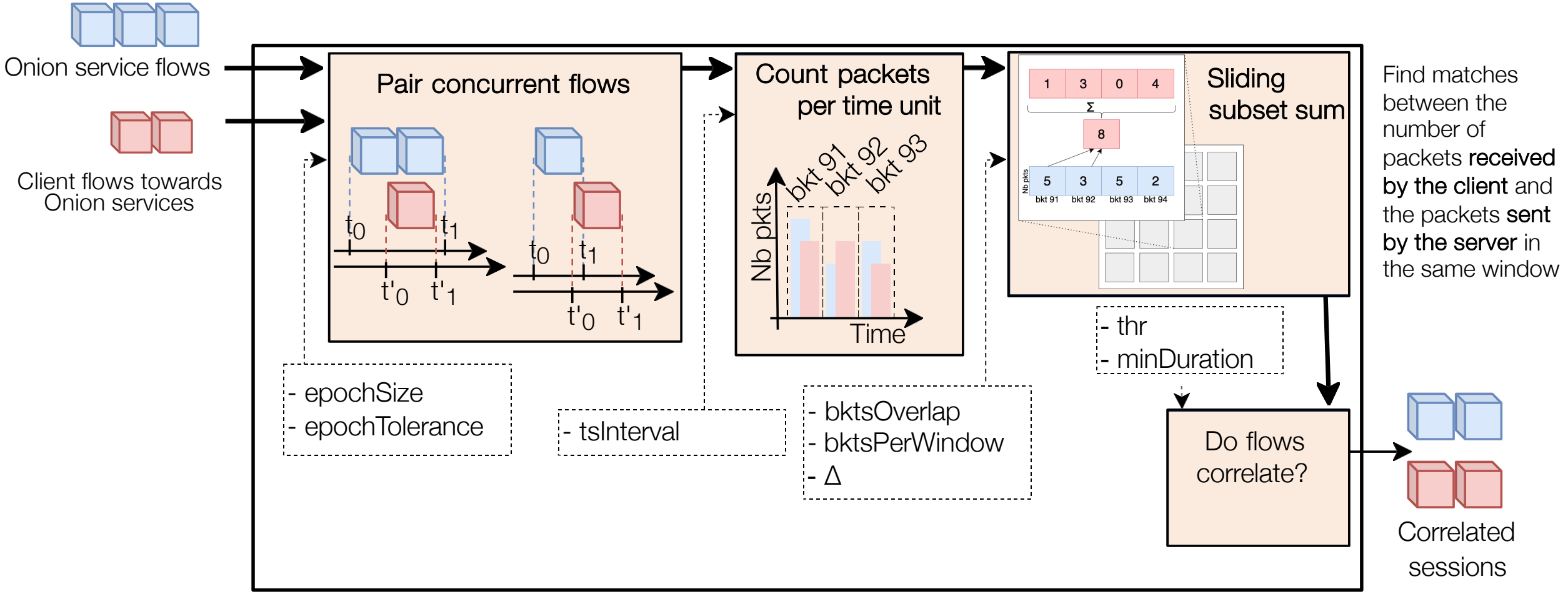


Filtering Phase (on local probes)

The SUMo Pipeline



Match clients with onion services



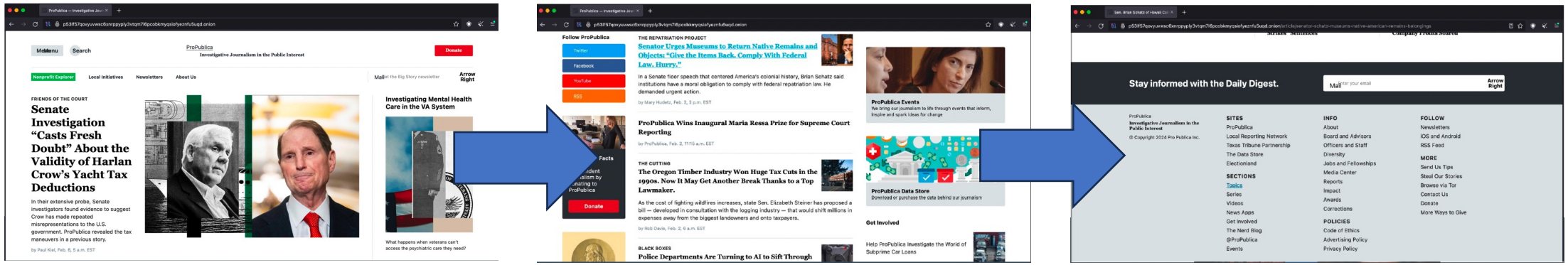
Find matches between the number of packets **received by the client** and the packets **sent by the server** in the same window

Matching Phase (on correlator)

How did we evaluate SUMo?

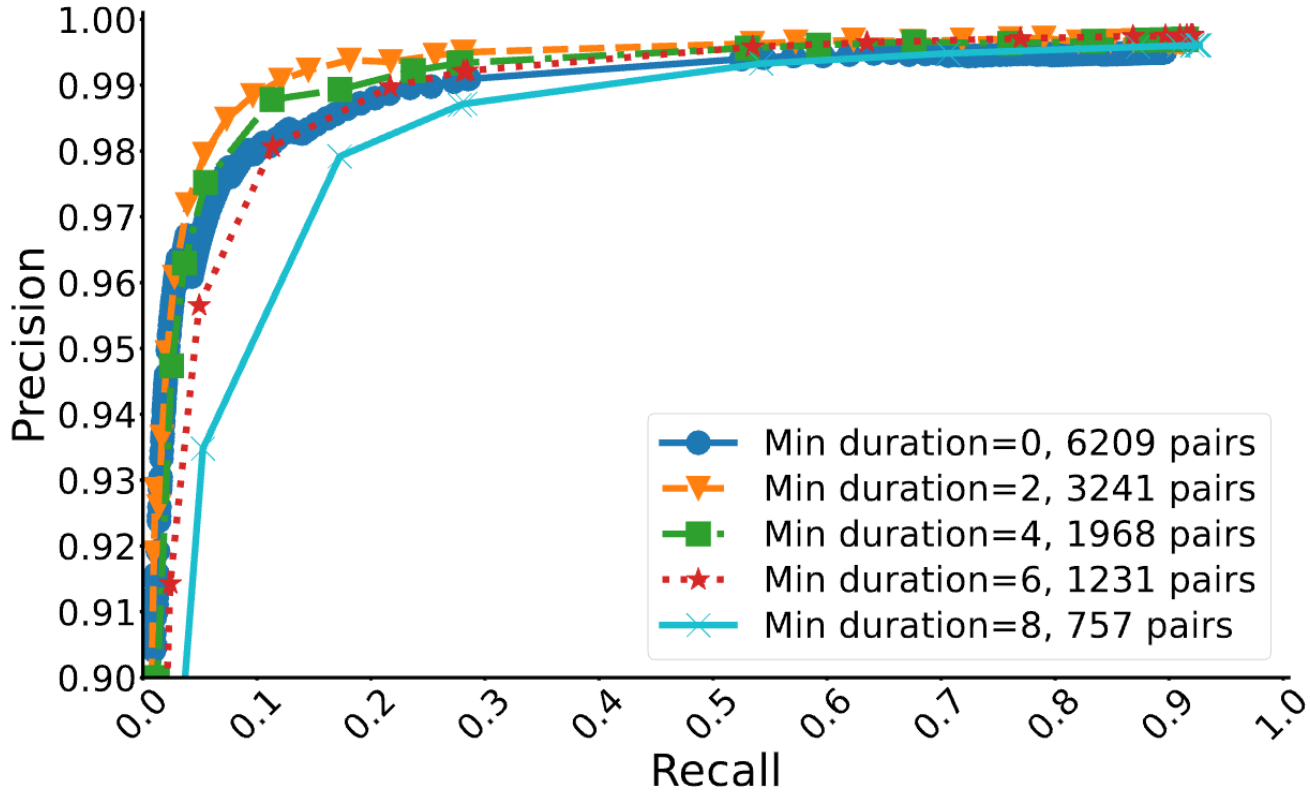
Experimental testbed

- Developed a framework to generate network traces (> 80,000 pcaps)
  - Geographical distribution of live nodes (clients and servers)
  - Hosted a set of webpages scraped from actual onion services
  - Modeled client requests' concurrency to .onions and Tranco top 150 sites
  - Emulated typical browsing behaviour to collect **browsing sessions**



SUMo can effectively correlate flow-pairs

- **Sessions of any duration**
  - 99.5% precision
  - 89.6% recall
- **Sessions > 6 minutes**
  - 99.8% precision
  - 92.1% recall



SUMo's pipeline is efficient

Phase	Stage	Training time (full data)	Inference Time (per flow)
Filtering	Source separation	4.3s	44.7ms
	Target separation	1.7s	35.0ms
Matching	Session correlation	-	32.6ms

Trains fast

Filtering classifiers are fast to train

Filters fast

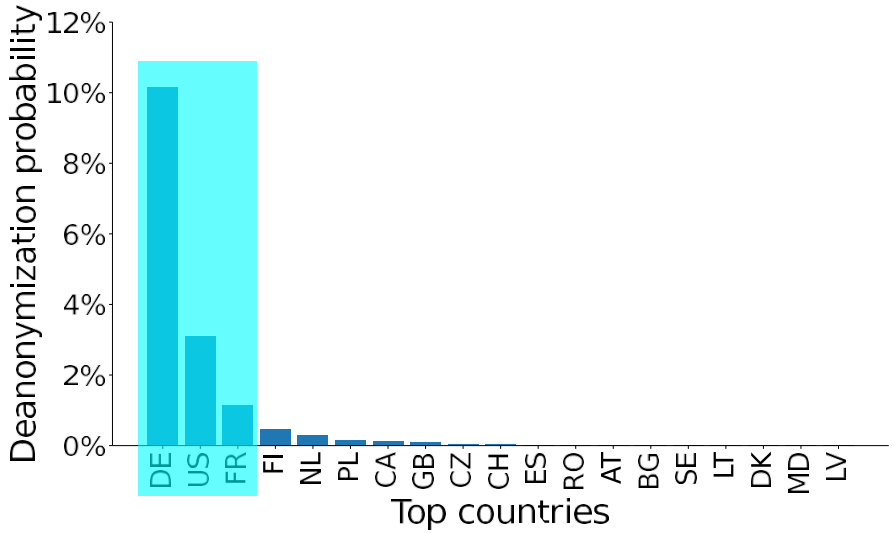
Filtering classifiers are fast to predict

Correlates fast

Sliding subset sum is fast to match (100x faster than SOTA correlation)

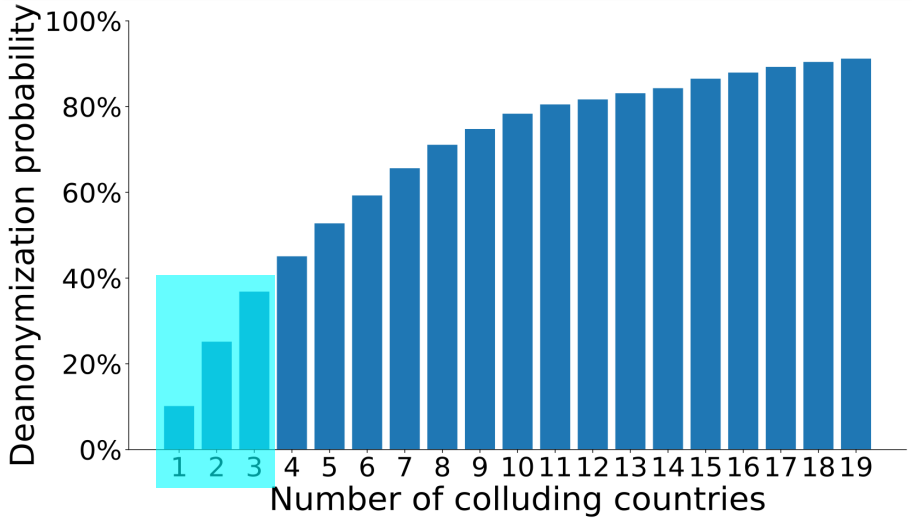
Traffic correlation on Tor is a realistic threat

- We verified client and onion service guard locations
  - for 40,000 random 6-hop circuits between our clients and Oses



Probability of both guard nodes being under the same jurisdiction

Guard node distribution is heavily skewed



Visibility over guard nodes under country collusion

Chances of deanonymization increase

Takeaways

- Onion services are instrumental for anonymity online
- The threat of traffic correlation on Tor is getting increasingly realistic
  - colluding entities can intercept a large fraction of Tor traffic
  - correlation algorithms are increasingly efficient and effective

- **Countermeasures**:
  - Increase guard node's geographical diversity
  - Pluggable transports (e.g. Brik [CoNEXT'23])
  - Client concurrency (e.g., fake client traffic)
  - Concurrent "multi-tab" requests

Diogo Barradas
diogo.barradas@uwaterloo.ca

Thank you!