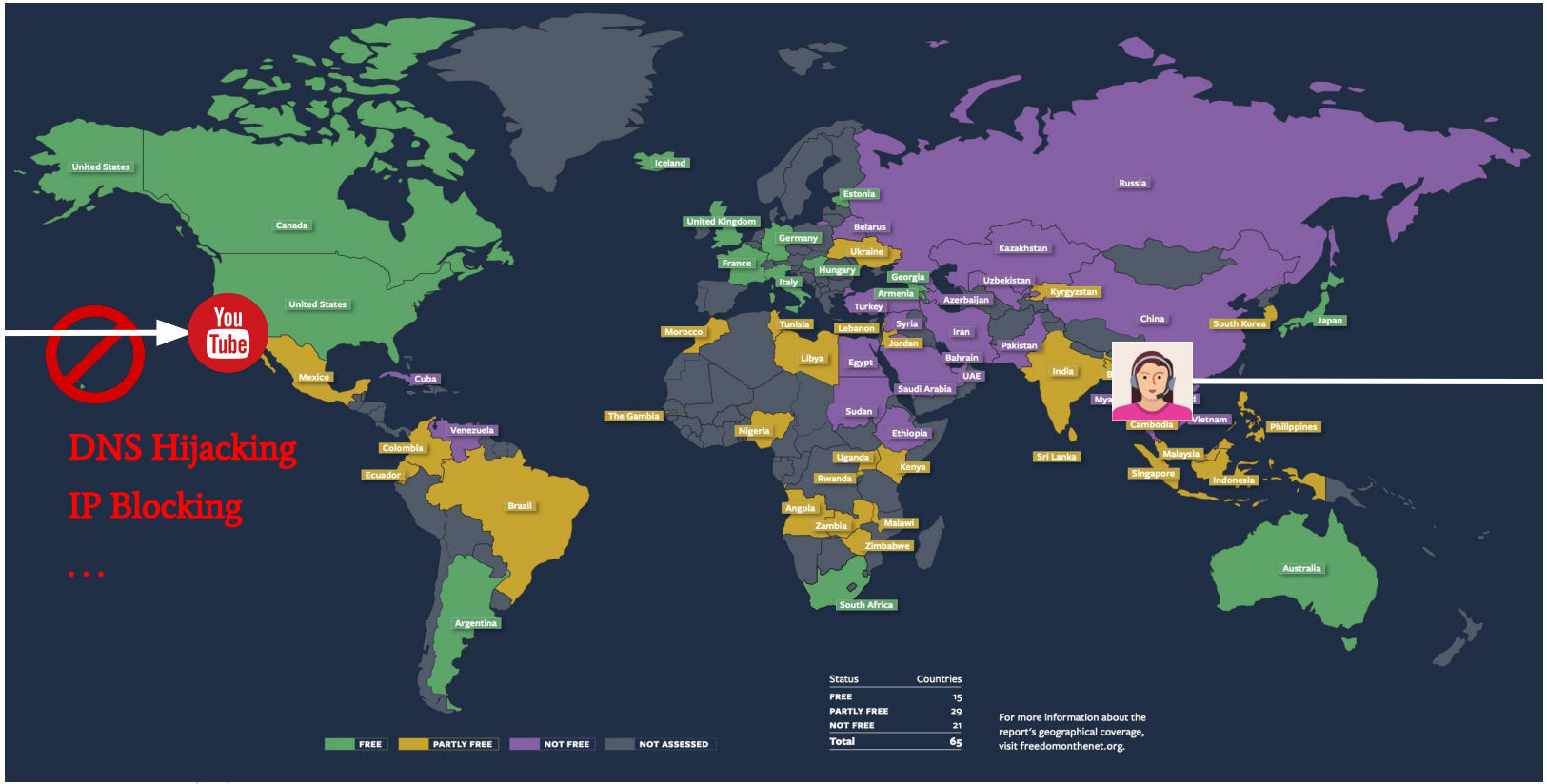
Towards a Scalable Censorship-Resistant Overlay Network based on WebRTC Covert Channels

Diogo Barradas Nuno Santos

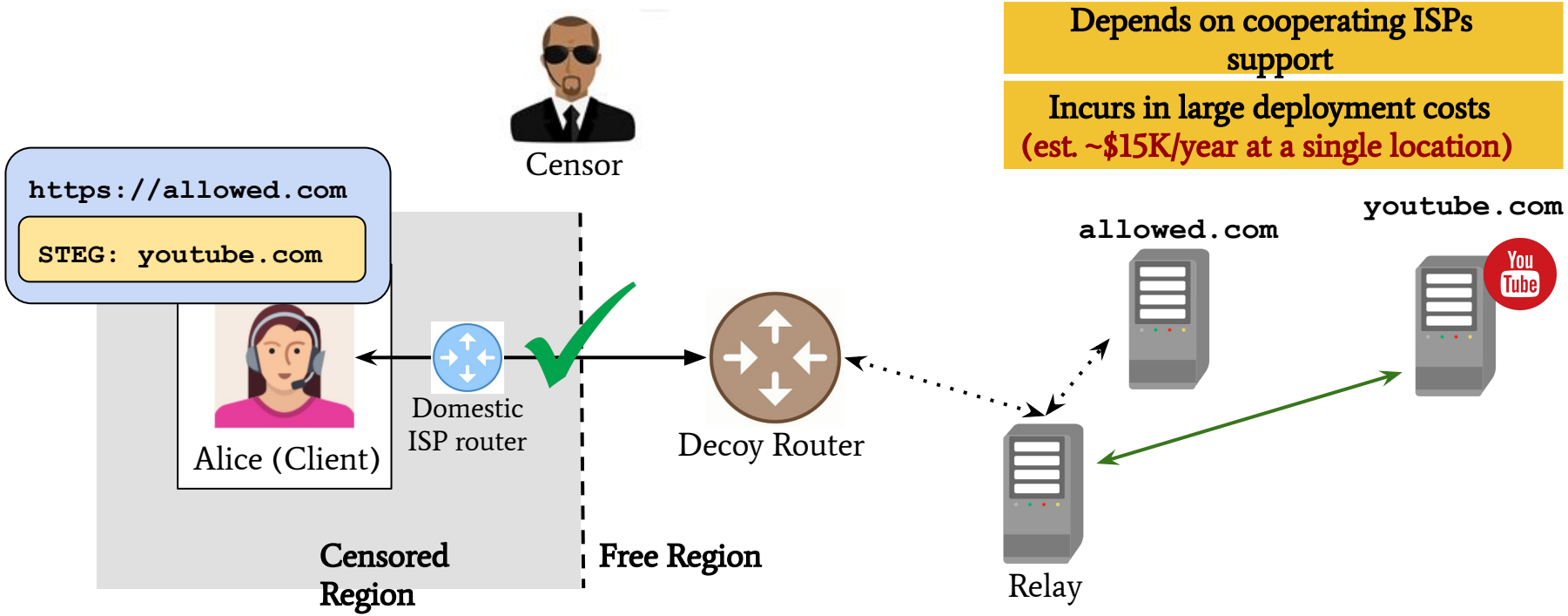
INESC-ID, Instituto Superior Técnico, Universidade de Lisboa

Internet Censorship is Widespread



Bypassing Censorship with Decoy Routing

e.g., TapDance [PETS'20]



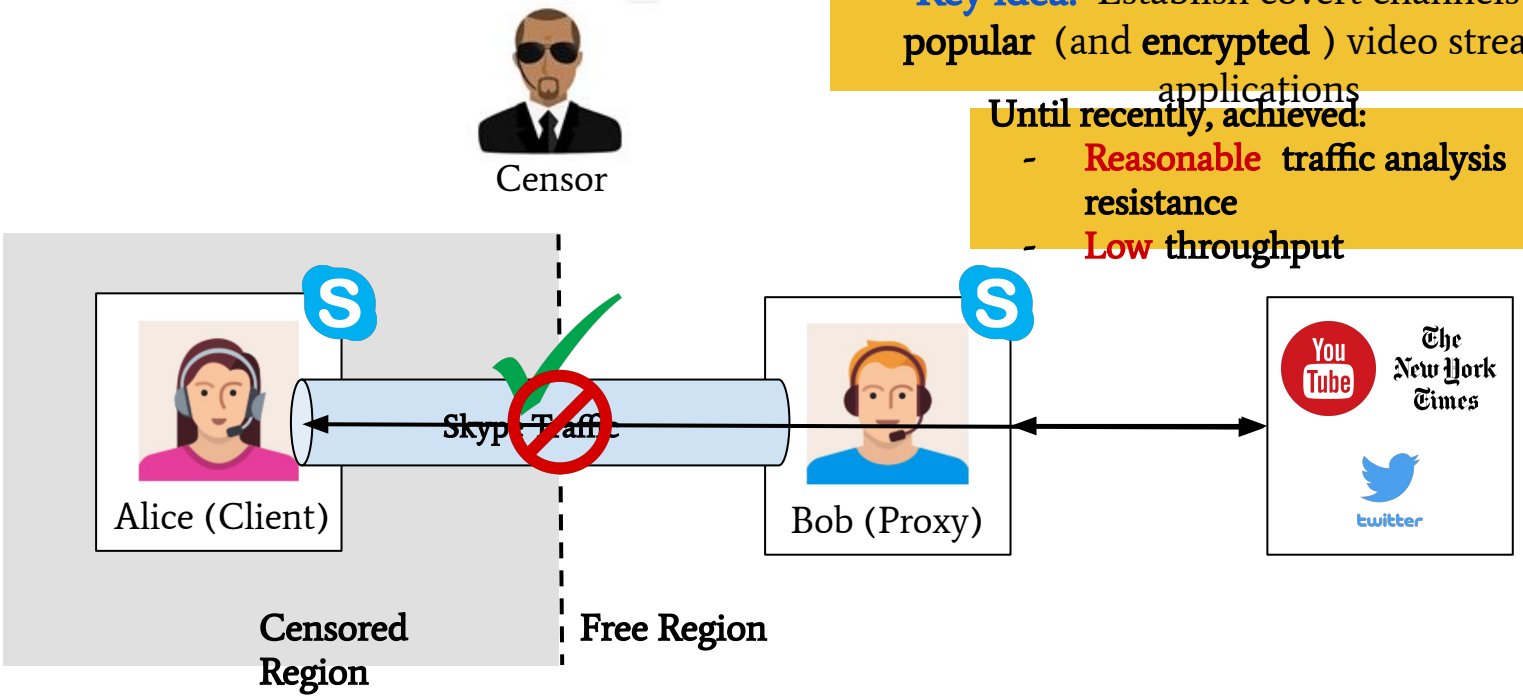
Bypassing Censorship with Multimedia Covert Channels

e.g. DeltaShaper[PETS'17]

Key idea: Establish covert channels over **popular** (and **encrypted**) video streaming applications

Until recently, achieved:

- **Reasonable traffic analysis resistance**
- **Low throughput**



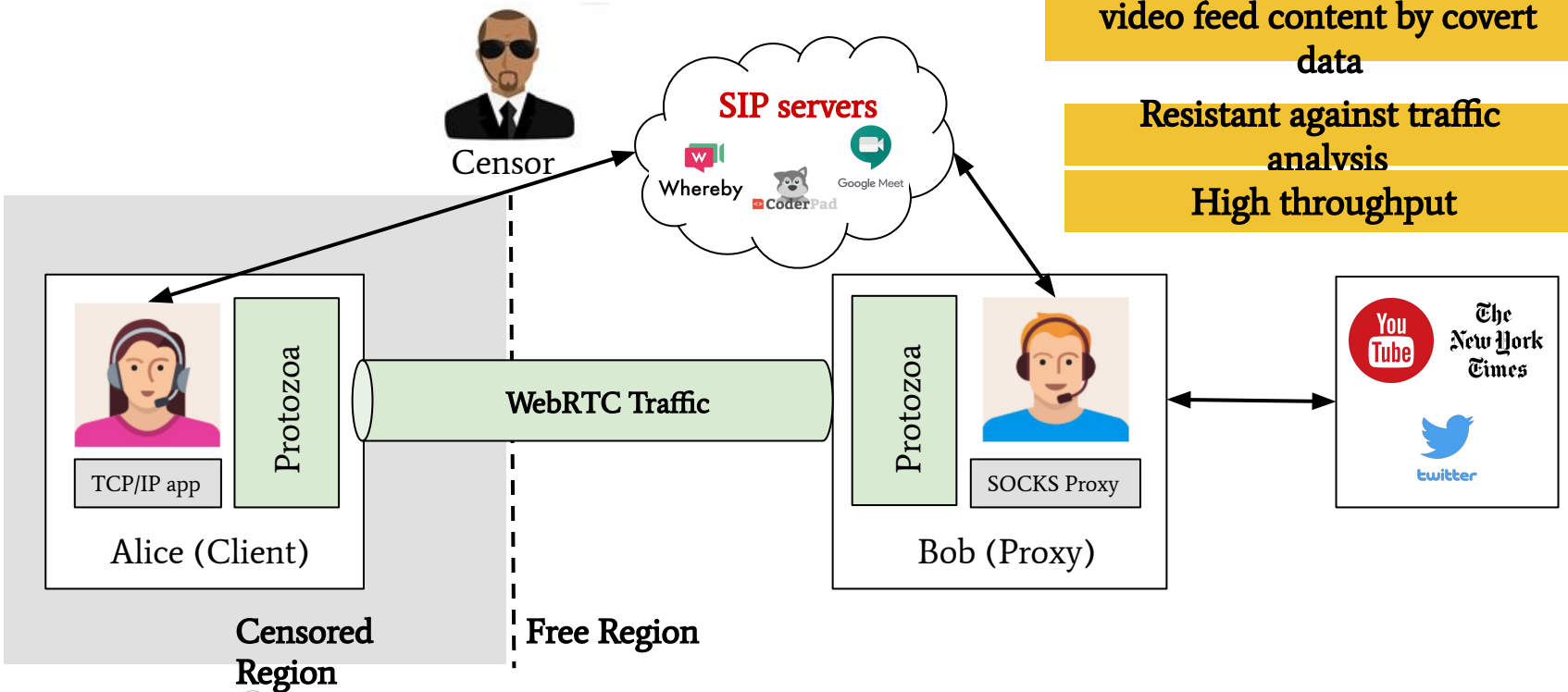
Bypassing Censorship with WebRTC

Protozoa [CCS'20]

Protozoa: Exchanges encoded video feed content by covert data

Resistant against traffic analysis

High throughput

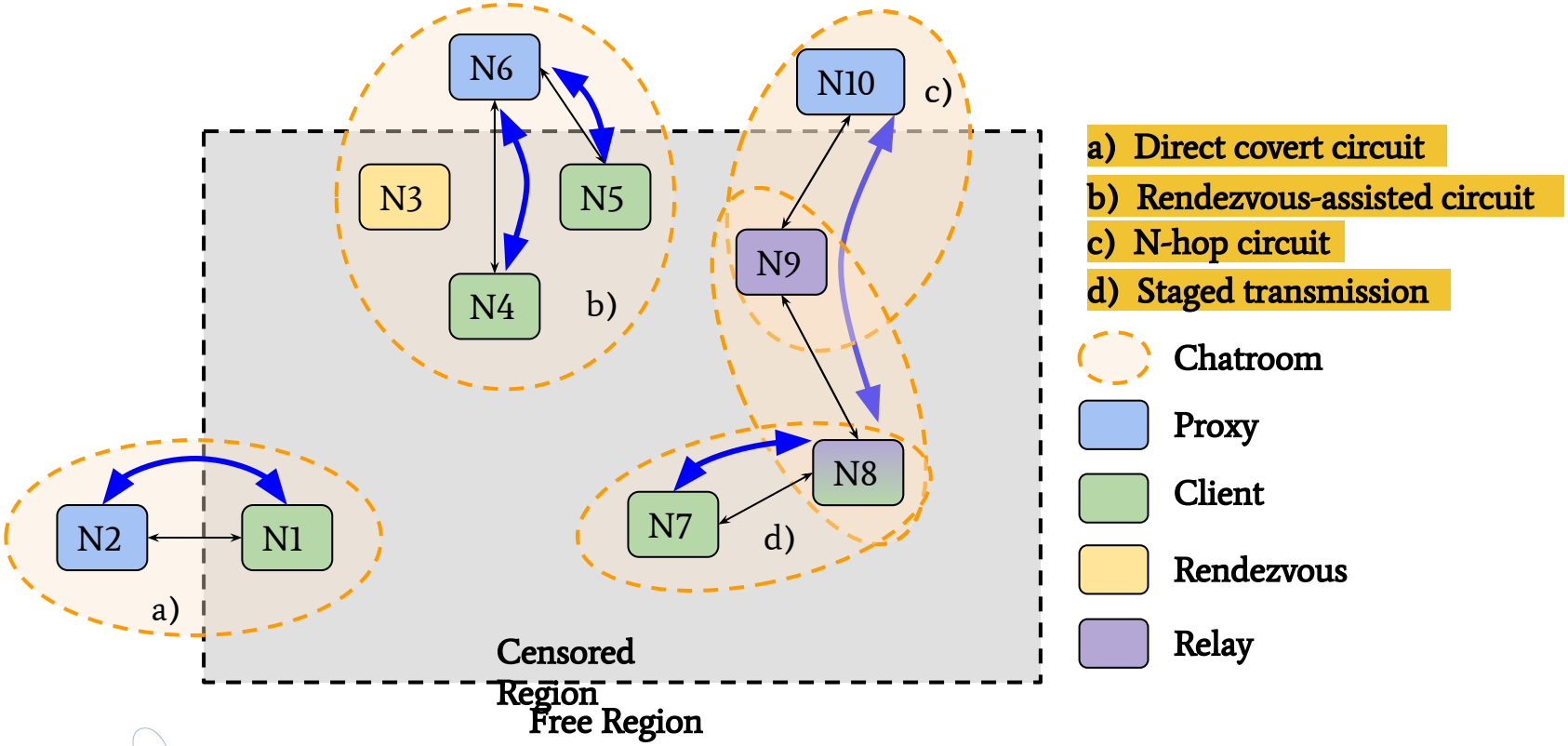


Limitations of Protozoa

- **Does not provide a mechanism for finding proxies**
- **Prone to censor attacks:**
  - Does not protect against **long-term user profiling**
  - Does not protect users against **censor-controlled WebRTC** services
  - Does not provide a defense mechanism against **Sybil nodes**

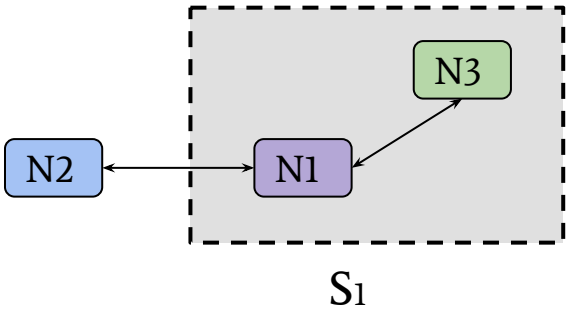
CRON: Censorship-Resistant Overlay Network

CRON Leverages Social Circles for Finding Proxies

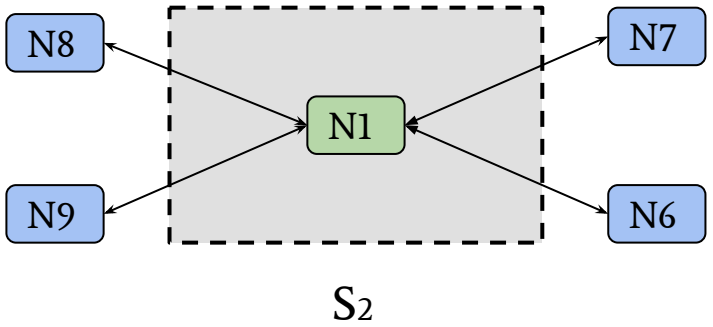


Attack Vector #1 - Long-Term Profiling of Users

- **Adversaries may build user profiles to identify uncommon behavior**
  - (S_1) Simultaneous video calls (relay nodes)
  - (S_2) Uncommon call parties
  - (S_3) Uncommon call times, frequency, and duration



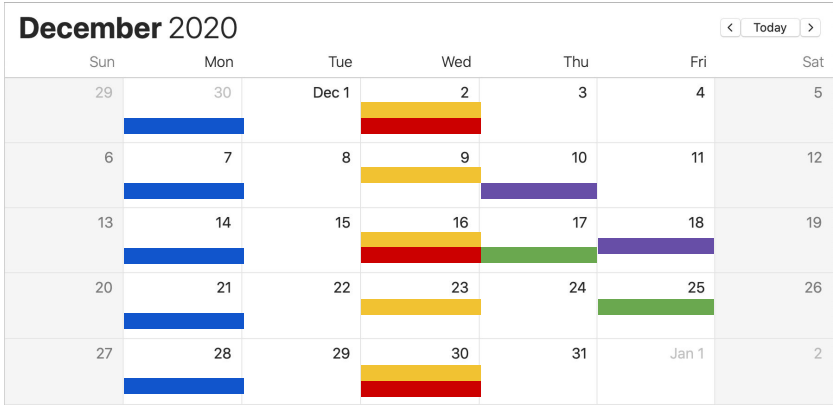
S_1



S_2

Prevent the Identification of Users due to Long-Term Profiling

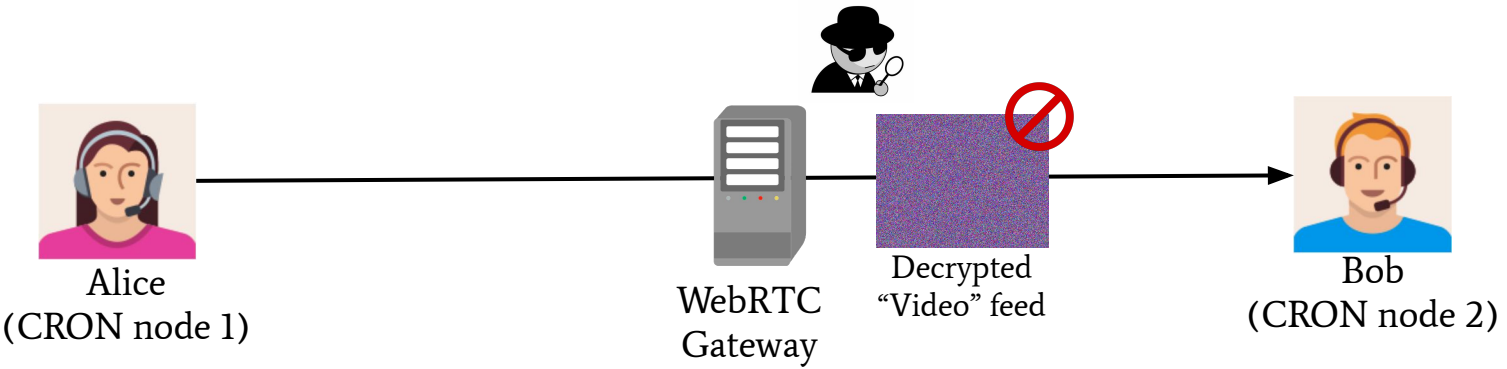
- **Passive Mode**
  - **Monitor user call patterns**
  - Explore **windows of opportunity**
    - e.g., weekly video meeting



- **Active Mode**
  - Take advantage of the **inherent variability** in user patterns
  - Introduce **bounded variability**
    - call times, frequency, duration

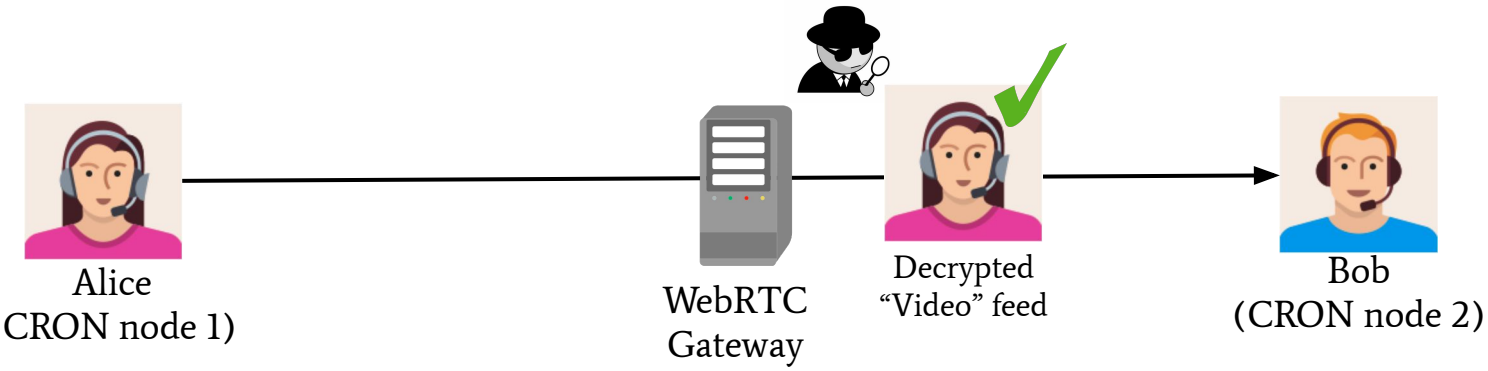
Attack Vector #2 - Censor-controlled WebRTC Services

- **Adversary-controlled WebRTC services are** **prone to MITM attacks**
  - Hijack user identity during call signalling / establishment phase
  - Force calls through WebRTC gateways
  - Allow an adversary to **decrypt / inspect media content**



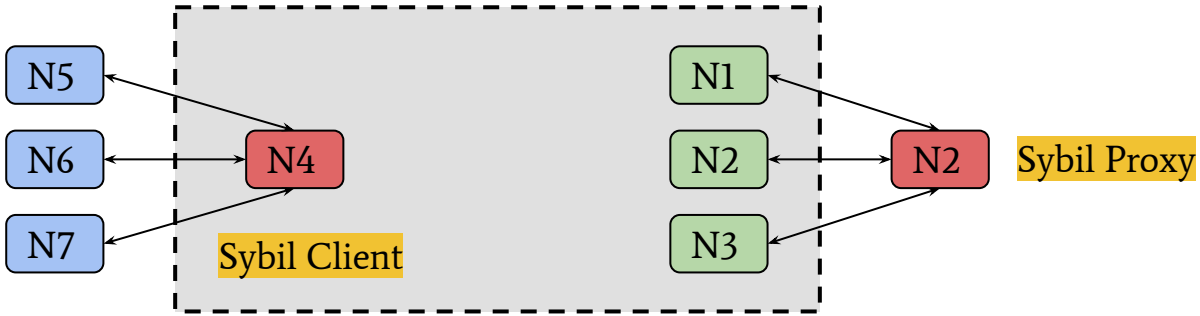
Prevent Adversarial WebRTC Apps from Detecting Covert Channels

- **New flavor of CRON circuits:** ***Stego*** **circuits**
  - Embed covert data in video frames using **video steganography** techniques
  - Protect steganographic content with public key exchanged out-of-band



Attack Vector #3 - Identifying CRON Users using Sybils

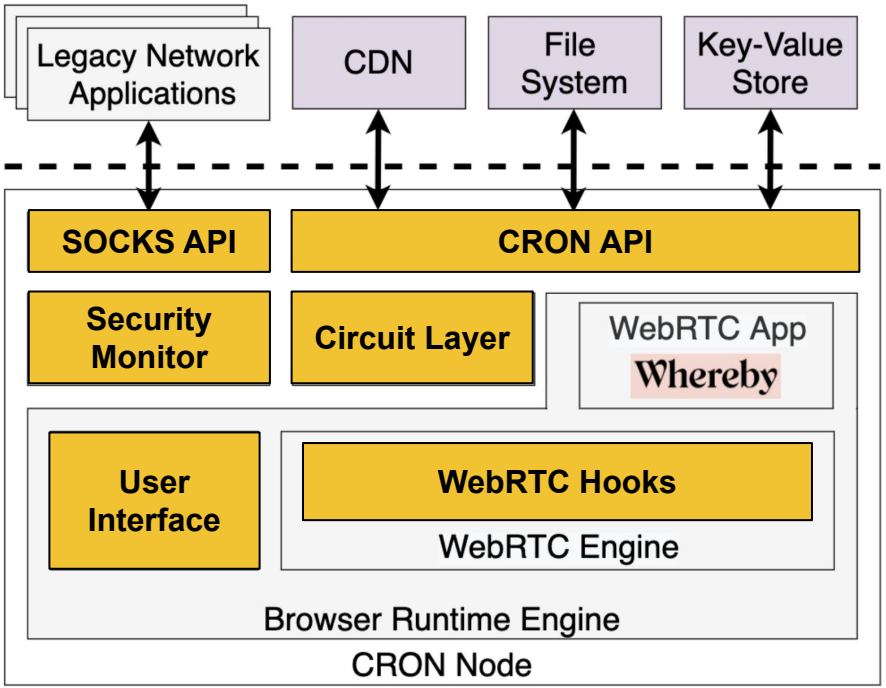
- **Adversaries can infiltrate state-controlled agents into the network**
  - Issue **fake client requests** to track down legitimate CRON proxy nodes
  - Offer **fake proxying or relaying services**



Prevent Users from Being Identified by Sybil Nodes

- Avoid indeliberate trust **in every CRON node**
  - Discretionary trust system
- **Trust establishment is centred in each user, in** two rings of trust
  - **1st degree trustees:** Nodes in a user's direct social circle
  - **2nd degree trustees:** Nodes that are "friends-of-a-friend"
- **Circuits are only established if all involved nodes are** mutually trusted
  - Circuit creation is not an unilateral decision
  - 2nd degree trustees can still be used for establishing N-hop circuits

Envisioned CRON Architecture



User Interface:

Set functioning mode (client / proxy)
Assign levels of trust

Circuit Layer:

Manage regular / stego circuits
Mitigate profiling attacks

Security Monitor:

Check whether nodes are trusted
Check location of proxies
Content whitelisting

CRON / SOCKS API:

Support distributed applications

Conclusions

- **We presented CRON (Censorship-Resistant Overlay Network)**
  - Distributed system of nodes interlinked by WebRTC video channels
  - **Goal:** Tackle multiple limitations of proxy-based multimedia covert channels
- **Exposes an API for building censorship-resistant distributed applications**
  - CDNs, distributed file systems, key-value stores, etc.

Discussion:

- How can we accurately profile WebRTC users across sessions?
- Will the performance impact of stego circuits disable some CRON apps?
- Can we detect Sybil nodes and make them accountable?

Thank You!

`https://web.ist.utl.pt/diogo.barradas`

Limitations of Protozoa

- **Does not provide a mechanism for finding trusted peers**
  - Users with no connections abroad are prevented from using the system
- **Does not provide a defense mechanism against Sybil nodes**
  - A censor can enumerate clients and proxies
- **Does not protect users against censor-controlled WebRTC services**
  - Opens the possibility for inspection of unencrypted covert traffic
- **Does not protect against long-term user profiling**
  - Adversaries can try to spot unusual client behavior