

# The Parrot Is Dead: Observing Unobservable Network Communications. (May 2013)

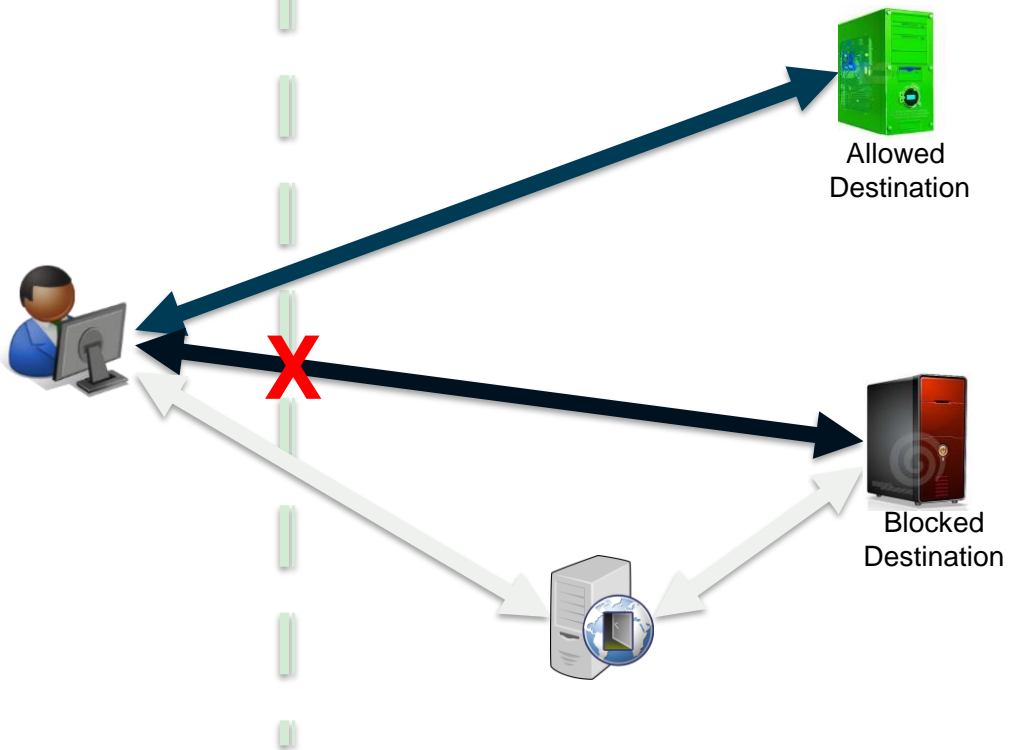
Amir Houmansadr Chad Brubaker Vitaly Shmatikov



**Censorship Region**



**The Internet**



Allowed  
Destination

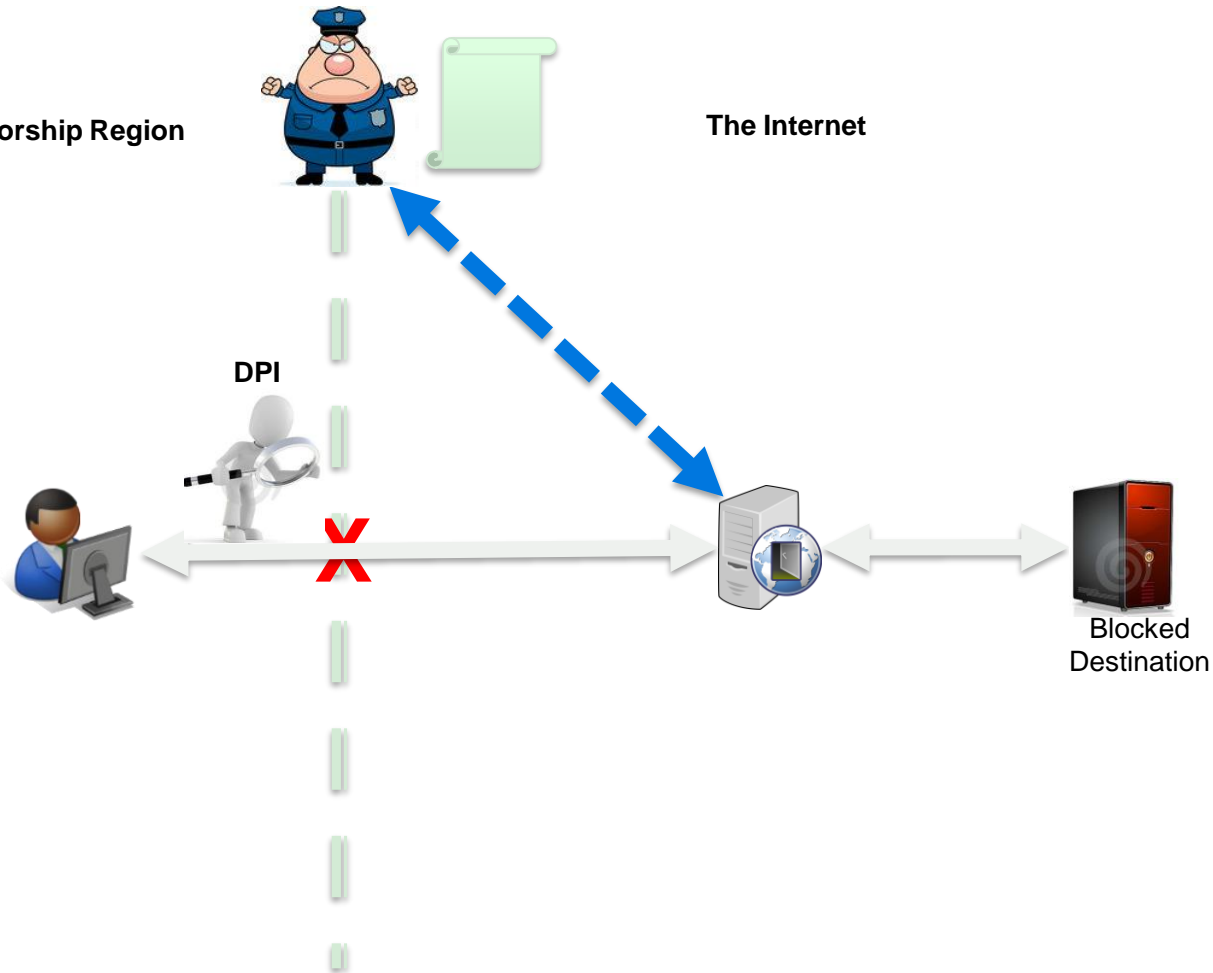
Blocked  
Destination

**Censorship Region**

**The Internet**

**DPI**

**Blocked  
Destination**



# Goal of censorship-resistance

**Increasingly used by people in non-democratic countries to bypass restrictions on Internet access, share information, browse websites prohibited by the regime, etc.**

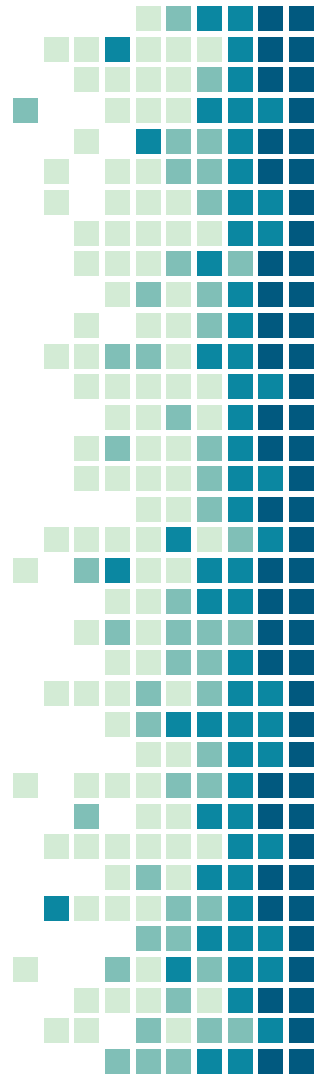
such as the Tor anonymity network

## **Motivations:**

- Unavailability of anonymous communication systems to users
- Tor patterns is still recognizable and frequently blocked

## **Challenges:**

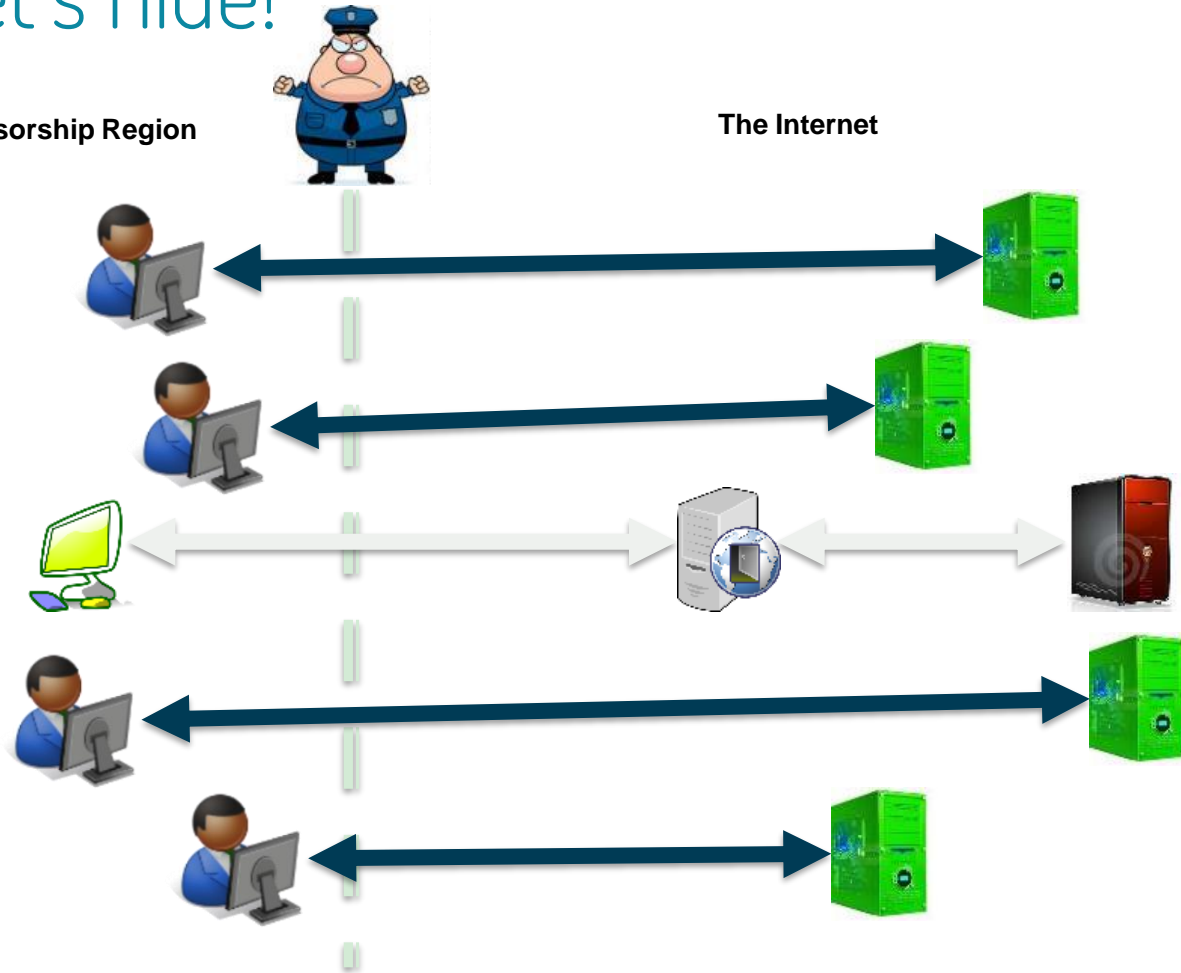
- Improved technical capabilities of government censors, i.e. real-time deep-packet inspection and traffic analysis
- Some tools were developed to detect Skype parrots



# Let's hide!

Censorship Region

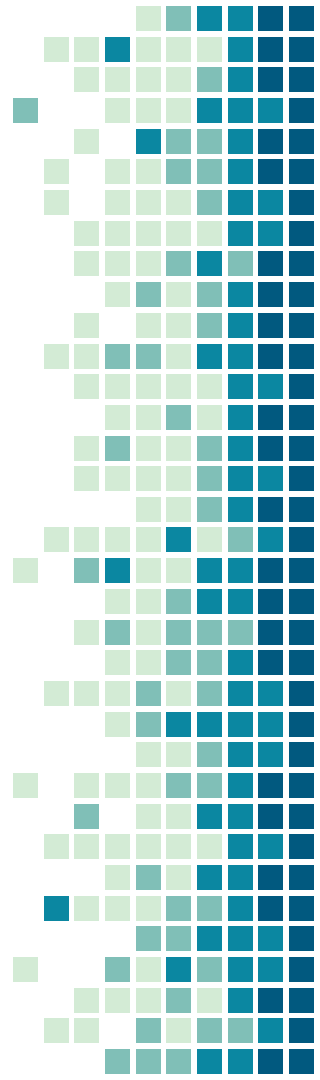
The Internet




# Current Solutions: parrot circumvention systems

## How they work:

- Depend on unobservability by imitating popular applications such as Web browsers and Skype clients
- Hide the traffic and make it indistinguishable from the protocol they are trying to imitate.
- Imitation targets must be common protocols.
  - Skype- Morph hides Tor traffic by mimicking Skype video calls
  - StegoTorus mimics Skype and/or HTTP
  - CensorSpoofer mimics SIP-based Voice-over-IP



“ *Unobservability: means that a censor can neither recognize the traffic generated by the circumvention system, nor identify the endpoints engaged in circumvention*



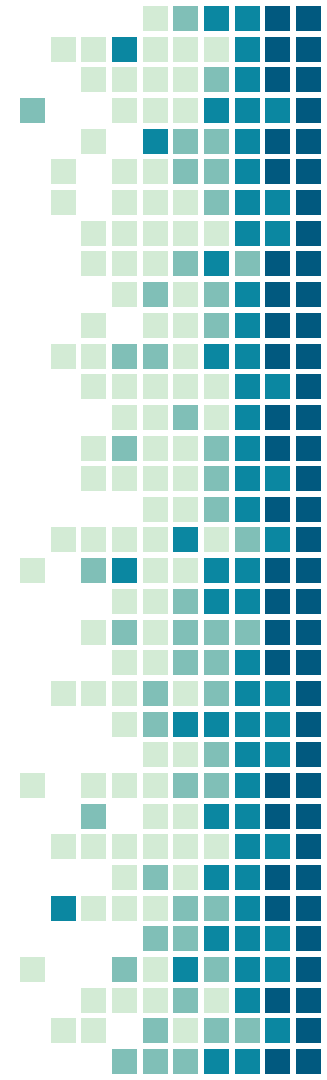
What's, uh...  
What's wrong with it?

'E's dead, that's what's wrong  
with it!



# These systems are not reliable

- Completely fail to achieve unobservability
- Can be recognized easily only by local network adversary
- Discrepancies between their imitation and the genuine protocol implementation, i.e. SkypeMorph and StegoTorus fails to mimic TCP channel.
- Has to mimic the protocol as well as the specific implementations
- Incomplete imitation = high detection probability

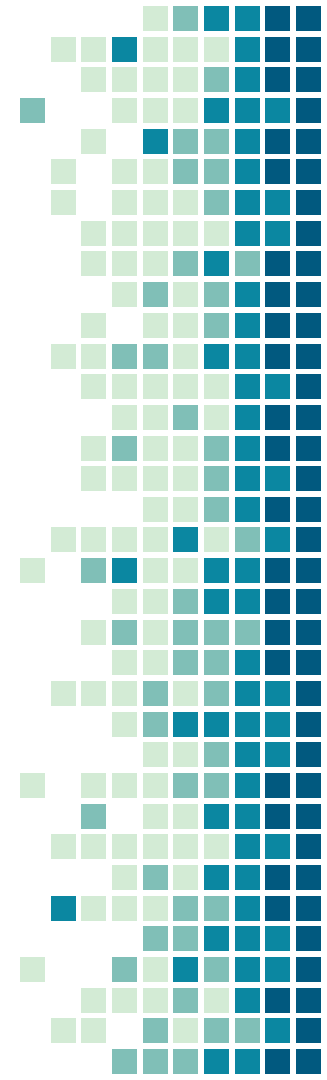


# PARROT CIRCUMVENTION SYSTEMS

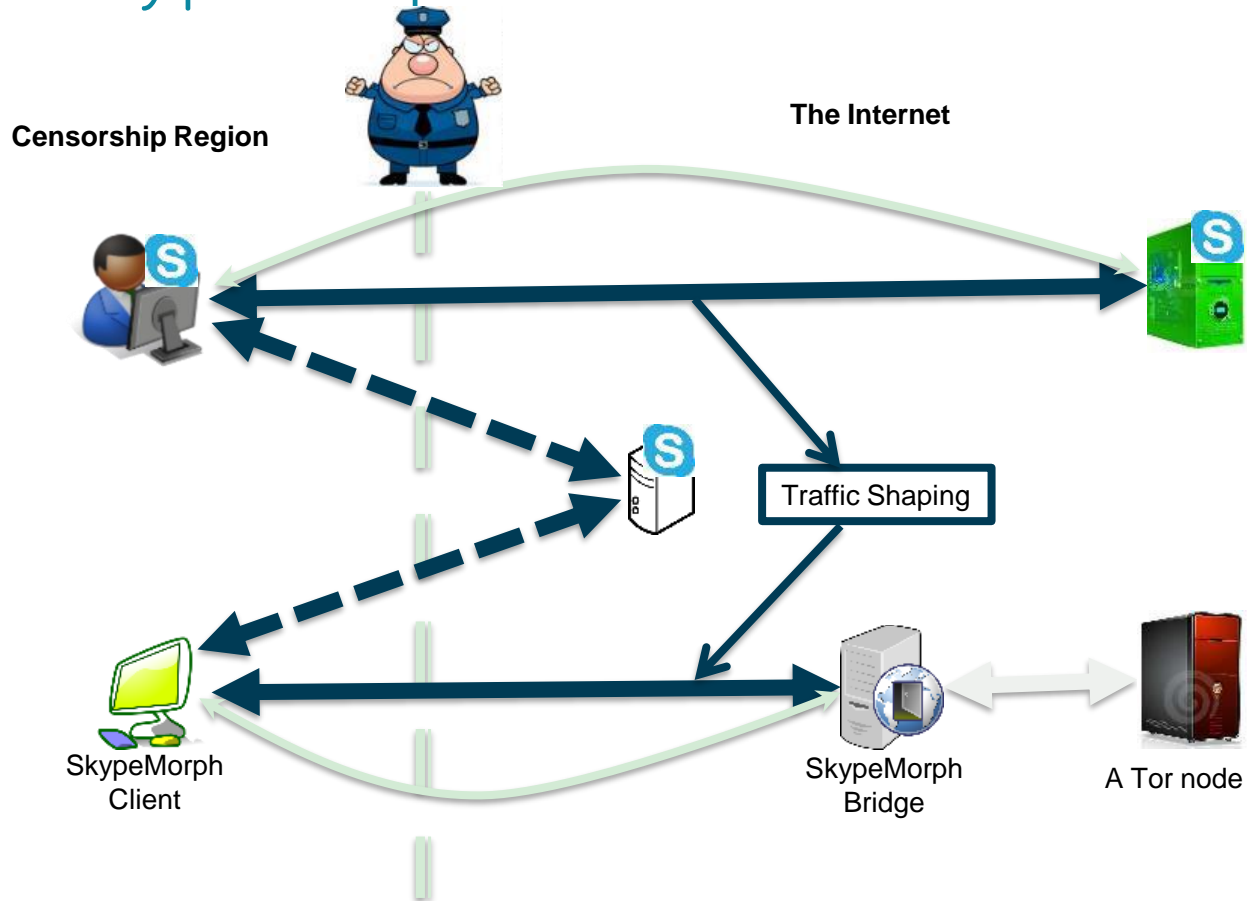


# 1. SkypeMorph

- A pluggable transport for Tor
- Intended to make the traffic between a Tor client and a Tor bridge look like a Skype video call
- Probes aimed quite directly at Tor bridges
- Fail even against the weakest censor (passive)



# SkypeMorph

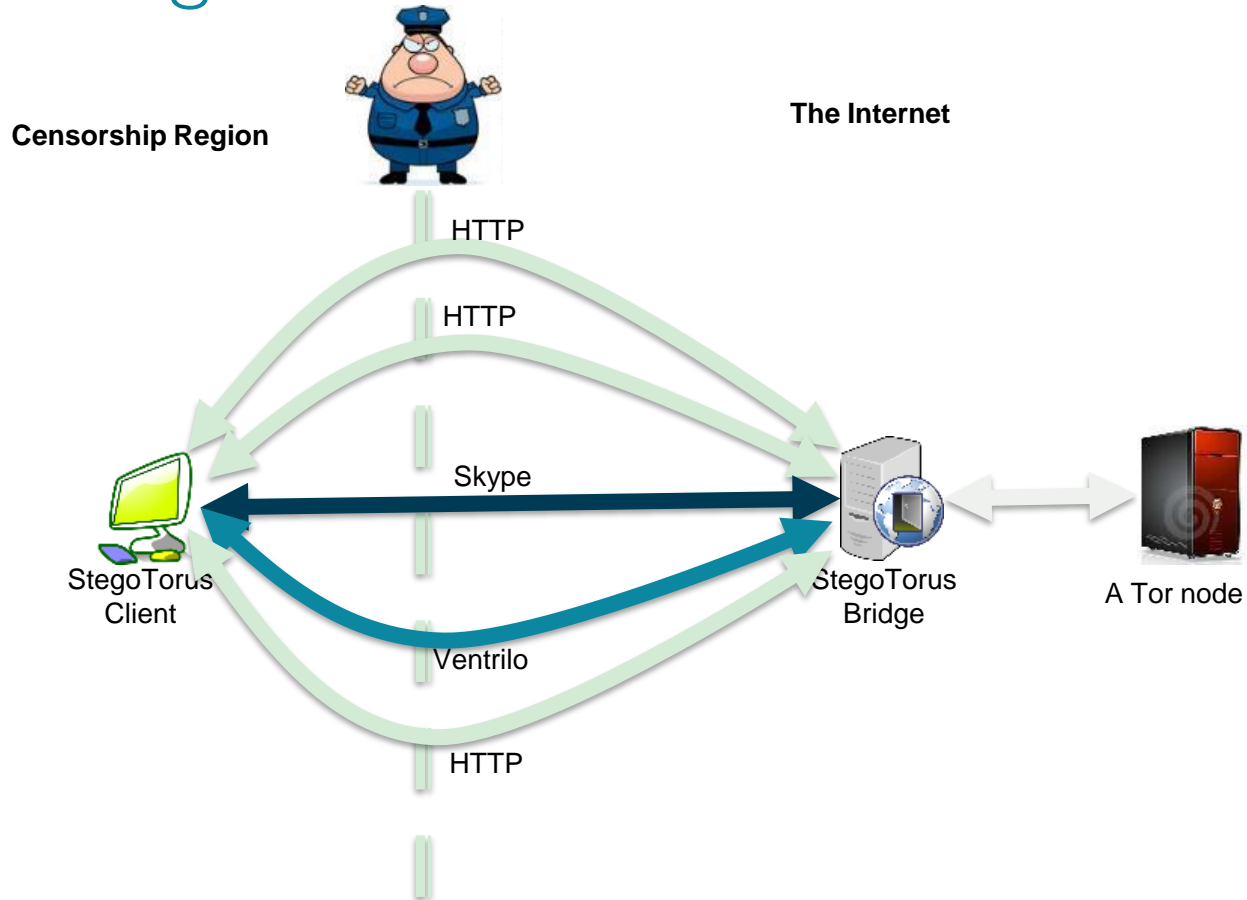


## 2. StegoTorus

- Adds chopping and steganography to Tor clients and bridges
- Uses a database of genuine, previously collected Skype and Ventrilo packet traces
- Two types:
  1. StegoTorus-Embed aims to mimic a P2P connection such as Skype or Ventrilo VoIP
  2. StegoTorus-HTTP aims to mimic unencrypted HTTP traffic, and hides data in files.
- Censors can perform IP, content, and statistical (OM)



# StegoTorus



# Session Initiation Protocol (SIP)

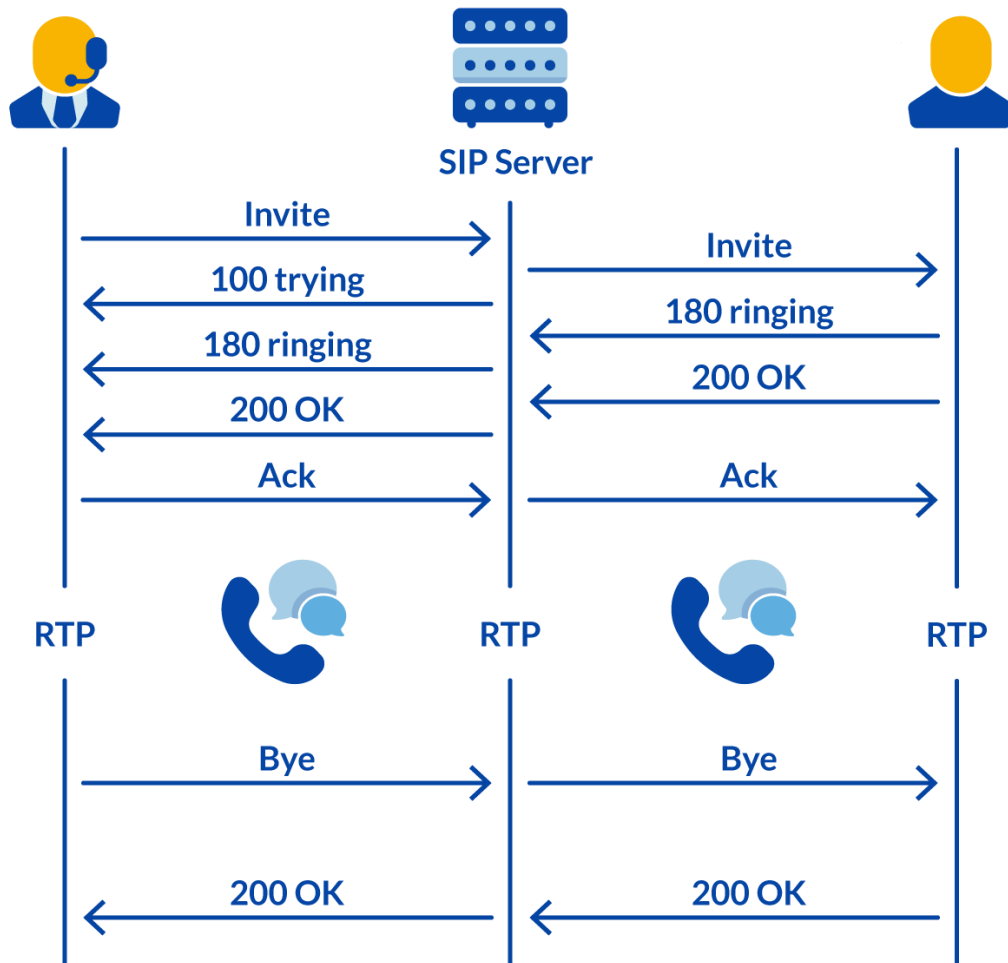
- SIP is an application-layer protocol
- runs over TCP or UDP.

## **Comprised of:**

- User agents: have registered SIP IDs and run SIP client software
- Location services: the VoIP provider's database
- Registrar servers: receive SIP registration requests
- Proxy servers: forward call requests

- Real-time Transport Protocol (**RTP**) is a standard for media transmission.
- Real-time Transport Control Protocol (**RTCP**): controls RTP connection
- Both run over UDP and have encrypted versions

# SIP

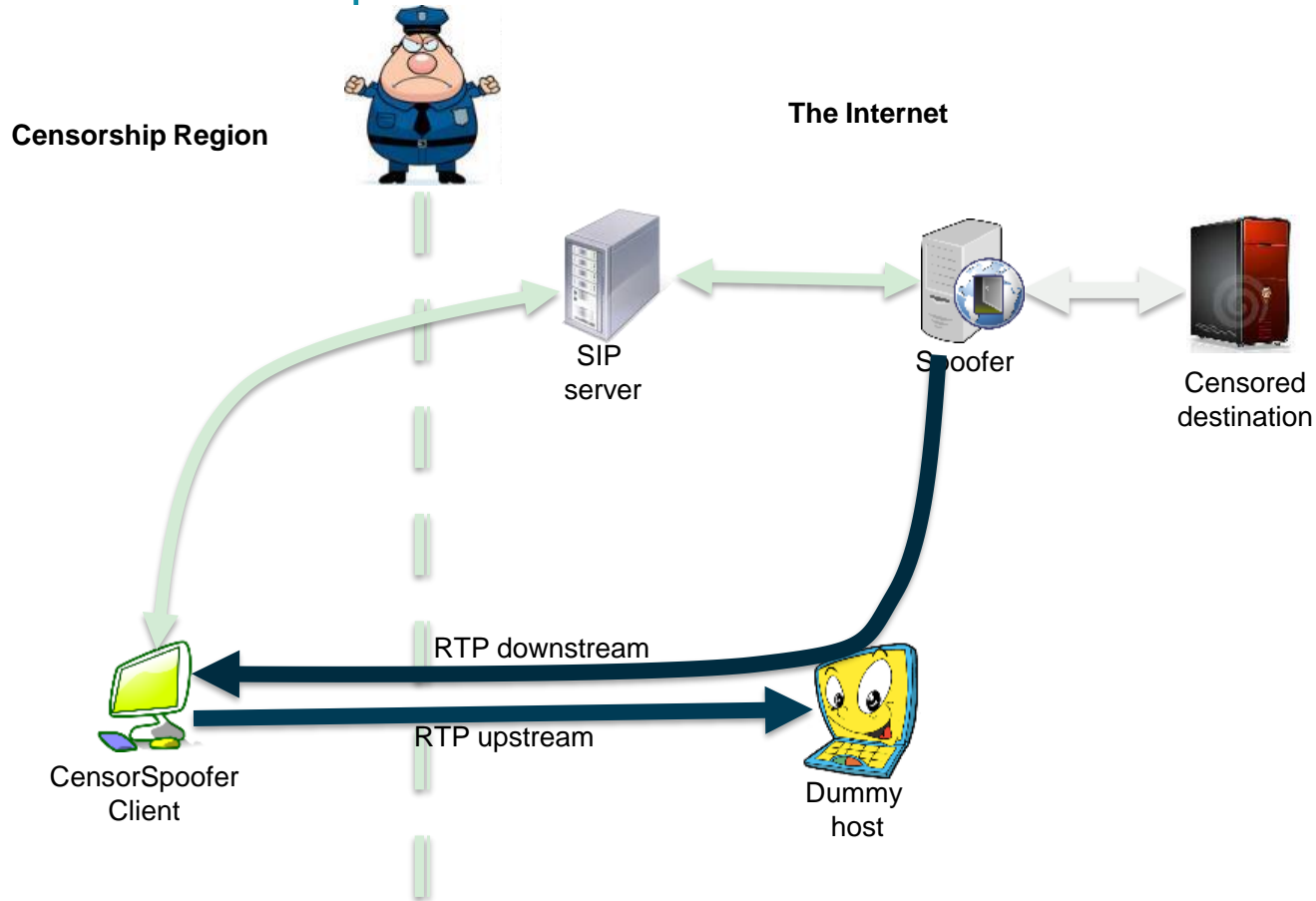




### 3. CensorSpoofer

- IP spoofing to obfuscate the server's identity
  - Mimics VoIP traffic to obfuscate traffic patterns
  - Mimics UDP-based VoIP traffic
  - mainly designed for censorship-resistant Web browsing
  - CensorSpoofer decouples upstream and downstream connections.
  - Principle: The server hides HTTP responses by mimicking P2P traffic from an oblivious dummy host.  
Dummy hosts are chosen by port-scanning random IPs
- considers a “state-level adversary”

# CensorSpoofer



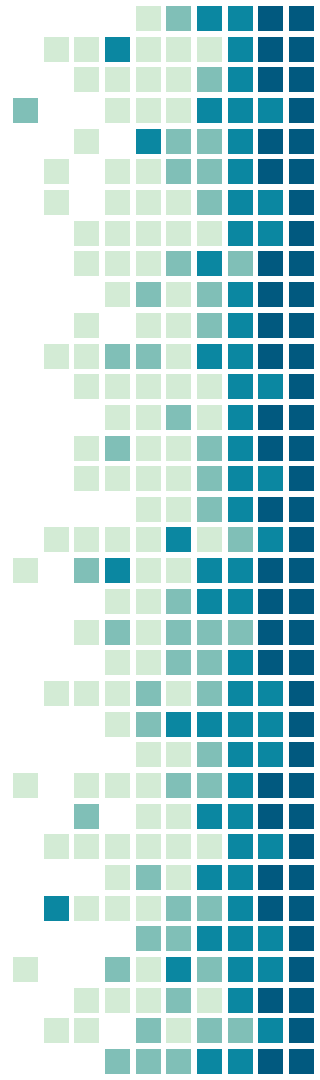
# Imitation Requirements

<b>Correct</b>	<b>SideProtocols</b>
<b>IntraDepend</b>	<b>InterDepend</b>
<b>Err</b>	<b>Network</b>
<b>Content</b>	<b>Patterns</b>
<b>Users</b>	<b>Geo</b>
<b>Soft</b>	<b>OS</b>

# Adversaries

- 1- Passive attacks: traffic analysis, deep packet inspection, and behavioral analysis
- 2- Active attacks: Typical techniques are delaying, dropping, or injecting packets into existing connections, modifying packet contents..etc.
- 3- Proactive attacks: probe to identify network entities involved

Can be: Local adversaries or State-level adversary



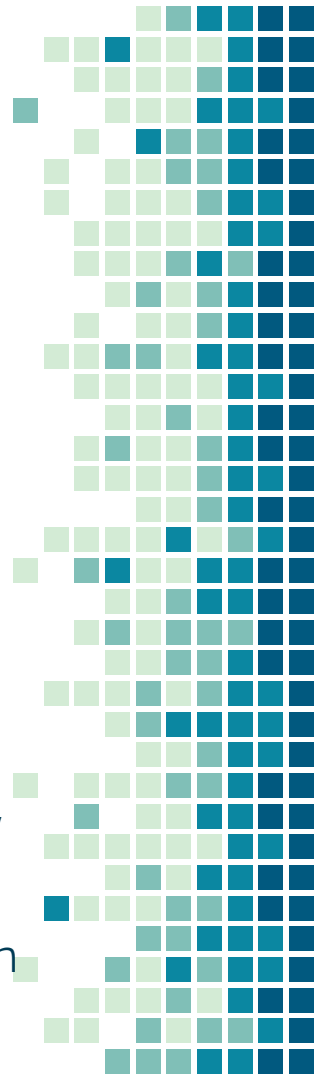
# EXPERIMENT

## Setup:

- Obtained latest implementations of all analyzed parrot systems and their imitation targets (Skype, Ekiga, etc.)
- Executed the software in VirtualBox3 virtual machines (VMs).
- Intel i5 CPU and 4GB of RAM
- A Tor bridge

## Goal:

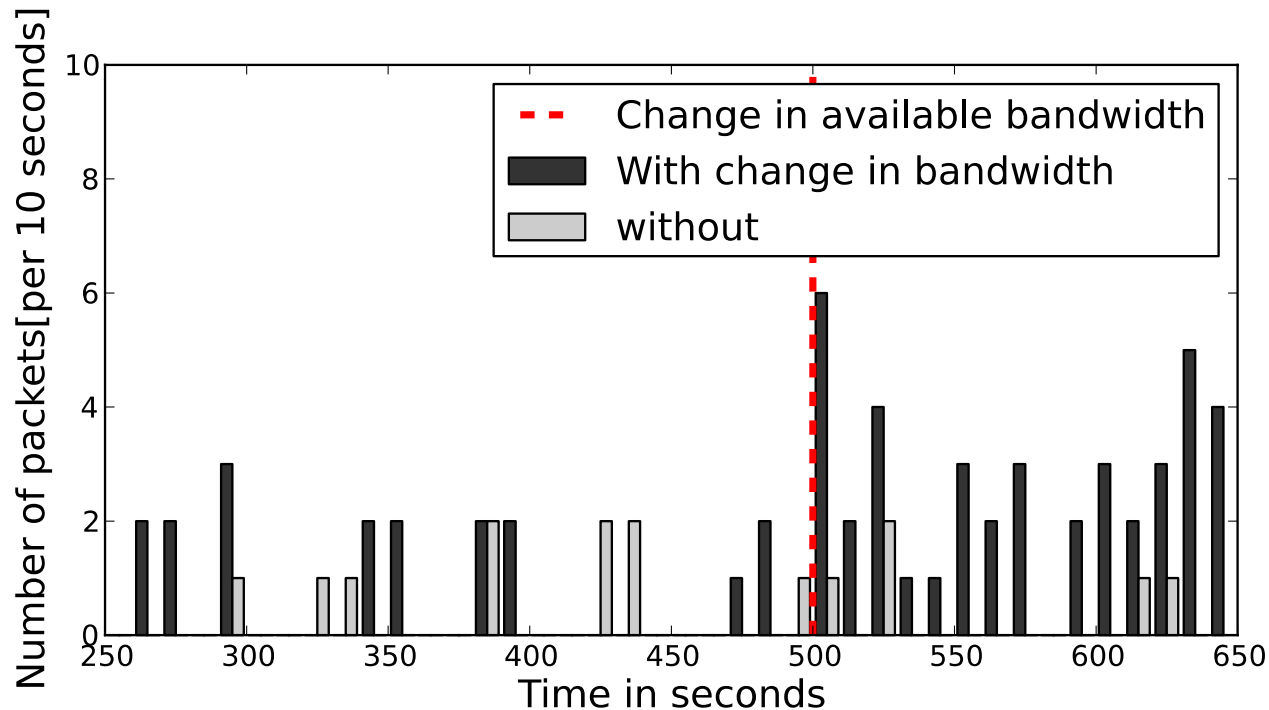
- Demonstrate that unobservability is not achieved
- Their imitation of Skype is incomplete and can thus be recognized even by low-cost, passive attacks
- Even hypothetical improved versions (SkypeMorph+ and StegoTorus+) can be easily distinguished.



# Both fail skype identification tests.

Test	Skype	SkypeMorph+ and StegoTorus+
Flush Supernode cache	Serves as a SN	Rejects all Skype messages
Drop UDP packets	Burst of packets in TCP control	No reaction
Close TCP channel	Ends the UDP stream	No reaction
Delay TCP packets	Reacts depending on the type of message	No reaction
Close TCP connection to a SN	Initiates UDP probes	No reaction
Block the default TCP port	Connects to TCP ports 80 and 443	No reaction

# Dropping UDP packets



# Attacks against StegoTorus-HTTP

- Does not look like a typical HTTP server!
- Most HTTP methods not supported!

HTTP request	Real HTTP server	StegoTorus's HTTP module
GET existing	Returns "200 OK" and sets Connection to keep-alive	Arbitrarily sets Connection to either keep-alive or Close
GET long request	Returns "404 Not Found" since URI does not exist	No response
GET non-existing	Returns "404 Not Found"	Returns "200 OK"
GET wrong protocol	Most servers produce an error message, e.g., "400 Bad Request"	Returns "200 OK"
HEAD existing	Returns the common HTTP headers	No response
OPTIONS common	Returns the supported methods in the Allow line	No response
DELETE existing	Most servers have this method not activated and produce an error message	No response
TEST method	Returns an error message, e.g., "405 Method Not Allowed" and sets Connection=Close	No response
Attack request	Returns an error message, e.g., "404 Not Found"	No response

Table: Httprecon tool was used to send requests to parrot servers



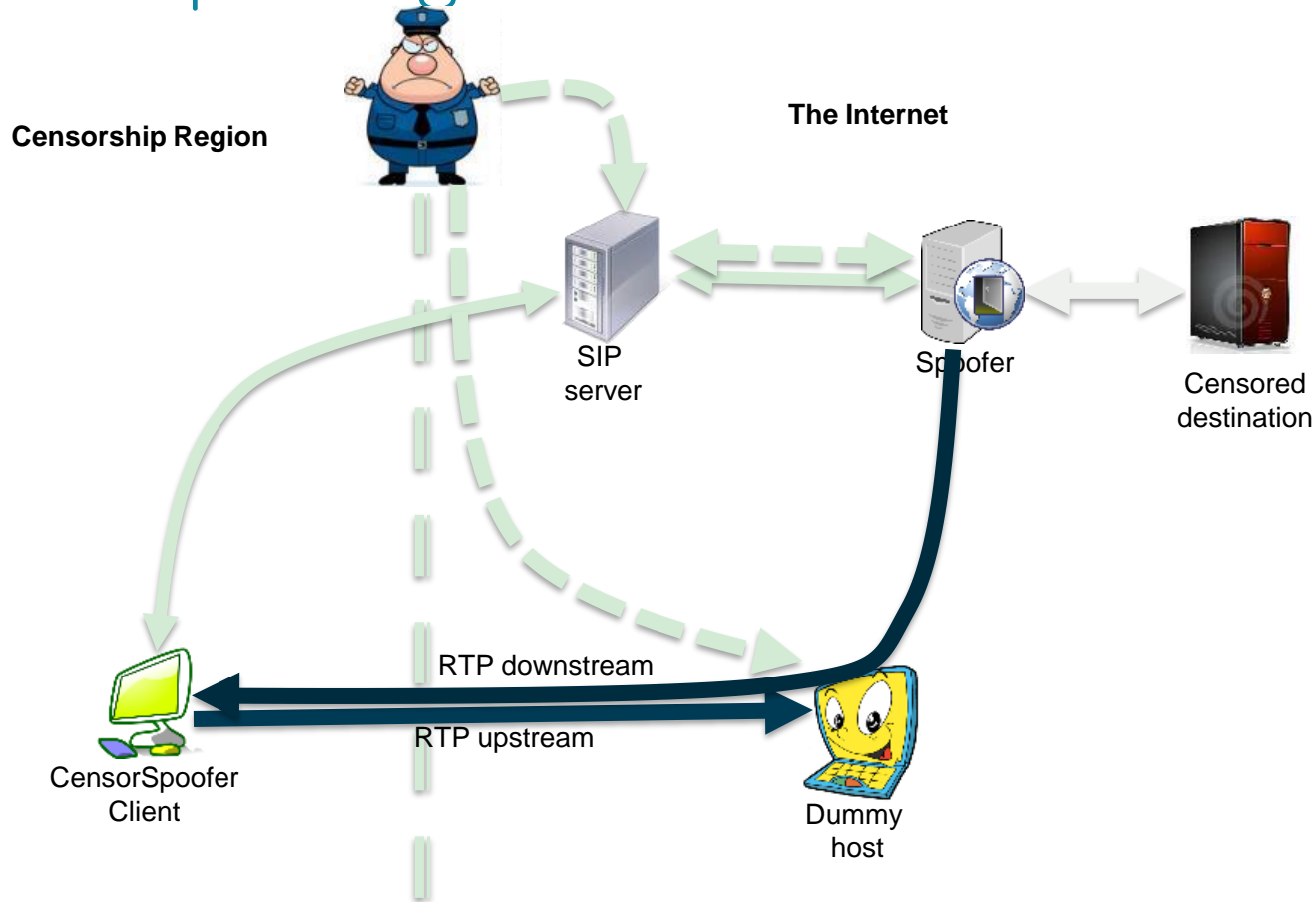
# Detecting CensorSpoofer

- SIP probing: Censor sends SIP messages to the callee IP address and checks whether a genuine SIP client is listening.
- Most HTTP methods not supported!
- Prevent probing by:
  - Change IP address selection algorithm (12.1% of 10, 000 are suitable)
  - Mimic a more popular proprietary service

Attack	Imitation requirement	Adversary	Typical SIP clients (e.g., Ekiga)	CensorSpoofer
Manipulate tag in SIP OK	Soft	LO/OB/OM	Nothing	Client closes the call
SIP INVITE to fakeID@suspiciousIP	SideProtocols Soft, Err	LO/OB/OM	Respond with “100 Trying” and “180 Ringing”, “483 Busy Here”, “603 Decline”, or “404 Not Found”	Nothing
SIP INVALID	SideProtocols,Err	LO/OB/OM	Respond “400 BadRequest”	Nothing
SIP BYE with invalid SIP-ID	SideProtocols Soft, Err	LO/OB/OM	Respond “481 Call Leg/Transaction Does Not Exist”	Nothing
Drop RTP packets (only for confirmation)	SideProtocols Soft, Network	LO/OB/OM	Terminate the call after a time period depending on the client, may change codec in more advanced clients.	Nothing

Table: DISTINGUISHING CENSORSPOOFER FROM GENUINE SIP CLIENTS.

# SIP probing



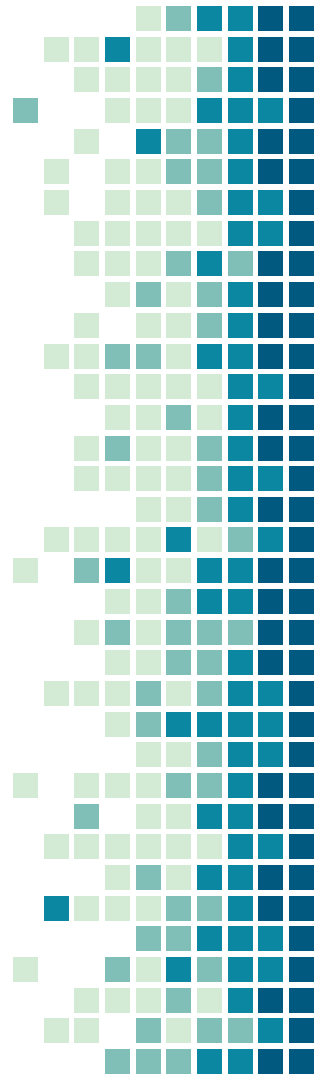
# Recommendations:

- The parrot must mimic a concrete implementation, including bugs!
- Conduct deeply understanding of the adversaries
- Partial imitation is worse than no imitation at all

## Alternative:

- Do not mimic, but run the actual protocol.  
A challenging Task!

After all, much research is needed.



# Summery

- Previously proposed systems such as Tor:
  - 1- Don't hide that a given user is participating in the system
  - 2- Easily blockable and thus not censorship-resistant.
- Parrots are very distant from perfect imitations

## Lessons Learned:

- Extra care must be taken to ensure privacy and anonymity protection.
- It is not enough to simply mimic a popular protocol
- Some imitation flaws are impossible to fix at any cost

# Thank You !

This is an ex-parrot!  
This parrot is no more  
This is a late parrot  
It's stone dead



# References:

- <https://www.nextiva.com/blog/sip-protocol.html>
- <https://slideplayer.com/slide/7309824/>

