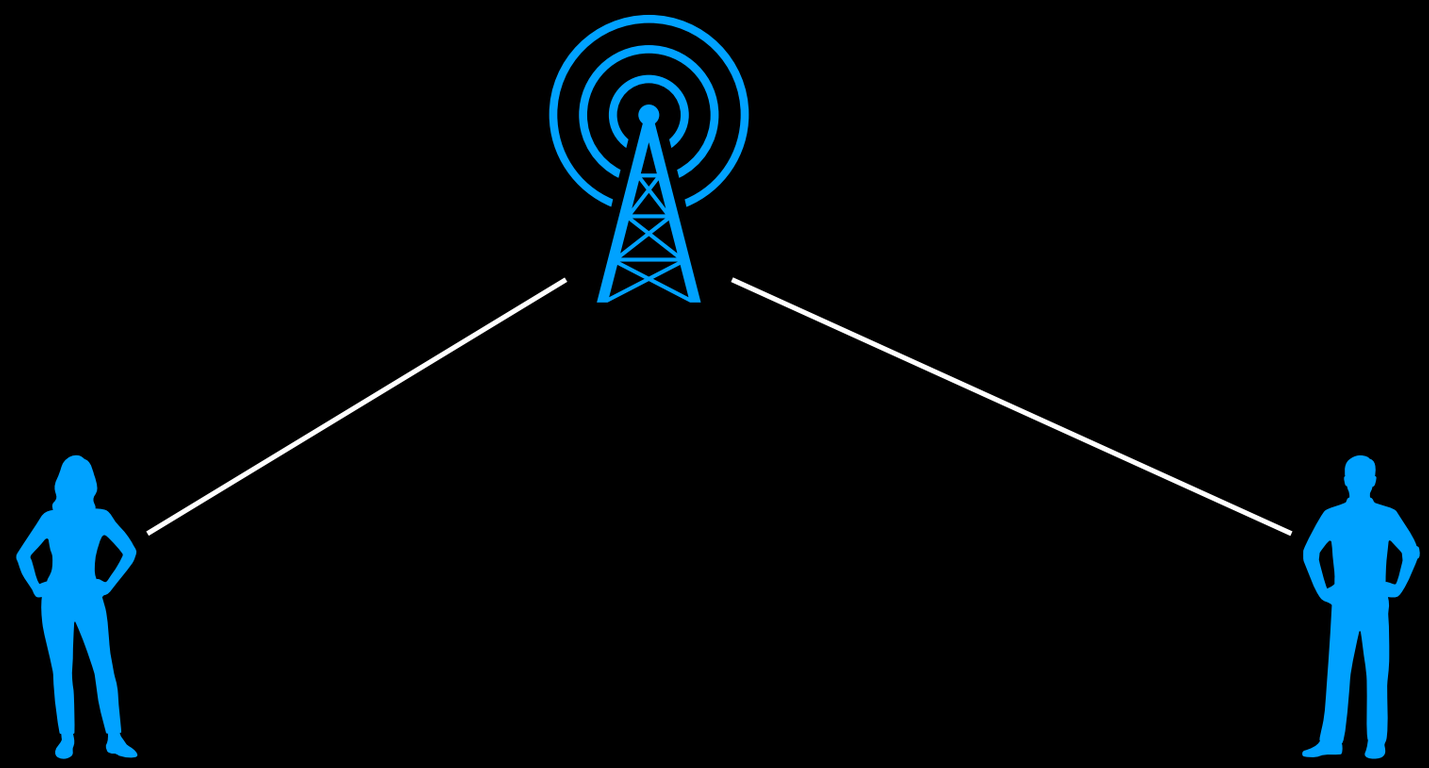
Messaging – Metadata

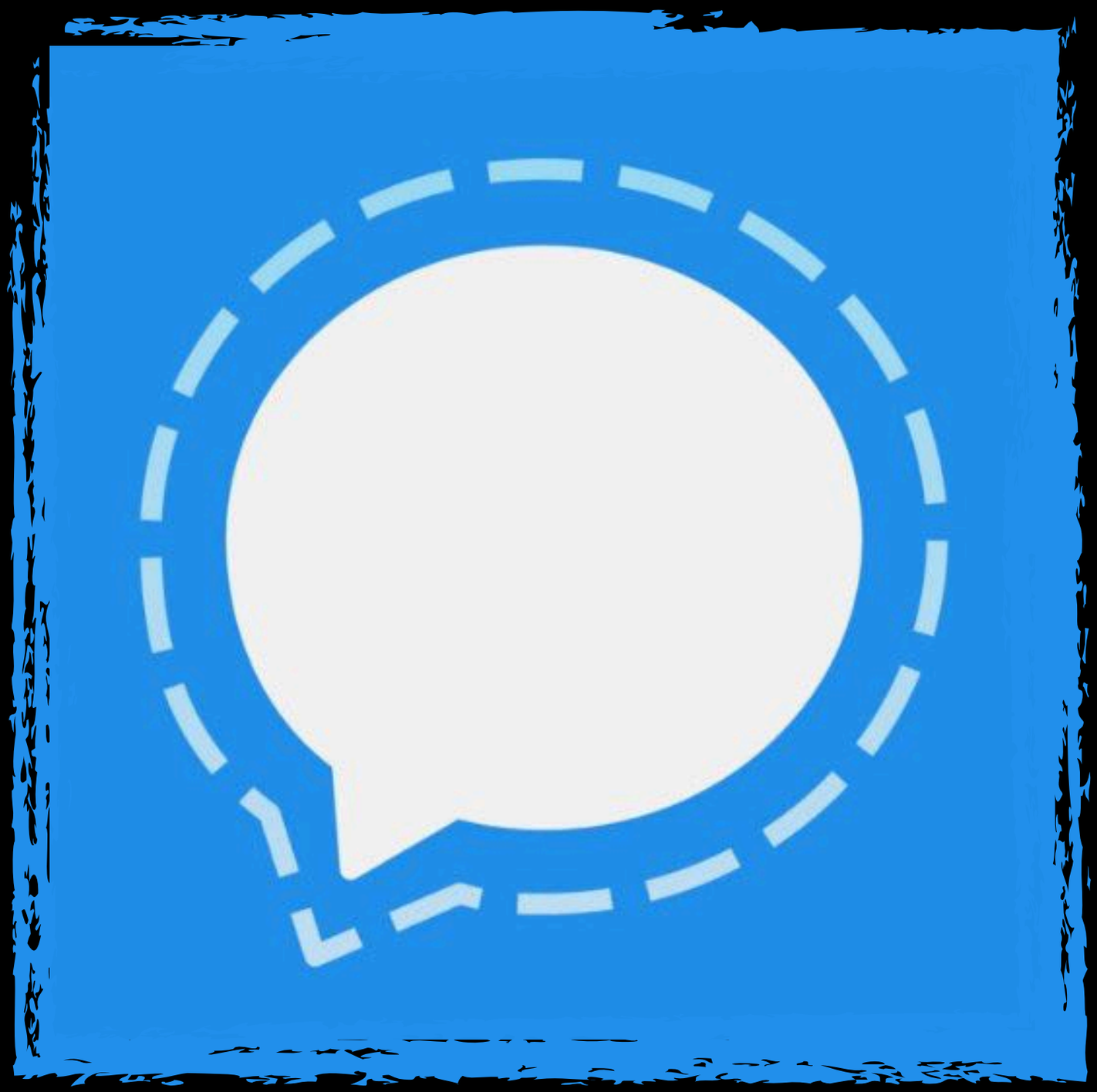
@WillScott

Messaging

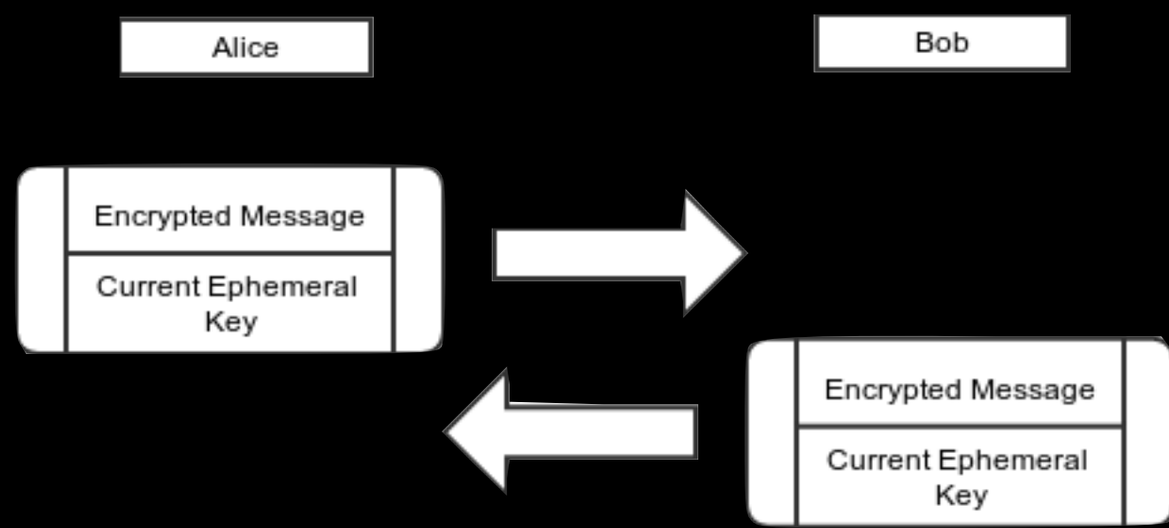


End-to-End Encryption

Signal

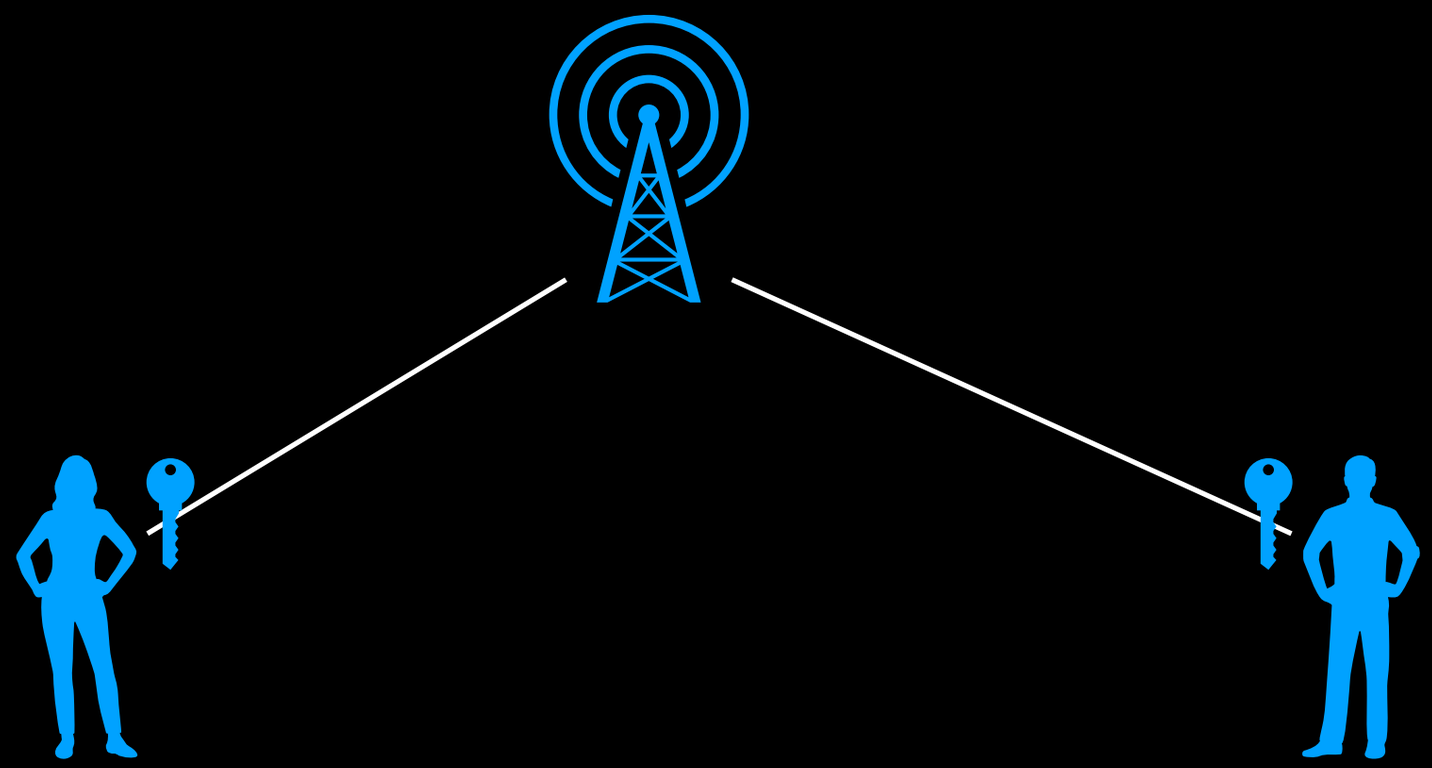


Signal


Alice
Bob
Encrypted Message
Current Ephemeral Key
Encrypted Message
Current Ephemeral Key

Signal

- Forward Secret
- Deniable
- Asynchronous



Threat Model

- Network or Server passively watching / monitoring traffic
- Surveillance of the contents of messages

Signal

Scaling

- (n+1)sec
- Flute
- mpENC

Metadata

- When a user is online
- Contact List
- When messages are sent
- Size of messages

Metadata

We kill people based on metadata

—*Michael Hayden, NSA/CIA*

Pond

GitHub, Inc. github.com/agl/pond

This repository | Search | Pull requests | Issues | Marketplace | Gist

agl / pond

Watch 113 | Star 845 | Fork 105

Code | Issues 59 | Pull requests 17 | Projects 0 | Insights

Pond

442 commits | 5 branches | 2 releases | 29 contributors | BSD-3-Clause

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

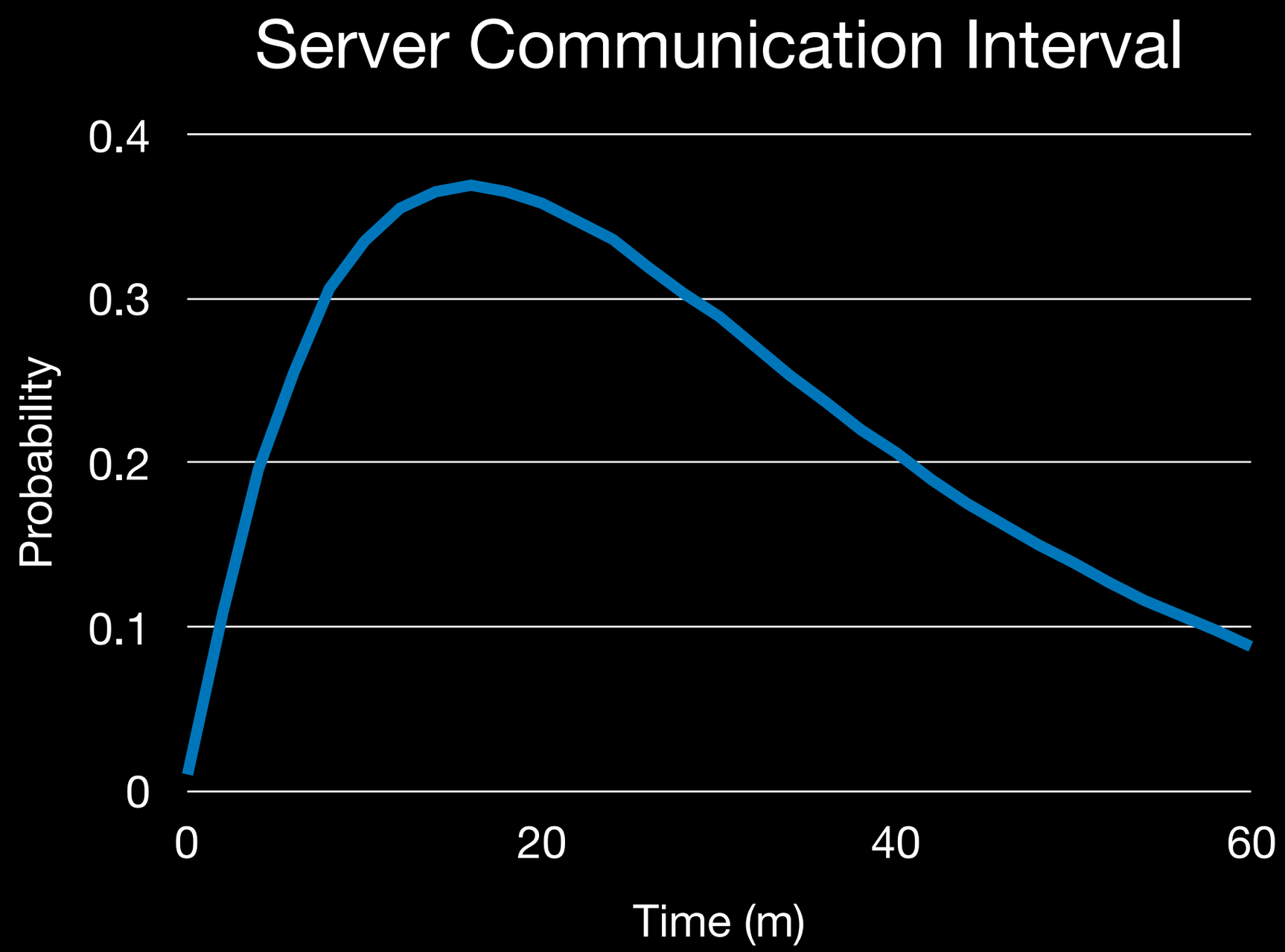
agl Mention state of the project in README.md. Latest commit 675020c on May 24, 2016

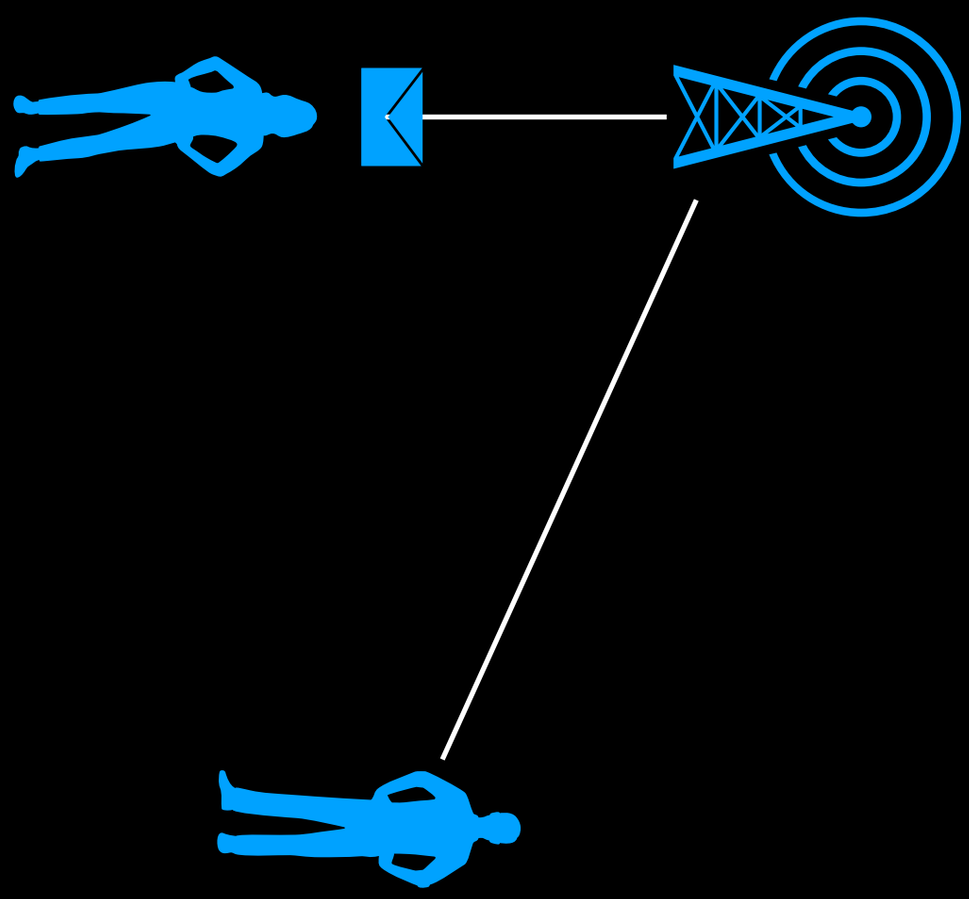
bbssig	Update references from code.google.com to GitHub.	3 years ago
bn256cgo	Update references from code.google.com to GitHub.	3 years ago
client	Set more Vim options to try and minimise any leakage.	2 years ago
decrypt	Update references from code.google.com to GitHub.	3 years ago
doc	Remove links to precompiled binaries.	3 years ago
editstate	Update references from code.google.com to GitHub.	3 years ago

Pond

- Fixed-size messages
- Asynchronous
- PANDA rendezvous

Pond


Server Communication Interval
0.4
0.3
0.2
0.1
0
Probability
0
20
40
60
Time (m)



Threat Model

- Network Adversary (NSA) watching patterns of traffic to understand who is talking to whom.

Loopix

www.usenix.org/system/files/conference/usenixsecurity17/sec17

The Loopix Anonymity System

Ania M. Piotrowska
University College London

Jamie Hayes
University College London

Tariq Elahi
KU Leuven

Sebastian Meiser
University College London

George Danezis
University College London

Abstract

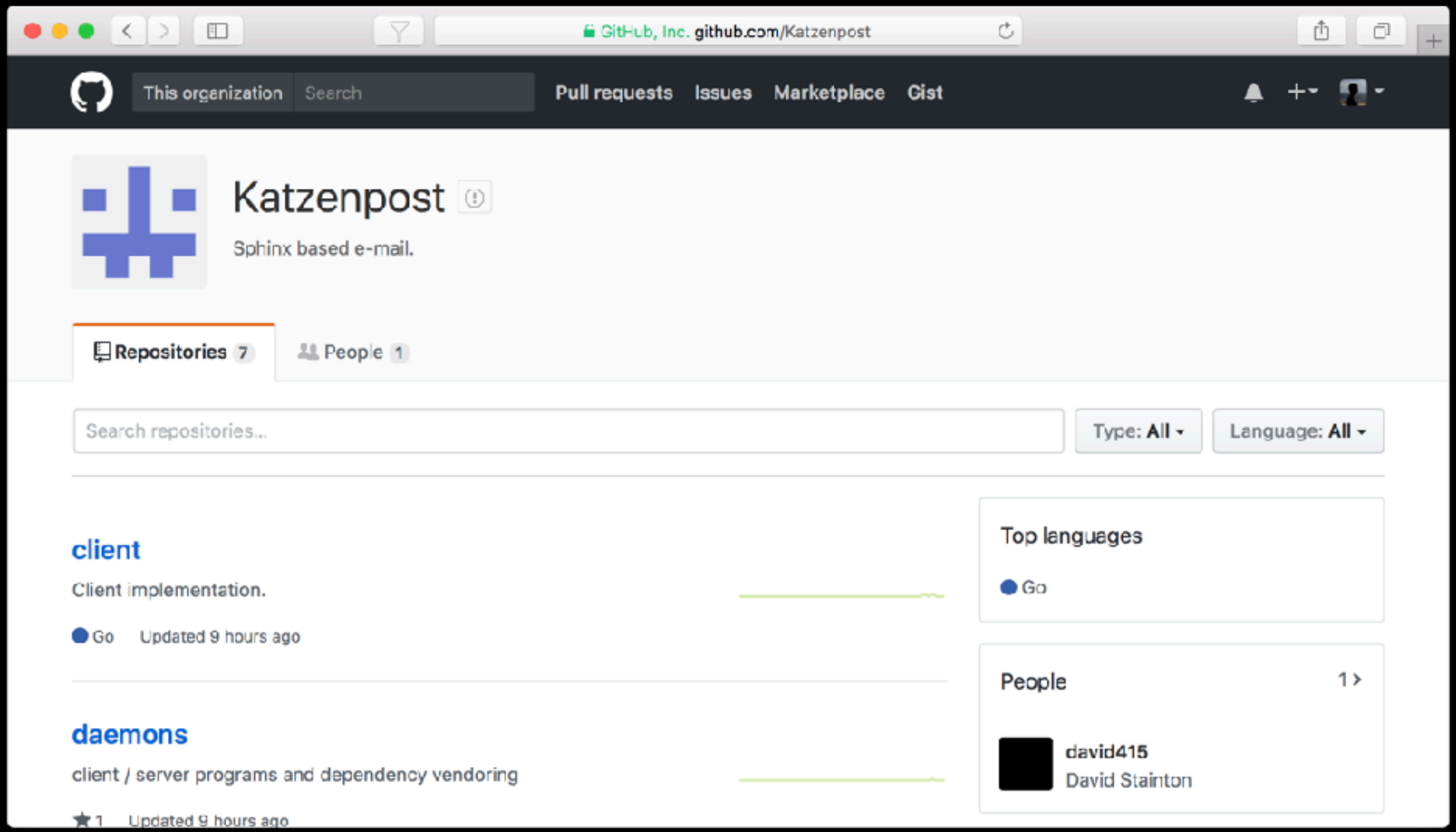
We present *Loopix*, a low-latency anonymous communication system that provides bi-directional 'third-party' sender and receiver anonymity and unobservability. Loopix leverages cover traffic and *Poisson mixing*—brief independent message delays—to provide anonymity and

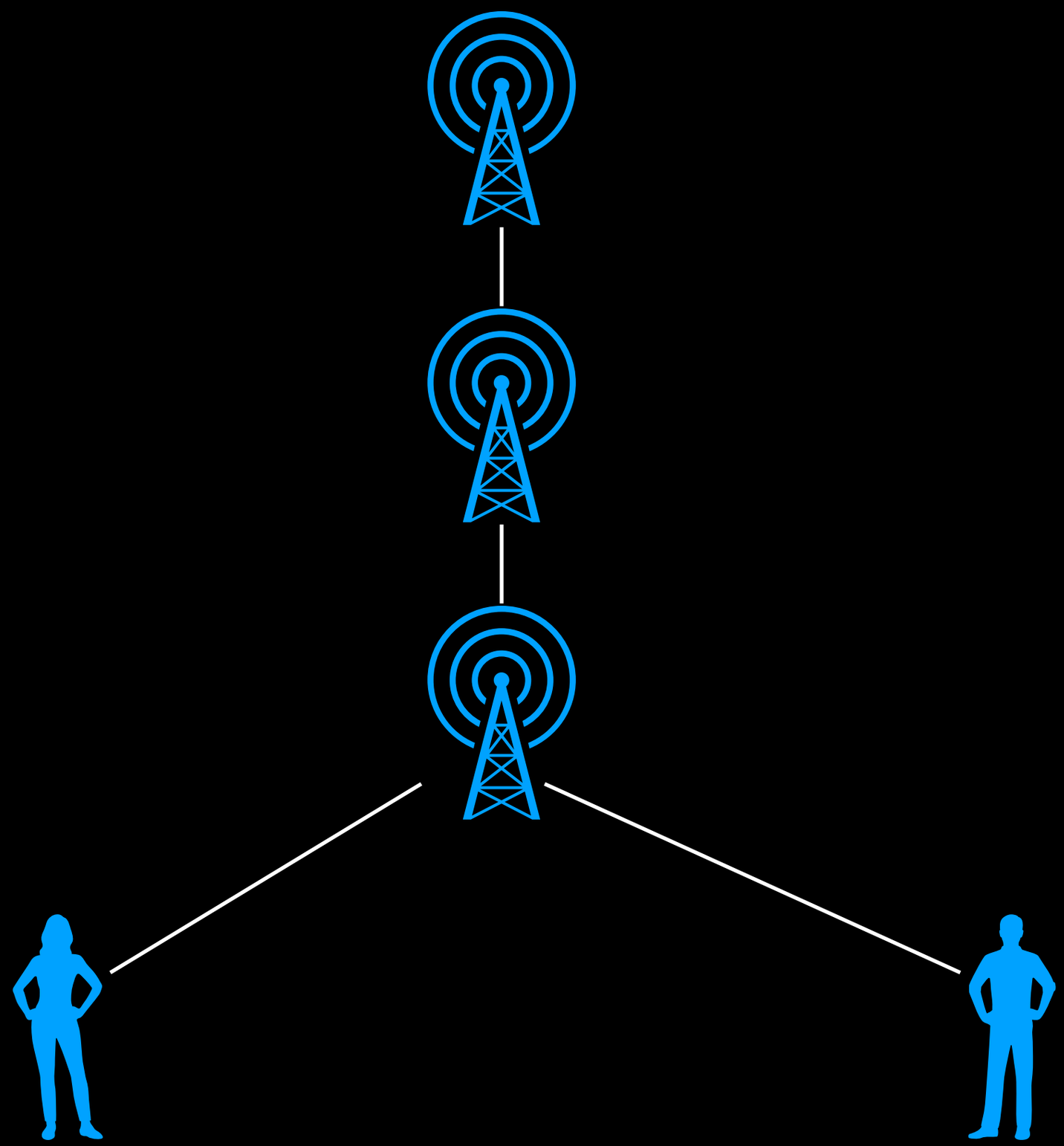
cent leaks of extensive mass surveillance programs[1], exposing such meta-data leads to significant privacy risks.

Since 2004, Tor [20], a practical manifestation of circuit-based onion routing, has become the most popular anonymous communication tool, with systems such as Herd [33], Riposte [11], HORNET [10] and Vuvuzela [46] extending and strengthening this paradigm

Loopix

Katzenpost


GitHub, Inc. github.com/Katzenpost
This organization
Search
Pull requests
Issues
Marketplace
Gist
Katzenpost
Sphinx based e-mail.
Repositories 7
People 1
Search repositories...
Type: All
Language: All
client
Client implementation.
Go
Updated 9 hours ago
daemons
client / server programs and dependency vendoring
1
Updated 9 hours ago
Top languages
Go
People
1
david415
David Stainton



Ricochet

ricochet.im

Ricochet

About

Latest changes

GitHub

Sponsors

Anonymous instant messaging for real privacy

Ricochet is a different approach to instant messaging that **doesn't trust anyone** in protecting your privacy.

- *Eliminate metadata.* Nobody knows who you are, who you talk to, or what you say.
- *Stay anonymous.* Share what you want, without sharing your identity and location.
- *Nobody in the middle.* There are no servers to monitor, censor, or hack.
- *Safe by default.* Security isn't secure until it's automatic and easy to use.

Get started

WINDOWS MAC LINUX

The latest version is **1.1.4** (November 5, 2016). You can also **build from source**.

How it works

Ricochet uses the **Tor network** to reach your contacts without relying on messaging servers. It creates a **hidden service**, which is used to rendezvous with your contacts without revealing your location or IP address.

Ricochet

GitHub, Inc. github.com/ricochet-im/ricochet-go

This repository | Search | Pull requests | Issues | Marketplace | Gist

ricochet-im / ricochet-go

Watch 9 | Star 18 | Fork 6

Code | Issues 0 | Pull requests 0 | Projects 1 | Insights

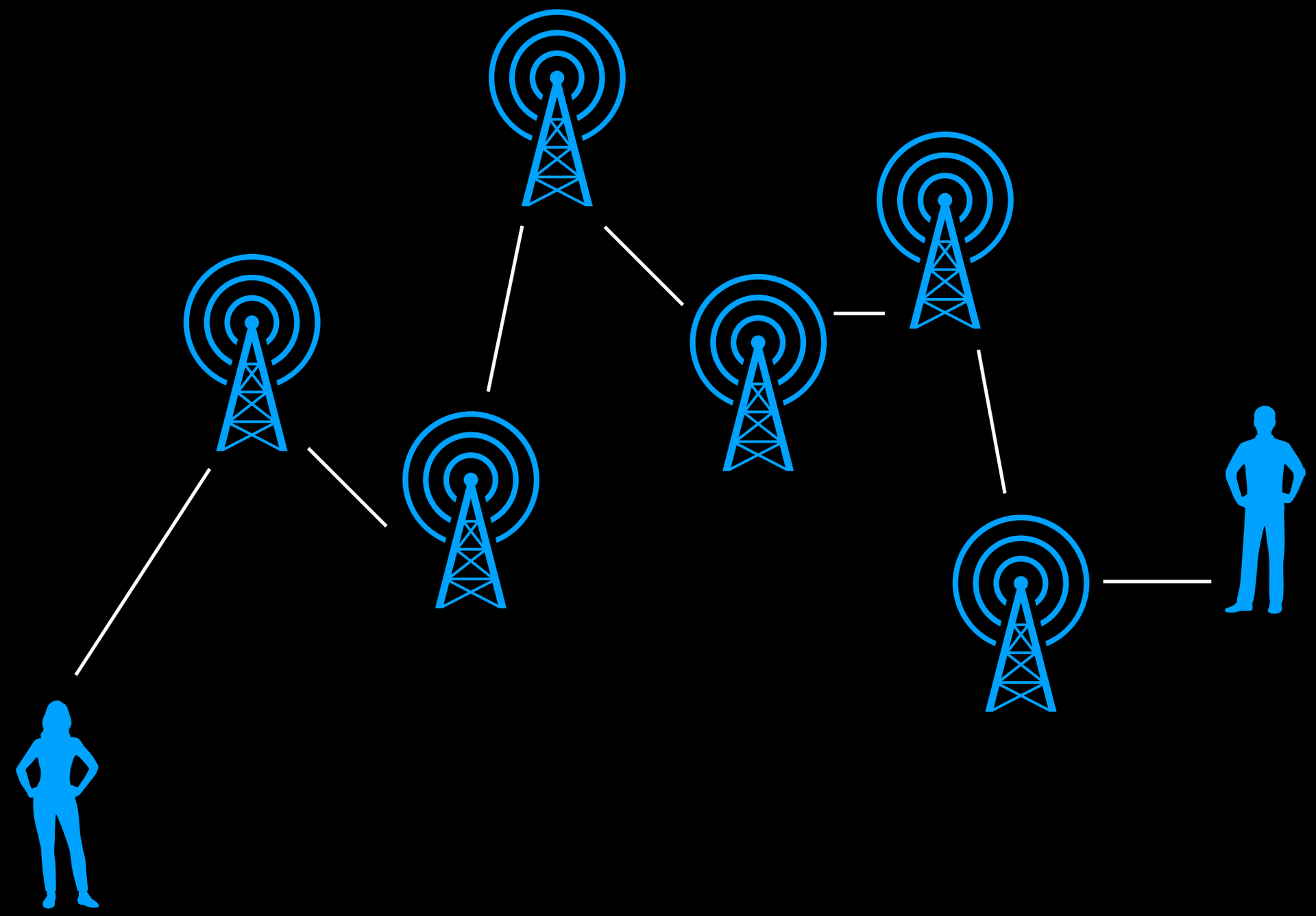
Experimental multiprocess Ricochet client in Go https://ricochet.im/

86 commits | 1 branch | 0 releases | 2 contributors

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

special core: Minor fixes for inbound contact requests — Latest commit 23e22f0 10 days ago

core	core: Minor fixes for inbound contact requests	10 days ago
ricochet-cli	cli: Only print 'new message' for inbound messages	10 days ago
rpc	core: Implement inbound contact requests	11 days ago
vendor	vendor: Switch go-ricochet back to upstream	7 months ago
.gitignore	Delete backend and rename cli to ricochet-cli	10 months ago
.travis.yml	Add minimal travis config	9 months ago



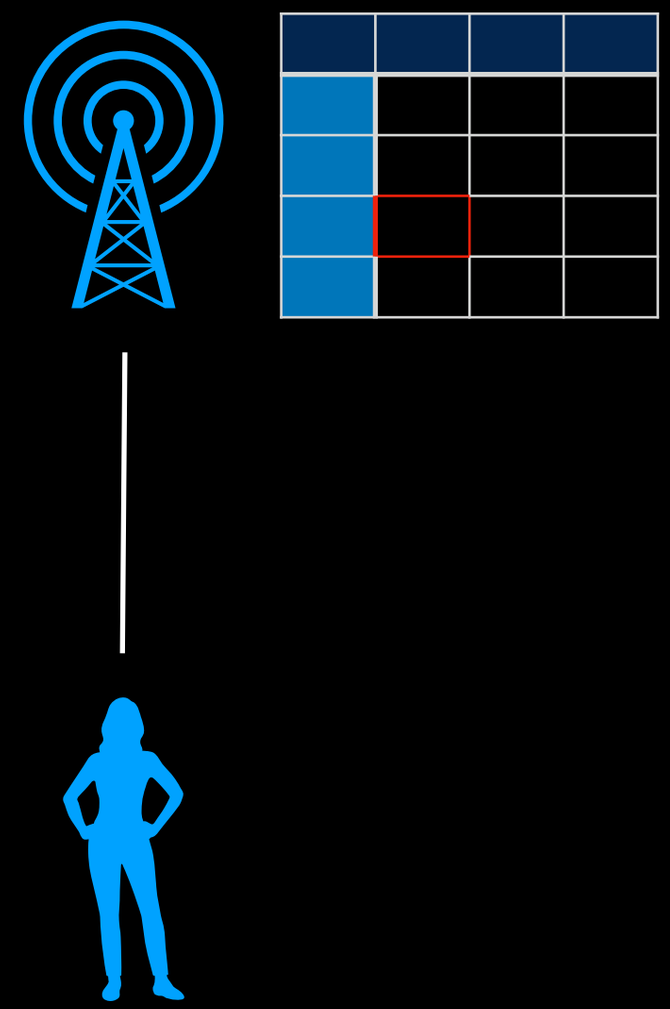
Threat Model

- Network Adversary, *or Server*, watching patterns of traffic to understand who is talking to whom.

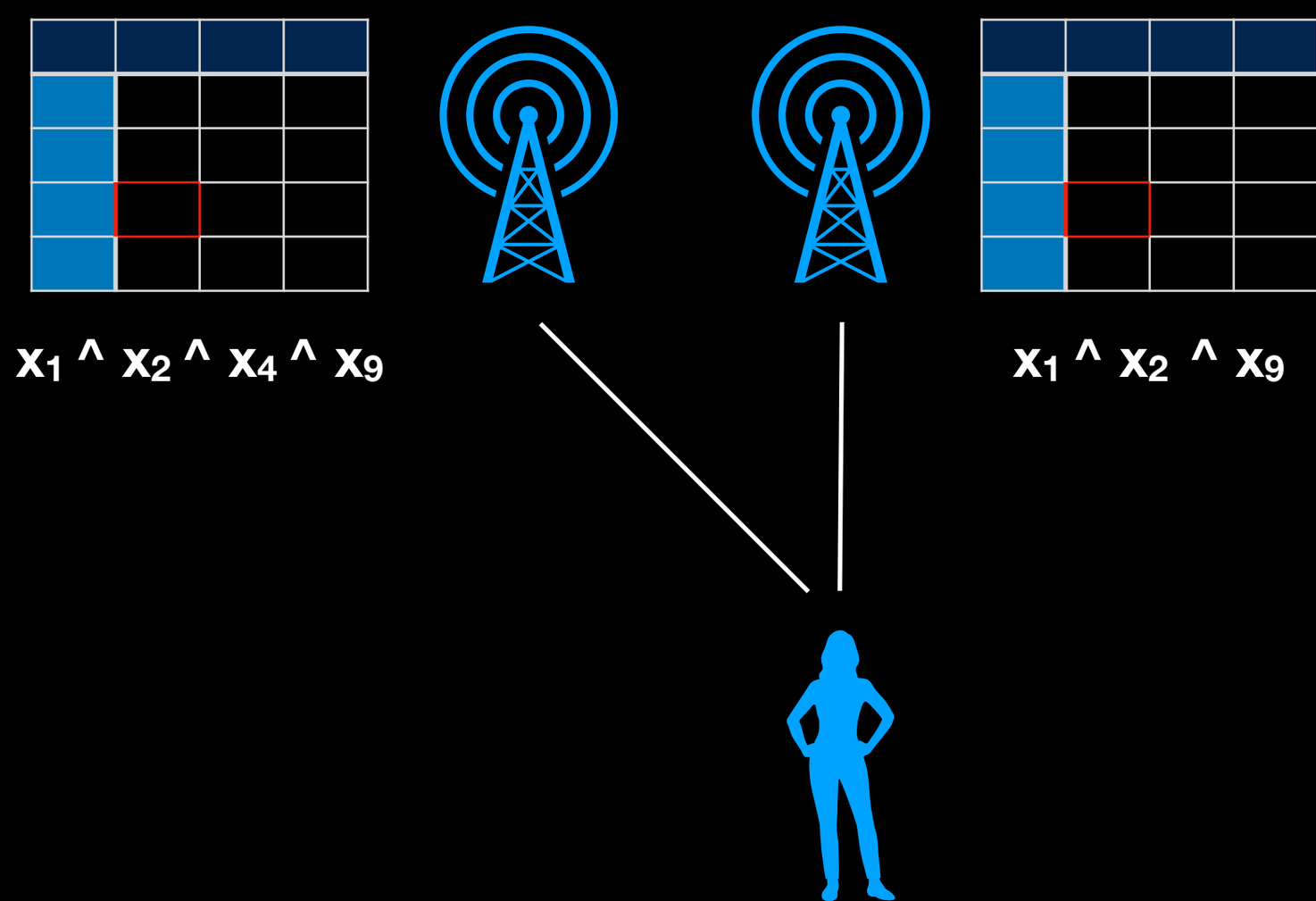
Private Information Retrieval

Can I retrieve a message from a server without the server knowing which message it gave me?

With reasonable network costs?



Private Information Retrieval



DP5

hyperelliptic.org/PSC/slides/dp5-psc.pdf

DP5: Privacy-preserving Presence Protocols

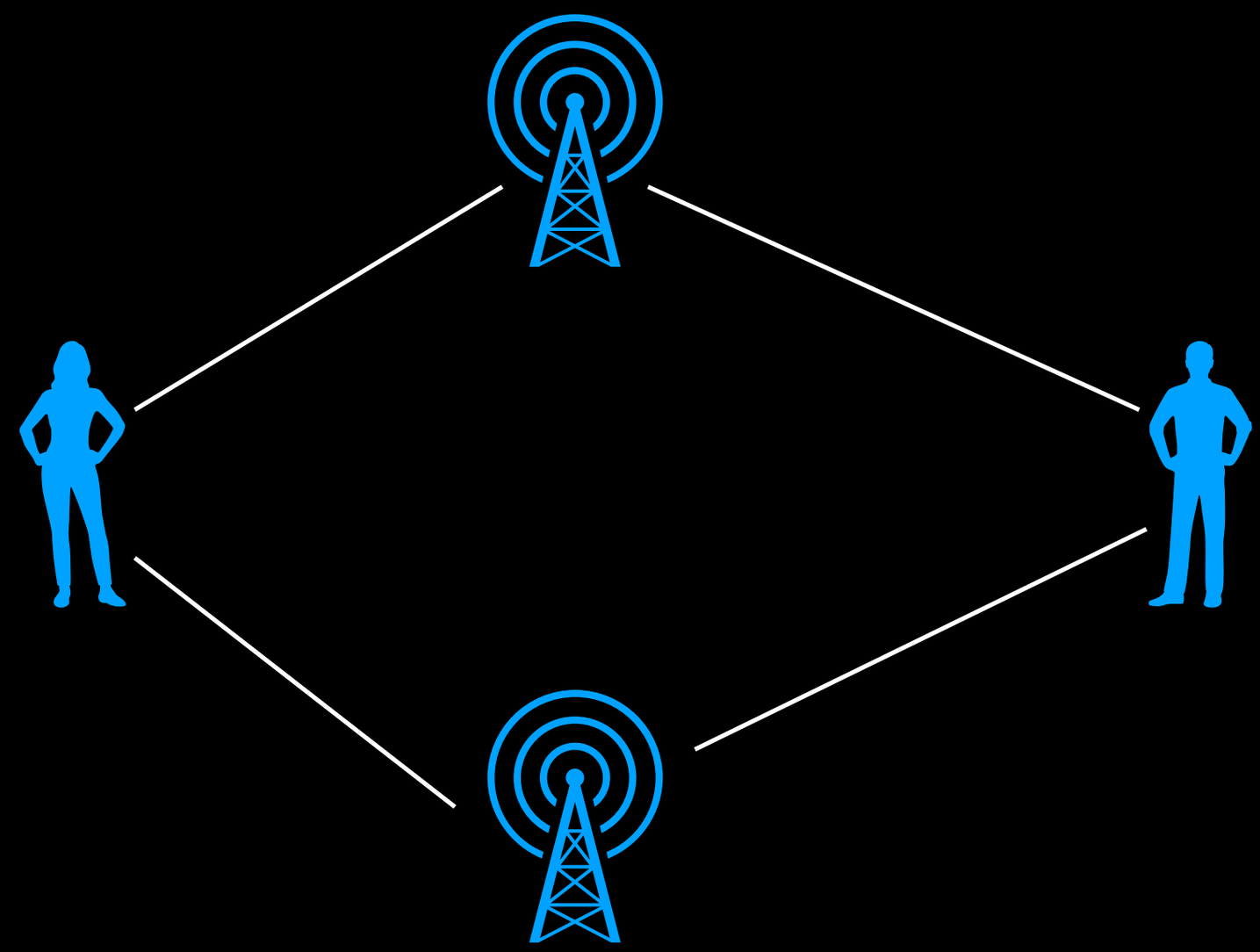
Ian Goldberg

joint work with Nikita Borisov, George Danezis

Cryptography, Security, and Privacy Research Lab
University of Waterloo

Post-Snowden Cryptography
Brussels, 10 December 2015

Messaging



DP5

GitHub, Inc. github.com/szechuen/dp5_proxy

szechuen/dp5_proxy: XMPP Proxy for DP5 | szechuen/CCC-Presale: Simple script to grab CCC Presale tickets

This repository | Search | Pull requests | Issues | Marketplace | Gist

szechuen / dp5_proxy

Watch 1 | Unstar 5 | Fork 1

Code | Issues 5 | Pull requests 0 | Projects 0 | Wiki | Insights

XMPP Proxy for DP5

445 commits | 1 branch | 1 release | 3 contributors

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

szechuen Not checking cert in health check for now — Latest commit cd4c99e on Mar 31

dp5	Fix static path to certificate	8 months ago
.gitignore	Exclude compiled binary	5 months ago
.gitmodules	Removed relicwrapper module	3 years ago
Dockerfile-client	Added Docker build for client proxy	5 months ago
Dockerfile-server	Not checking cert in health check for now	5 months ago
README.md	Added Docker build for client proxy	5 months ago

Threat Model

- *Global Passive Network Adversary*, or Server, watching patterns of traffic to understand who is talking to whom.

Talek

raymondcheng.net/download/papers/talek-tr.pdf

Talek: a Private Publish-Subscribe Protocol

Raymond Cheng†, William Scott†, Bryan Parno‡, Irene Zhang†
Arvind Krishnamurthy†, Thomas Anderson†
†*University of Washington*, ‡*Carnegie Mellon University*

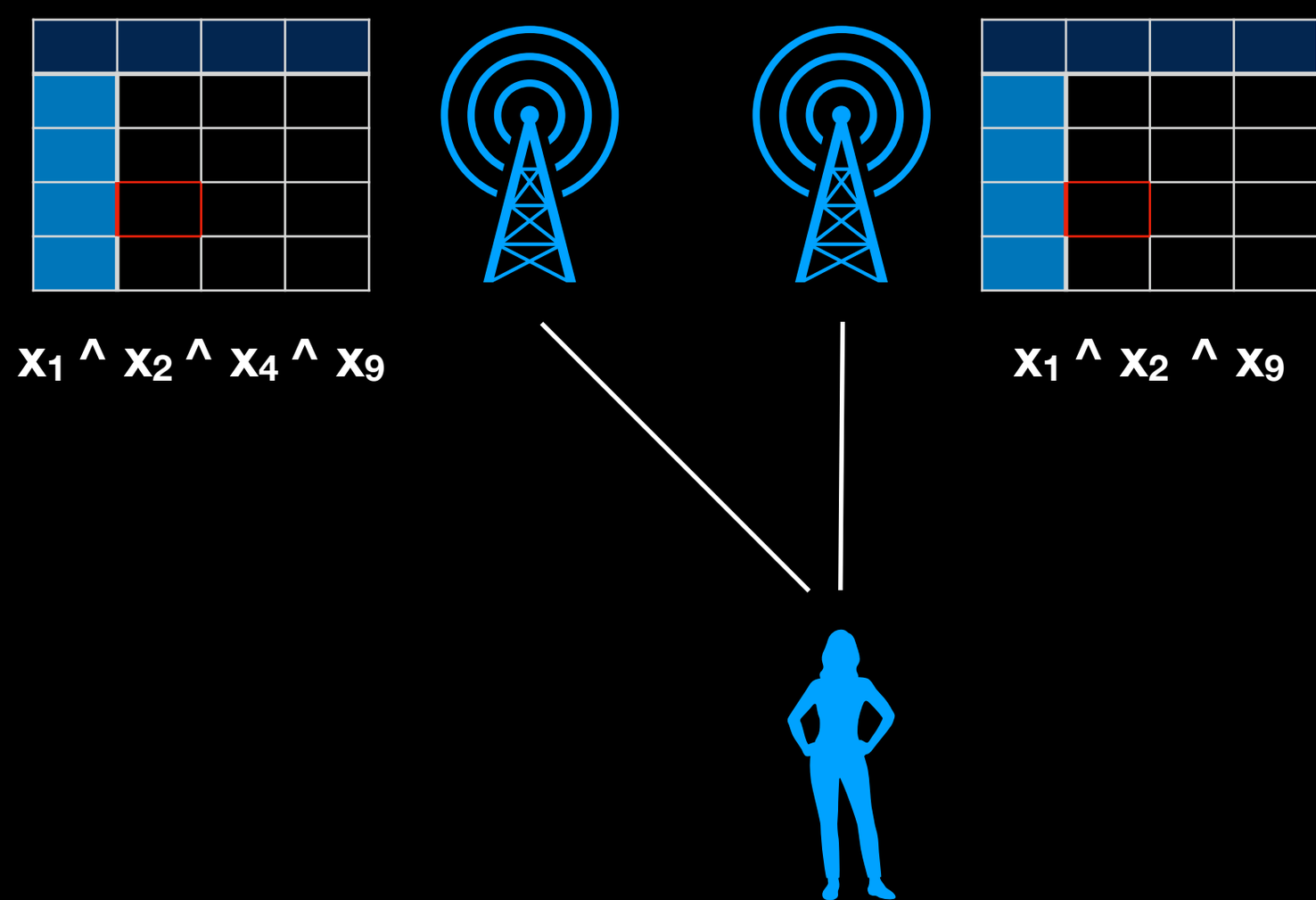
Abstract

Modern applications share user-generated data over the cloud, often exposing sensitive information. Talek is a private publish-subscribe (pub/sub) system that shares user data through potentially untrustworthy servers, while hiding both data content and the communication patterns among its users. Talek is designed with two goals that distinguish it from the prior work in private messaging. First, Talek is designed with the strong security goal of *access sequence indistinguishability*, where clients leak no information to adversarial servers that might help an adversary distinguish between two arbitrary-length client access sequences. Second, our system aims to be practical for general-purpose work-

range of applications, including group messaging, news feeds, and data synchronization. Publishers create message logs, which groups of trusted subscribers read at a later time. As long as clients and at least one server are uncompromised and running authentic versions of the software, Talek prevents a cloud operator from learning anything about the communication patterns of the users. Combined with encryption, developers conceal both the *contents* and *metadata* of users' application usage without losing the reliability and availability of the cloud.

Recent research has advanced both one-to-one private messaging [2, 3, 52, 59] and anonymous broadcasting [17–19, 42, 60]. These systems offer security guarantees rooted in k-anonymity [57], plausible deniabil-

Talek



Talek

- Deterministic-Random Writes
- Dense packing of messages (Cuckoo hashing)
- Efficient lookup (GPU)

Talek

- 5,000 Lines of Go
- (+500 lines for XMPP shim; talexmpp)
- 3 Components
  - App client library: Libtalek
  - Talek Server: 'Replica'
  - Coordinator: 'Frontend'

Talek

https://wills.co.tt/talexmpp

Trade-offs

- Signal - Server trust
- Mix Networks - Latency
- Tor - Global passive adversary. Asynchronous
- PIR - Computation

Messaging Without Metadata

https://github.com/privacylab/
@WillScott