Gonçalo Pestana

https://gpestana.com

https://hashmatter.com

@gpestana

Engineer & researcher at **hashmatter**

Privacy in P2P networks

Privacy in P2P networks with focus on Distributed Hash Tables

The future we are building

Privacy Enhancing Technology for DHTs and P2P networks

Where to go from here

Distributed Hash Tables

Collaborative, P2P overlay network

Decentralized **key-value** storage

Content based hashing

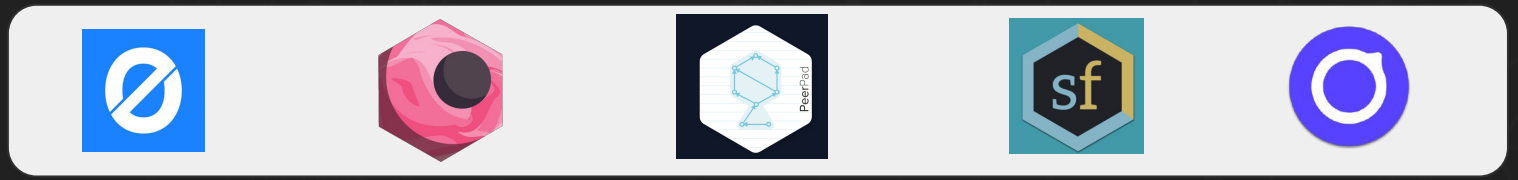
```
get(content_id)
store(content)
```

Distributed Hash Tables

DHT can be used as the **scaffolding** for decentralized applications

Distributed Hash Tables

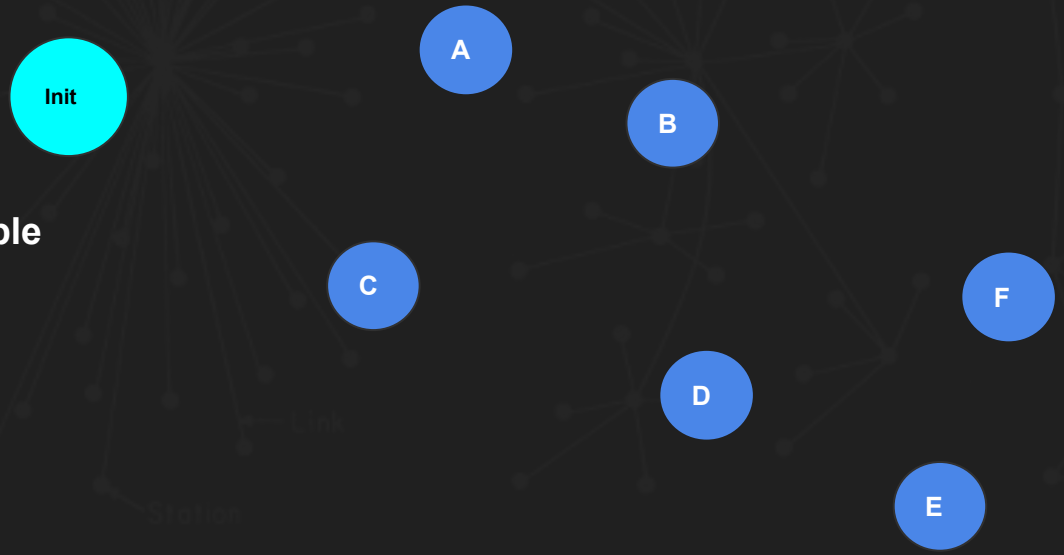
Application layer



Privacy in P2P networks

Distributed Hash Tables

DHT.get(E)

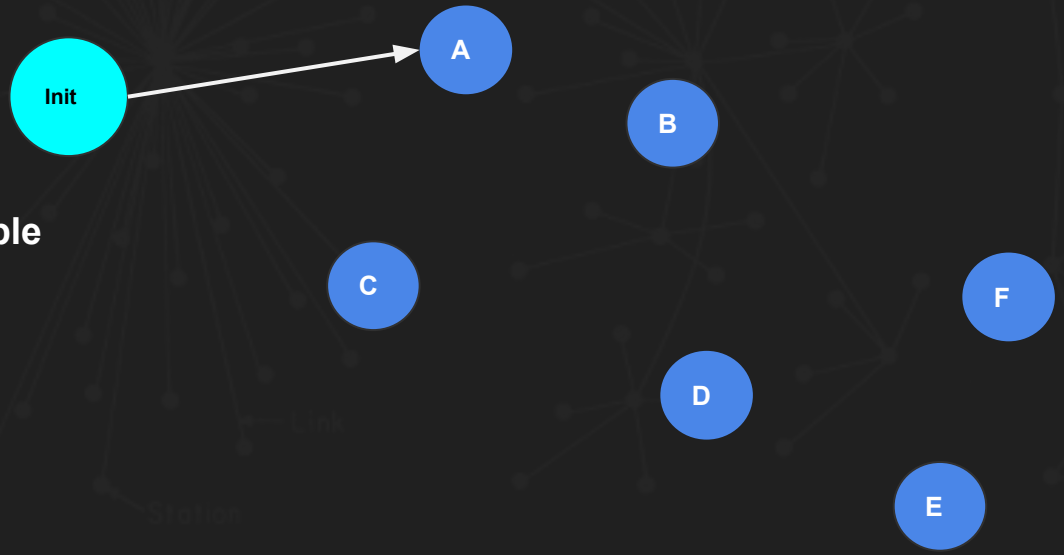


Routing Table
A: ***addr***

Distributed Hash Tables

DHT.get(E)

Routing Table
D: *addr*

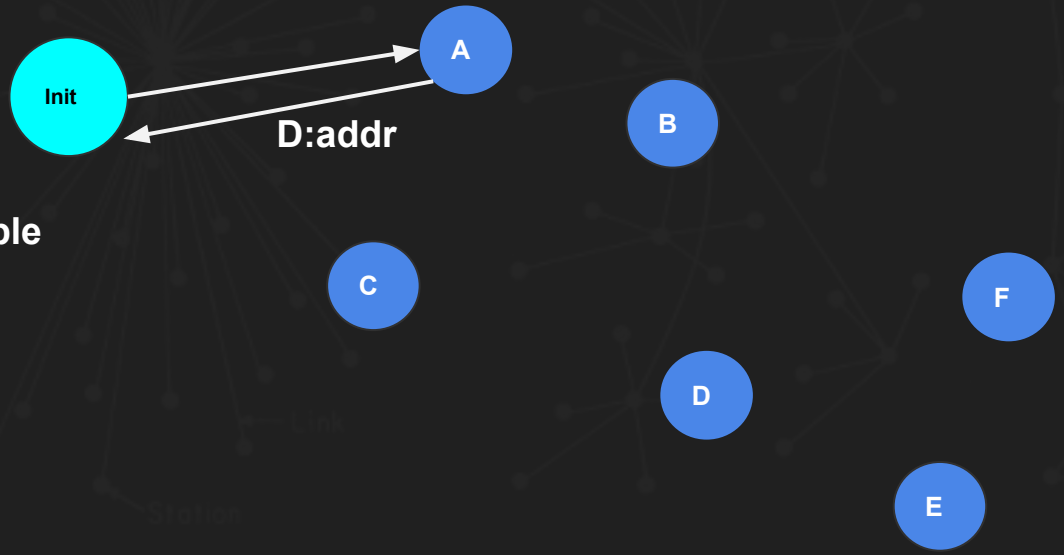


Routing Table
A: *addr*

Distributed Hash Tables

DHT.get(E)

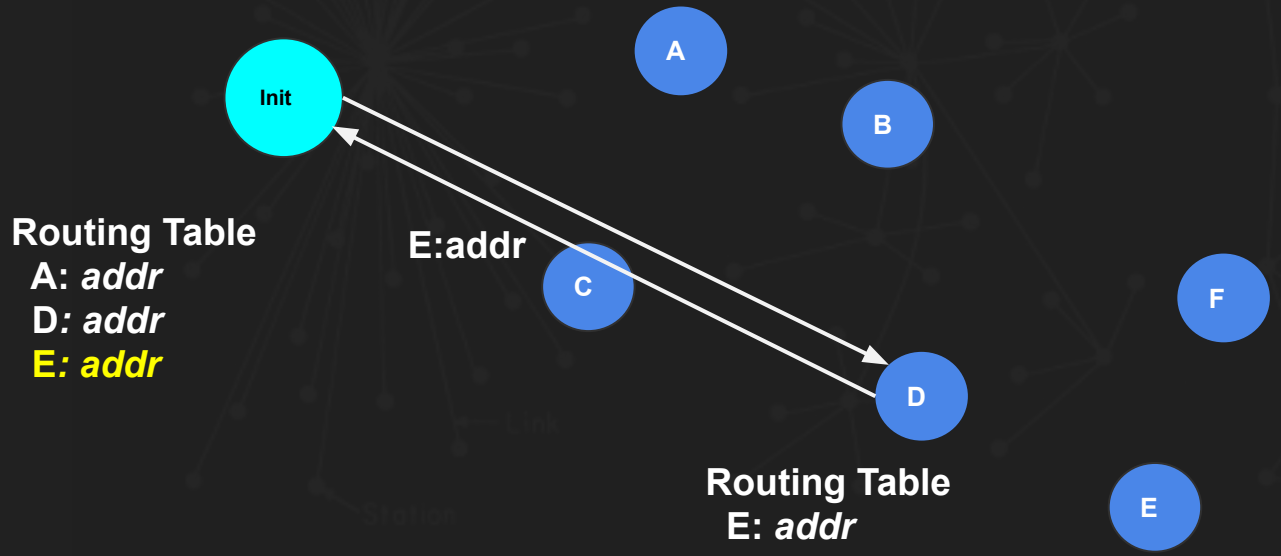
Routing Table
D: *addr*



Routing Table
A: *addr*
D: *addr*

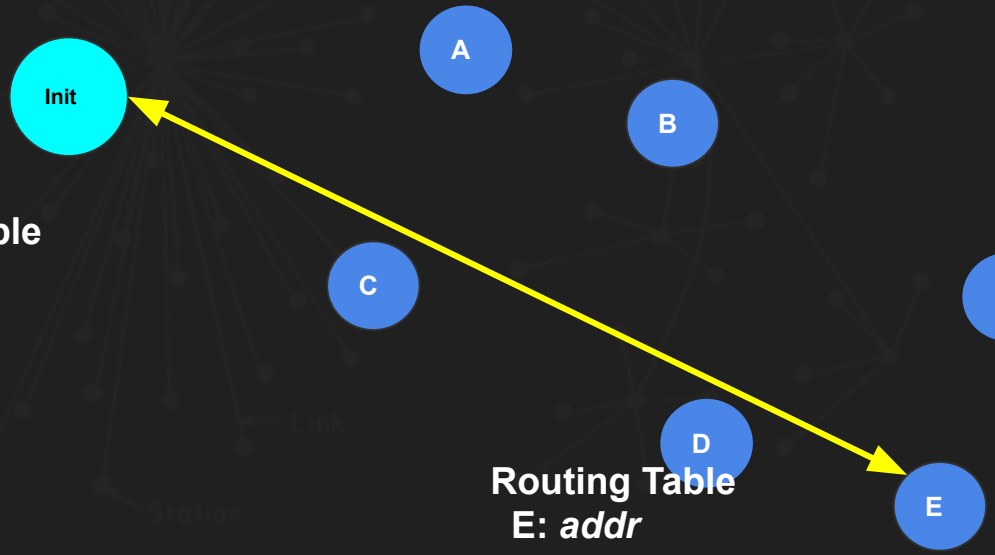
Distributed Hash Tables

DHT.get(E)



Distributed Hash Tables

DHT.get(E)



Routing Table
A: *addr*
D: *addr*
E: *addr*

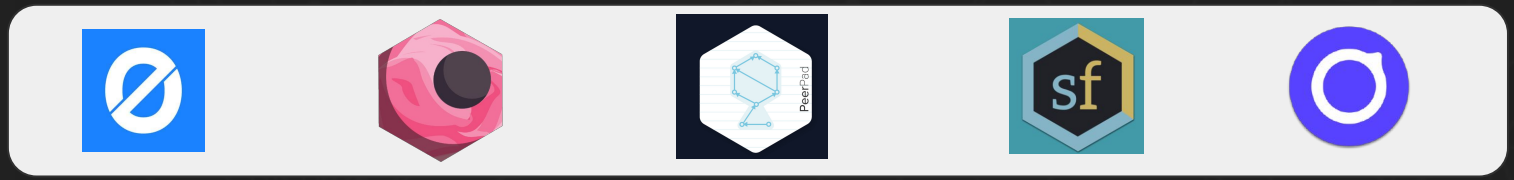
Routing Table
E: *addr*

How about privacy?

The properties that make DHTs a great building block for the decentralized web, also makes them **vulnerable to privacy attacks** [m,n,j,l,k,.. (many more)]

Distributed Hash Tables

Application layer



Vulnerability propagation

Privacy in P2P networks

Thought experiments

We live in a world where P2P infra is as mainstream as client-server and centralized infrastructure

dINSTA

dYTUBE

dSITES

Case 1) **Content provider tracking**

DHT is used to store and retrieve **decentralized web content**

URL → **`content_ID`** `(eg. QmSJqTBmUD3gLTqFoqmEJ3yXKrxgvXu54fUMjhuTpEeHk2)`

→ Bob adds his personal website to the network and caches it in his own devices

Case 1) **Content provider tracking**

What if an adversary periodically queries the IP Bob's webpage provider?

```
t0: provIP_t0 = DHT.findProviders(bob_pageID)
t1: provIP_t1 = DHT.findProviders(bob_pageID)
...
tm: provIP_tm = DHT.findProviders(bob_pageID)
```

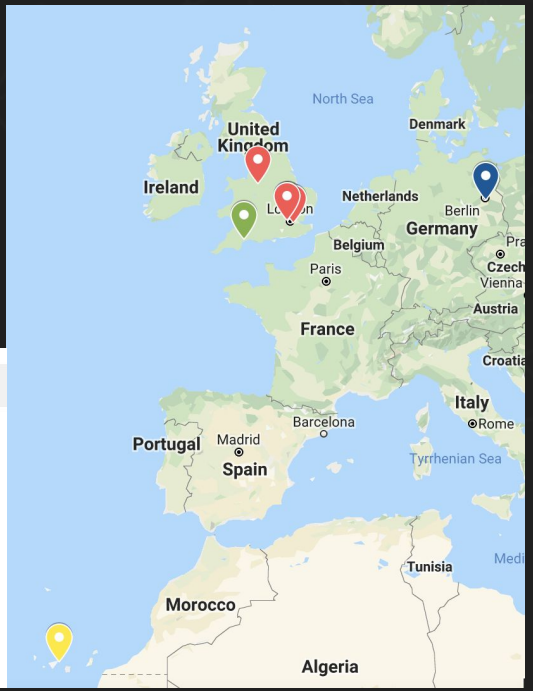
Case 1) **Content provider tracking**

dSITES

What if an adversary periodically queries the IP Bob's webpage provider?

```
t0: provIP_t0 = DHT.findProviders(bob_pageID)
t1: provIP_t1 = DHT.findProviders(bob_pageID)
...
tm: provIP_tm = DHT.findProviders(bob_pageID)
```

- 24 April 2019 08:17:06
- 24 April 2019 08:20:29
- 24 April 2019 15:35:16
- 1 May 2019 10:00:17
- 29 April 2019 11:51:37
- 16 May de 2019 13:00:12
- 17 May 2019 18:50:30



https://github.com/gpestana/dht-sneak

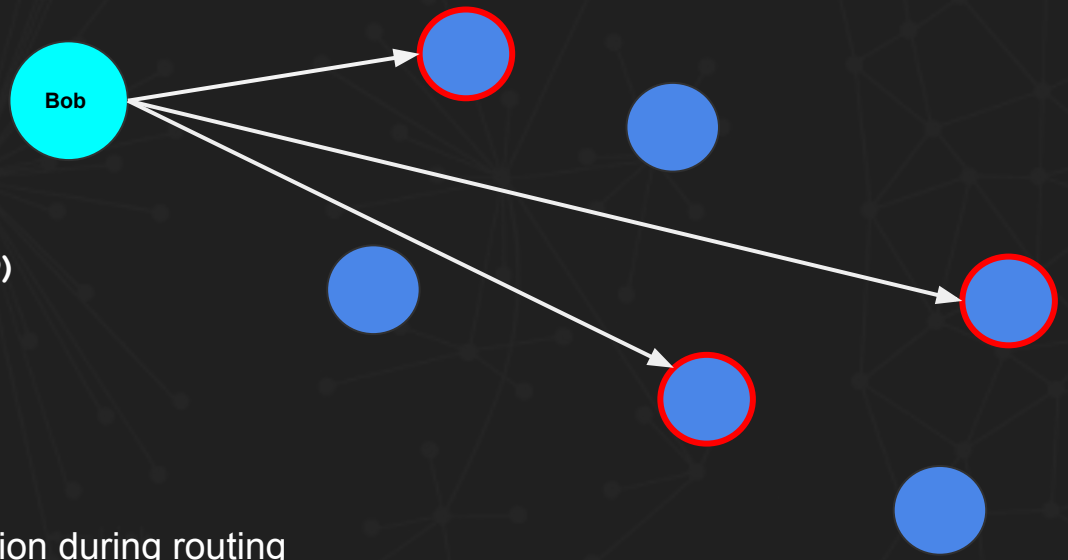
Case 2) **Leaking routing metadata and interest extrapolation**

Decentralized photo sharing app built on top of DHT

Each photo has an unique ID `(eg. QmSJqTBmUD3gLTqFoqmEJ3yXKrxgvXu54fUMjhuTpEeHk2)`

→ What if Bob requests for Alice's photo?

Case 2) **Leaking routing metadata and interest extrapolation**



`DHT.get(Alice_photoID)`

Bob leaks interest information during routing

Trivial to monitor, infer social graph, track Bob

Case 3) **Content providers inference attacks**

Decentralized video caching and streaming network

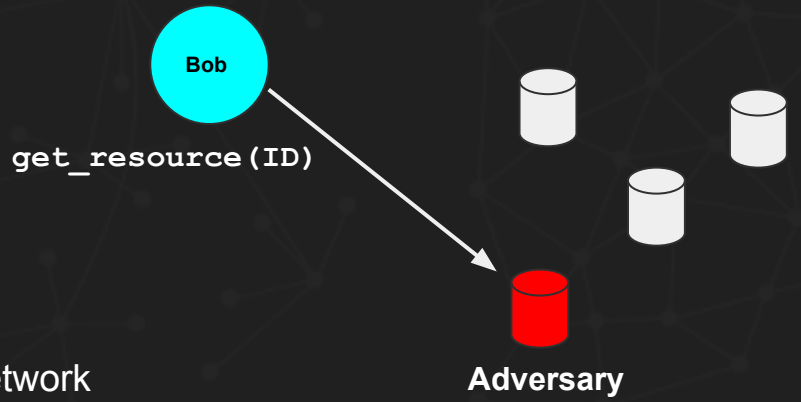
Each video has an unique ID `(eg. QmSJqTBmUD3gLTqFoqmEJ3yXKrxgvXu54fUMjhuTpEeHk2)`

Anyone can replicate and serve videos in the network

Case 3) **Content providers inference attacks**

dYTUBE

Rogue government caches and provider *opposition* video and **tracks who is consuming it**



→ Bob's government learns that he's watching gov. opposition videos just by being part of the network

→ corps. trying to understand customers track content consumption

Encryption and private computation won't help

Collaboration protocols **leak metadata**

This future is no better than the present

dINSTA

dYTUBE

dSITES

Encryption and private computation won't help

Collaboration protocols **leak metadata**

Centralized → disclose to one entity

Decentralized → disclose to **everyone**

This future is no better than the present

dINSTA

dYTUBE

dSITES

Privacy Goals

Initiator anonymity given a lookup request, initiator **??**

Target anonymity given a lookup initiator, target **??**

Lookup unlikeability given multiple lookups, same initiator **??**

Replication and interest unlikeability storing content **!=** interest (plausible deniability)

[j] Octopus: A Secure and Anonymous DHT Lookup. (Qiyan Wang, Nikita Borisov)

Privacy Goals

Initiator anonymity given a lookup request, initiator **??**

Target anonymity given a lookup initiator, target **??**

Lookup unlikeability given multiple lookups, same initiator **??**

Replication and interest unlikeability storing content **!=** interest (plausible deniability)

Low latency
Decentralized
Scalability

[j] Octopus: A Secure and Anonymous DHT Lookup. (Qiyan Wang, Nikita Borisov)

Threat model

Local adversary

Passive and active attacks

Controls up to 20% of the network nodes

Threat model

Local adversary

Passive and active attacks

Controls up to 20% of the network nodes

How about **global adversaries**?

In DHTs, latency is key

Privacy engineering for P2P networks

As a **P2P application developer/designer**, how can I improve

privacy of the users? *(based on our goals and threat model)*

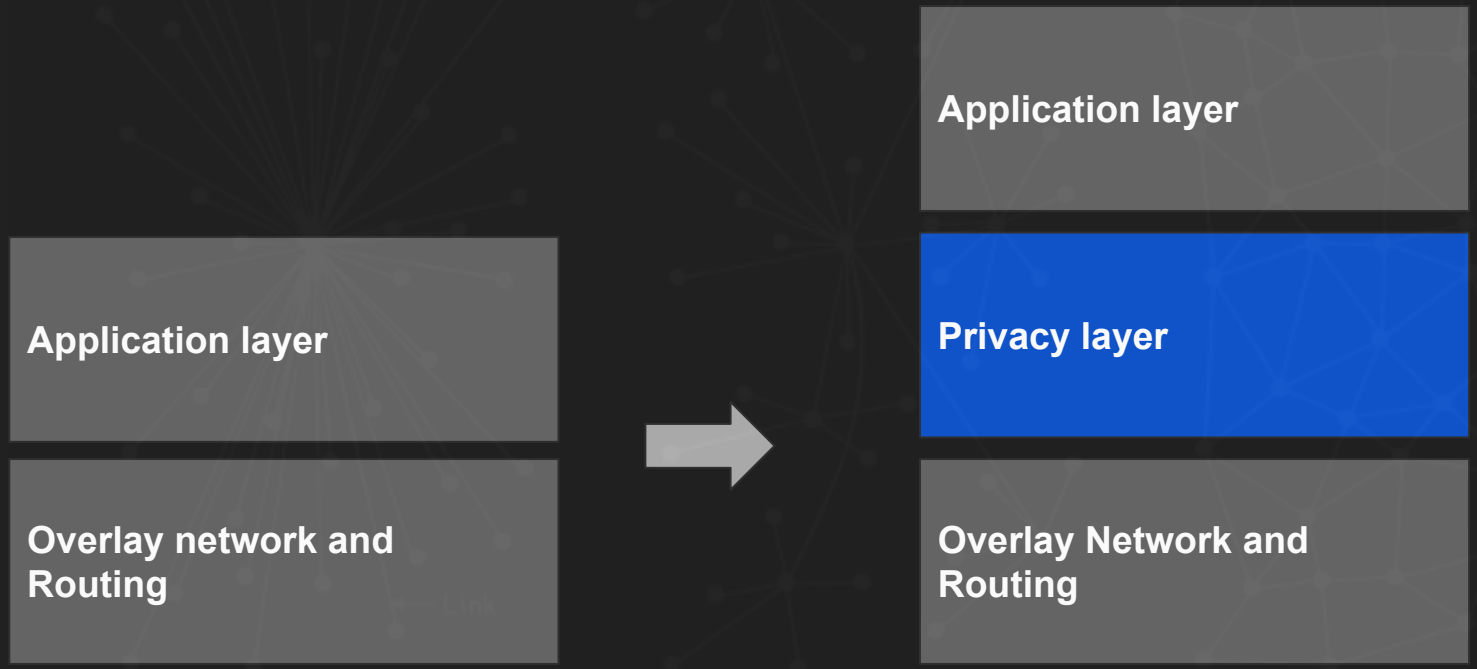
Delegated encrypted requests

Plausible deniability through noise

PIR and Oblivious Transfer

Octopus DHT lookup

Privacy engineering for P2P networks



Delegated encrypted requests

Plausible deniability through noise

PIR and Oblivious Transfer

Octopus DHT lookup

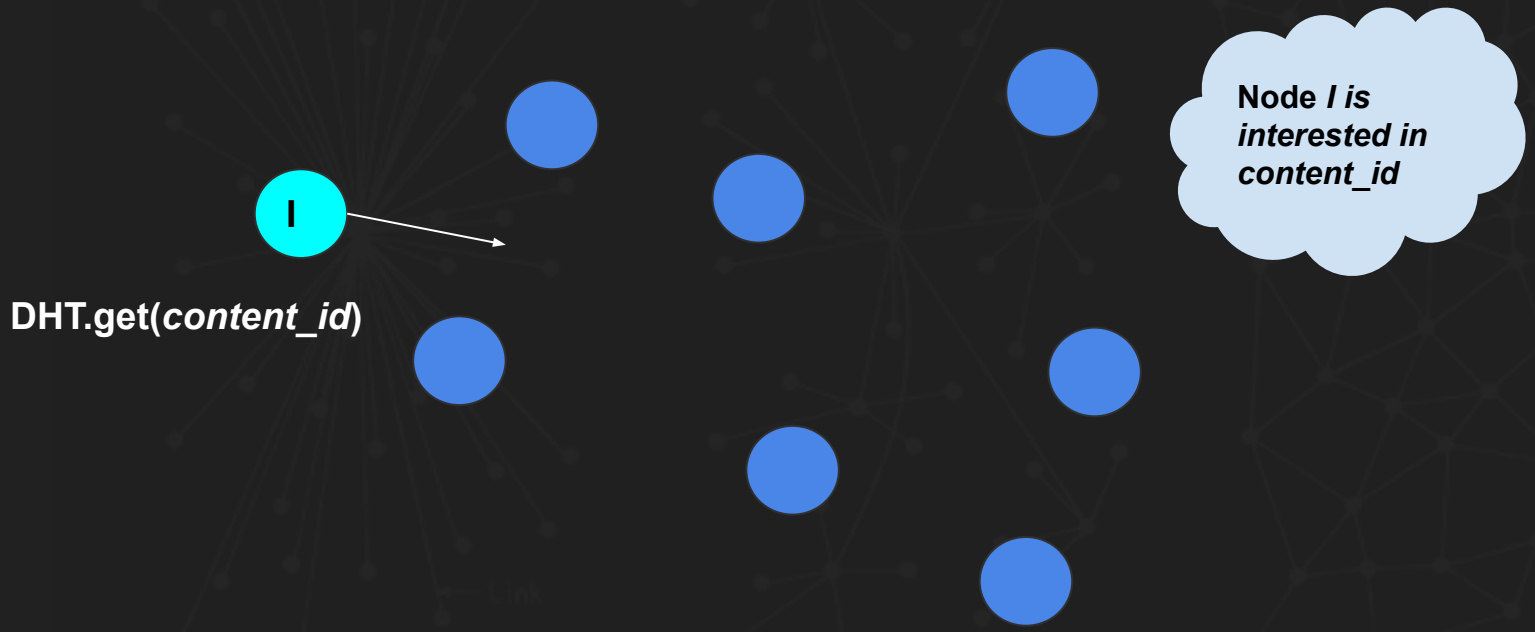
Delegated encrypted requests

Onion routing on top of the DHT → network peers are onion relayers

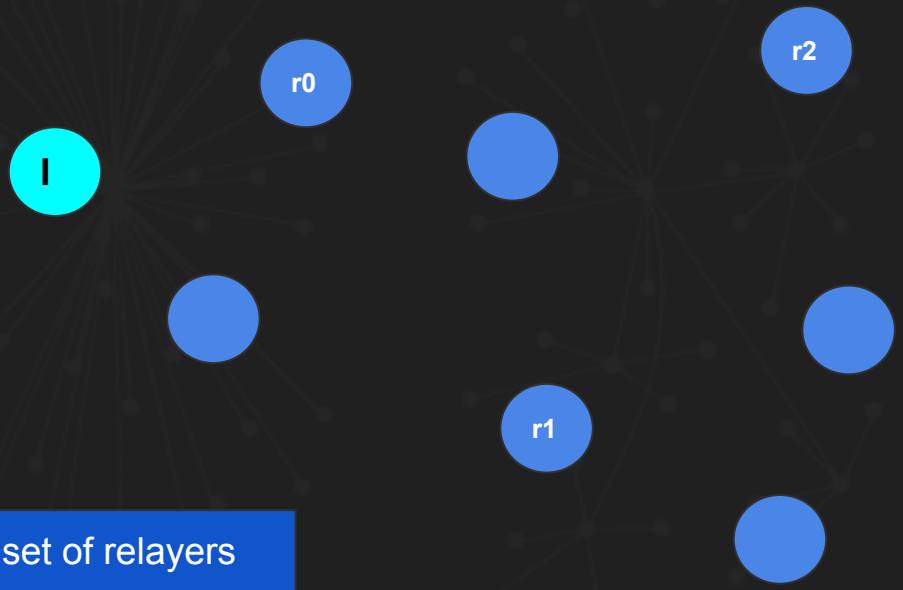
Lookup initiator (/) wraps the lookup request in an onion packet which decrypted by relayers **before the lookup takes place**

→ No relayer learn enough information on who is **initiator**

Delegated encrypted requests



Delegated encrypted requests

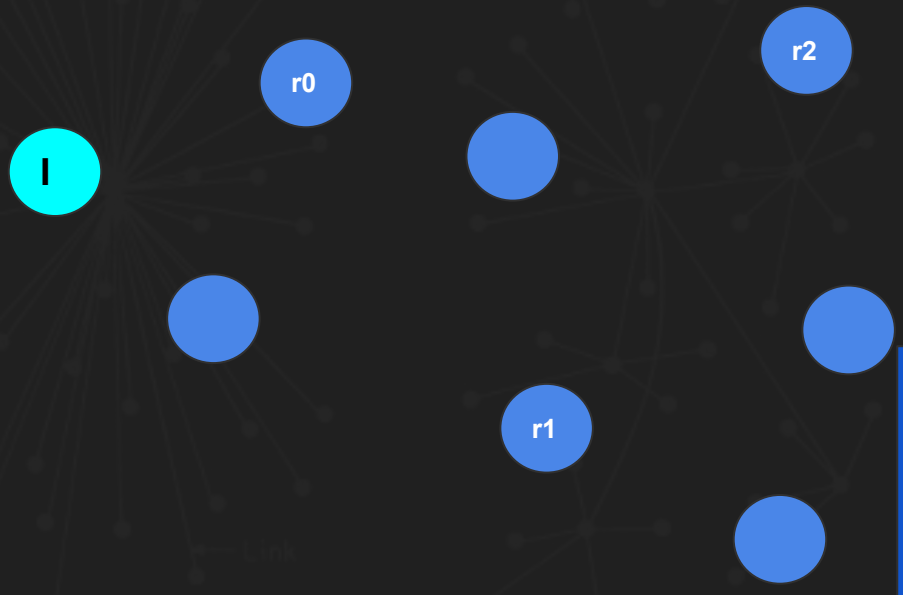


Relayer table
r0: [pubkey, addr]
r1: [pubkey, addr]
r2: [pubkey, addr]
...

Initiator selects a set of relayers

(r0, r1, r2)

Delegated encrypted requests



Relayer table
r0: [pubkey, addr]
r1: [pubkey, addr]
r2: [pubkey, addr]
...

Derives shared keys for each relayer, encrypts the payload in layers with routing info of next relayer

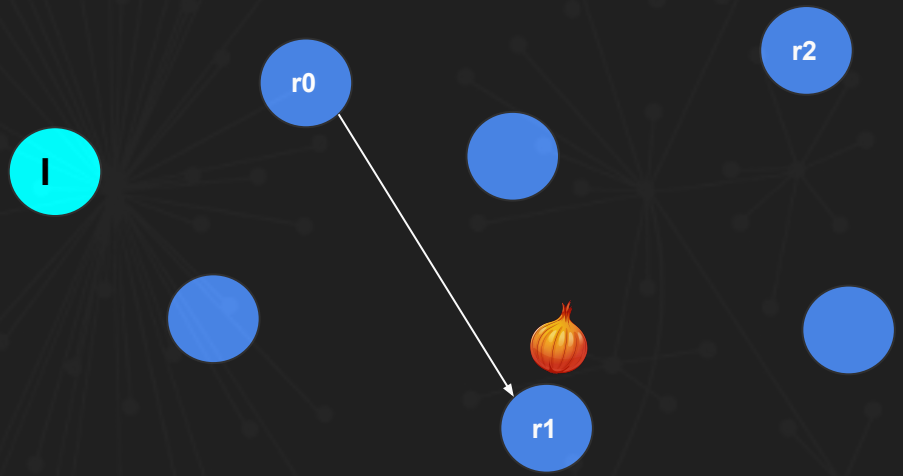
= **newOnionPacket**(relayAddrs, relayPubKeys, **requestPayload**)

Delegated requests with cryptography

Relayer table
r0: [pubkey, addr]
r1: [pubkey, addr]
r2: [pubkey, addr]
...

nextRelay,  **= processPacket(**  **)**

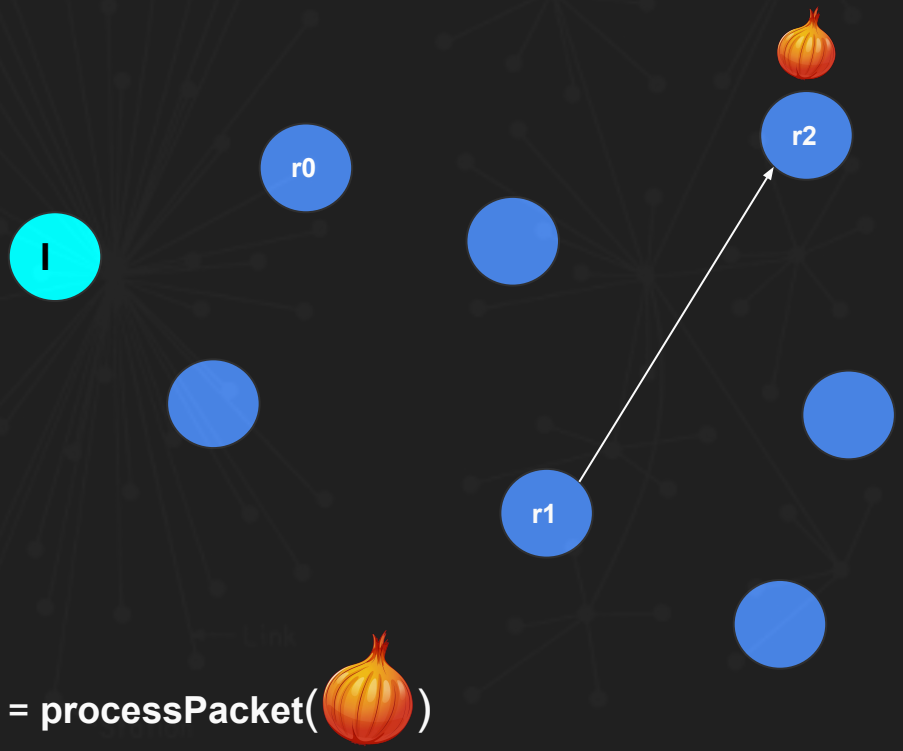
Delegated requests with cryptography



Relayer table
r0: [pubkey, addr]
r1: [pubkey, addr]
r2: [pubkey, addr]
...

nextRelay,  **= processPacket(**  **)**

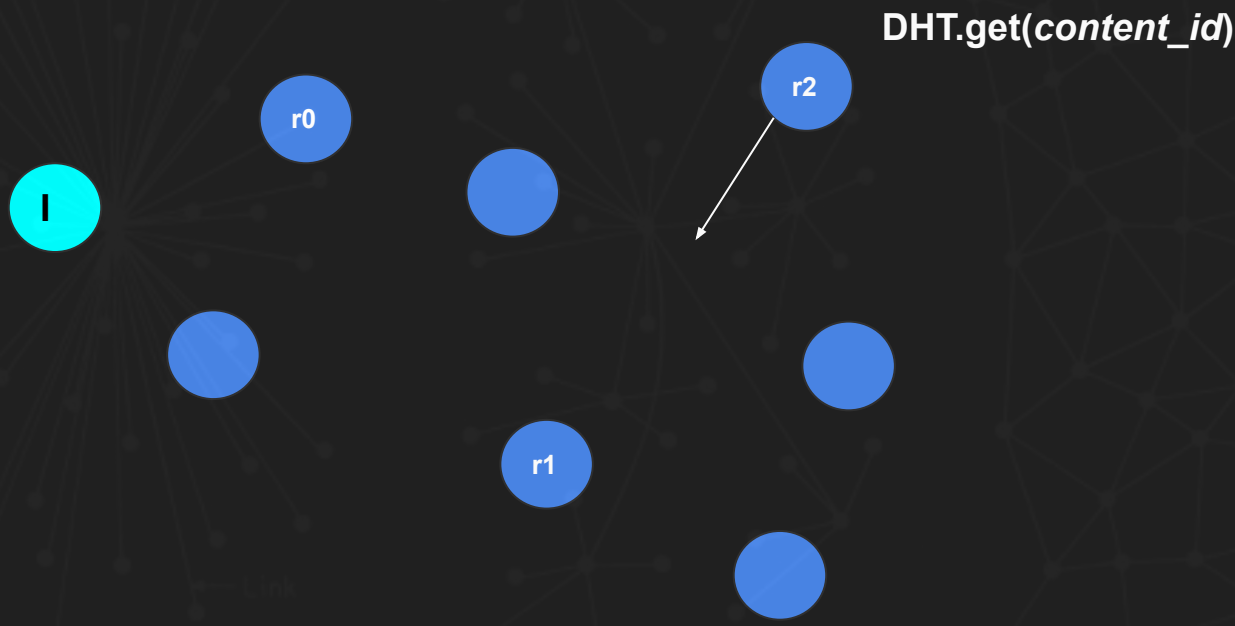
Delegated requests with cryptography



Relayer table
r0: [pubkey, addr]
r1: [pubkey, addr]
r2: [pubkey, addr]
...

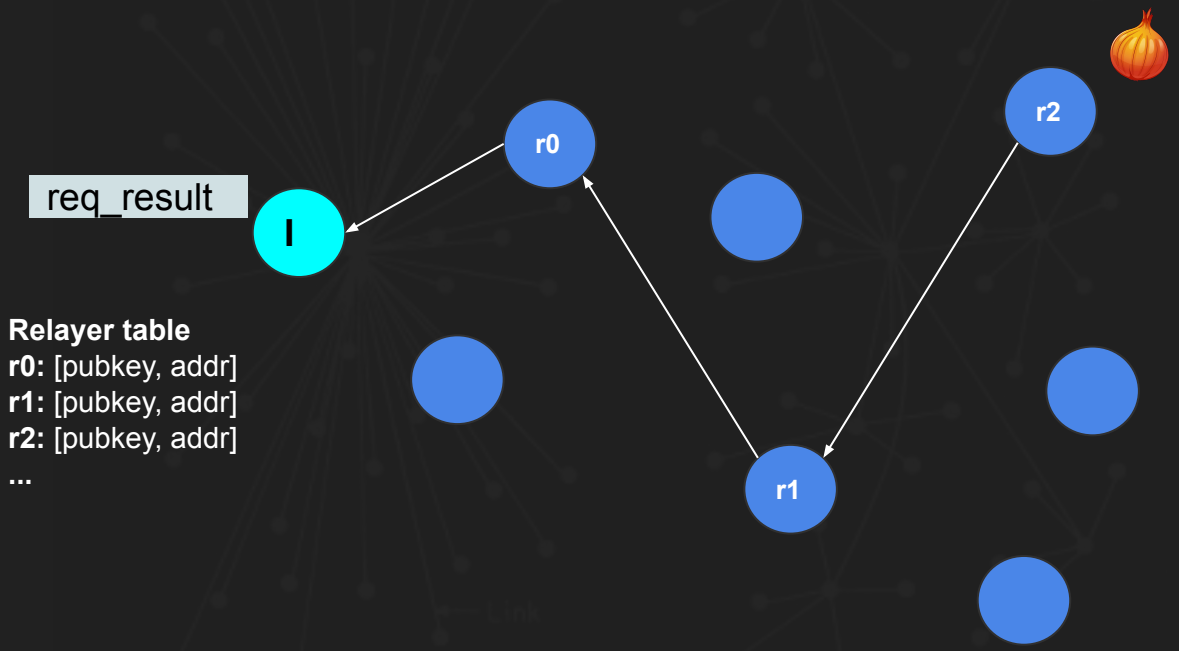
requestPayload = processPacket( )

Delegated encrypted requests



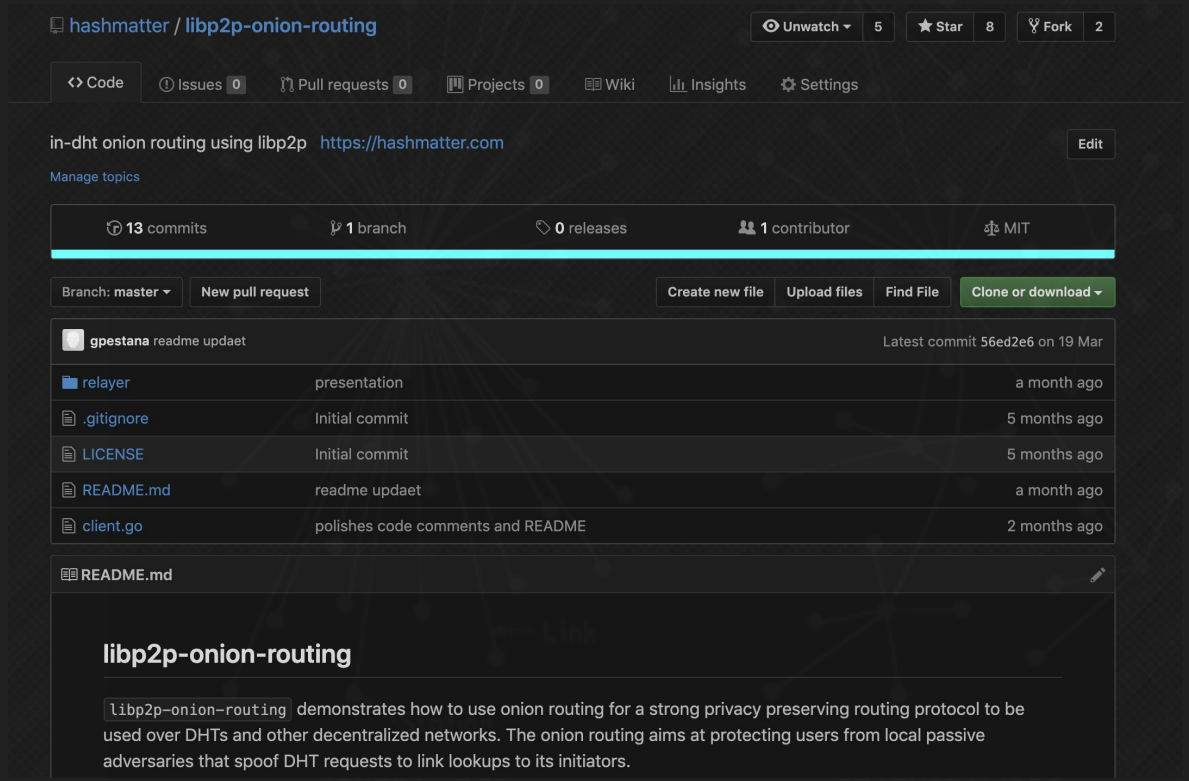
Relayer table
r0: [pubkey, addr]
r1: [pubkey, addr]
r2: [pubkey, addr]
...

Delegated encrypted requests



Relayer table
r0: [pubkey, addr]
r1: [pubkey, addr]
r2: [pubkey, addr]
...

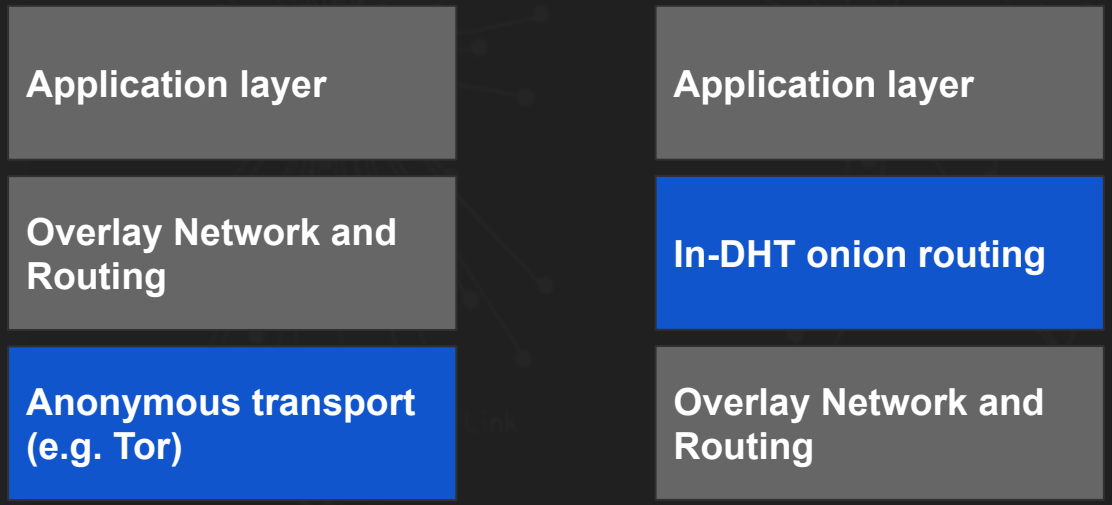
Delegated encrypted requests



Privacy in P2P networks

Delegated encrypted requests

How this different from using anon. networks (e.g Tor)**?**



Delegated encrypted requests

How this different from using anon. networks (e.g Tor)?

- Better latency
- Better usability for user
- More control and flexibility for developer (user)
- Easier to add incentives to the protocol

Delegated encrypted requests

Caveats & open questions

- How to anonymously **select the available relays**
- Anonymous **incentives** for relays *(crypto is free, privacy is not)*
- Decentralized PKI infrastructure
- Entropy and how to measure privacy? I
- Overhead?

Delegated encrypted requests

Plausible deniability through noise

PIR and Oblivious Transfer

Octopus DHT lookup

Plausible deniability through noise

"**peerA** sent me a request, but there is probability of less than 30% that **peerA** was the original lookup initiator"

Protocols that make it impossible to be sure about things

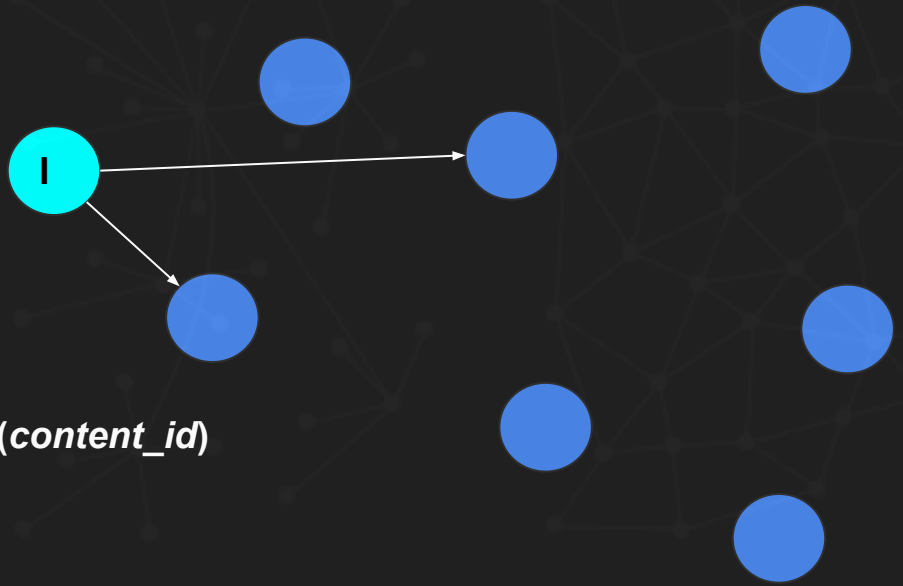
Plausible deniability with ***ρ, k*** parameters***:***

⟶ Lookup could have been requested by at least ***k*** peers

→ There is a probability $P < \rho$ that the lookup request does not map to peer interests

Plausible deniability through noise

Example: Coin Flipping Request Delegation (CFRD)



GET_VALUE(*content_id*)

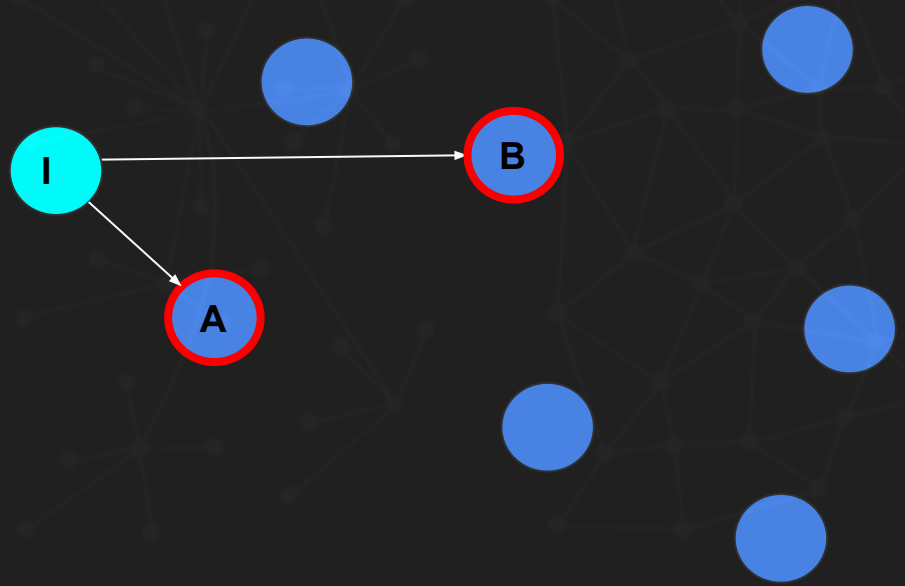
Plausible deniability through noise

Example: Coin Flipping Request Delegation (CFRD)

Peers **A** and **B**:

- Start a new request with probability *k*
- Proceed as defined by protocol with probability (*1-k*)

1 > k > 0

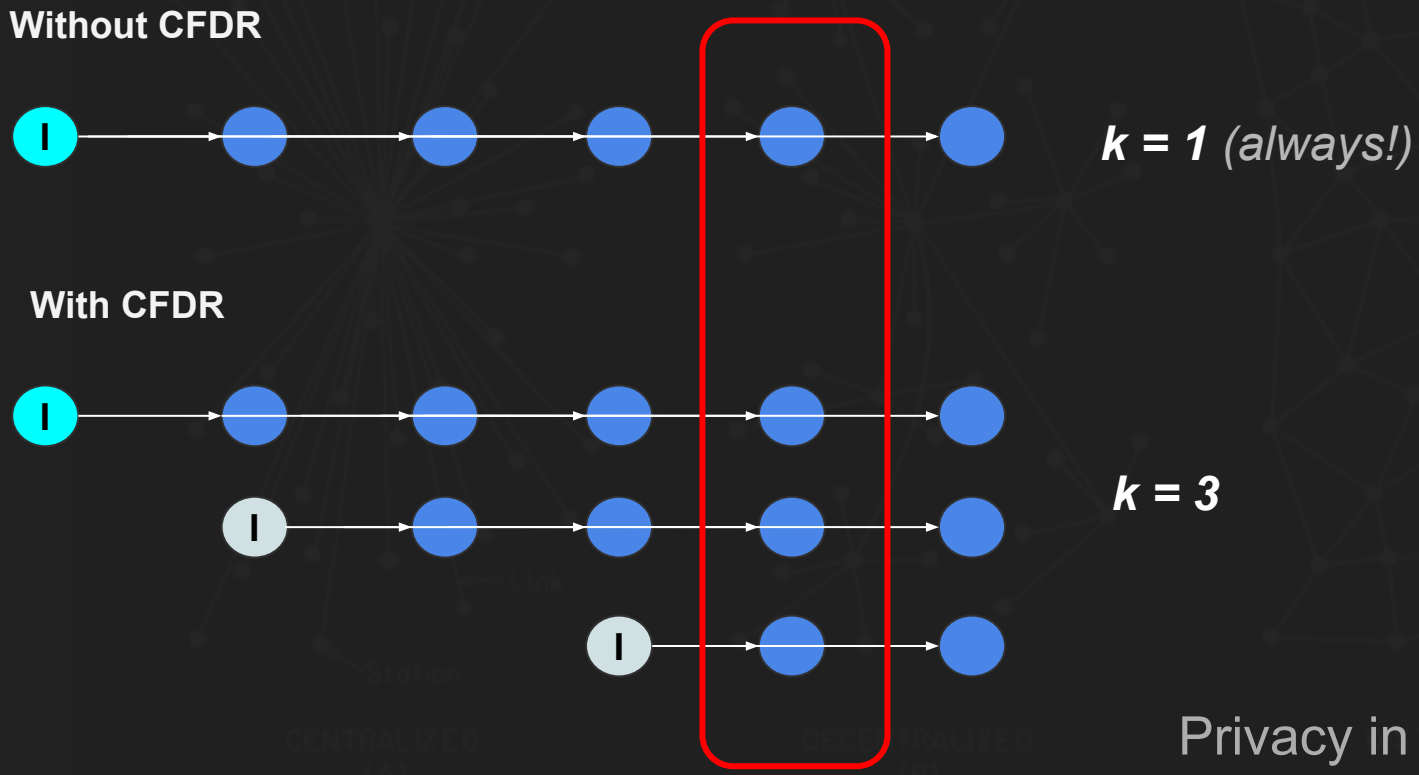


Plausible deniability through noise

Without CFDR

Plausible deniability through noise

Remember: not global adversary



k = 1 *(always!)*

k = 3

Plausible deniability through noise

More sophisticated protocols using **plausible deniability** **p, k, d**

Can be used for

- Plausible deniable requests
- Plausible deniable content caching

Plausible deniability through noise

What about replication-interest unlikeability?

Interest Obfuscation through Random Replication peer replicating ***resource_A*** can deny interest, up to a plausible deniable parameter ***k***

→ peer replicates *(k-1)* extra resources for each "interesting" resource

Plausible deniability through noise

Caveats/ to think about:

- How to measure privacy?
- Random noise vs **useful noise**
- How to ensure nodes behave as expected? (aka Incentives)

Delegated encrypted requests

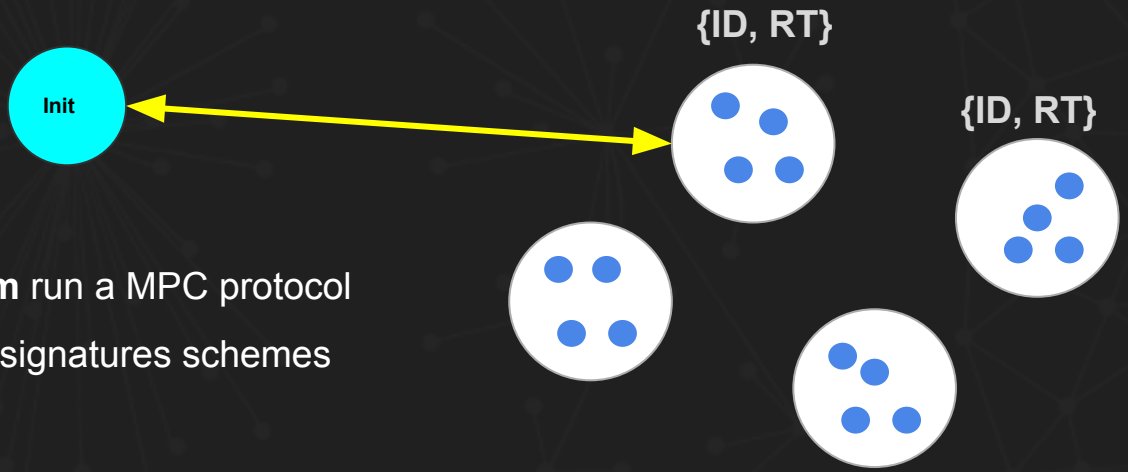
Plausible deniability through noise

PIR and Oblivious Transfer

Octopus DHT lookup

Oblivious transfer for query privacy

Based on Quorum Topology and robust* DHT protocols [k]



Peers in the **quorum** run a MPC protocol

based on threshold signatures schemes

→ private query

* Robust DHT can tolerate bynzantine faults and resist spam (RCP-I [r] and RCP-II [s])

[k] Adding Query Privacy to Robust DHTs (Michael Backes, et. al.)

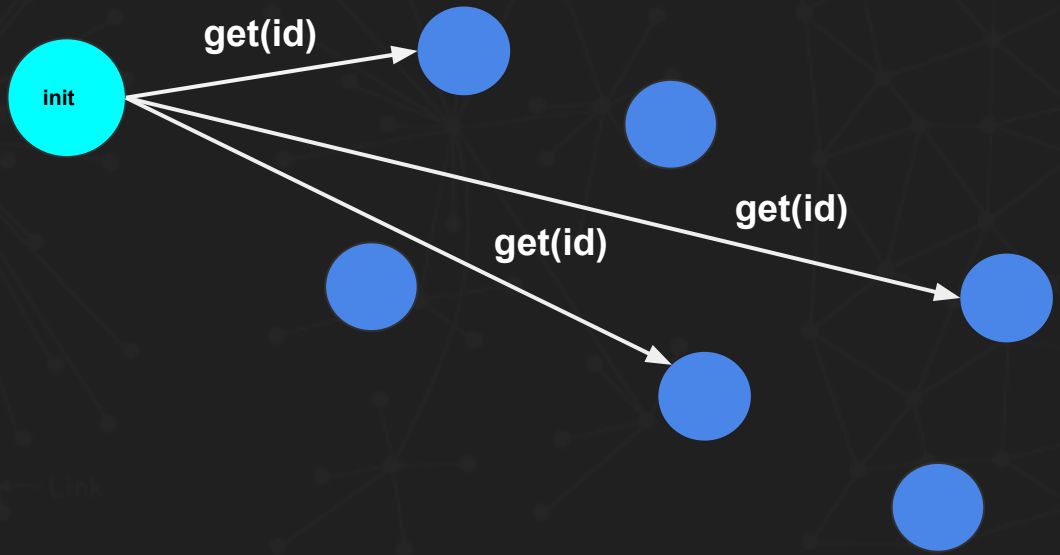
Delegated encrypted requests

Plausible deniability through noise

PIR and Oblivious Transfer

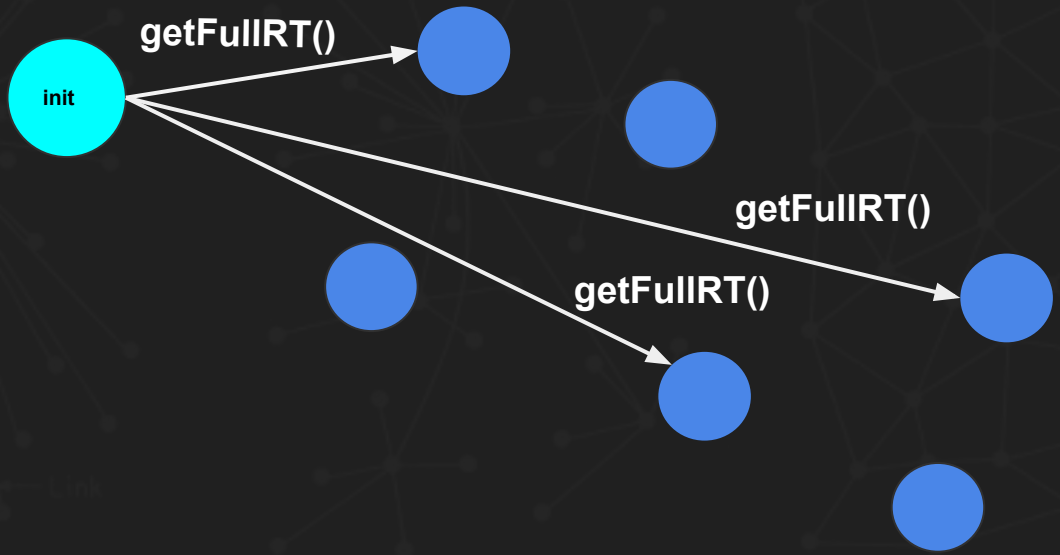
Octopus DHT lookup

What if..

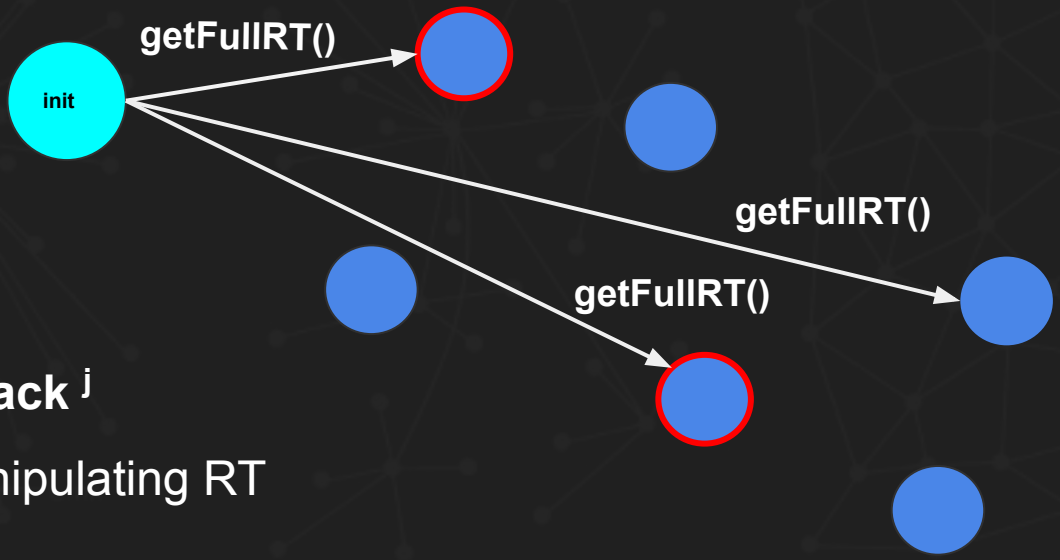


What if..

For target anonymity, peers request full RT table



What if..



Range estimation attack [j]

Active attacks by manipulating RT

Octopus DHT Lookup [j]

Initiator requests **full routing table**

Multiple disjoint lookup paths

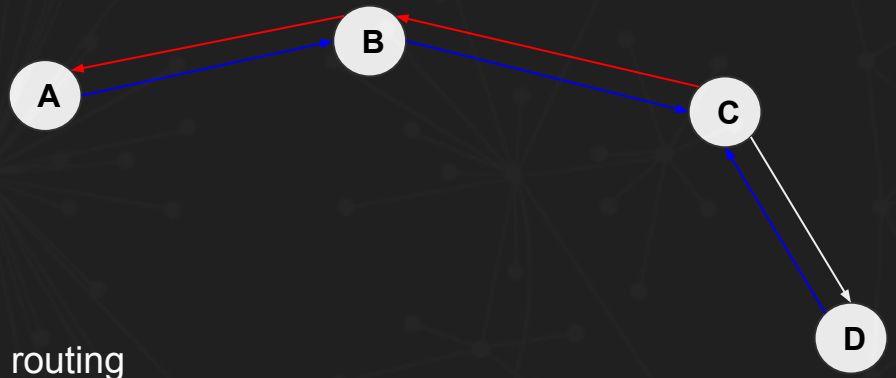
Dummy queries *(noise)*

Assumptions: active adversaries are removed from the network

(shadow nodes checks if routing tables are correct)

[j] Octopus: A Secure and Anonymous DHT Lookup. Qiyan Wang, Nikita Borisov

Bonus: **Restricted routing topology**



Friend to Friend (F2F) routing

Finger table with only **trusted** peers

Small world network principle shortest path to any node in the network is (logarithmically) "small"

	Initiator priv.	Target priv.	Query unlink.	Caching-interest unlink.
F2F routing				
Onion Routing				
Coin Flipping Lookup				
Random replication				
Oblivious Transfer				
Octopus DHT lookup				

Privacy in P2P networks

breakpoint >

How and why DHT (and P2P networks) leak user private information

Mechanisms and protocols to mitigate privacy vulnerabilities

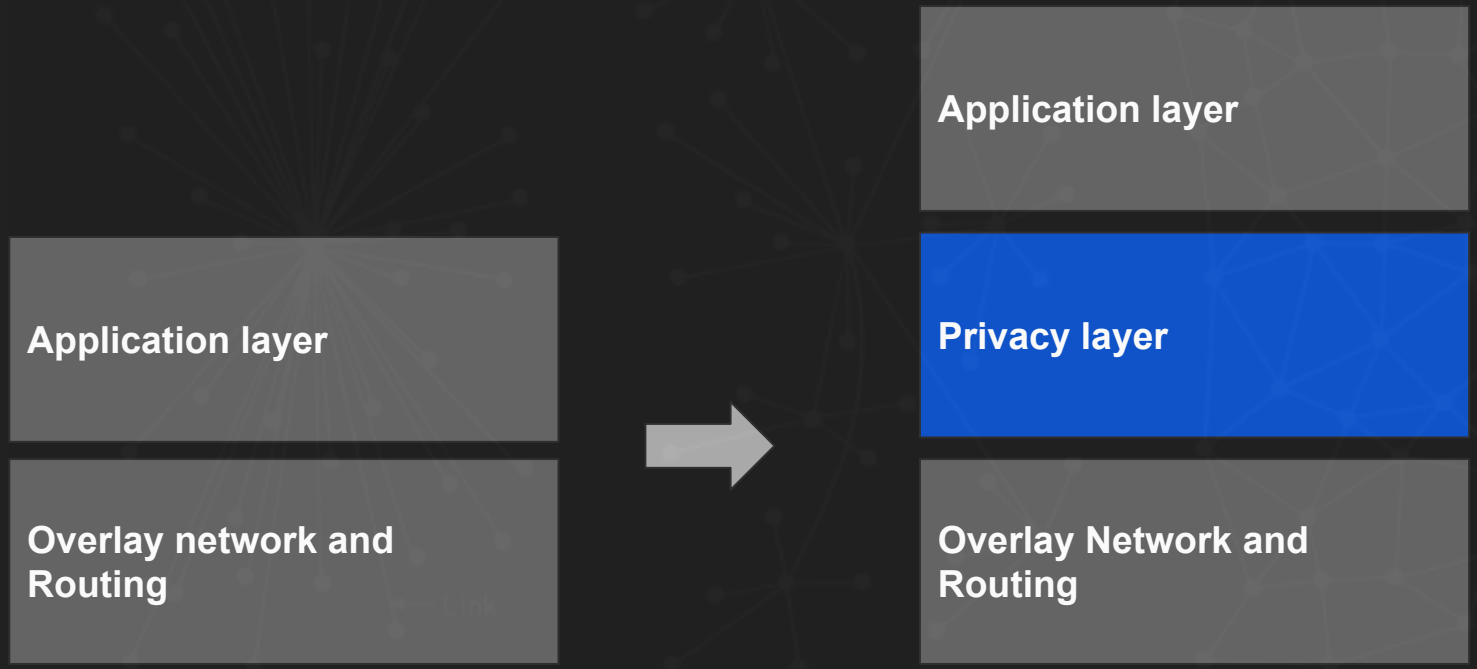
breakpoint >

How and why DHT (and P2P networks) leak user private information

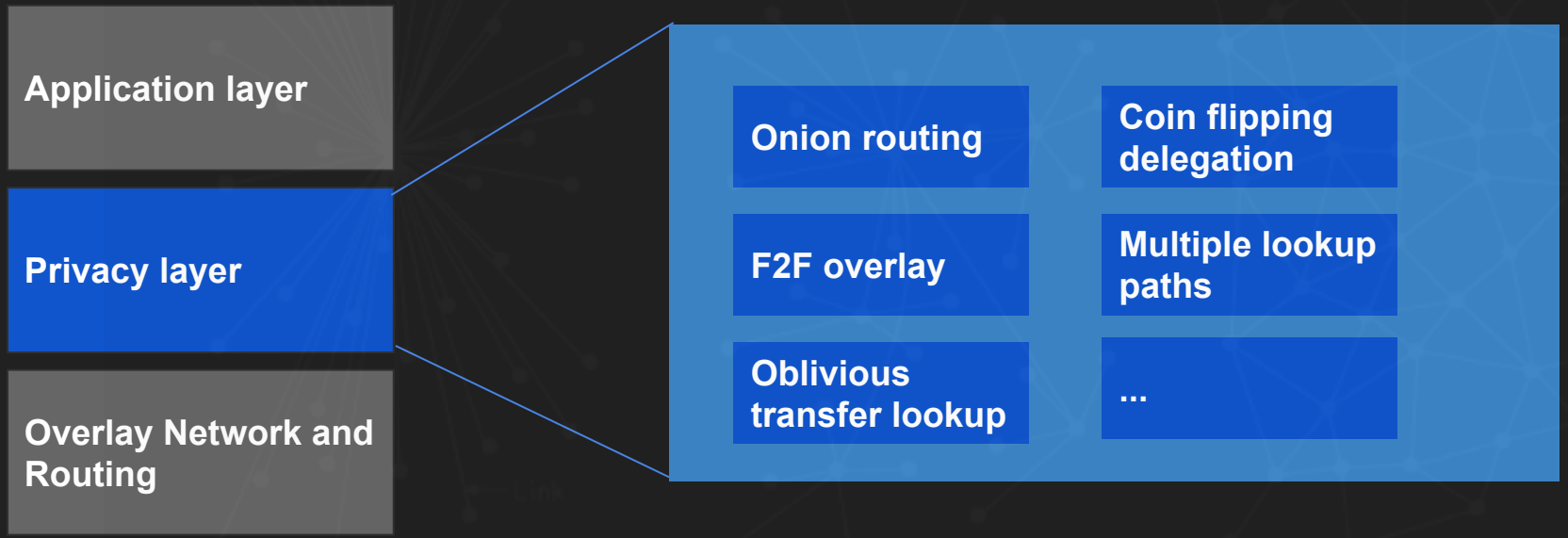
Mechanisms and protocols to mitigate privacy vulnerabilities

Privacy Engineering → putting all together in practice

Privacy engineering for P2P networks



Privacy engineering for P2P networks



Privacy engineering for P2P networks

p3lib https://github.com/hashmatter/p3lib

The toolbox for engineers to enhance privacy in P2P networks

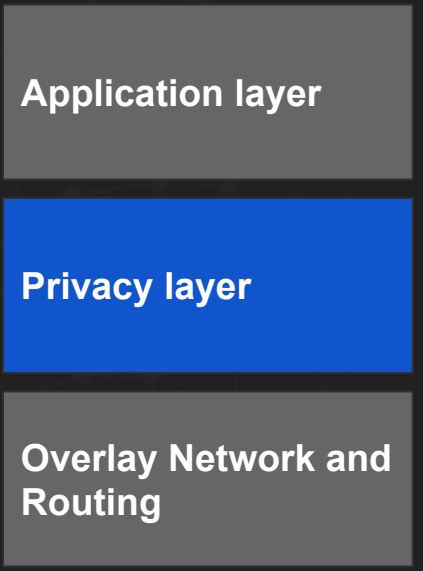
p3lib-sphinx all purpose onion routing implementation

p3lib-cfdr plausible deniability for DHT lookups

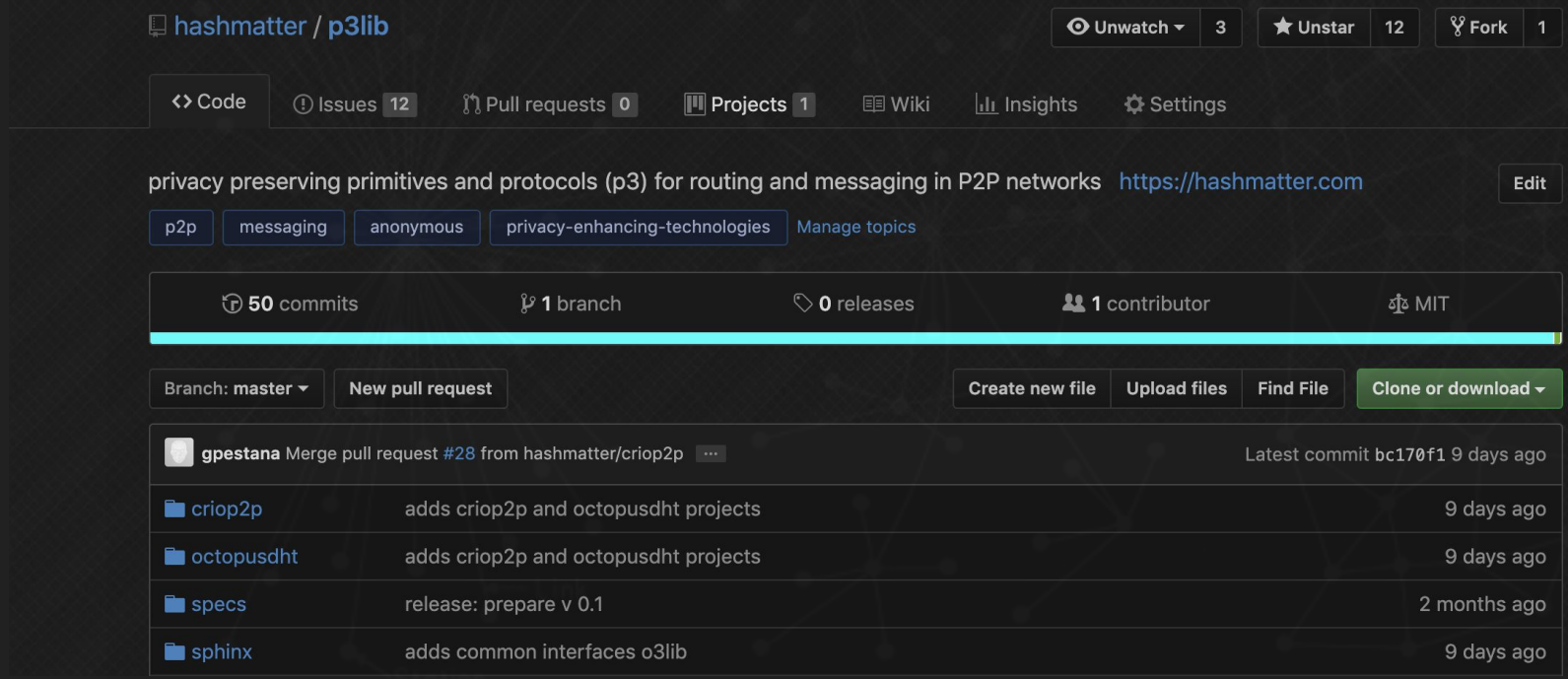
p3lib-octopusdht multipath lookup mechanism with noise for DHT

more..?

p3lib ❤ **libp2p**



Call for action!!



Privacy engineering for P2P networks

interesting problems to be solved, lots of research and engineering open questions

https://hashmatter.com

https://github.com/gpestana/p2psec

Incentives for “private work"

Active attacks detection / prevention

Primitives and protocol development

Scalable and secure PKI infra for OT

Oblivious transfer in practice

Measuring privacy

...

How does the future we're building look like?

data **ownership**

open, **inclusive** and collaborative protocols

no centralized and external stewardship

respects **privacy**

References

[n] Anonymity Trilemma: Strong Anonymity, Low Bandwidth Overhead, Low Latency—Choose Two

[m] Routing in the Dark: Pitch Black. Nathan S. Evans, et al.

[j] Octopus: A Secure and Anonymous DHT Lookup. Qiyan Wang, Nikita Borisov

[l] Why I'm not an Entropist, Paul Syverson

[k] Adding Query Privacy to Robust DHTs (Michael Backes, et. al.)

[r] Practical Robust Communication in DHTs Tolerating a Byzantine Adversary (Maxwell Youngs et. al)

[s] Towards Practical Communication in Byzantine-Resistant DHTs (Maxwell Youngs et. al)