



Gonalo Pestana

<https://gpestana.com>

<https://hashmatter.com>



@gpestana

Engineer & researcher at **hashmatter**

CENTRALIZED
(A)

DECENTRALIZED
(B)

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)

CENTRALIZED
(A)

DECENTRALIZED
(B)

Privacy in P2P networks

Distributed Hash Tables

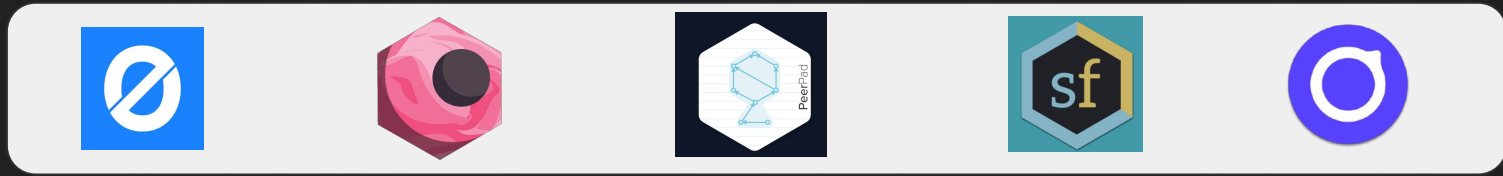
DHT can be used as the **scaffolding** for decentralized applications



Privacy in P2P networks

Distributed Hash Tables

Application layer



Privacy in P2P networks

Distributed Hash Tables

DHT.get(E)



Routing Table
A: addr



CENTRALIZED
(A)

DECENTRALIZED
(B)

Privacy in P2P networks

Distributed Hash Tables

DHT.get(E)

Routing Table
D: *addr*



Routing Table
A: *addr*

Privacy in P2P networks

Distributed Hash Tables

DHT.get(E)

Routing Table
D: *addr*



D: *addr*

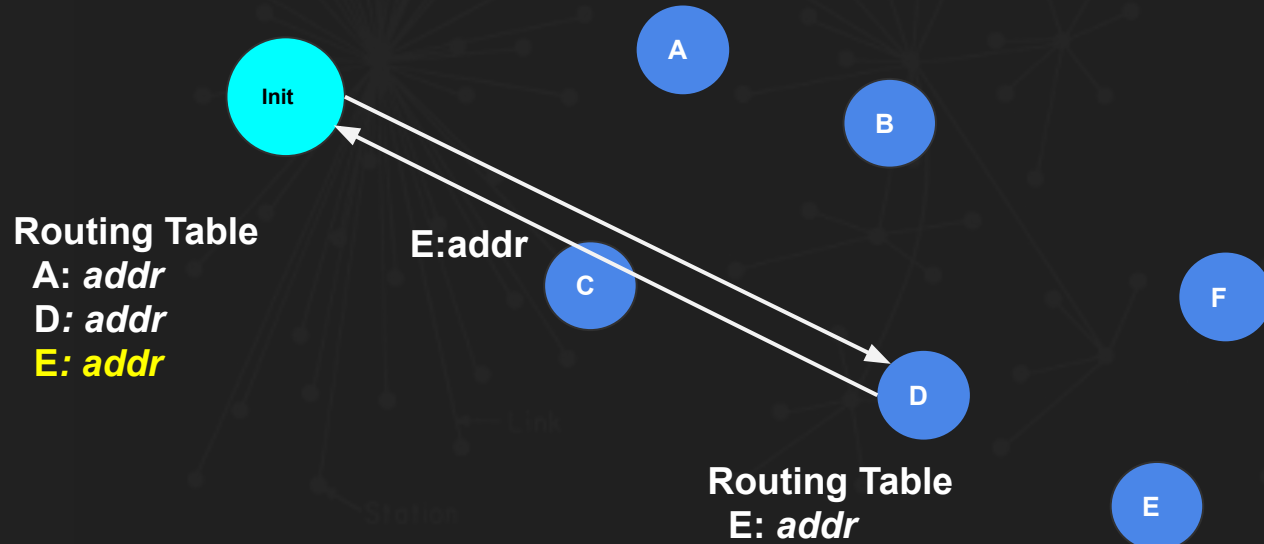
Routing Table
A: *addr*
D: *addr*



Privacy in P2P networks

Distributed Hash Tables

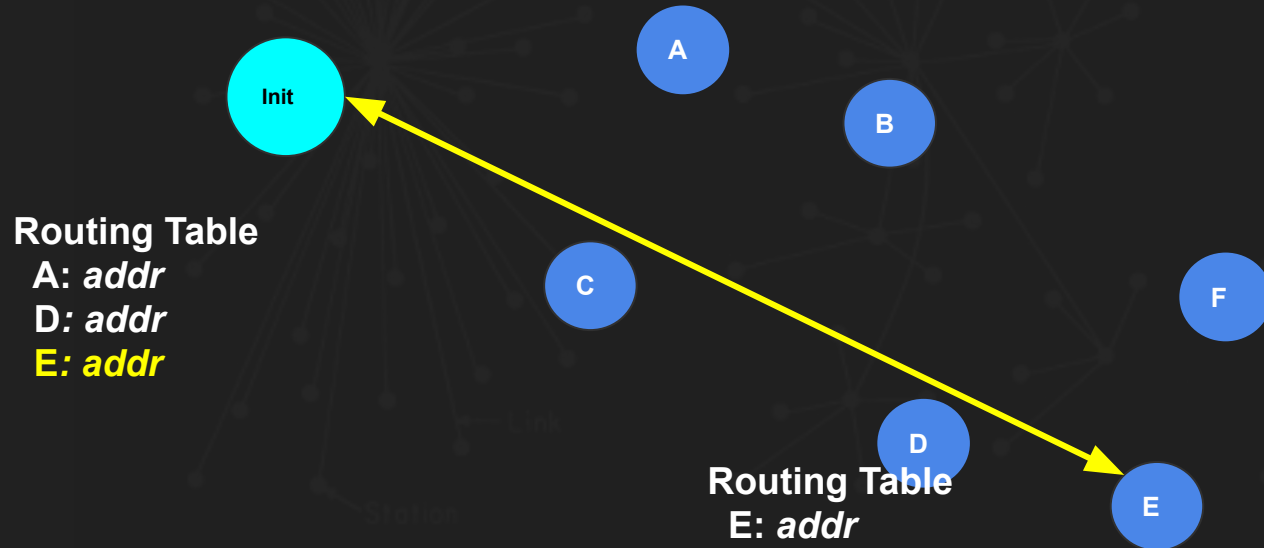
DHT.get(E)



Privacy in P2P networks

Distributed Hash Tables

DHT.get(E)



Privacy in P2P networks

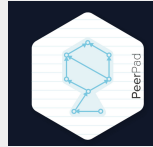


How about privacy?

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

CENTRALIZED
(A)

DECENTRALIZED
(B)

Privacy in P2P networks

(C)

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)
```



Privacy in P2P networks

Case 1) Content provider tracking

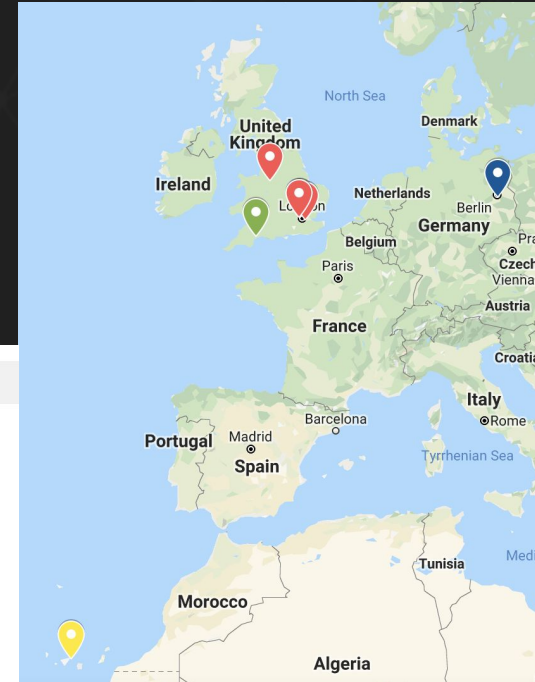


What if an adversary periodically queries the IP Bob's webpage provider?

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

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

- 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





dINSTA

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?



CENTRALIZED
(A)



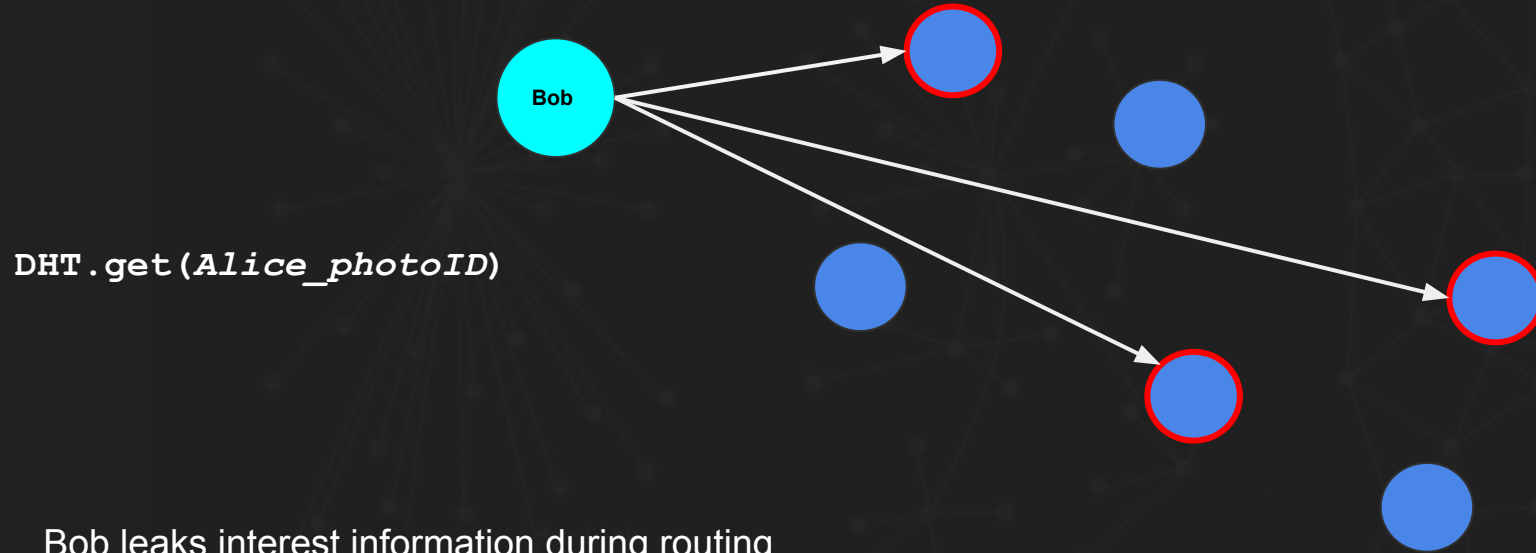
DECENTRALIZED
(B)

Privacy in P2P networks

Case 2) Leaking routing metadata and interest extrapolation



dINSTA



Bob leaks interest information during routing

Trivial to monitor, infer social graph, track Bob

Privacy in P2P networks

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



CENTRALIZED
(A)



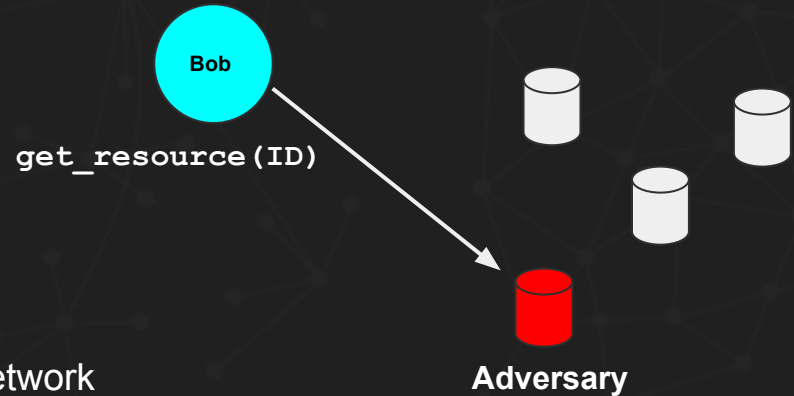
DECENTRALIZED
(B)

Privacy in P2P networks

Case 3) Content providers inference attacks



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

Privacy in P2P networks

Encryption and private computation won't help

Collaboration protocols **leak metadata**

This future is no better than the present



dINSTA



dYTUBE



dSITES

CENTRALIZED
(A)

DECENTRALIZED
(B)

Privacy in P2P networks

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 in P2P networks

Privacy Goals

Initiator anonymity given a lookup request, initiator ??

Target anonymity given a lookup initiator, target ??

Lookup unlikeness given multiple lookups, same initiator ??

Replication and interest unlikeness storing content != interest (plausible deniability)

Privacy Goals

Initiator anonymity given a lookup request, initiator ??

Target anonymity given a lookup initiator, target ??

Lookup unlikenability given multiple lookups, same initiator ??

Replication and interest unlikenability storing content != interest (plausible deniability)

Low latency

Decentralized

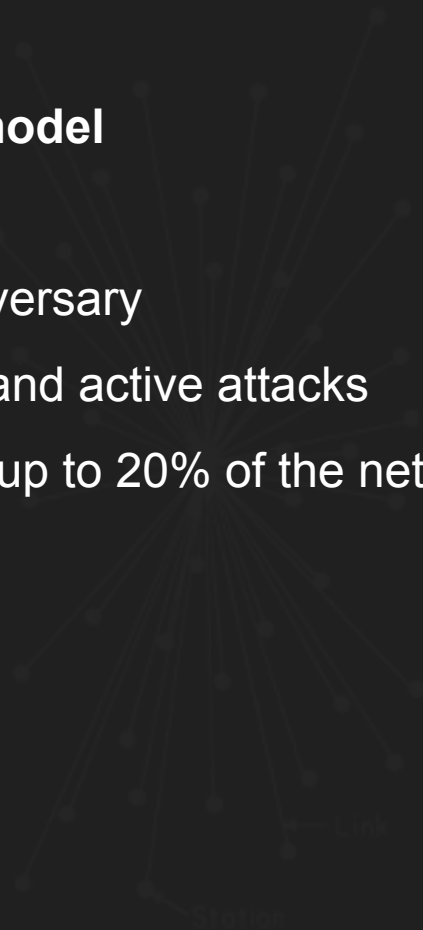
Scalability

Threat model

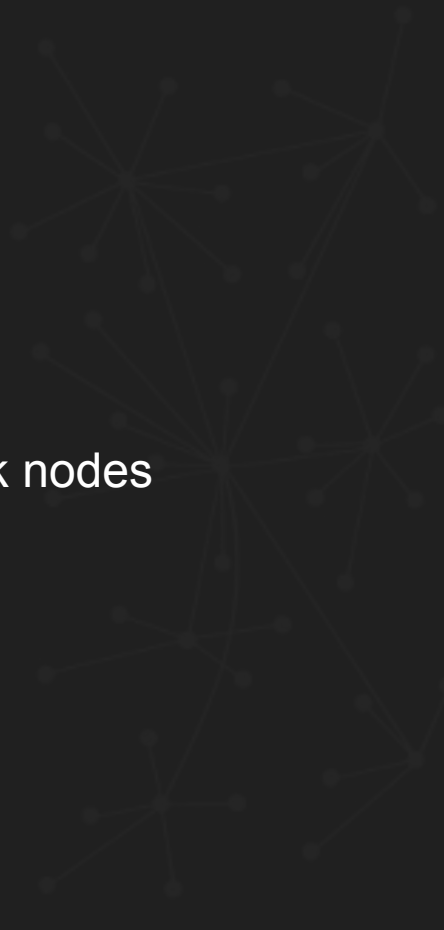
Local adversary

Passive and active attacks

Controls up to 20% of the network nodes



CENTRALIZED
(A)



DECENTRALIZED
(B)



P2P
(C)

Privacy in P2P networks

Threat model

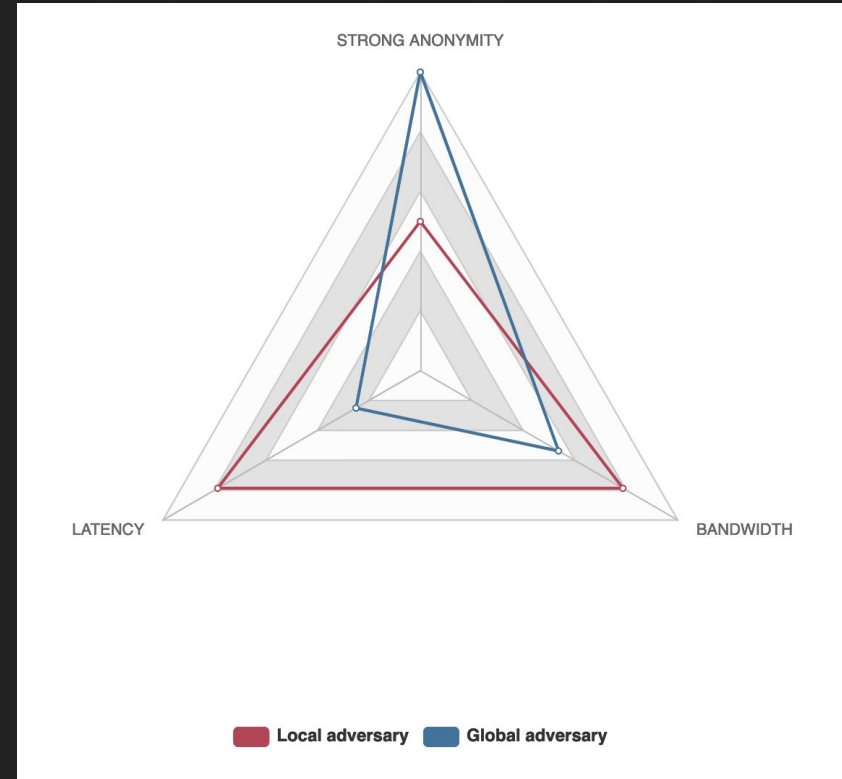
Local adversary

Passive and active attacks

Controls up to 20% of the network nodes

How about **global adversaries**?

In DHTs, latency is key



Privacy in P2P networks

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)*



Privacy in P2P networks

The background features three faint network diagrams. On the left, a 'CENTRALIZED (A)' diagram shows a central node connected to many peripheral nodes. In the middle, a 'DECENTRALIZED (B)' diagram shows a more distributed network with multiple hubs. On the right, a 'P2P (C)' diagram shows a dense, mesh-like network of interconnected nodes.

Delegated encrypted requests

Plausible deniability through noise

PIR and Oblivious Transfer

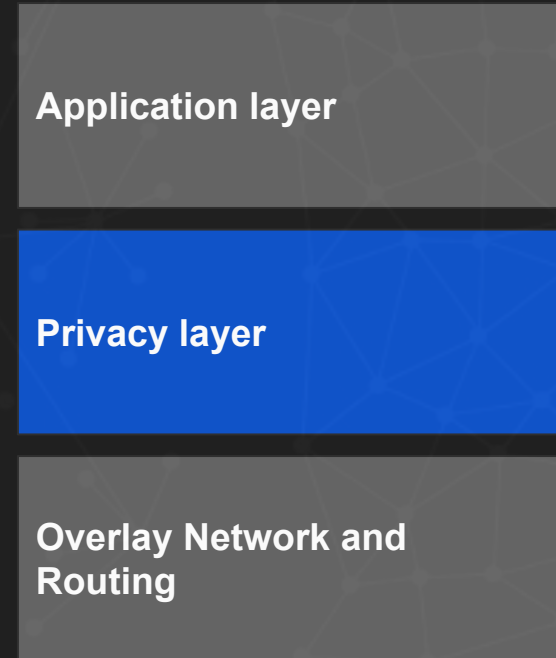
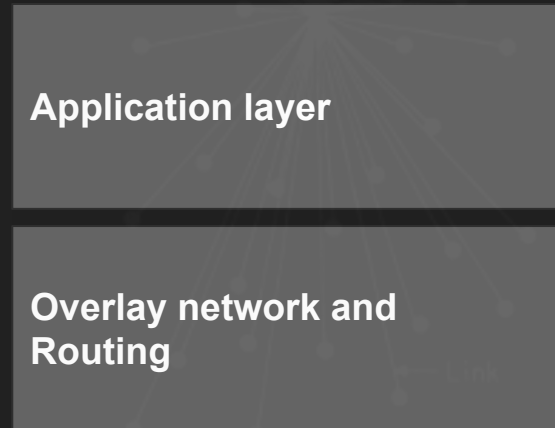
Octopus DHT lookup

CENTRALIZED
(A)

DECENTRALIZED
(B)

Privacy in P2P networks
(C)

Privacy engineering for P2P networks



CENTRALIZED
(A)

DECENTRALIZED
(B)

Privacy in P2P networks

The background features three faint network diagrams. On the left, a 'CENTRALIZED (A)' network shows a single central node connected to many peripheral nodes. In the middle, a 'DECENTRALIZED (B)' network shows a more distributed structure with multiple hubs. On the right, a 'P2P (C)' network shows a dense, mesh-like structure where nodes are interconnected in a decentralized manner.

Delegated encrypted requests

Plausible deniability through noise

PIR and Oblivious Transfer

Octopus DHT lookup

CENTRALIZED
(A)

DECENTRALIZED
(B)

Privacy in P2P networks
(C)

Delegated encrypted requests

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

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

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

Delegated encrypted requests



CENTRALIZED
(A)

DECENTRALIZED
(B)

Privacy in P2P networks

Delegated encrypted requests

I

Relayer table

r0: [pubkey, addr]

r1: [pubkey, addr]

r2: [pubkey, addr]

...

Initiator selects a set of relayers
(r0, r1, r2)

r0

r2

r1

CENTRALIZED
(A)

DECENTRALIZED
(B)

Privacy in P2P networks

Delegated encrypted requests



Relayer table

r0: [pubkey, addr]

r1: [pubkey, addr]

r2: [pubkey, addr]

...

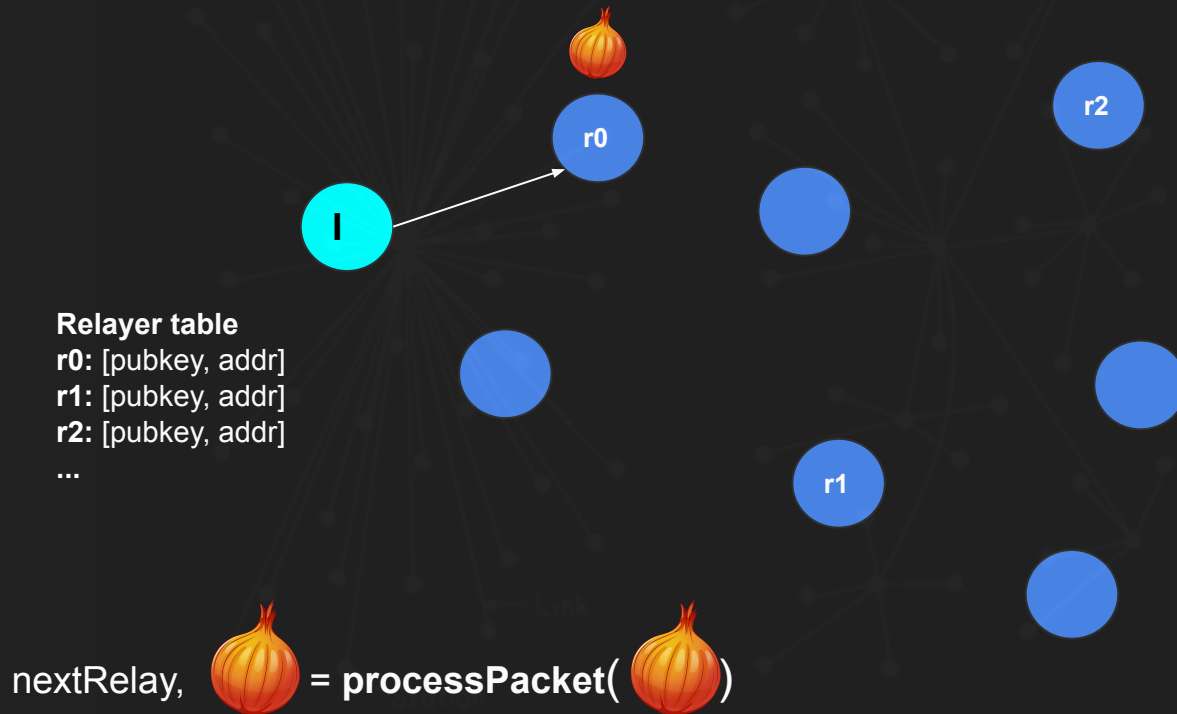
Derives shared keys for each relayer, encrypts the payload in layers with routing info of next relayer



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

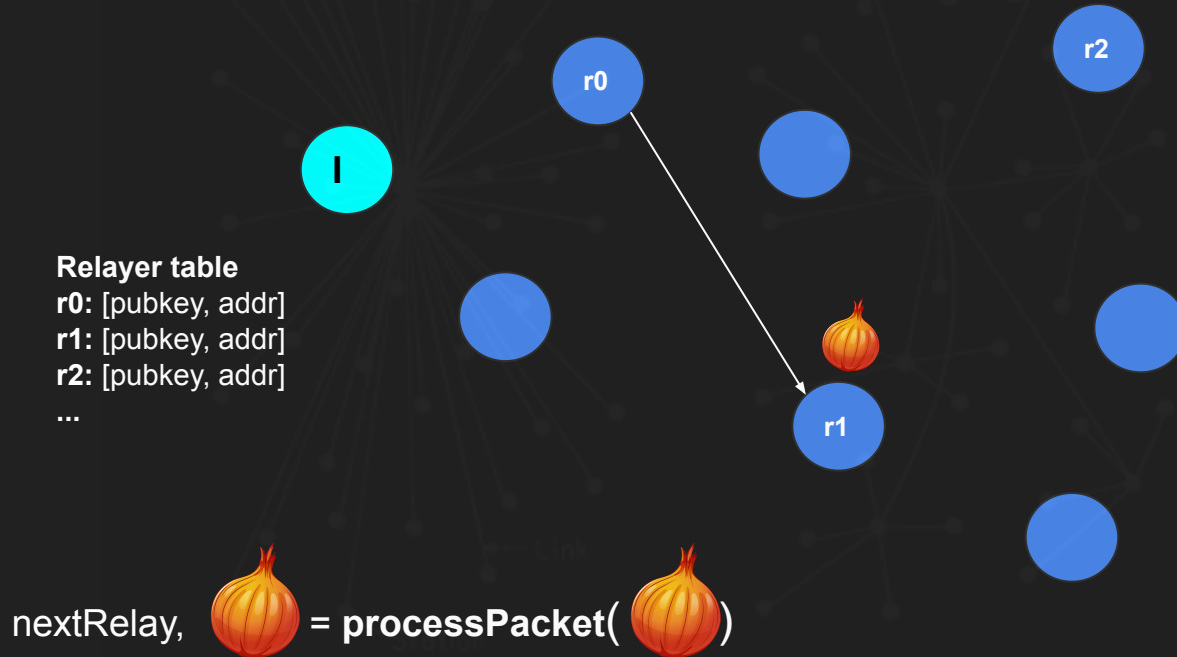
Privacy in P2P networks

Delegated requests with cryptography



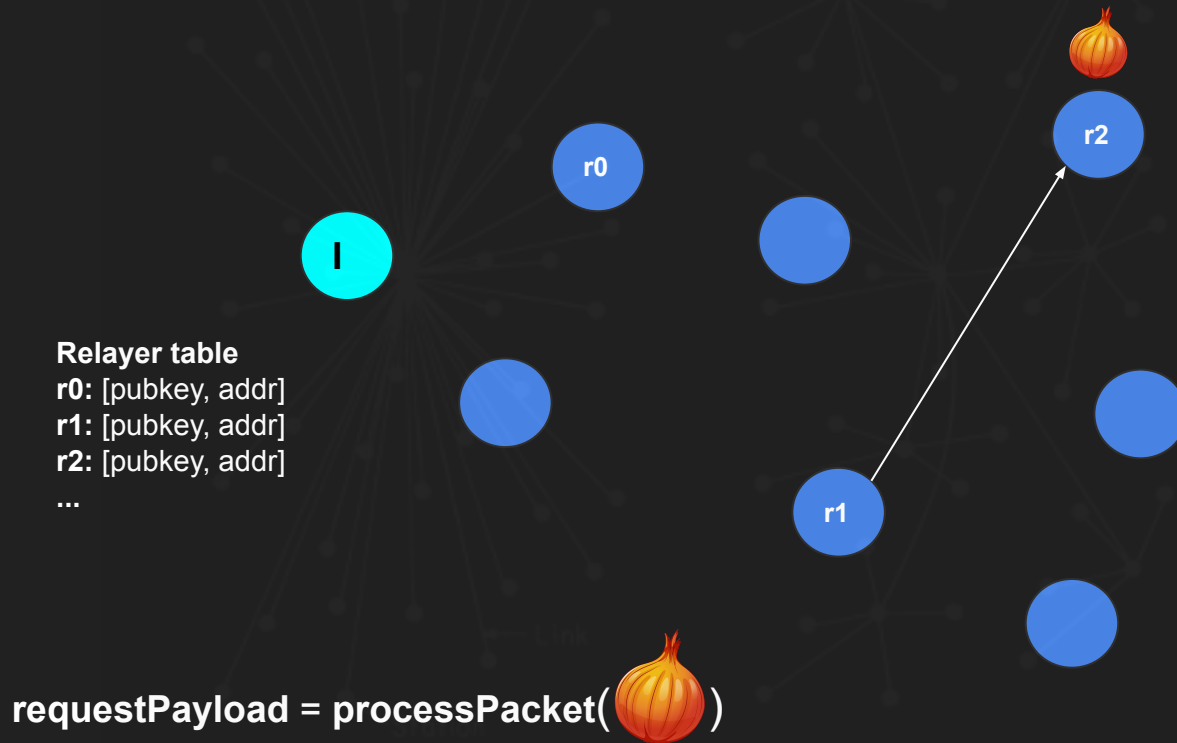
Privacy in P2P networks

Delegated requests with cryptography



Privacy in P2P networks

Delegated requests with cryptography



Privacy in P2P networks

Delegated encrypted requests

Relayer table

r0: [pubkey, addr]

r1: [pubkey, addr]

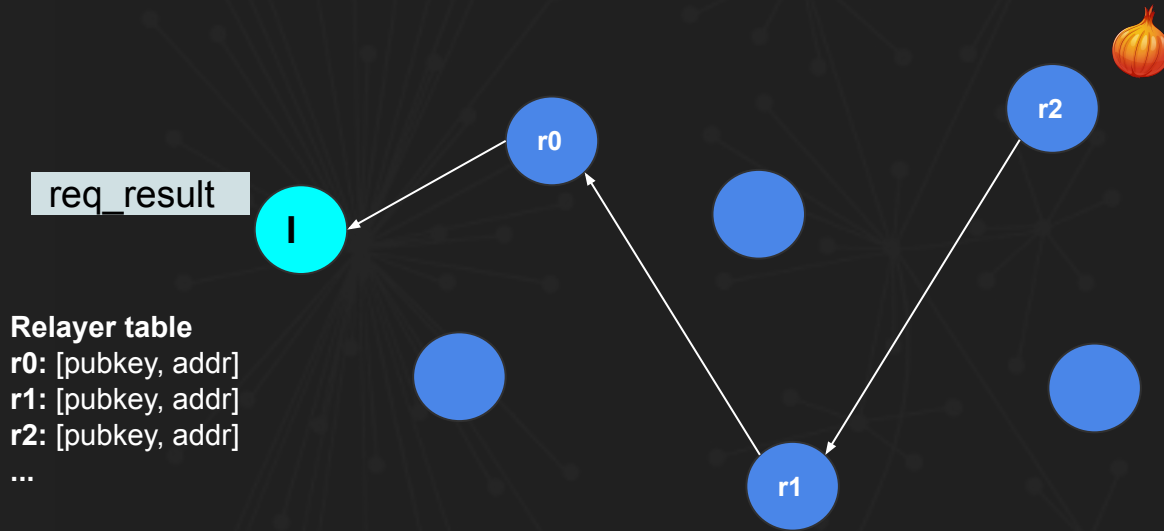
r2: [pubkey, addr]

...

DHT.get(*content_id*)

Privacy in P2P networks

Delegated encrypted requests



CENTRALIZED
(A)

DECENTRALIZED
(B)

Privacy in P2P networks

Delegated encrypted requests

hashmatter / libp2p-onion-routing

Unwatch 5 Star 8 Fork 2

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

in-dht onion routing using libp2p <https://hashmatter.com> Edit

Manage topics

13 commits 1 branch 0 releases 1 contributor MIT

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

gpestana readme updaet Latest commit 56ed2e6 on 19 Mar

relayer	presentation	a month ago
.gitignore	Initial commit	5 months ago
LICENSE	Initial commit	5 months ago
README.md	readme updaet	a month ago
client.go	polishes code comments and README	2 months ago

README.md

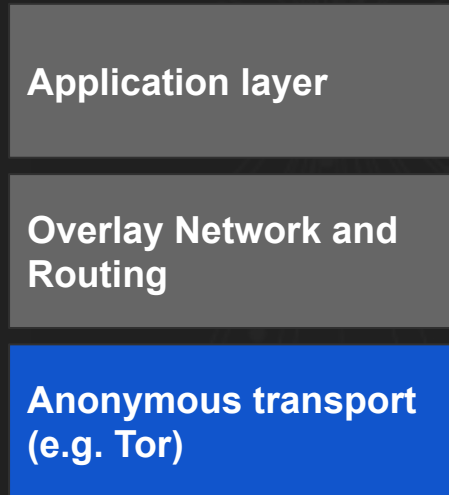
libp2p-onion-routing

`libp2p-onion-routing` demonstrates how to use onion routing for a strong privacy preserving routing protocol to be used over DHTs and other decentralized networks. The onion routing aims at protecting users from local passive adversaries that spoof DHT requests to link lookups to its initiators.

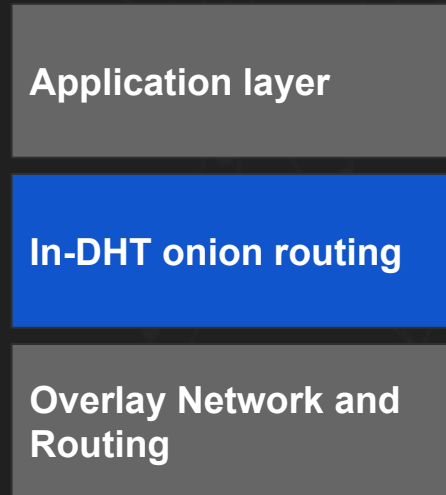
Privacy in P2P networks

Delegated encrypted requests

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



CENTRALIZED
(CA)



DECENTRALIZED
(DHT)

Privacy in P2P networks

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? ¹
- Overhead?

The background features three faint network diagrams. On the left, a 'CENTRALIZED (A)' diagram shows a central node connected to many peripheral nodes. In the middle, a 'DECENTRALIZED (B)' diagram shows a more distributed network with multiple hubs. On the right, a 'P2P (C)' diagram shows a highly interconnected mesh of nodes.

Delegated encrypted requests

Plausible deniability through noise

PIR and Oblivious Transfer

Octopus DHT lookup

CENTRALIZED
(A)

DECENTRALIZED
(B)

Privacy in P2P networks
(C)

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

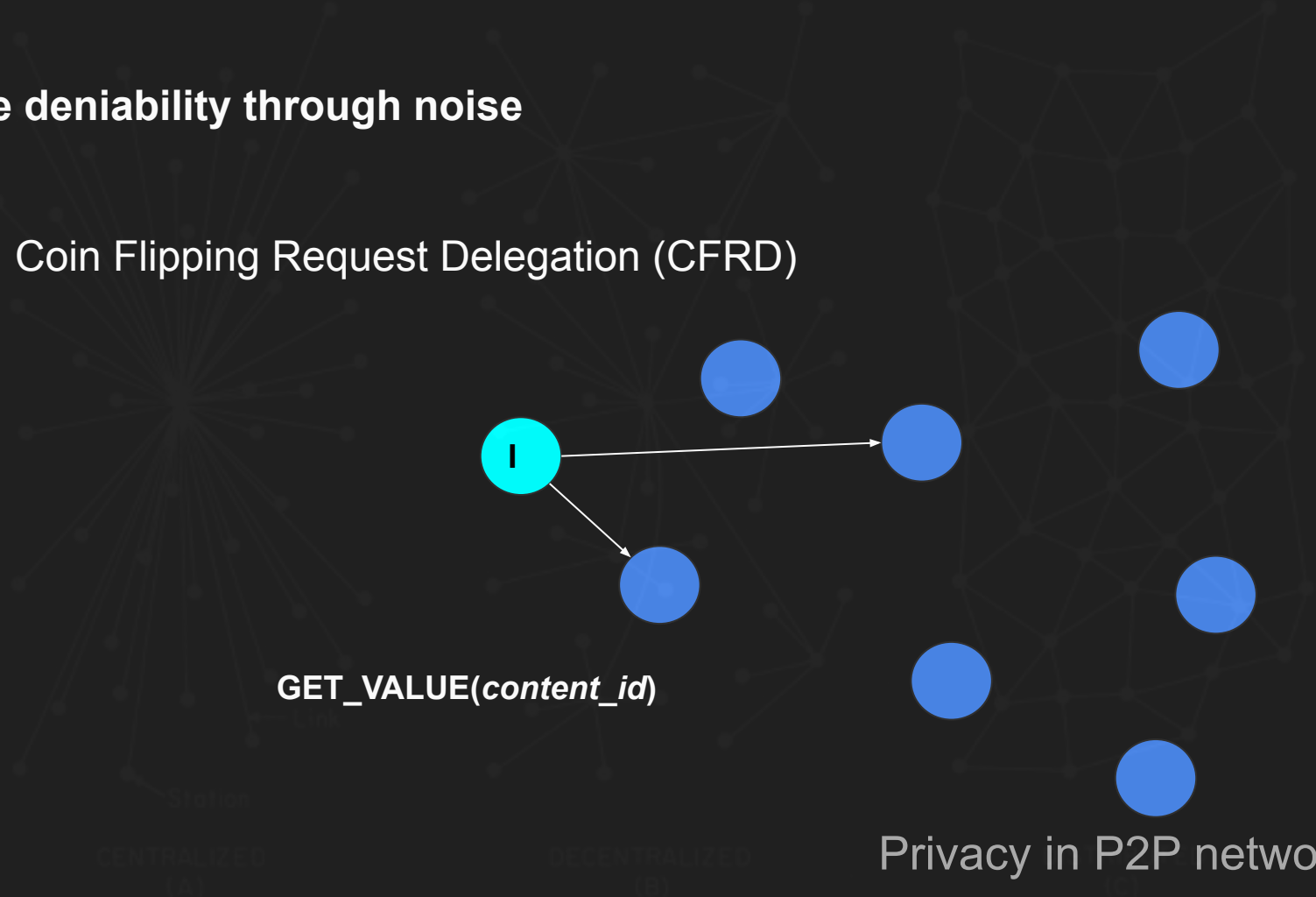
→ There is a probability $P < \rho$ that the lookup request does not map to peer interests

Plausible deniability through noise

Example: Coin Flipping Request Delegation (CFRD)

`GET_VALUE(content_id)`

Privacy in P2P networks



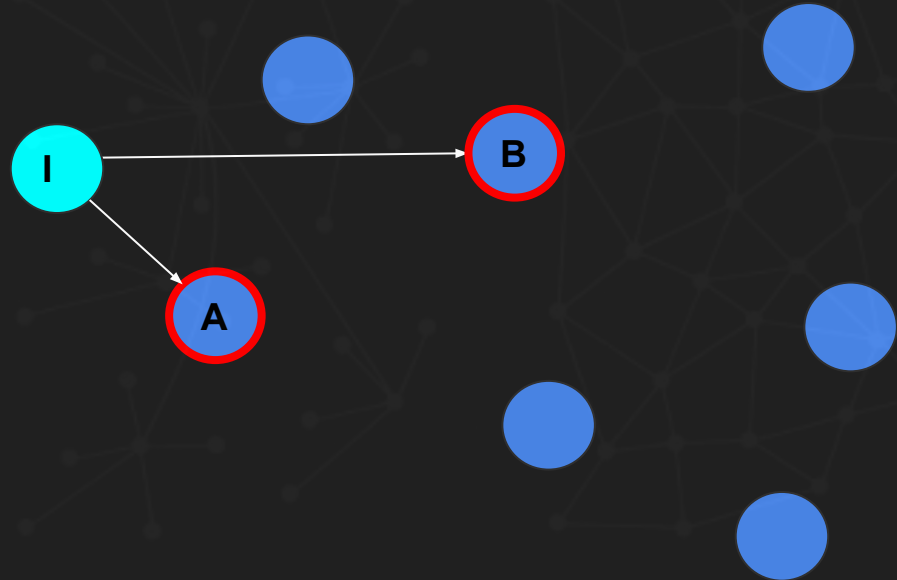
Plausible deniability through noise

Example: Coin Flipping Request Delegation (CFRD)

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

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

$$1 > k > 0$$



Privacy in P2P networks

Plausible deniability through noise

Without CFDR



CENTRALIZED
(A)

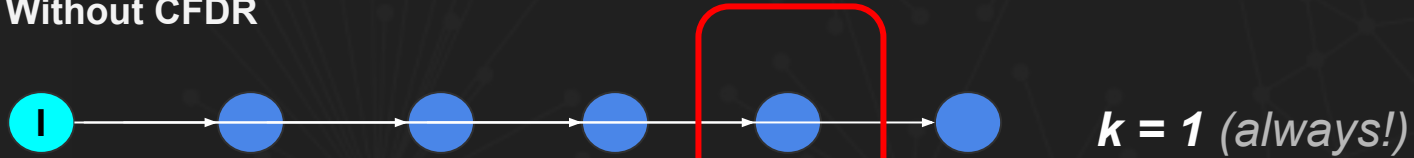
DECENTRALIZED
(B)

Privacy in P2P networks

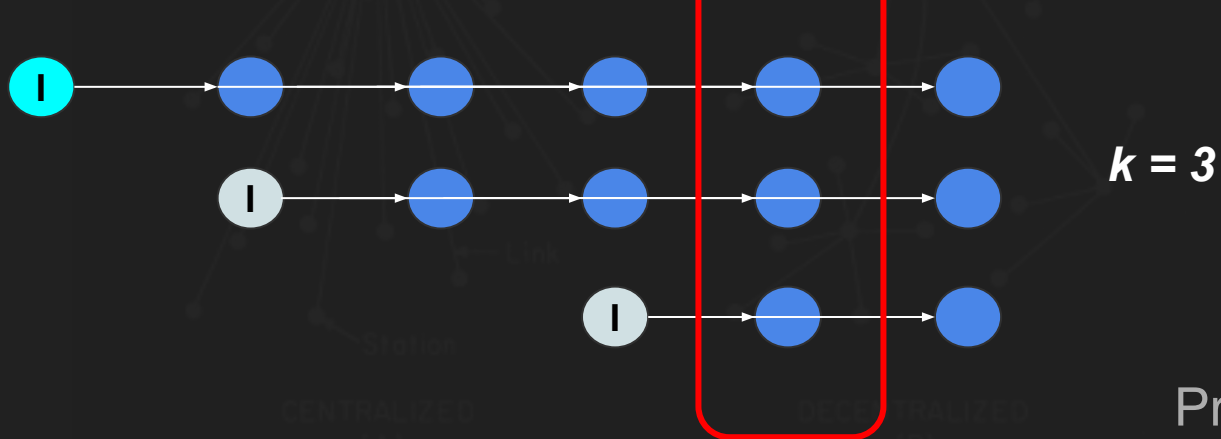
Remember: not global adversary

Plausible deniability through noise

Without CFDR



With CFDR



Privacy in P2P networks

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)

The background features three faint network diagrams. On the left, a 'CENTRALIZED (A)' diagram shows a single central node connected to many peripheral nodes. In the middle, a 'DECENTRALIZED (B)' diagram shows a more complex, interconnected web of nodes. On the right, a 'P2P (C)' diagram shows a highly dense, mesh-like network structure.

Delegated encrypted requests

Plausible deniability through noise

PIR and Oblivious Transfer

Octopus DHT lookup

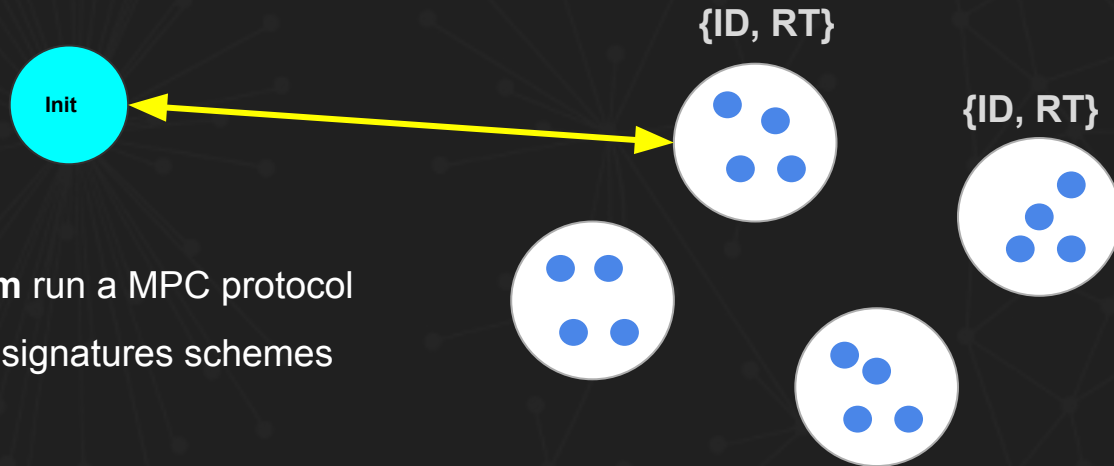
CENTRALIZED
(A)

DECENTRALIZED
(B)

Privacy in P2P networks
(C)

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 byzantine faults and resist spam (RCP-I [r] and RCP-II [s])

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

The background features three faint network diagrams. On the left, a 'CENTRALIZED (A)' diagram shows a central node connected to many peripheral nodes. In the middle, a 'DECENTRALIZED (B)' diagram shows a more distributed network with multiple hubs. On the right, a 'P2P (C)' diagram shows a dense, mesh-like network where every node is connected to many others.

Delegated encrypted requests

Plausible deniability through noise

PIR and Oblivious Transfer

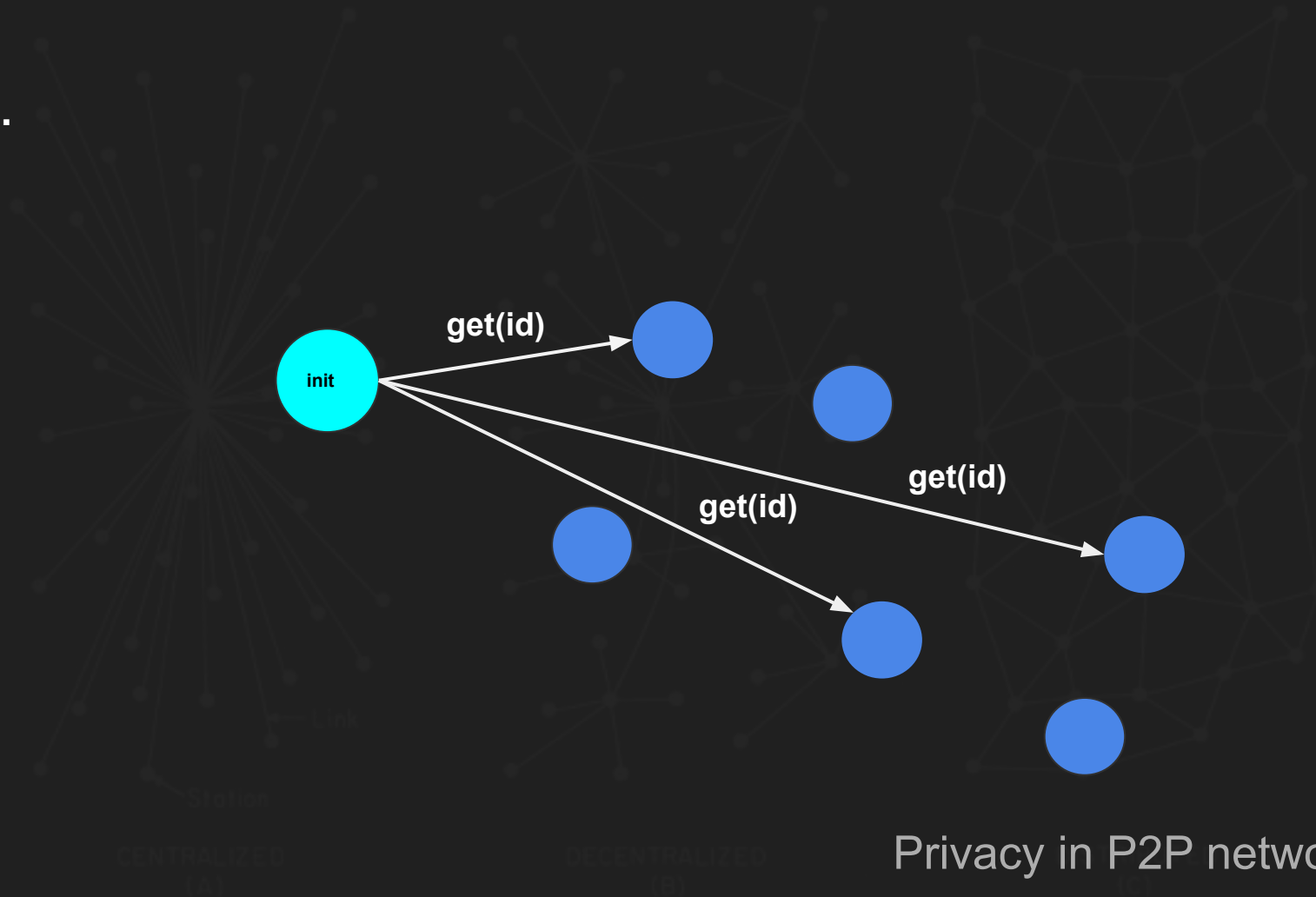
Octopus DHT lookup

CENTRALIZED
(A)

DECENTRALIZED
(B)

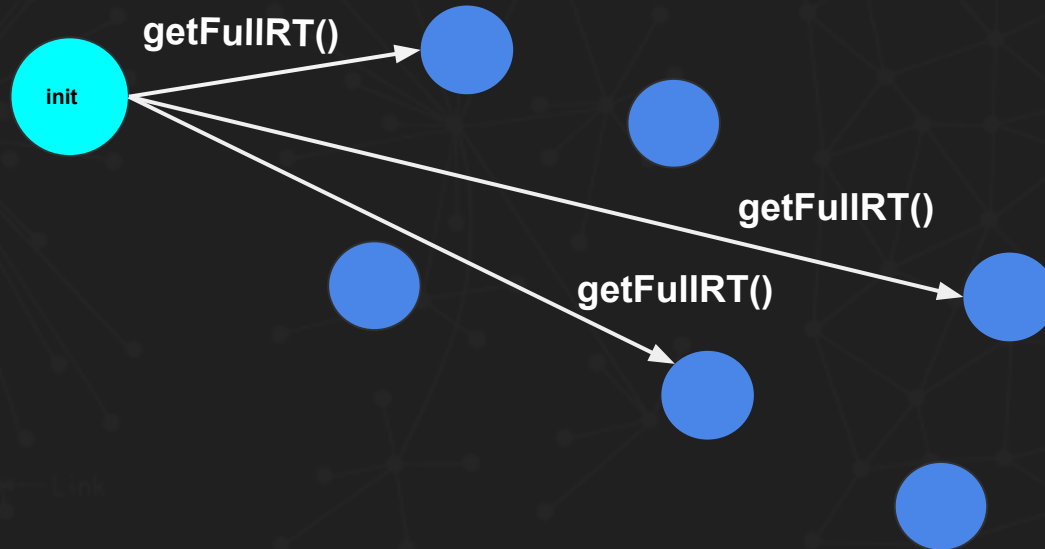
Privacy in P2P networks
(C)

What if..



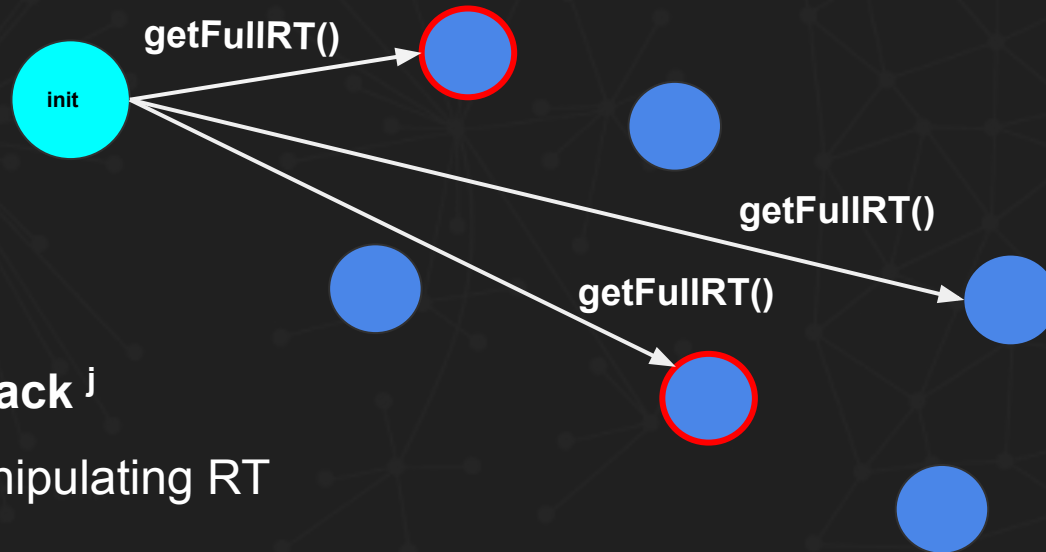
What if..

For target anonymity, peers request full RT table



Privacy in P2P networks

What if..



Range estimation attack ^j

Active attacks by manipulating RT

Privacy in P2P networks

Octopus DHT Lookup ^j

Initiator requests **full routing table**

Multiple disjoint lookup paths

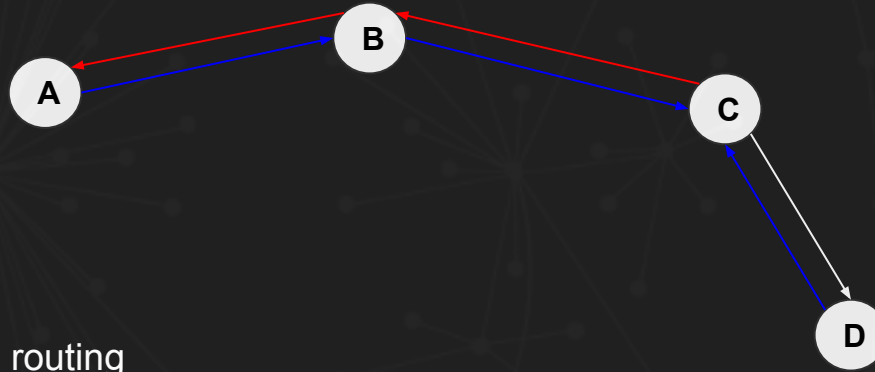
Dummy queries (*noise*)

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"

Privacy in P2P networks

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

Privacy in P2P networks



breakpoint >

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

Mechanisms and protocols to mitigate privacy vulnerabilities

CENTRALIZED
(A)

DECENTRALIZED
(B)

Privacy in P2P networks



breakpoint >

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

Mechanisms and protocols to mitigate privacy vulnerabilities

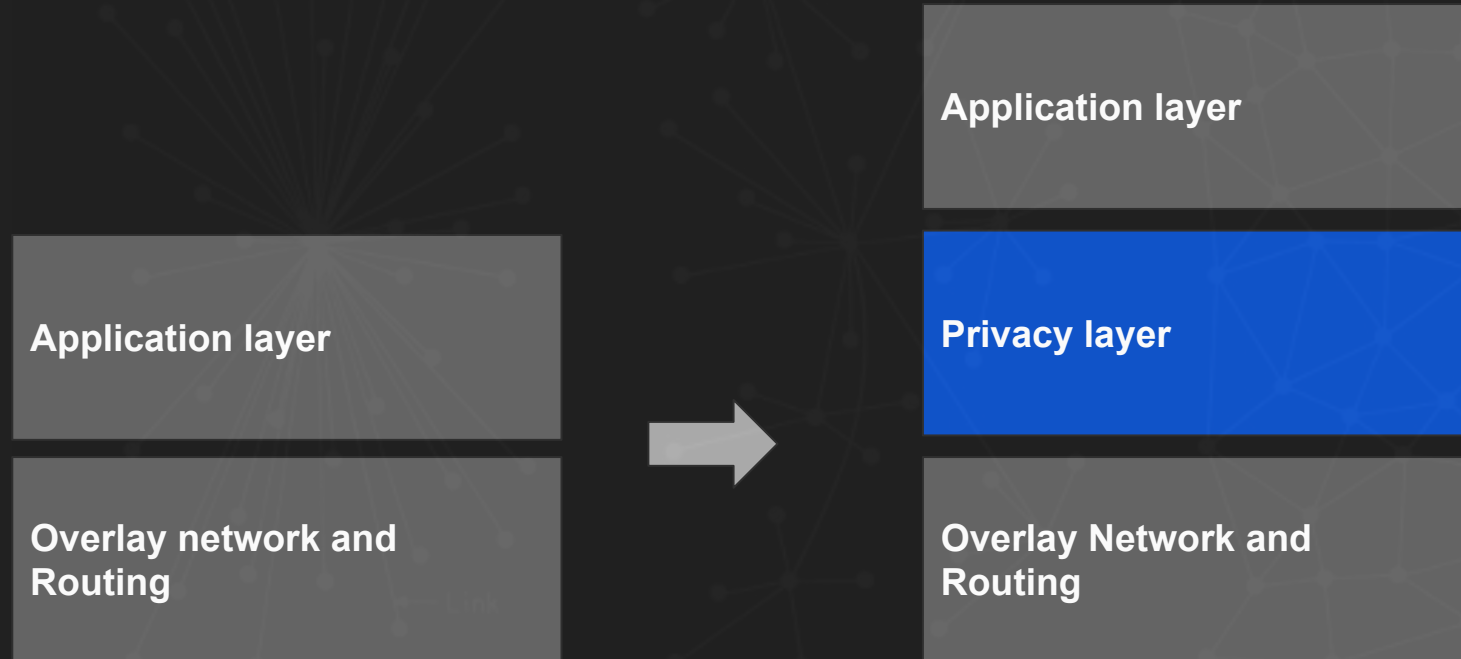
Privacy Engineering → putting all together in practice

CENTRALIZED
(A)

DECENTRALIZED
(B)

Privacy in P2P networks

Privacy engineering for P2P networks

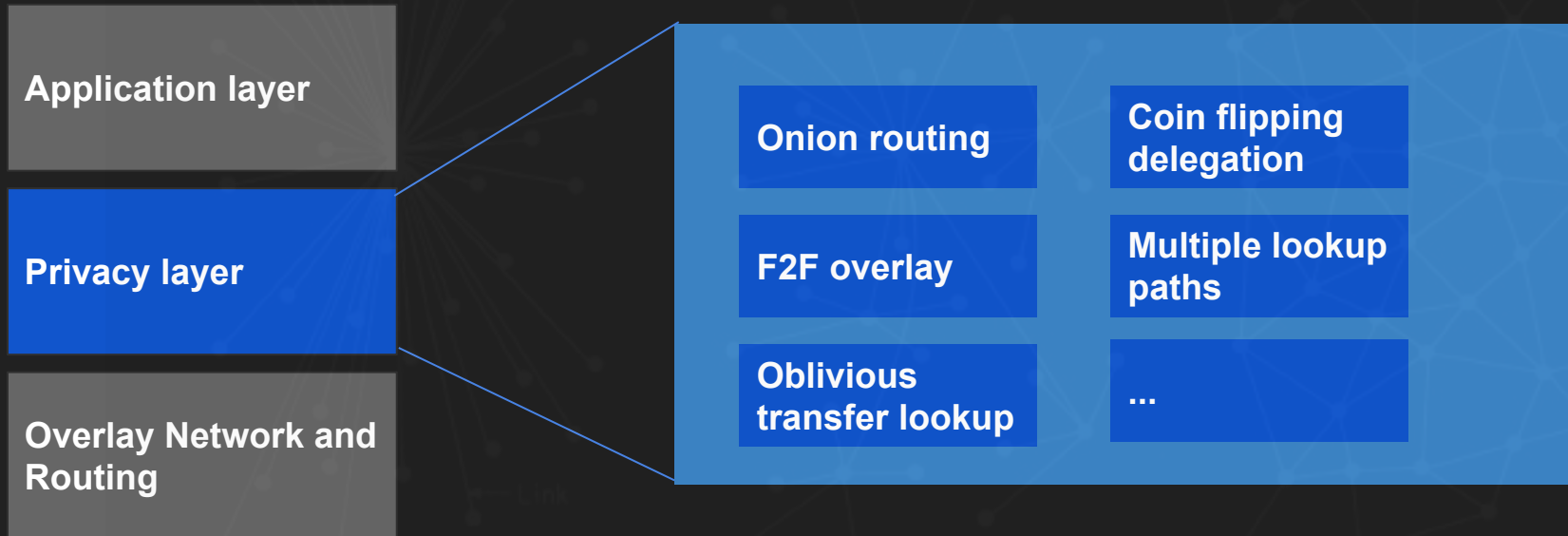


CENTRALIZED
(A)

DECENTRALIZED
(B)

Privacy in P2P networks

Privacy engineering for P2P networks



Privacy in 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..?

Application layer


Privacy layer


Overlay Network and
Routing


p3lib  libp2p


Privacy in P2P networks


Call for action!!


 **hashmatter** / **p3lib**


 Unwatch ▾ 3


 Unstar 12


 Fork 1


 Code


 Issues 12


 Pull requests 0

 Projects 1


 Wiki


 Insights


 Settings


privacy preserving primitives and protocols (p3) for routing and messaging in P2P networks <https://hashmatter.com> 


p2p messaging anonymous **privacy-enhancing-technologies** [Manage topics](#)


 50 commits


 1 branch


 0 releases


 1 contributor


 MIT



Branch: master ▾  New pull request

 Create new file





 Upload files

 Find File

 Clone or download ▾

 **gpestana** Merge pull request [#28](#) from hashmatter/criop2p 

Latest commit [bc170f1](#) 9 days ago

 criop2p	adds criop2p and octopusdht projects	9 days ago
 octopusdht	adds criop2p and octopusdht projects	9 days ago
 specs	release: prepare v 0.1	2 months ago
 sphinx	adds common interfaces o3lib	9 days ago

Privacy in P2P networks

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

...

Privacy in P2P networks

How does the future we're building look like?

data **ownership**

open, inclusive and collaborative protocols

no centralized and external stewardship

respects **privacy**

CENTRALIZED
(A)

DECENTRALIZED
(B)

Privacy in P2P networks



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)

