PRIVACY IN P2P NETWORKS: DHTs AND IPFS

Gonçalo Pestana
@gpestana

IPFS Camp

Privacy in P2P networks

IPFS and Distributed Hash Tables

hashmatter

Distributed Hash Tables

Collaborative, P2P overlay network

Decentralized **key-value** storage

Content based hashing

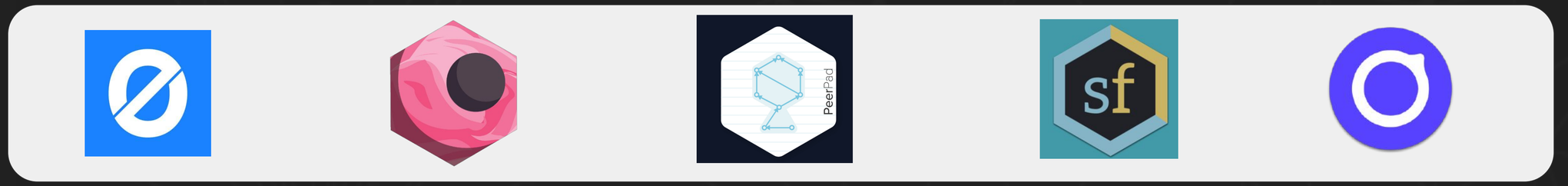
IPFS enables peers to store, discover and replicate resources on IPFS


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

Privacy in P2P networks

Distributed Hash Tables

Application layer

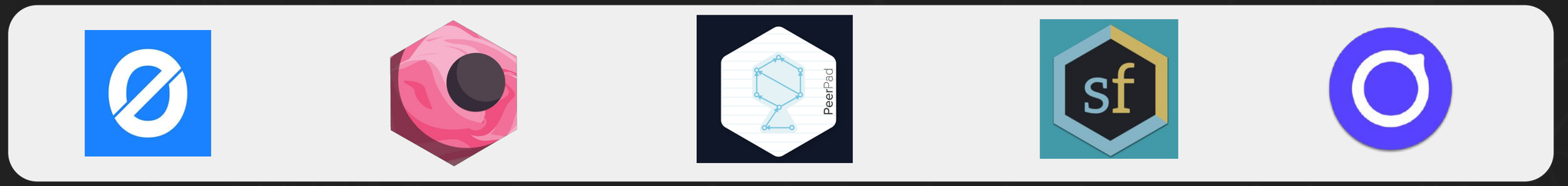


How about privacy?

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

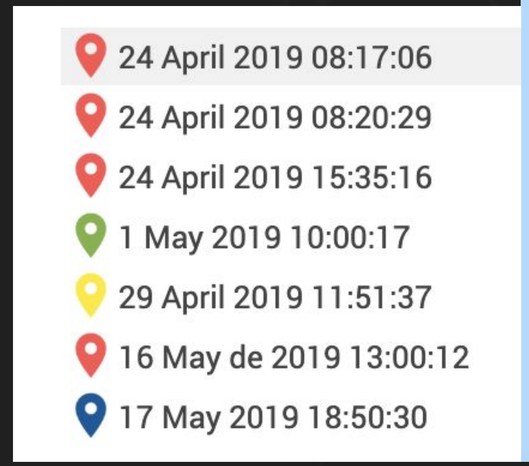
Privacy in P2P networks

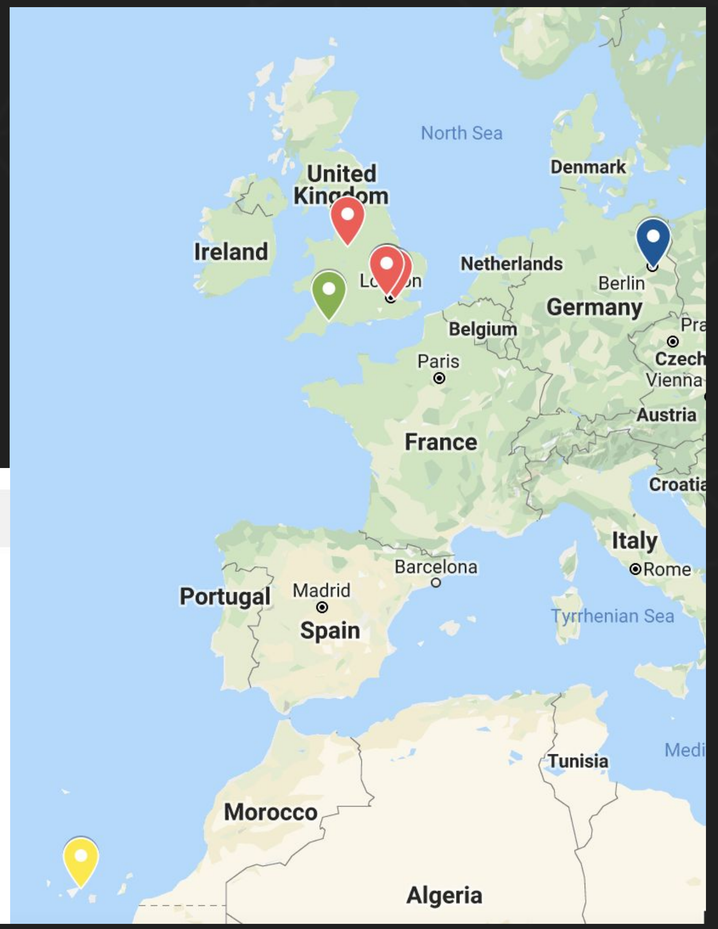
Attack: **Content provider tracking**

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)
```





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

Centralized services

One entity

(naïve) Decentralized networks

Every participant in the network

Your neighbor
Ads companies
Governments
...

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

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

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

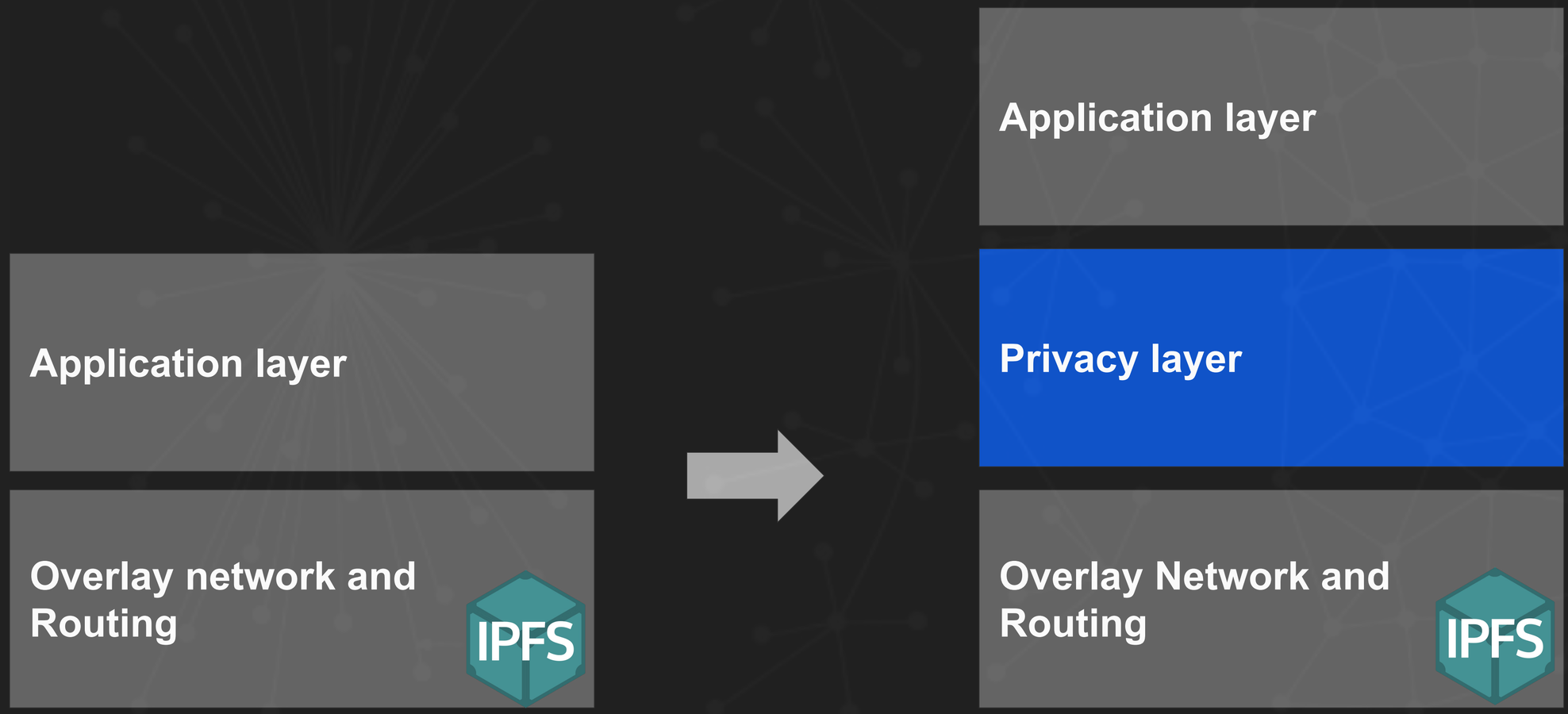
Low latency

Decentralized

Scalability

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

Privacy engineering for P2P networks



Delegated encrypted requests

Plausible deniability through noise

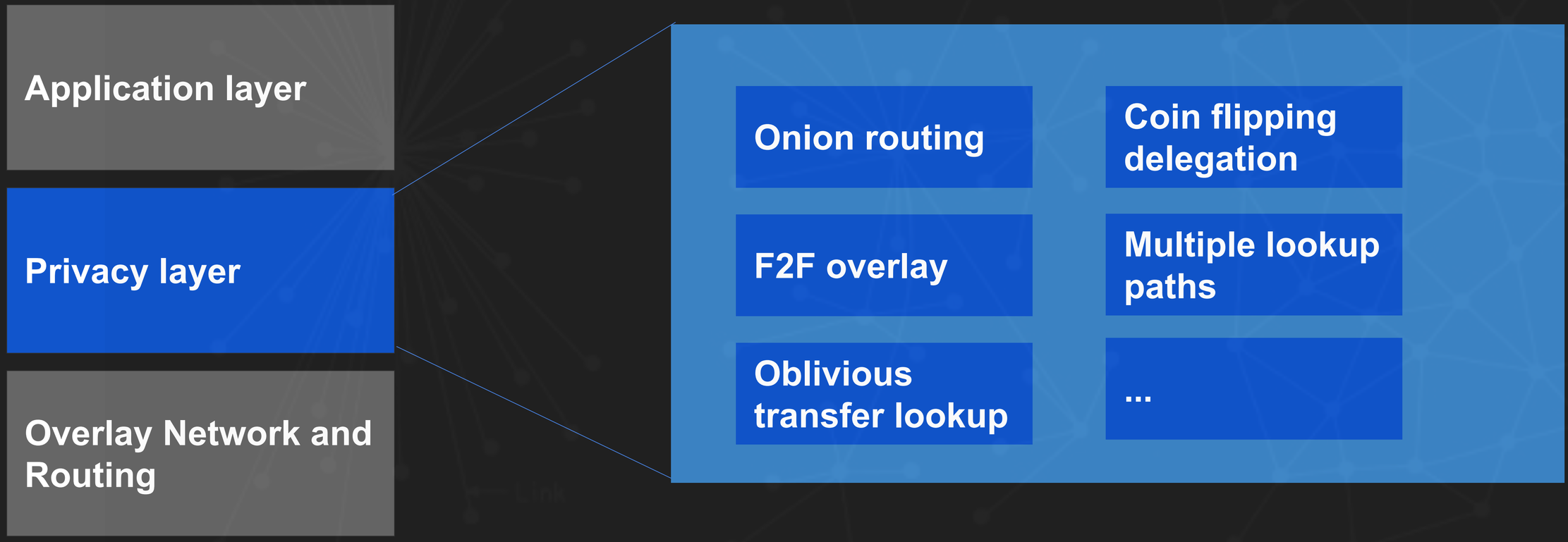
PIR and Oblivious Transfer

Octopus DHT lookup

	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

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

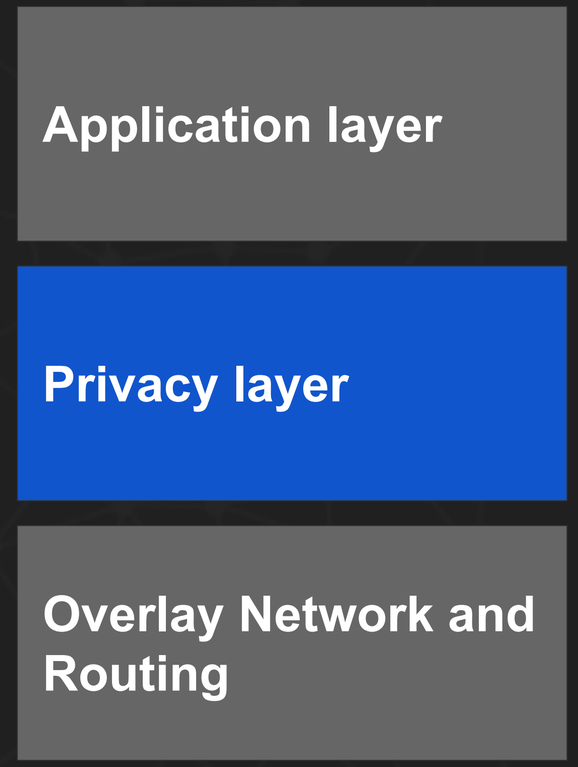
p3lib-sphinx all purpose onion routing implementation

p3lib-cfdr plausible deniability for DHT lookups

p3lib-octopusdht multipath lookup mechanism with noise for DHT

more..?

p3lib ❤ **libp2p**



Delegated encrypted requests (DEMO if time/later on)

Plausible deniability through noise

PIR and Oblivious Transfer

Octopus DHT lookup

Privacy engineering for P2P networks

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

https://hashmatter.com

Roadmap Q3 Q4

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

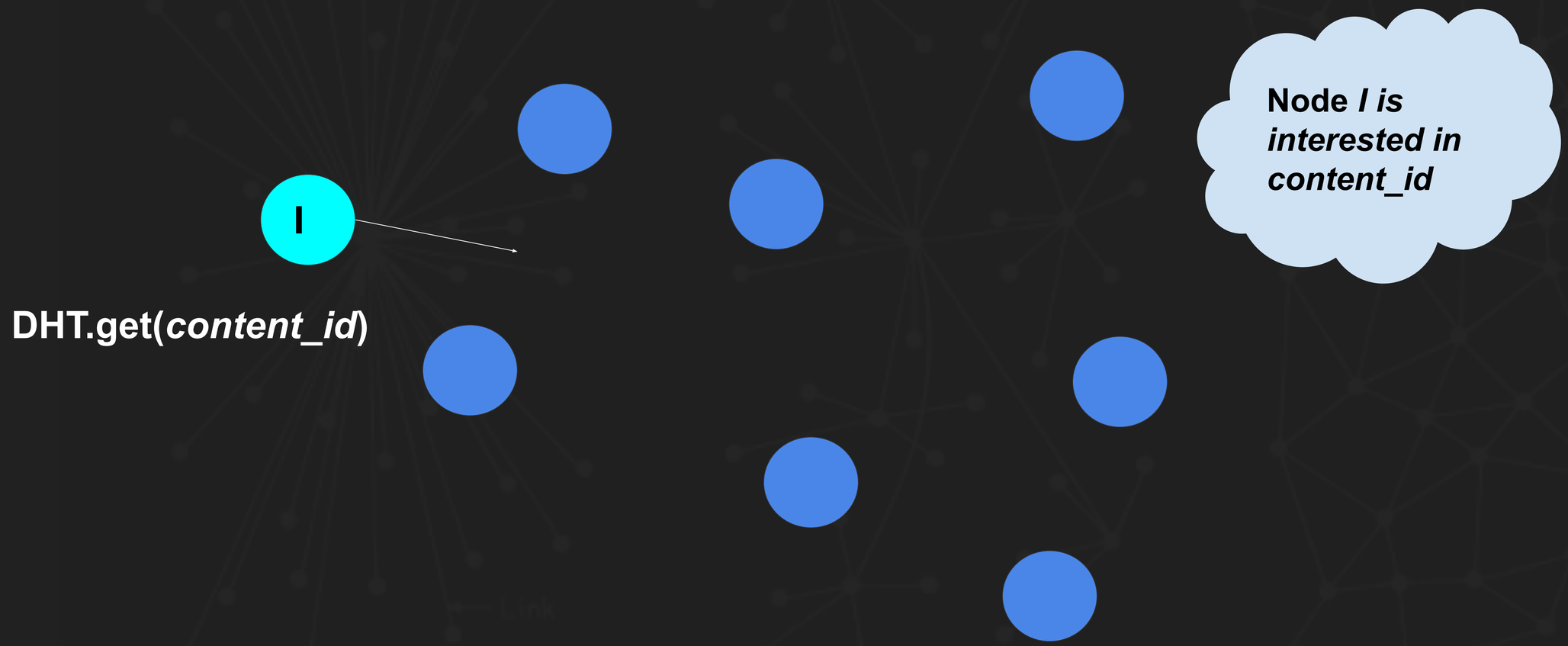
...

Demo! (bonus if time :D)

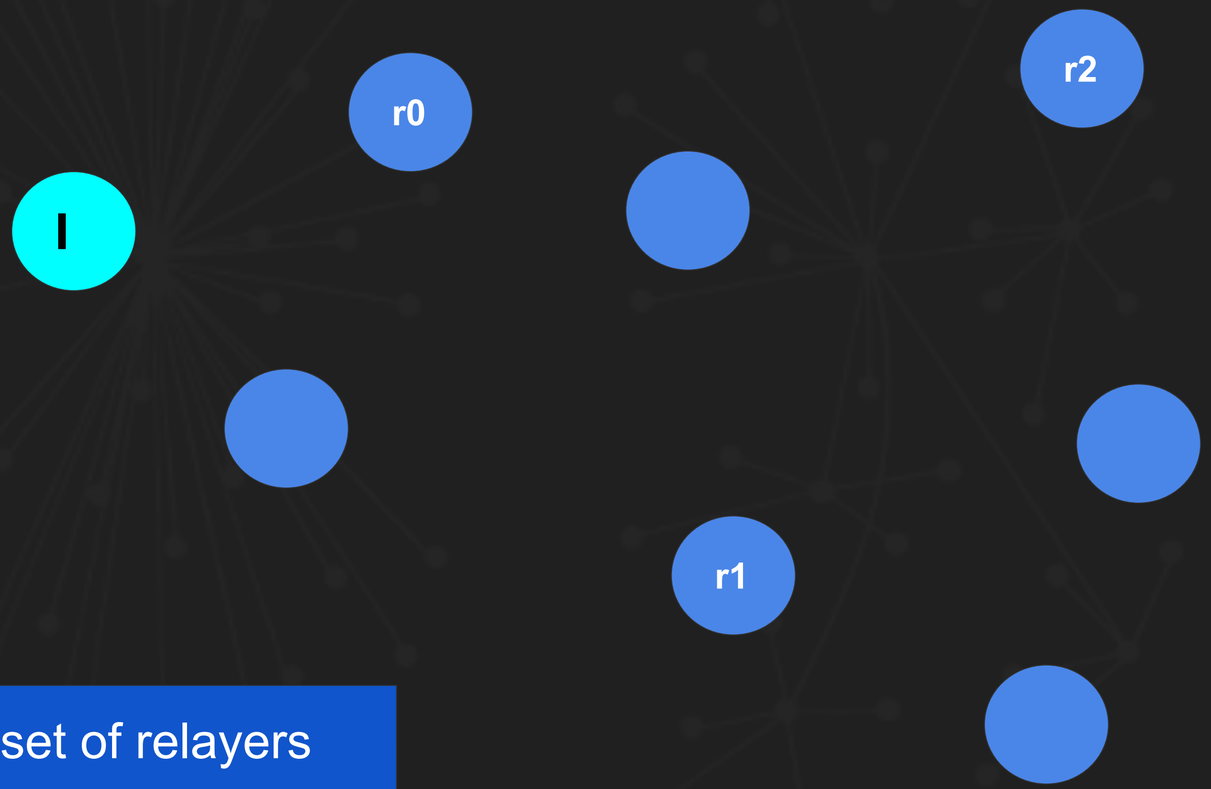
Short demo of **p3lib-sphinx** working with **libp2p**

https://youtu.be/j64C5CTb8J8

Delegated encrypted requests



Delegated encrypted requests

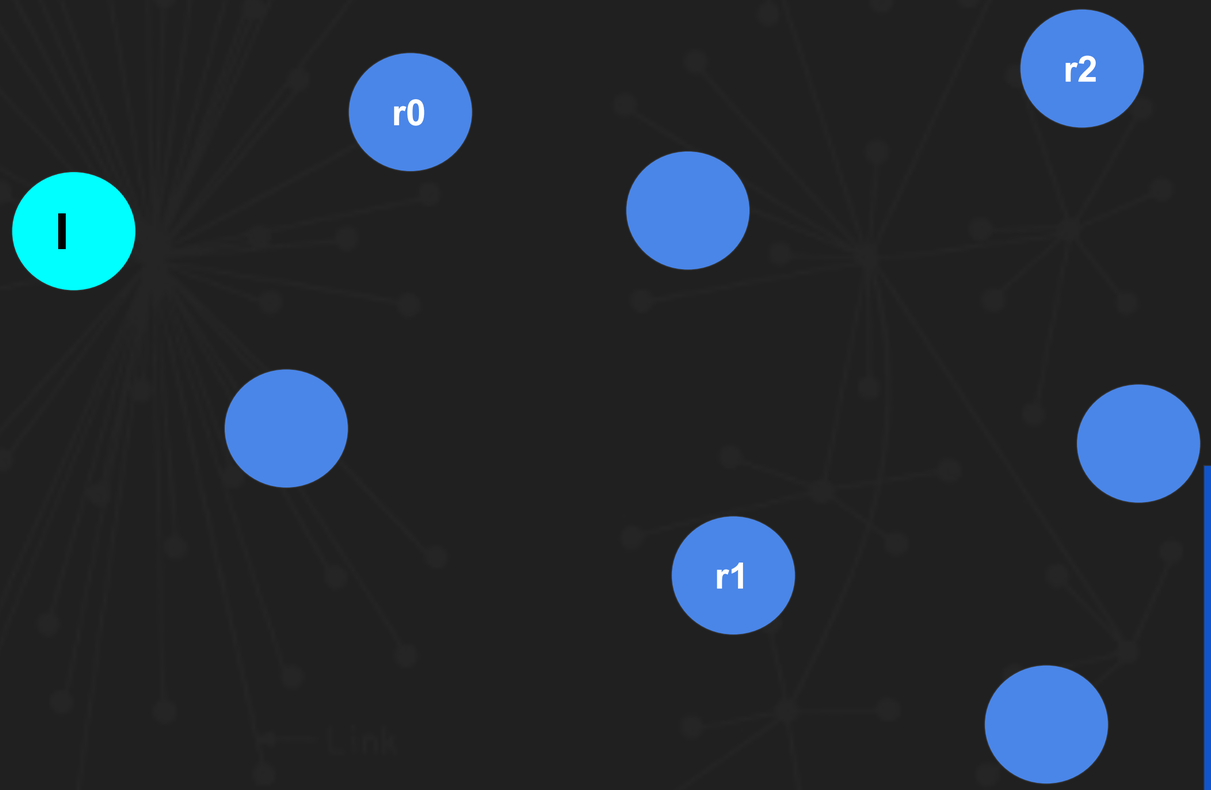


Relayer table
r0: [pubkey, addr]
r1: [pubkey, addr]
r2: [pubkey, addr]
...

Initiator selects a set of relayers

(r0, r1, r2)

Delegated encrypted requests

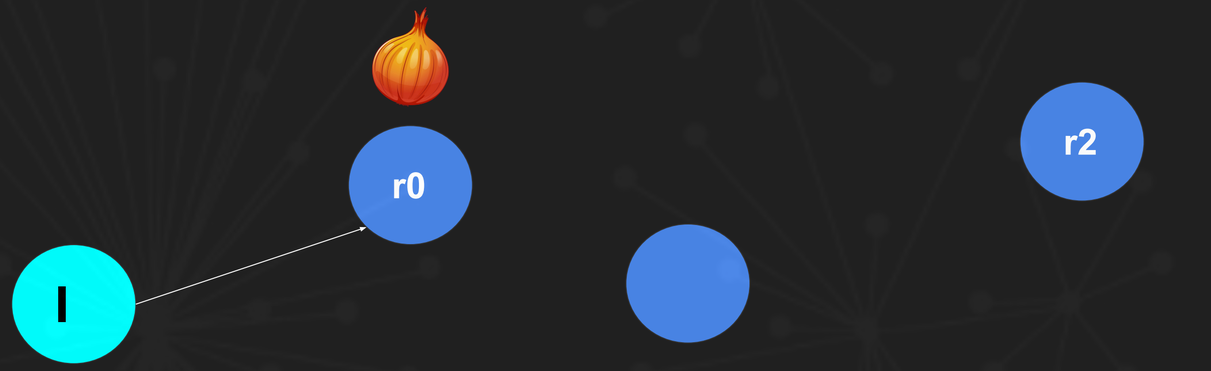


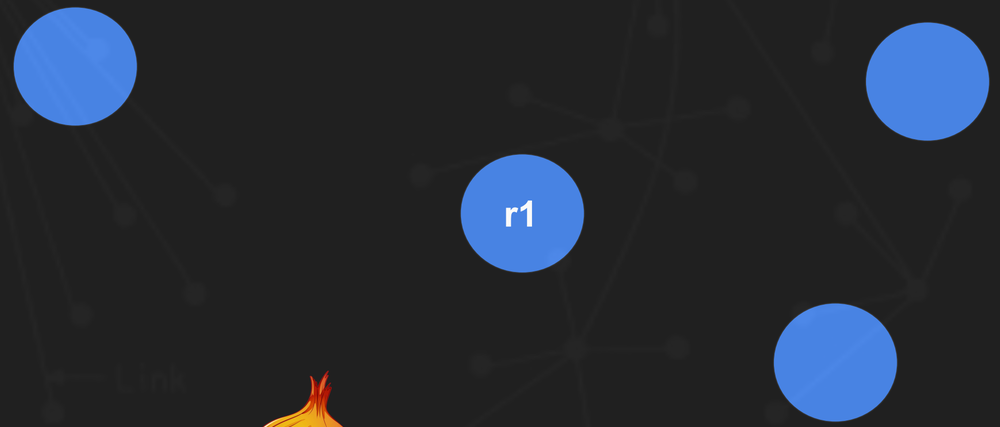
Relayer table
r0: [pubkey, addr]
r1: [pubkey, addr]
r2: [pubkey, addr]
...

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

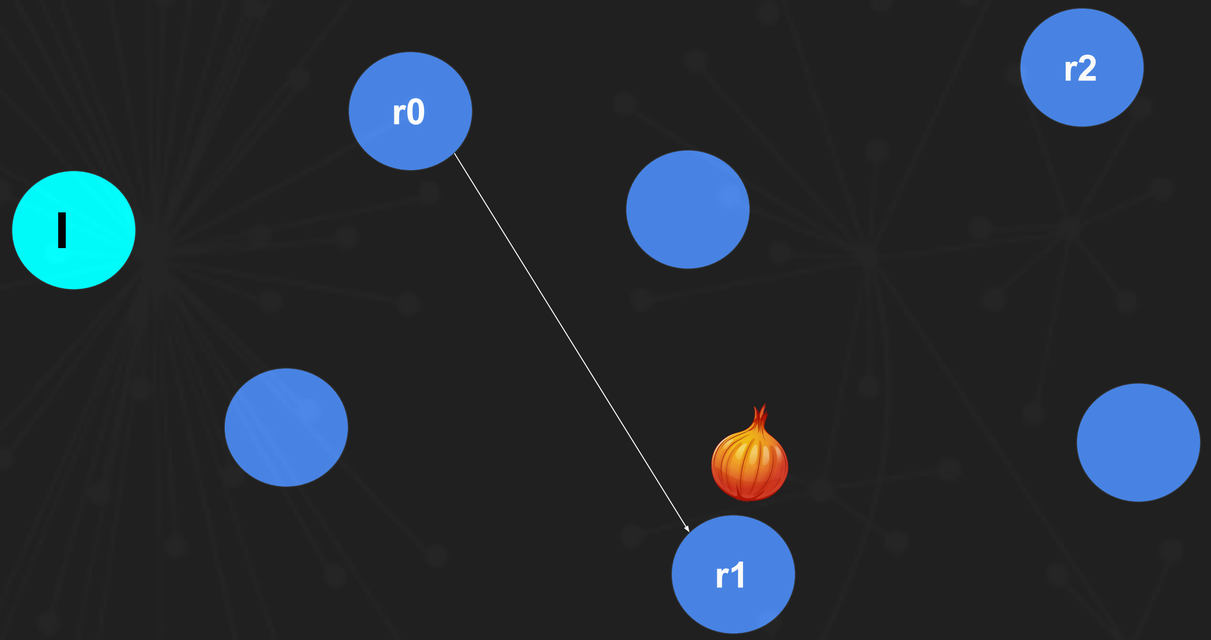




Relayer table
r0: [pubkey, addr]
r1: [pubkey, addr]
r2: [pubkey, addr]
...

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

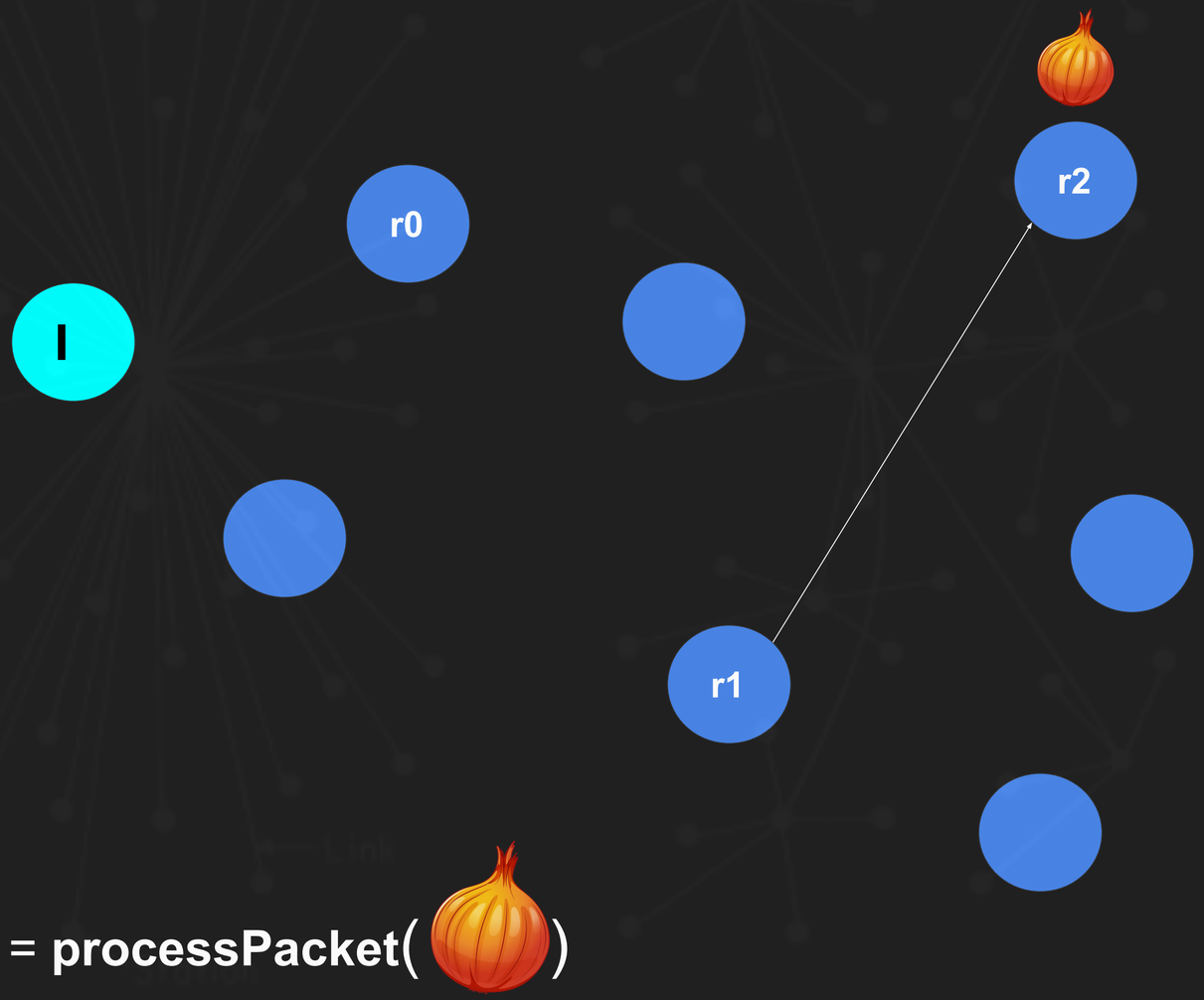
Delegated requests with cryptography



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

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

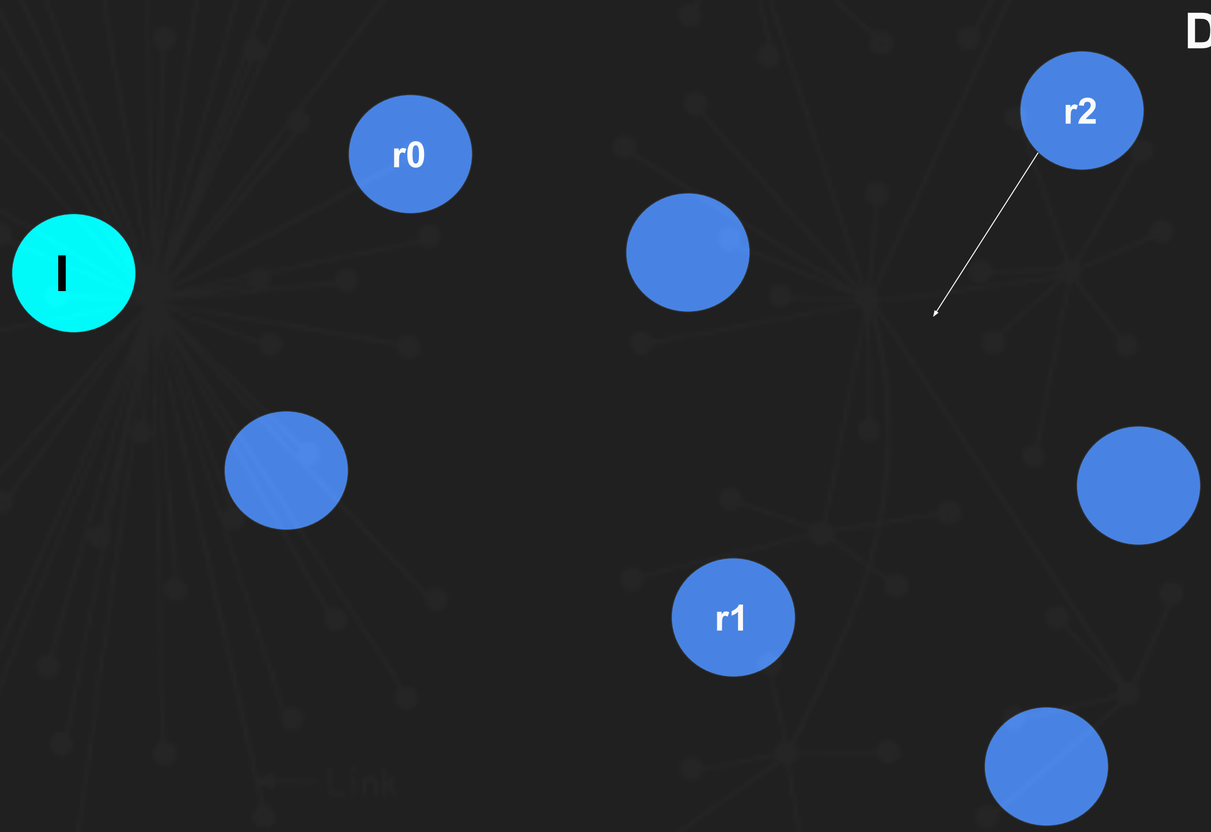
Delegated requests with cryptography



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

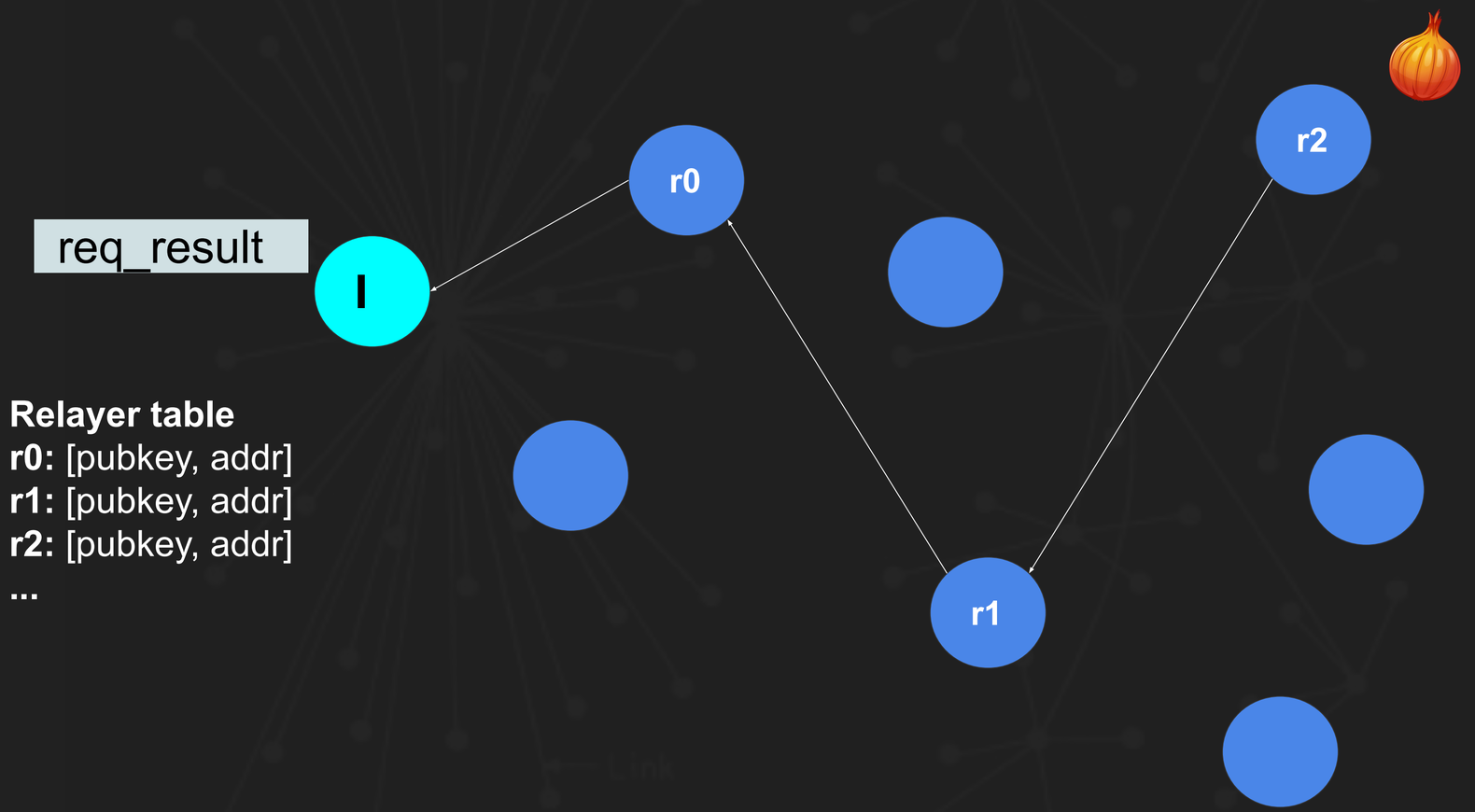
requestPayload = processPacket( )

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]
...