Making use of Double Hashing for Privacy

Guillaume Michel
@guissou

ProbeLab,
Protocol Labs

IPFS þing
14th July 2022

Work from the libp2p privacy discussion group

- Yiannis Psaras - Protocol Labs
- Will Scott - Protocol Labs
- Srivatsan Sridhar - Stanford University
- Guillaume Michel - Protocol Labs
- Florian Tschorsch - TU Berlin
- Erik Daniel - TU Berlin
- Elizabeth Binks - Chainsafe

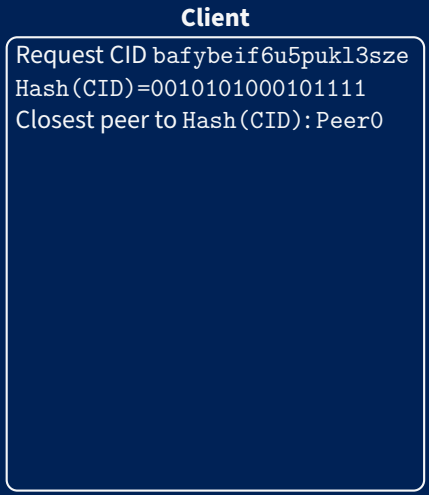
Double Hashing in IPFS

Hash() $\rightarrow$ 00100111011001001110001 10

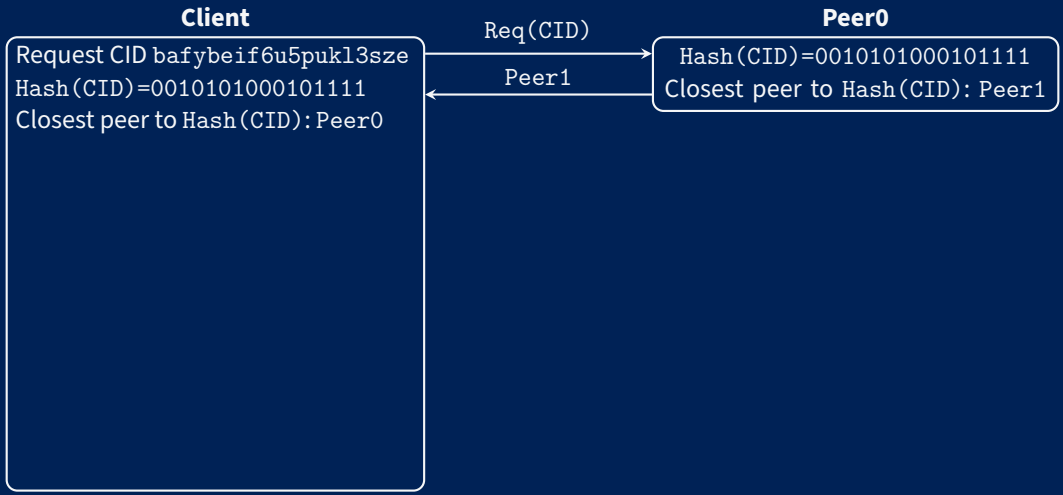
CID $\rightarrow$ bafybeiaqr6csdcnxrkpx23oithpjt

Hash(CID) $\rightarrow$ 111010000011001011110101

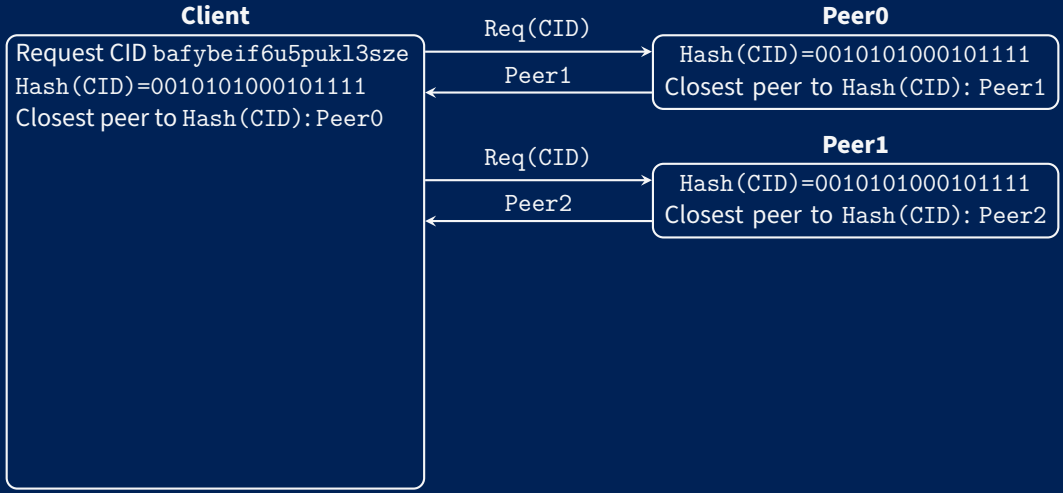
IPFS content lookup (simplified)



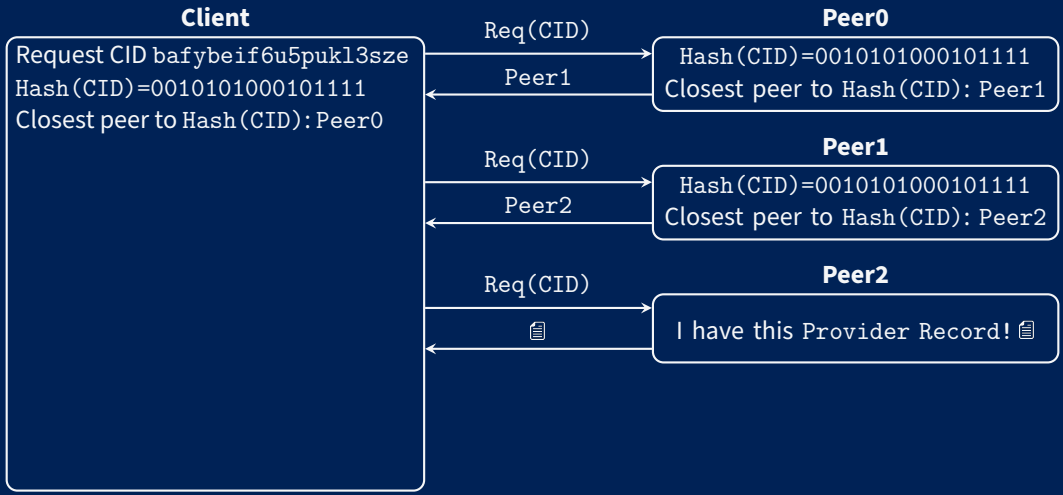
IPFS content lookup (simplified)



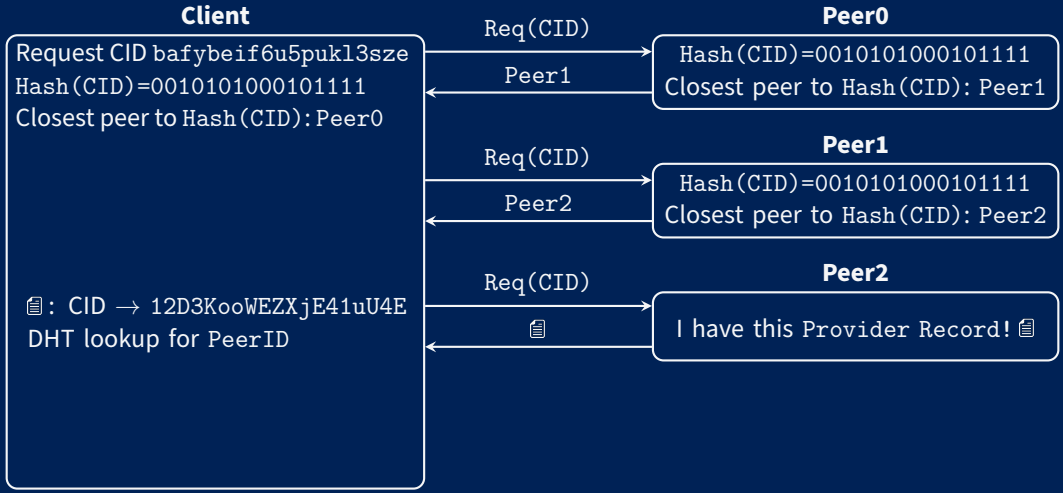
IPFS content lookup (simplified)



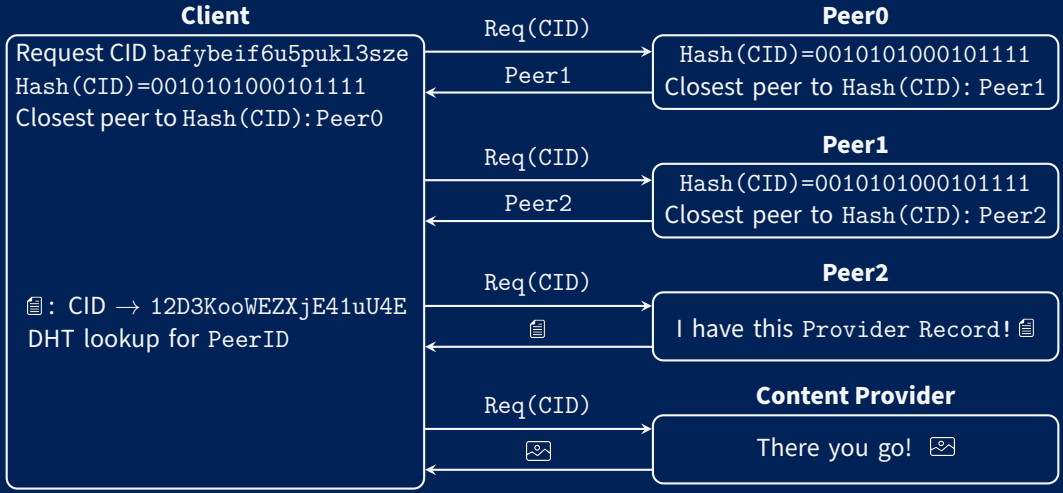
IPFS content lookup (simplified)



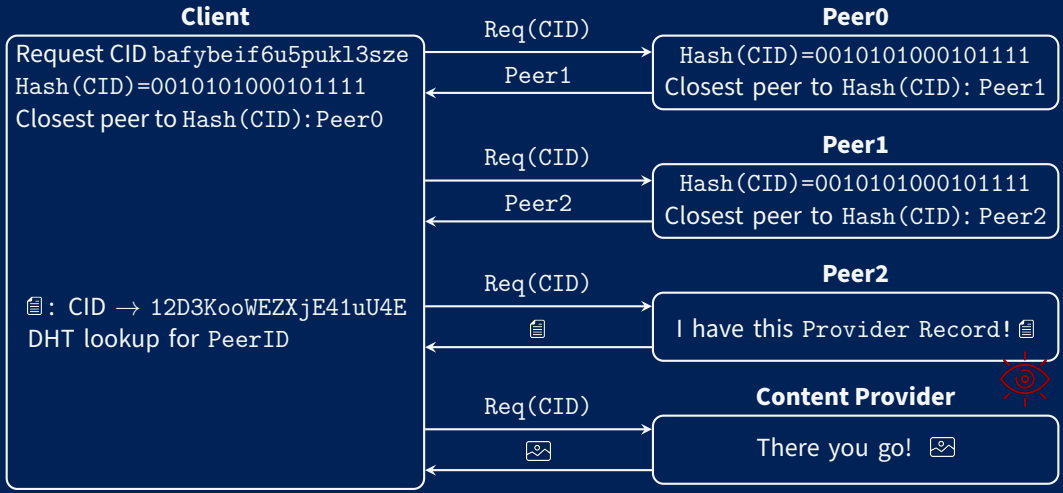
IPFS content lookup (simplified)



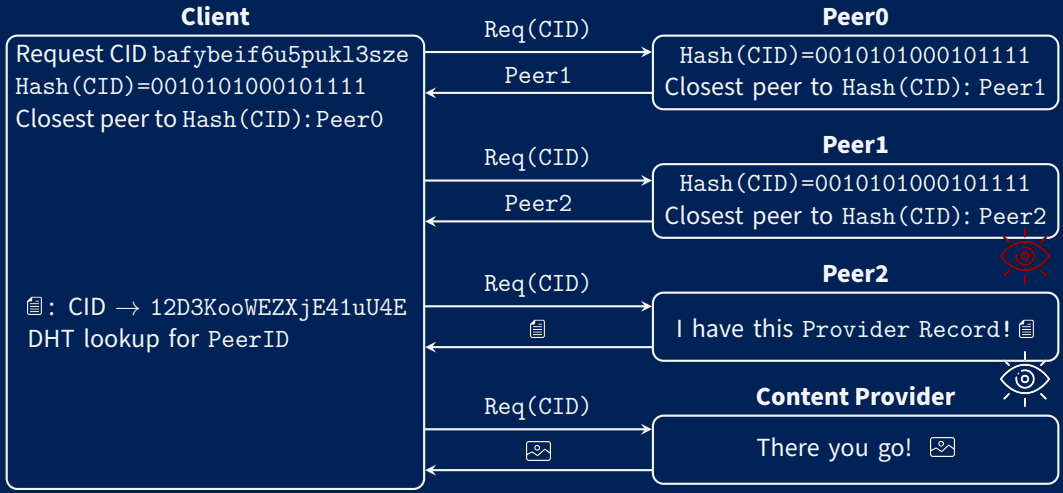
IPFS content lookup (simplified)



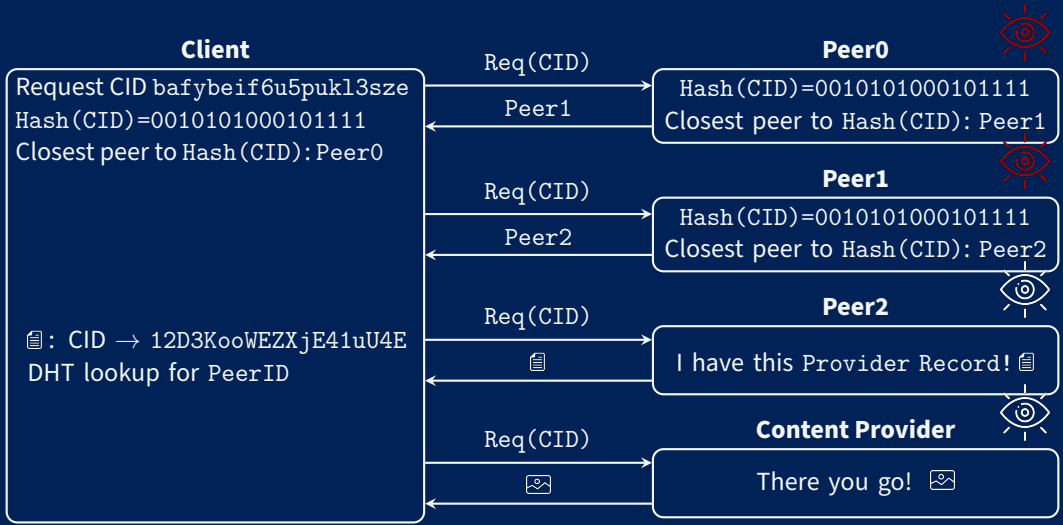
Who can see my requests?



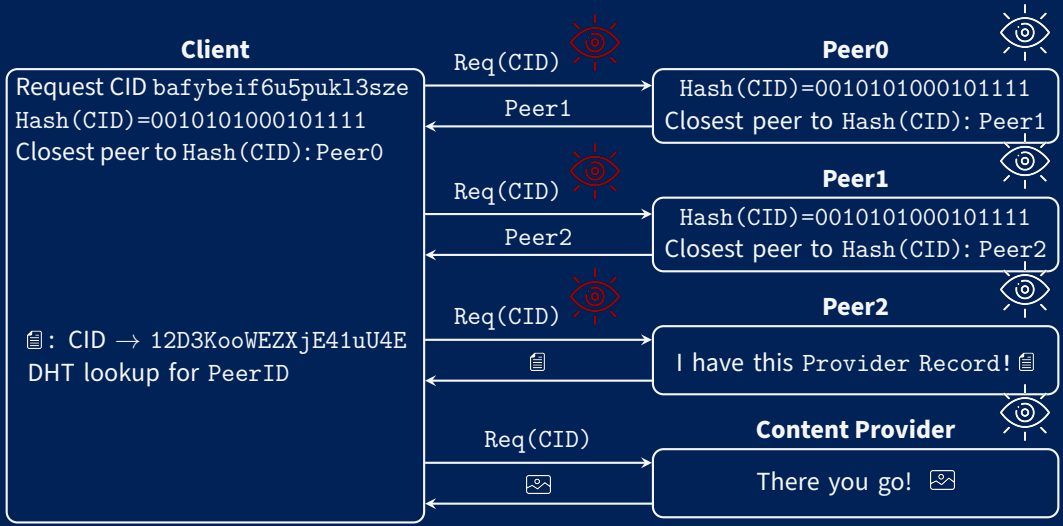
Who can see my requests?



Who can see my requests?



Who can see my requests?



Problem definition

We want to improve *Client Privacy* in the DHT. We want to hide Content Requests from the DHT nodes and passive observers.

We do not address:

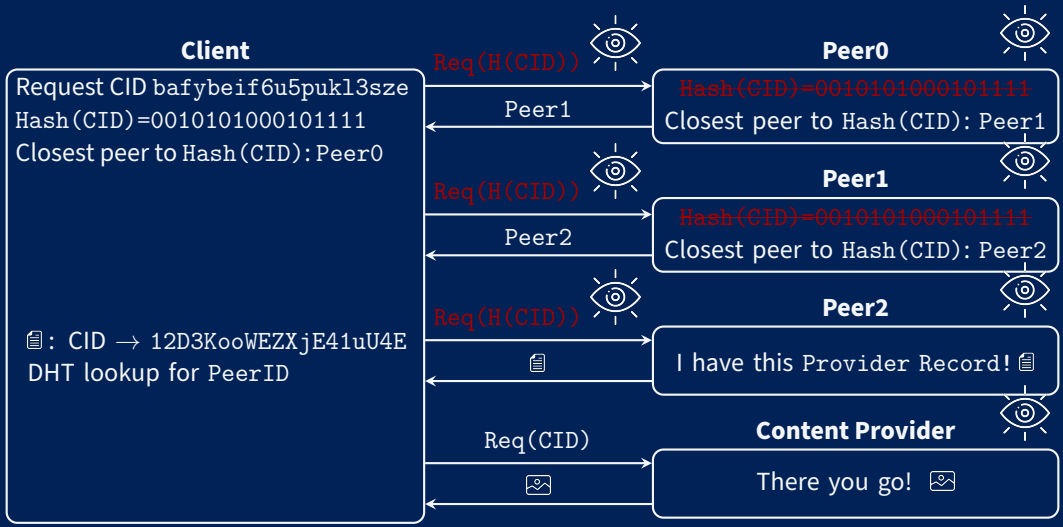
- Content Provider Privacy
- Bitswap Privacy
- Client Privacy from the Content Provider
- Client Privacy from the Gateways

Prefix lookup

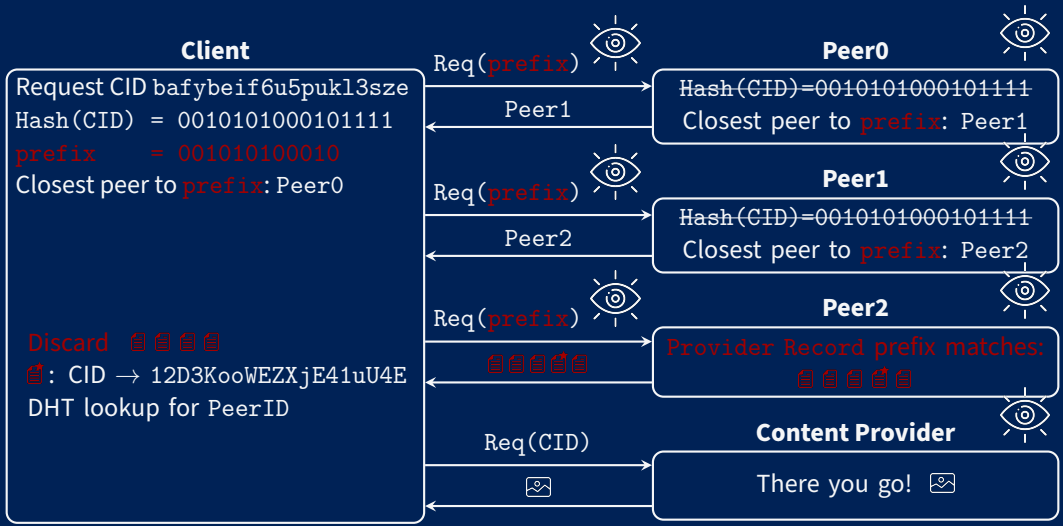
We want to request a prefix of the content to hide the exact CID. But:

- `Hash(bafybeif6u5pukl3szе)` $\neq$ `Hash(bafybeif6u5puk)`
- Request cannot be routed in the DHT $\rightarrow$ the content cannot be accessed
- We want to request a prefix of `Hash(bafybeif6u5pukl3szе)`
- DHT Routing process has to be adapted

First change: Request Hash(CID)

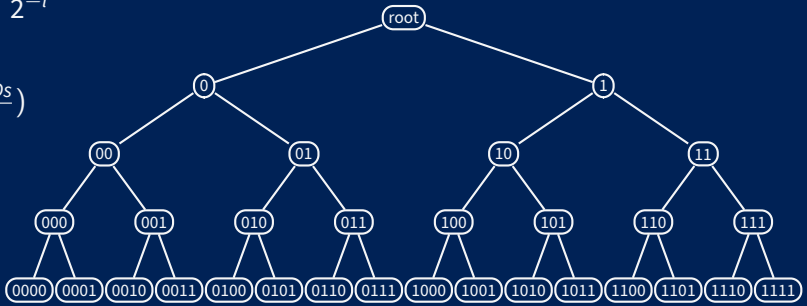


Second change: Request a prefix of Hash(CID)



Prefix Length Selection

- Prefix Length: l
- `k-anonymity`: The requested Provider Record can not be distinguished from at least $k - 1$ other Provider Records
- $k = \#CIDs \times 2^{-l}$
- $\frac{\#CIDs}{k} = 2^l$
- $l = log_2(\frac{\#CIDs}{k})$



Privacy gains

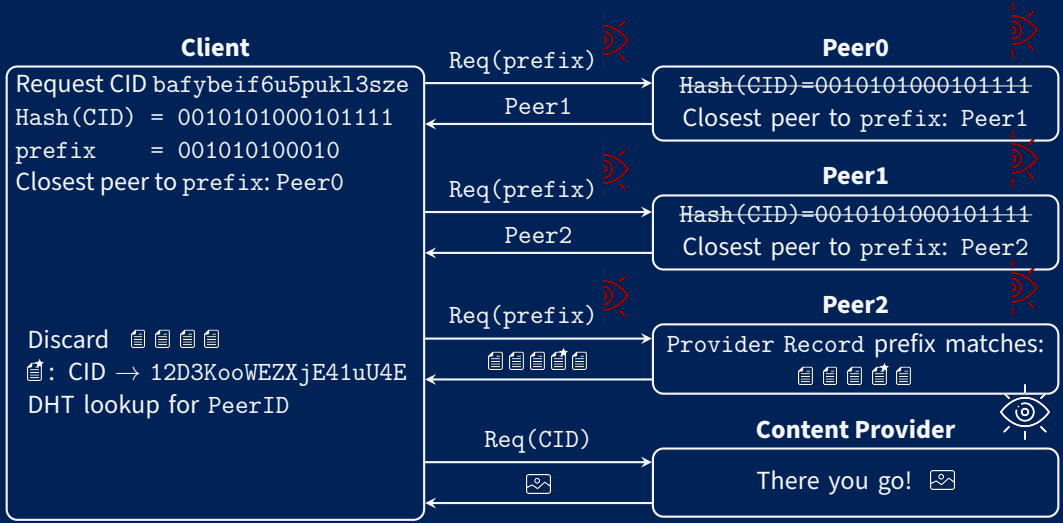
`k-anonymity` and plausible deniability from:

- DHT routing nodes
- DHT node storing the Provider Record
- Passive observers

But:

- No `l-diversity` nor `t-closeness`
- Network overhead: transmit *k* Provider Records instead of 1
- It is easy for observers to replay the same prefix request, and resolve all Provider Records matching this prefix

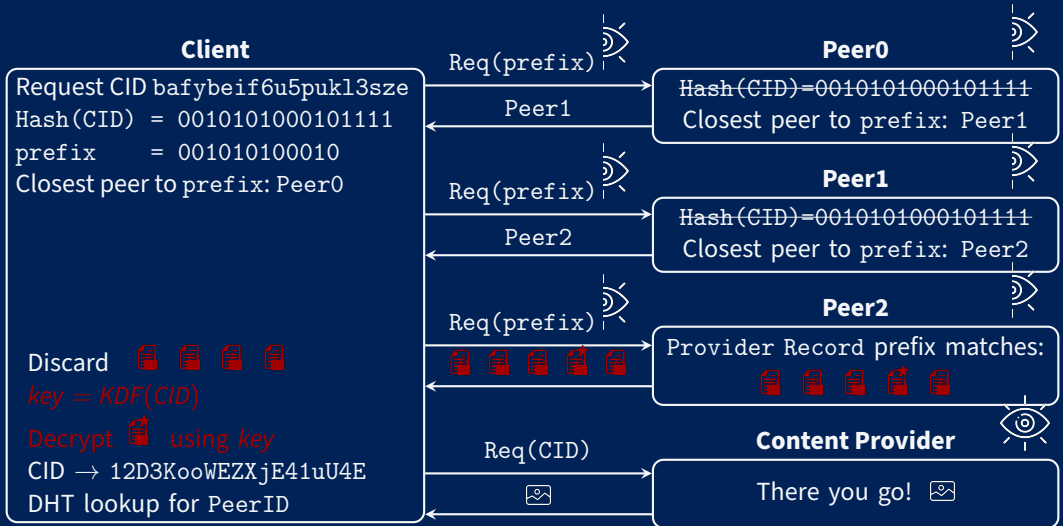
Privacy gains



Provider Records Encryption

- Provider Records encrypted before pinning
- Symmetric encrypted e.g `AES-256`
- $key = KDF(CID)$
- Having access to `Hash(CID)` is not enough to read the Provider Record
- Only nodes knowing the CID can read the Provider Record

Provider Records Encryption

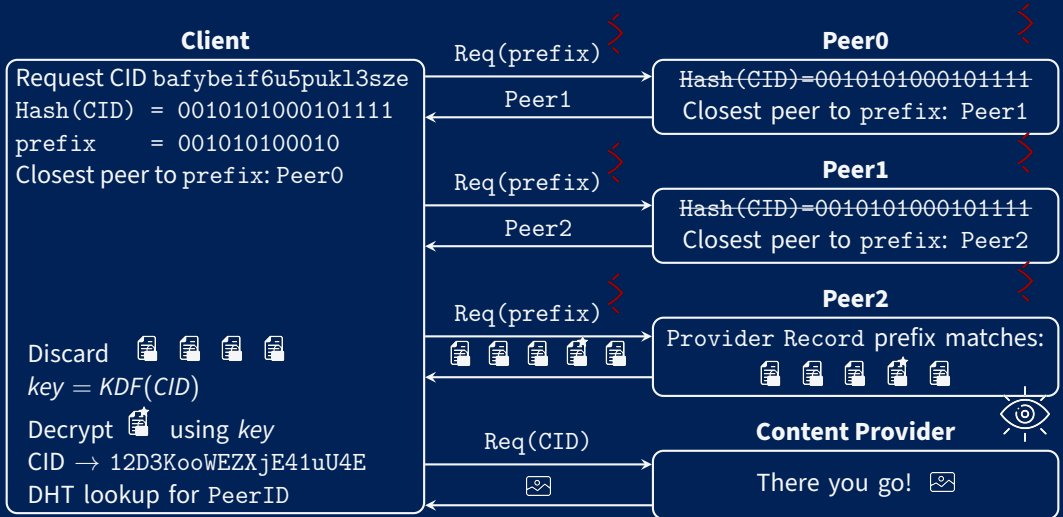


Privacy Gains

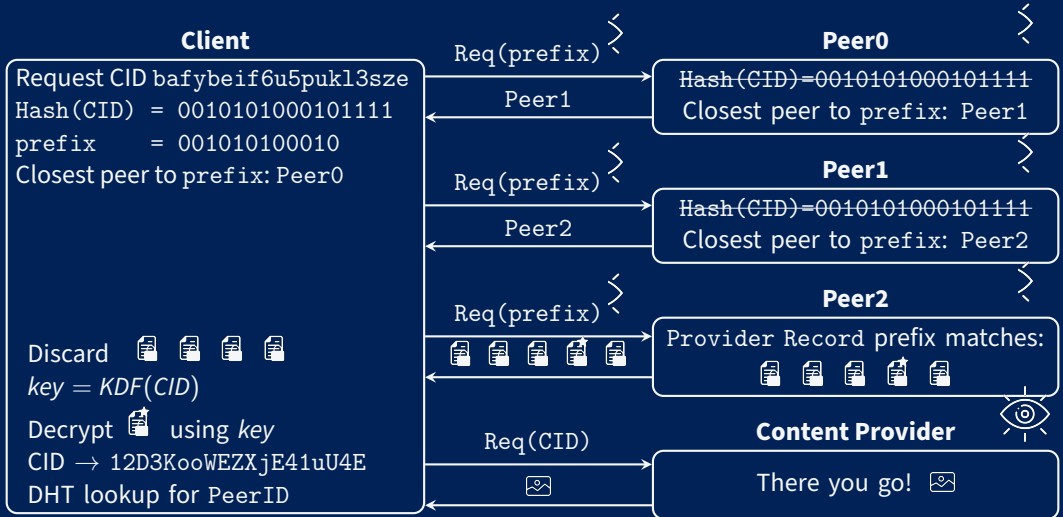
- Observers can only decrypt the Provider Records of which they know the CID
- DHT nodes storing the Provider Records don’t know the accessed Provider Record
- Bonus: The Content Provider gains privacy from the DHT nodes storing the Provider Record

- Downside: The client has to perform one decryption operation

Privacy Gains



Privacy Gains



Conclusion

- We can significantly improve privacy in the DHT!
- DHT servers don’t need to hash the CID for every request
- Network overhead: sending *k* Provider Records instead of 1
- Computation overhead: one symmetric decryption for the Provider Record
- Require to modify the server code and republish all Provider Records
- Illusion of privacy