Sybil Attack Strikes Again: Denying Content Access in IPFS with a Single Computer

Thibault Cholez, Claudia Ignat

Inria, Université de Lorraine, France
claudia.ignat@inria.fr

InterPlanetary File System (IPFS)

Largest decentralized peer-to-peer file system

- 20,000 peers active every day
- >20 million requests per day

Key storage platform for decentralized apps (dApps) and NFTs

Open participation

Research Question

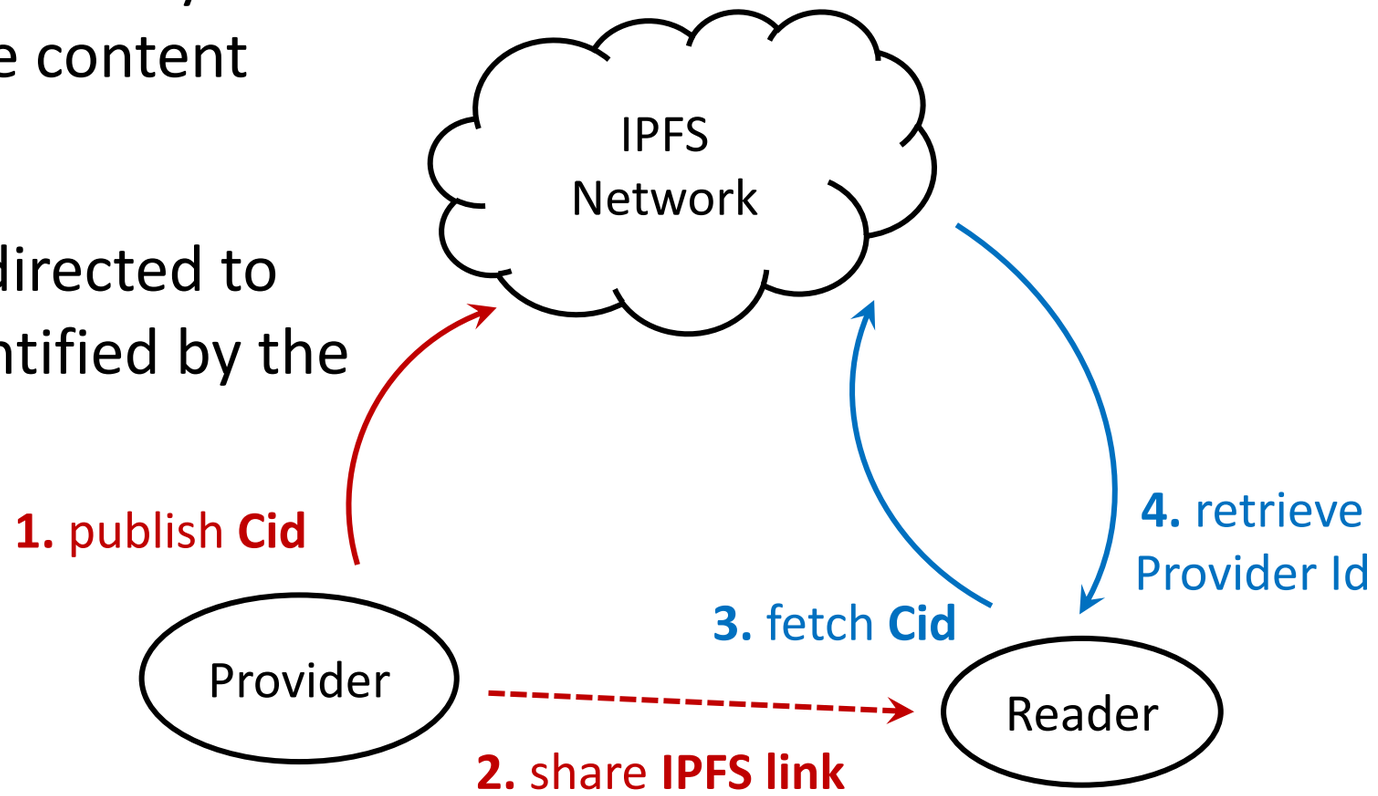
- IPFS (libP2P) relies on Kademlia Distributed Hash Table (DHT) for publishing/fetching a content
- But, DHTs are vulnerable to Sybil Attack that can lead to peer and/or content censorship
- Several effective mitigation strategies have been proposed and implemented over time

Is IPFS resilient to Sybil Attack?

Publishing/fetching content in IPFS

Providers publish a **Document** identified by a **Content Identifer (Cid)** based on the content hash and shared out of band

A reader interested in a **Cid** will be directed to the **Provider** that stores the file identified by the **Cid**



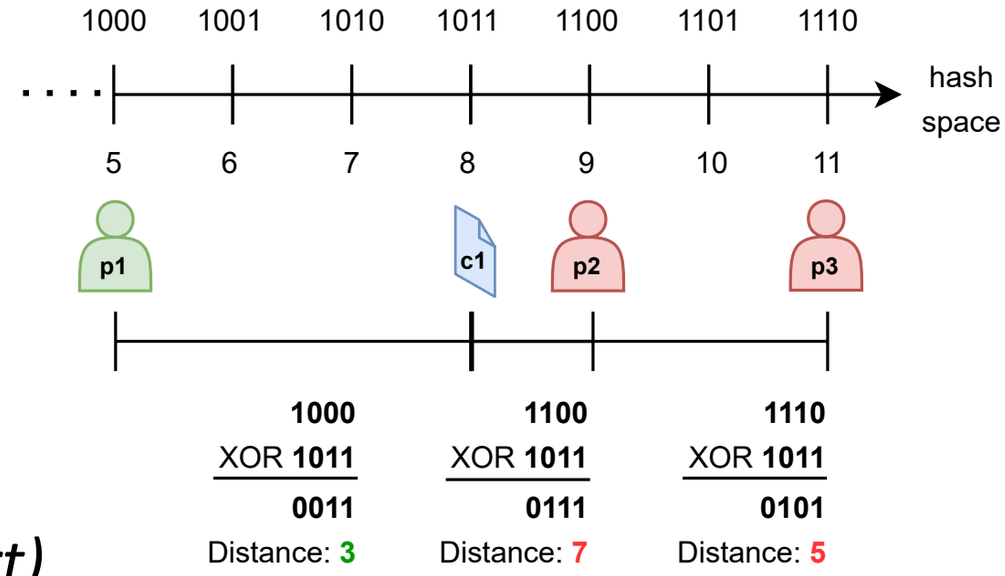
Peers and content discovery

Peers identified by a **PeerID** (hash of the public key)

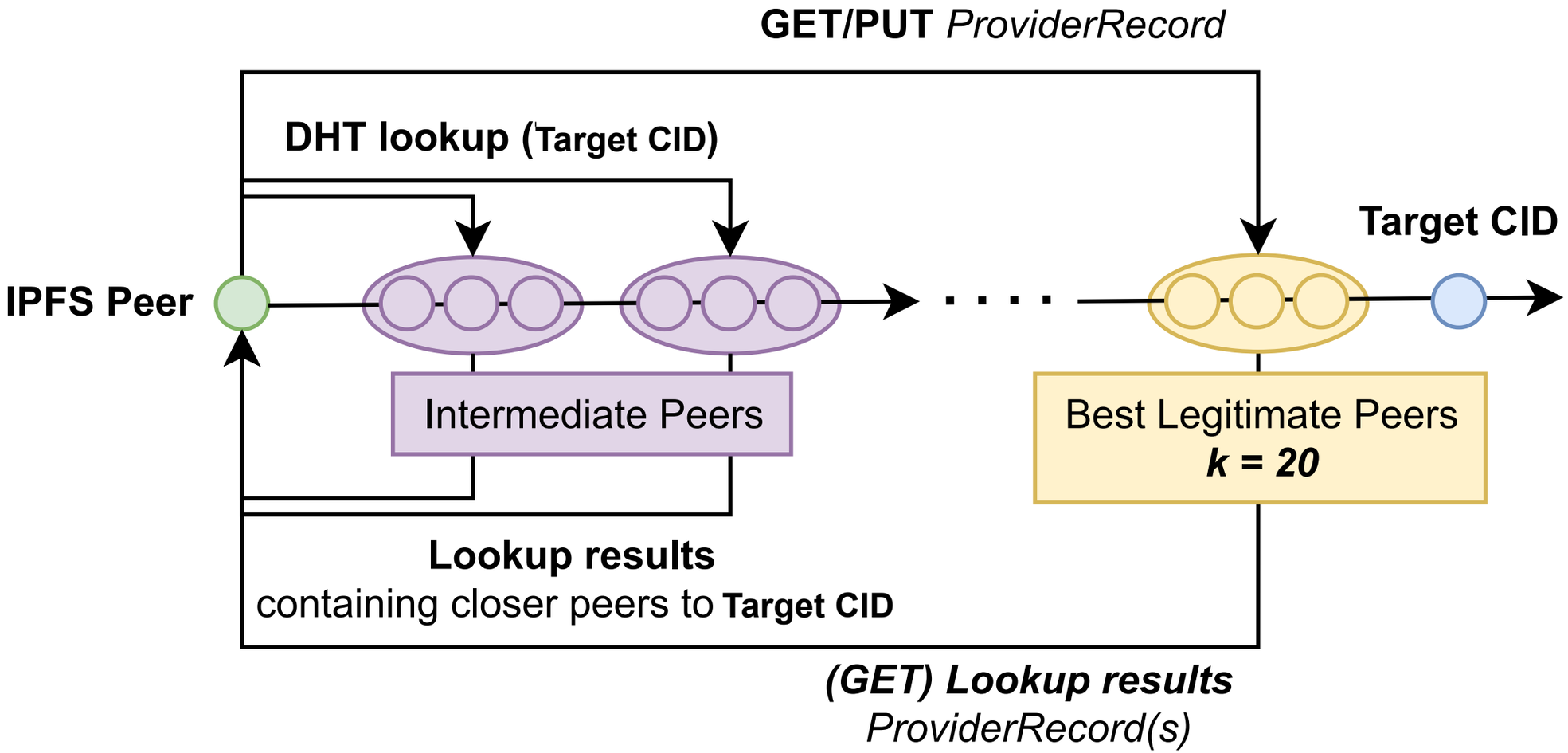
Distance between identifiers computed by XOR

Records published on the DHT

- **Provider Record**: (**PeerID**, **Cid**)
- **Peer Record**: (**PeerID**, **Multiaddress**)
  i.e. information to connect to a peer (@IP, port)



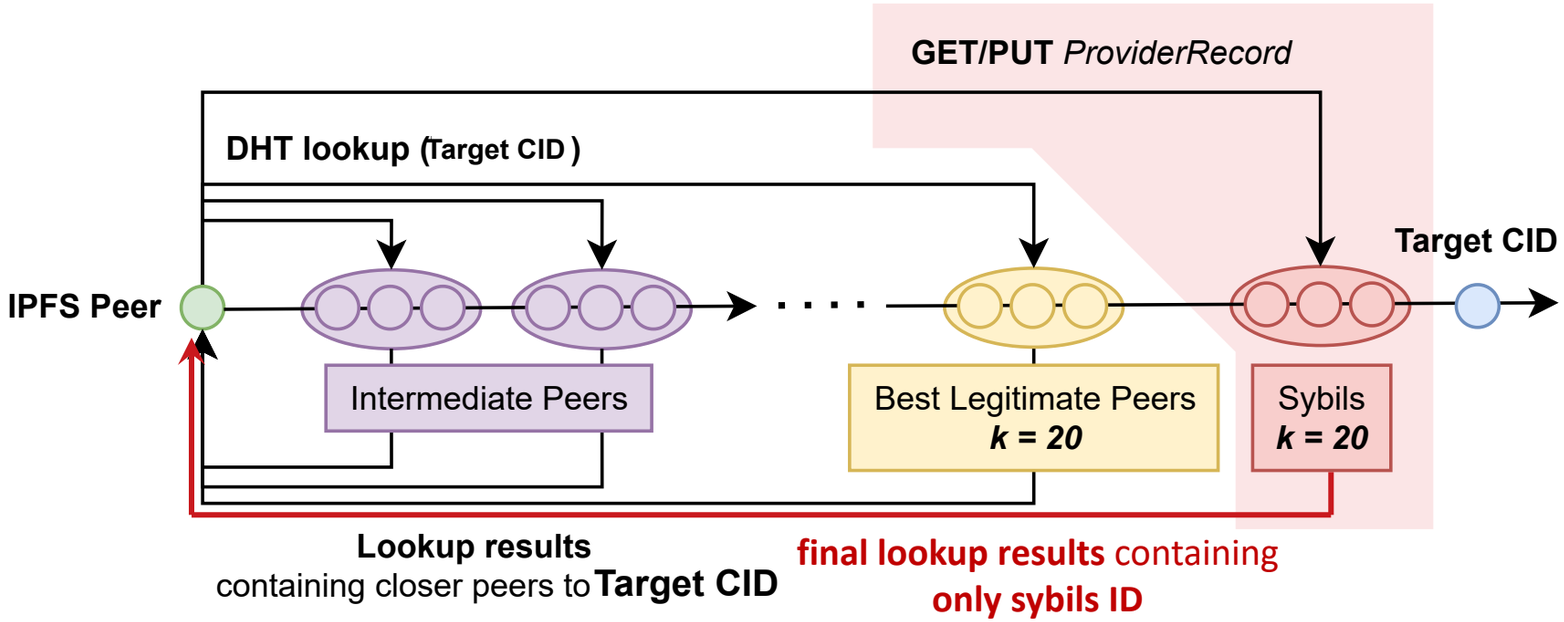
DHT Lookup


GET/PUT ProviderRecord
DHT lookup (Target CID)
Target CID
IPFS Peer
Intermediate Peers
Best Legitimate Peers
k = 20
Lookup results
containing closer peers to Target CID
(GET) Lookup results
ProviderRecord(s)

Sybil Attack Design

Objective: place 20 or more Sybils to be the closest to a given **Target CID** to store all the related **Records**

Passive attacker scenario: attracting all PUT requests but denying GET requests eclipses a Peer/Content in IPFS



Sybil Attack Design

Difficulty: **PeerIDs** are constrained (hash of a cryptographic key), so an attacker must first pre-compute Sybils' PeerID

- Monitored the network with 200 probes during 3 days
- Counted 6,800 **PeerIDs** and 3,500,000 **Cids**
- Estimated empirically that placing Sybils at a maximum distance of 2^{230} to a TargetID is close enough to get control of 99.95% of **Cids**
- Took 1h30 on a 8 cores desktop computer to **brute force** the 20 Sybil's **PeerID**
- All generated **PeerIDs** can be saved for other attacks

Implementation and experiment setup

Implementation

- Sybil client is a sightly modified IPFS Kubo client
- Behaves normally except for the target *Cid*
- Sybils advertise each other during the lookup process

Experiment setup

- Generate a random “target” file and share it in IPFS with a regular client
- Start Sybils and let them 15 minutes to be connected
- Try to retrieve the file with another regular client

Evaluation

- Attack success is the inability to retrieve the targeted file
- Upon attack failure we investigate how many records were captured by Sybils out of 20

Kubo vers.	Nb sybils	Nb IP@	Nb attack success	Nb attack failure	Nb Records intercepted in case of failure
19.2	27	27	9/11	2/11	19 and 19/20
20	27	27	10/12	2/12	17 and 19/20
20	20	1	11/11	0/11	-
20	20	1	12/12	0/12	-

- Attack is very effective overall
- IP-level distribution is not enforced. Running all Sybils on a single computer achieves 100% attack success

Possible defenses

Many defenses strategies exist against Sybil Attacks, we only consider here:

- Defenses considering the lookup process
- Applicable to an open fully-distributed P2P network
- Incrementally deployable

First layer: IP address restrictions

- Allow a single peer per IPv4 /24 subnet to be considered during a given lookup
- Forces attacker to distribute the Sybils at the network level

Possible defenses

Second layer: Hardening the lookup algorithm

- S/Kademlia proposal: run 3 independent parallel lookups (never stepping on a same peer)
- Avoid the attack to succeed when a Sybil is on the path

Third layer: Statistical distributions of PeerIDs

- Estimate PeerID’s distribution with lookups to random IDs
- Compare the distribution around an ID with the theoretical distribution to detect attacks (Sybils insertion create a bias)

Conclusion

We showed that IPFS lookup process is unprotected and can be abused by a Sybil Attack (even with a single computer)

We proposed a list of known mitigation mechanisms

Future work

- Didactic survey of P2P security mechanisms
- Consider active attacker scenario in IPFS