DATASHARDS

A presentation to the W3C CCG

Christopher Lemmer Webber

Website: https://dustycloud.org/

Fediverse: https://octodon.social/@cwebber

Birdsite: https://twitter.com/dustyweb

DATASHARDS

Secure, distributed storage primitives for the web

DATASHARDS

Secure, distributed storage primitives for the web

- Only intended recipients can see messages
- Safer for node operators to help network
- Secure against size-of-file-attacks

DATASHARDS

Secure, distributed storage primitives for the web

- Many different "stores"
  - A "Secure Data Hub" web store!
  - Your local filesystem!
  - USB keys!
  - Distributed over a gossip network/DHT!
- Nodes can host content, but don't know what it is

DATASHARDS

Secure, distributed storage primitives for the web

- Meant to be foundational, the way http(s) is foundational for live content
- Two new URI schemas:
  - idsc: Immutable, unchanging data
  - mdsc: Mutable, updateable data
- Data can have the same name, but can live in many places

But why not IPFS/Dat/etc?

Datashards encrypts everything!

This is critical for:

- **Users:** People deserve built-in privacy. Privacy should be the default, everywhere.
- **Node operators:** Being able to "see" content makes you more liable for its contents. Sometimes it's better not to know.
  - Volunteer postal workers analogy goes here!
  - This point means that "layering" encryption after the fact is unsafe. Cleartext data is toxic to the network.

IDSC example usage

Real implementations exist, today!
(racket-datashards, pydatashards)

Uploading:

```
$ raco idsc --verbose --upload friend-picture.png
DEBUG: posted urn:sha256d:i-UOOq5uoBZlmAYNe2CM_JQ_oObfbcsbVs3wO4_4IkA
DEBUG: posted urn:sha256d:hjL2-_Y4Tk9noJAW2gLu0XmRa3REnSmuKREp_ysgMyg
DEBUG: posted urn:sha256d:iGXzHXi_9NuyDgBhQcbLdOxQiaHXOzkQwA41JmDjJuQ
DEBUG: posted urn:sha256d:ZPB08Pl9ekrqihwQXoPdDx0Afkf9SWUc-Awi7hoGPW0
DEBUG: posted urn:sha256d:x9ZO2FiOy7rtf5bcMoUnU_IeMHTvobiAeH3tcc9W_OE
idsc:0p.x9ZO2FiOy7rtf5bcMoUnU_IeMHTvobiAeH3tcc9W_OE.UF84o-DrREcdP5McSk6YPJJDSzp4h9TEbAi35WXmJPE
```

Downloading:

```
$ raco idsc --verbose --get \
       idsc:0p.x9ZO2FiOy7rtf5bcMoUnU_IeMHTvobiAeH3tcc9W_OE.UF84o-DrREcdP5McSk6YPJJDSzp4h9TEbAi35WXmJPE \
       > got-friend-picture.png
DEBUG: got urn:sha256d:x9ZO2FiOy7rtf5bcMoUnU_IeMHTvobiAeH3tcc9W_OE
DEBUG: got urn:sha256d:i-UOOq5uoBZlmAYNe2CM_JQ_oObfbcsbVs3wO4_4IkA
DEBUG: got urn:sha256d:hjL2-_Y4Tk9noJAW2gLu0XmRa3REnSmuKREp_ysgMyg
DEBUG: got urn:sha256d:iGXzHXi_9NuyDgBhQcbLdOxQiaHXOzkQwA41JmDjJuQ
DEBUG: got urn:sha256d:ZPB08Pl9ekrqihwQXoPdDx0Afkf9SWUc-Awi7hoGPW0
```

The client is uploading/downloading to/from a store, but the store doesn’t understand what the data is

IDSCs seem fine for cat photos

But we demand content we can update!

(... right?)

Enter MDSCs!

- You can update MDSCs!
- Every MDSC is built on top of a new public/private key-pair
- The public key *is* the name of the MDSC!
- For every MDSC, there are three ”levels” of access:
  - Verify
  - Read (+ Verify)
  - Write (+ Read, Verify)

MDSC: consistency not included!

We want to support "garbage collectable" content, so we do not try to provide consistency out of the box

However, you can add it yourself:

- A blockchain
- A centralized oracle that registers "official" MDSC updates (eg, a government identity agency)

Example uses

- Any document you would otherwise keep on the web
- A more resilient federated social network
  - Content which survives nodes going down (Demo written! See "Spritely Golem")
  - Users can change where their profiles live
- Storing VCs
- As a DID mechanism: **`did:ds:<mdsc-verify-cap-info>`**
- As the foundation for secure data hubs, but not limited to secure data hubs

Conclusions

We need Datashards because:

- The "live web" is too fragile, too much data people care about lost
- Existing options don't support privacy and are unsafe for node administrators
- Having a fundamental secure CAS primitive allows us to build other things on top of it

https://datashards.net