paperback

a digital will and backup system for the reasonably paranoid

Aleksa Sarai
cyphar.com · github.com/cyphar/paperback

Digital Assets and Wills

Very difficult to handle digital assets with wills:

- Often no legally-mediated process for transfer.
  - “Just put your passwords in your will.”
- Relies very heavily on lawyers and their opsec.
  - Unlike most secrets held by lawyers, there is often no practical recourse for theft (obvious examples are cryptocurrency wallets).
  - There have been (rare) cases of lawyers themselves stealing money.
- I am not a lawyer. Speak to an actual lawyer about wills.

Digital Assets and Backups

Secure backups of digital assets is also difficult:

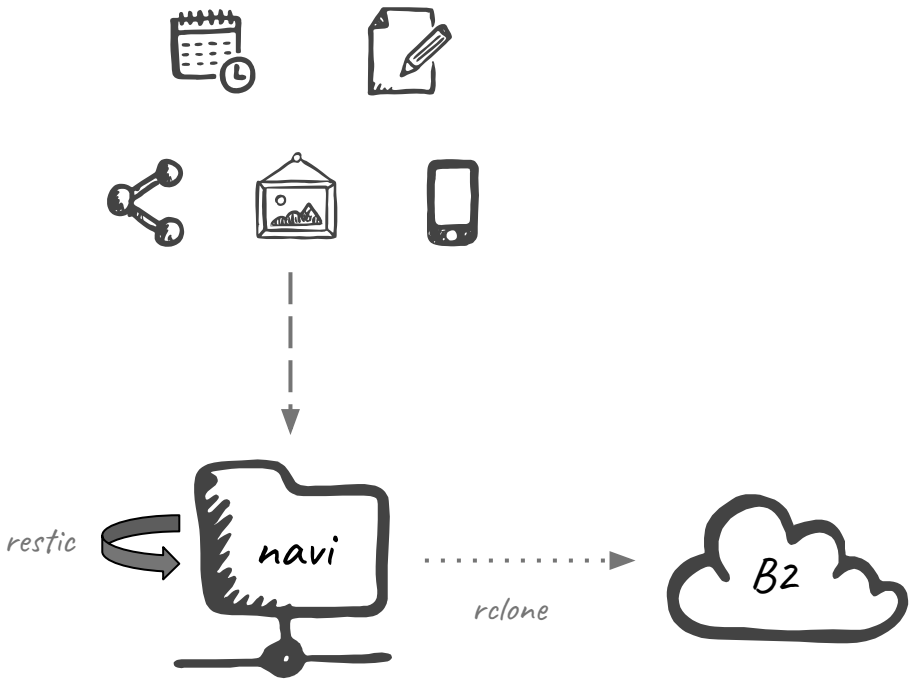
- Backups usually need to be distributed and encrypted.
  - How will you make the key both secure and recoverable in the case you lose all of your hardware?

- Similar opsec issues, except now it's your opsec.
  - … and the opsec of everyone you gave a copy to.

My Backups

- NextCloud front-end for uploading photos and documents to my home server.
- Restic for **encrypted** and highly-deduplicated backups.
- Backups uploaded to BackBlaze B2 with rclone daily.

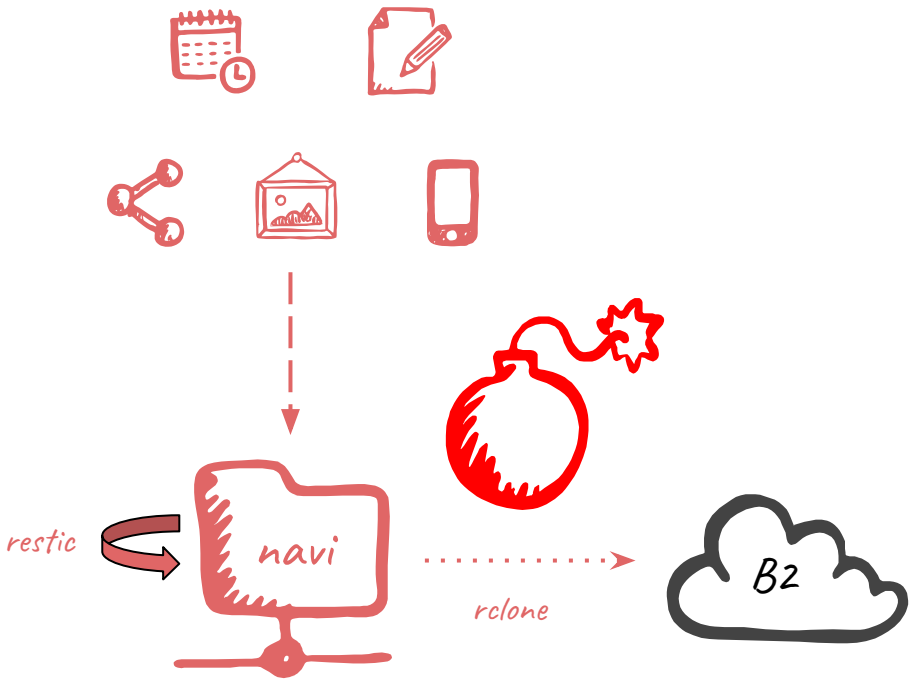
In order to recover my backups, I need a passphrase or keyfile.



How to Recover?

In order to recover my backups, I need a passphrase or keyfile.

- Cannot use a simple key, otherwise prone to attacks.
- A complicated key means you need to have a backup of some kind.
- But then how do we make *that* backup sec— … ah.



Shamir Secret Sharing

The Gist

Using some fairly rudimentary algebra, you can take a secret and split it into **K** pieces (shards) such that you require at least **N** shards to recover the secret.

> *Unlike the naive approach, an attacker with N-1 shards has no more information than if they had 0 shards. The scheme has **perfect security**, similar to a one-time pad.*

The only requirement is that less than **N** shard holders will conspire against you.

Lagrange Polynomials

The only maths really required is the following:

> *For any n+1 points there is a unique n^{th} order polynomial that passes through those points. In order to reconstruct the equation of an n^{th} order polynomial you need n+1 points.*

Lagrange polynomials are a relatively straightforward (though fairly slow) method of polynomial interpolation.

$$\ell_j(x) := \prod_{\substack{0 \le m \le k \\ m \ne j}} \frac{x - x_m}{x_j - x_m} = \frac{(x - x_0)}{(x_j - x_0)} \cdots \frac{(x - x_{j-1})}{(x_j - x_{j-1})} \frac{(x - x_{j+1})}{(x_j - x_{j+1})} \cdots \frac{(x - x_k)}{(x_j - x_k)}$$

$$L(x) := \sum_{j=0}^{k} y_j \ell_j(x)$$

Shamir Secret Sharing

Simply construct a polynomial with random coefficients and the constant is the secret. Give each holder a distinct (x, y) pair then use Lagrange polynomials to reconstruct the secret.

For an (**N**, **K**) scheme, make an **N**-1 degree polynomial and hand out **K** points.

Perfectly secure (in finite fields) because an infinite number of n^{th}-degree polynomials pass through any set of n points.

$$f(x) = R_0 x^n + R_1 x^{n-1} + \cdots + R_{n-1} x + Secret$$

$$L(0) = \sum_{j=0}^{k-1} y_j \prod_{\substack{m=0 \\ m \neq j}}^{k-1} \frac{x_m}{x_m - x_j} = Secret$$

A Few Words of Warning

- Perfect security sounds nice, but relies on working code.
  - No standard widely-used library which is well-audited.
  - Most libraries I looked at had some kind of critical flaw.

- Shamir Secret Sharing requires the secret to be in a single place.
  - Potential for a single-point-of-failure at creation and recovery time.
  - If you need to do recoveries often, there might be better solutions (e.g. Bitcoin Multisig).

paperback

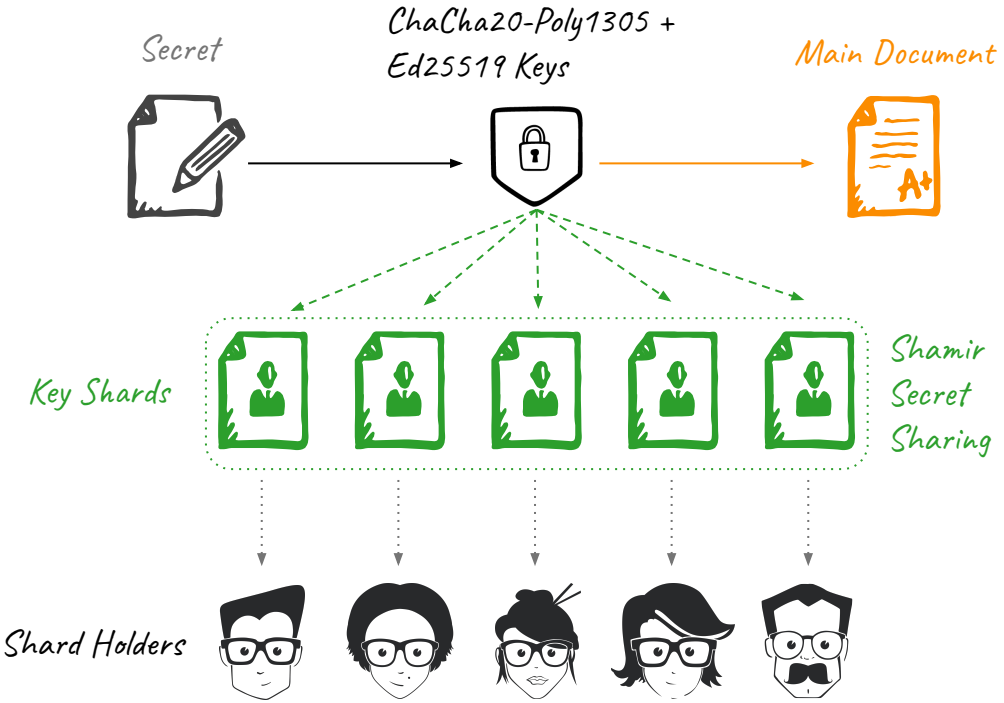
General Design

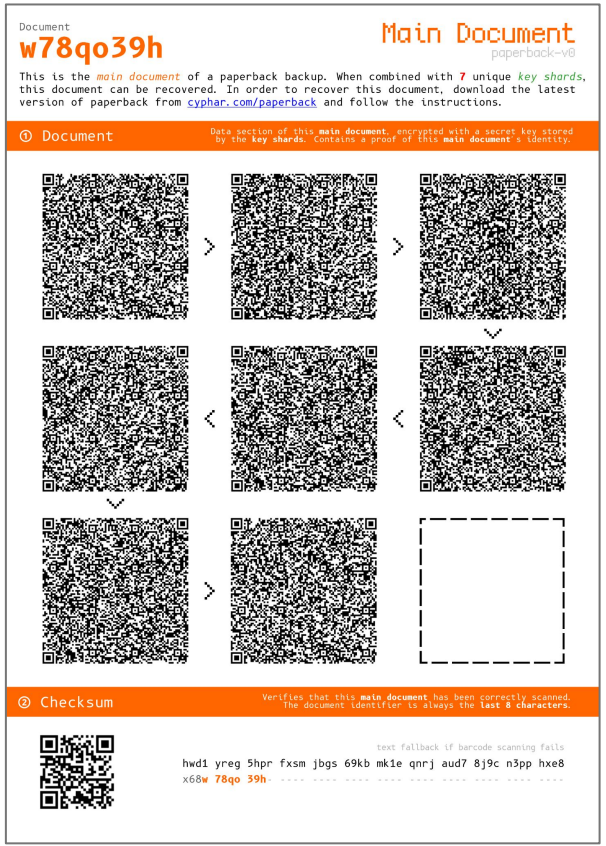
Secret is encrypted using random key with ChaCha20-Poly1305. Two document types:

- Main Document
- Key Shard

All documents are signed using Ed25519. Both keys are split using Shamir Secret Sharing into Key Shards. Public key and main document hash is added to every document. Hashes are Blake2b-256.

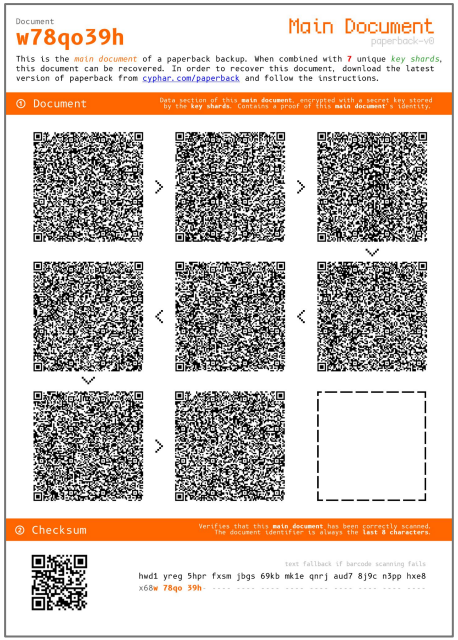
Threat model is described in the repo.





Paperback Documents (Design Mockups)

Main Document: Header



Document

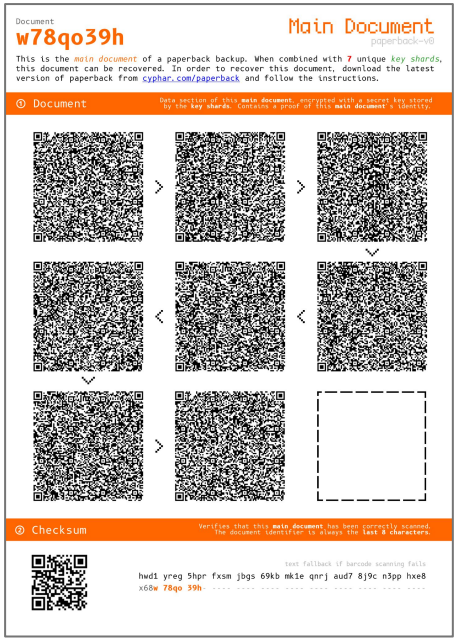
w78qo39h

Main Document

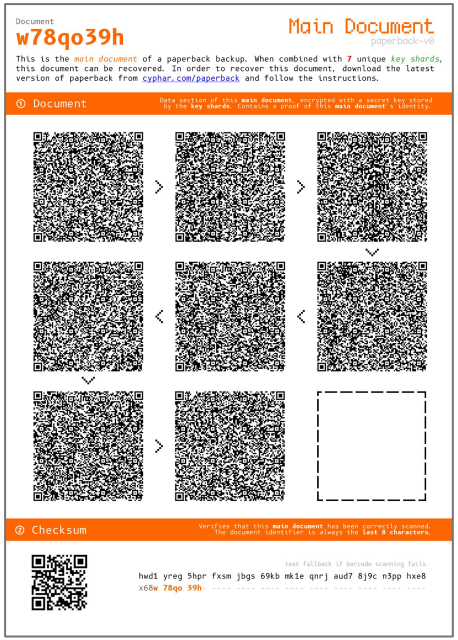
paperback-v0

This is the *main document* of a paperback backup. When combined with **7** unique *key shards*, this document can be recovered. In order to recover this document, download the latest version of paperback from cyphar.com/paperback and follow the instructions.

Main Document: Document


Document
w78qo39h
Main Document
paperback-v0
This is the main document of a paperback backup. When combined with 7 unique key shards, this document can be recovered. In order to recover this document, download the latest version of paperback from cyphar.com/paperback and follow the instructions.
① Document
Data section of this main document, encrypted with a secret key stored by the key shards. Contains a proof of this main document's identity.
② Checksum
Verifies that this main document has been correctly scanned. The document identifier is always the last 8 characters.
text fallback if barcode scanning fails
hwd1 yreg 5hpr fxsm jbgs 69kb mk1e qnrj aud7 8j9c n3pp hxe8
x68w 78qo 39h

Main Document: Checksum



Key Shard: Header

Shard
hxdisc4y
Document
w78qo39h
Key Shard
paperback-v0
This is a key shard of a paperback backup.
See cyphar.com/paperback for more details.

Key Shard: Shard

Shard
hxdisc4y
Document
w78qo39h

Key Shard
paperback-v0

This is a *key shard* of a paperback backup.
See cyphar.com/paperback for more details.

① Shard — **Key shard** data, encrypted with the **shard codewords** below. Contains a proof of this **key shard**'s identity.

text fallback if barcode scanning fails

hzim h1pi c8r7 rnuk qntc 9rcb acte makm kyqp
h6b9 ciey s33i dz67 ue37 8uek 44nw 1hxx bkjy
7h3n m1z1 x3mx 3y9m i138 imdm 6kfs nyxx pygi
co1s fr6q 94a7 gyyh ymsz ya31 ac8t yadg py4j
sddj pdpd izx6 p75a bey9 6tyw r114 5epq 3i3t
u4s8 xuqj 9hoz jq3j mnms g8c8 wsi9 q6mn uqe6
5r5e nsh8 bo91 tf78 1w8b 647g k55p uofp k8ai
zzsq 98g7 n7ro ---- ---- ---- ---- ---- ----

② Checksum — Verifies that the **key shard** has been correctly scanned.

text fallback if barcode scanning fails

hwd1 yrer fdwz y6oq kbts 7sij 1q6p nrnn n4pc
scep euc6 kh8t 8njj q7rj edr- ---- ---- ----

③ Codewords — Encrypts the **key shard** contents. Can optionally be cut off and stored separately from the **key shard**.

Shard
hxdisc4y
Document
w78qo39h

chair dash possible april simple cabin
similar pave swear alien spin elder media
ensure bachelor cage project dress spike
want stereo rural element permit

① Shard — **Key shard** data, encrypted with the **shard codewords** below. Contains a proof of this **key shard**'s identity.

text fallback if barcode scanning fails

hzim h1pi c8r7 rnuk qntc 9rcb acte makm kyqp
h6b9 ciey s33i dz67 ue37 8uek 44nw 1hxx bkjy
7h3n m1z1 x3mx 3y9m i138 imdm 6kfs nyxx pygi
co1s fr6q 94a7 gyyh ymsz ya31 ac8t yadg py4j
sddj pdpd izx6 p75a bey9 6tyw r114 5epq 3i3t
u4s8 xuqj 9hoz jq3j mnms g8c8 wsi9 q6mn uqe6
5r5e nsh8 bo91 tf78 1w8b 647g k55p uofp k8ai
zzsq 98g7 n7ro ---- ---- ---- ---- ---- ----

Key Shard: Checksum


Shard
hxdisc4y
Document
w78qo39h

Key Shard
paperback-v0

This is a *key shard* of a paperback backup.
See cyphar.com/paperback for more details.

① Shard — **Key shard** data, encrypted with the **shard codewords** below. Contains a proof of this **key shard**'s identity.

text fallback if barcode scanning fails

hzim h1pi c8r7 rnuk qntc 9rcb acte makm kyqp
h6b9 ciey s33i dz67 ue37 8uek 44nw 1hxx bkjy
7h3n m1z1 x3mx 3y9m i138 imdm 6kfs nyxx pygi
co1s fr6q 94a7 gyyh ymsz ya31 ac8t yadg py4j
sddj pdpd izx6 p75a bey9 6tyw r114 5epq 3i3t
u4s8 xuqj 9hoz jq3j mnms g8c8 wsi9 q6mn uqe6
5r5e nsh8 bo91 tf78 1w8b 647g k55p uofp k8ai
zzsq 98g7 n7ro ---- ---- ---- ---- ---- ----

② Checksum — Verifies that the **key shard** has been correctly scanned.

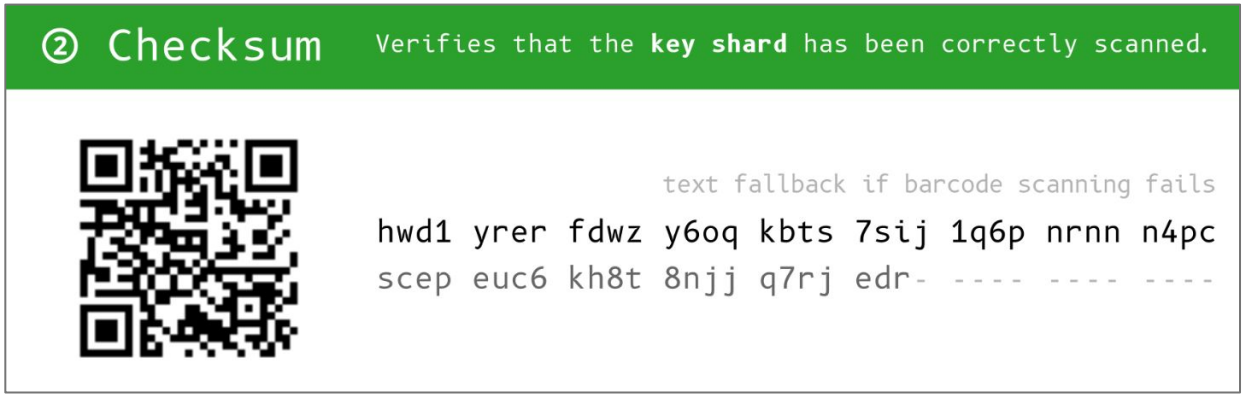
text fallback if barcode scanning fails

hwd1 yrer fdwz y6oq kbts 7sij 1q6p nrnn n4pc
scep euc6 kh8t 8njj q7rj edr- ---- ---- ----

③ Codewords — Encrypts the **key shard** contents. Can optionally be cut off and stored separately from the **key shard**.

Shard
hxdisc4y
Document
w78qo39h

chair dash possible april simple cabin
similar pave swear alien spin elder media
ensure bachelor cage project dress spike
want stereo rural element permit


② Checksum — Verifies that the **key shard** has been correctly scanned.

text fallback if barcode scanning fails

hwd1 yrer fdwz y6oq kbts 7sij 1q6p nrnn n4pc
scep euc6 kh8t 8njj q7rj edr- ---- ---- ----

Key Shard: Codewords


Shard
hxdisc4y
Document
w78qo39h

Key Shard
paperback-v0

This is a *key shard* of a paperback backup.
See cyphar.com/paperback for more details.

① Shard — **Key shard** data, encrypted with the **shard codewords** below. Contains a proof of this **key shard**'s identity.

text fallback if barcode scanning fails

hzim h1pi c8r7 rnuk qntc 9rcb acte makm kyqp
h6b9 ciey s33i dz67 ue37 8uek 44nw 1hxx bkjy
7h3n m1z1 x3mx 3y9m i138 imdm 6kfs nyxx pygi
co1s fr6q 94a7 gyyh ymsz ya31 ac8t yadg py4j
sddj pdpd izx6 p75a bey9 6tyw r114 5epq 3i3t
u4s8 xuqj 9hoz jq3j mnms g8c8 wsi9 q6mn uqe6
5r5e nsh8 bo91 tf78 1w8b 647g k55p uofp k8ai
zzsq 98g7 n7ro ---- ---- ---- ---- ---- ----

② Checksum — Verifies that the **key shard** has been correctly scanned.

text fallback if barcode scanning fails

hwd1 yrer fdwz y6oq kbts 7sij 1q6p nrnn n4pc
scep euc6 kh8t 8njj q7rj edr- ---- ---- ----

③ Codewords — Encrypts the **key shard** contents. Can optionally be cut off and stored separately from the **key shard**.

Shard
hxdisc4y
Document
w78qo39h

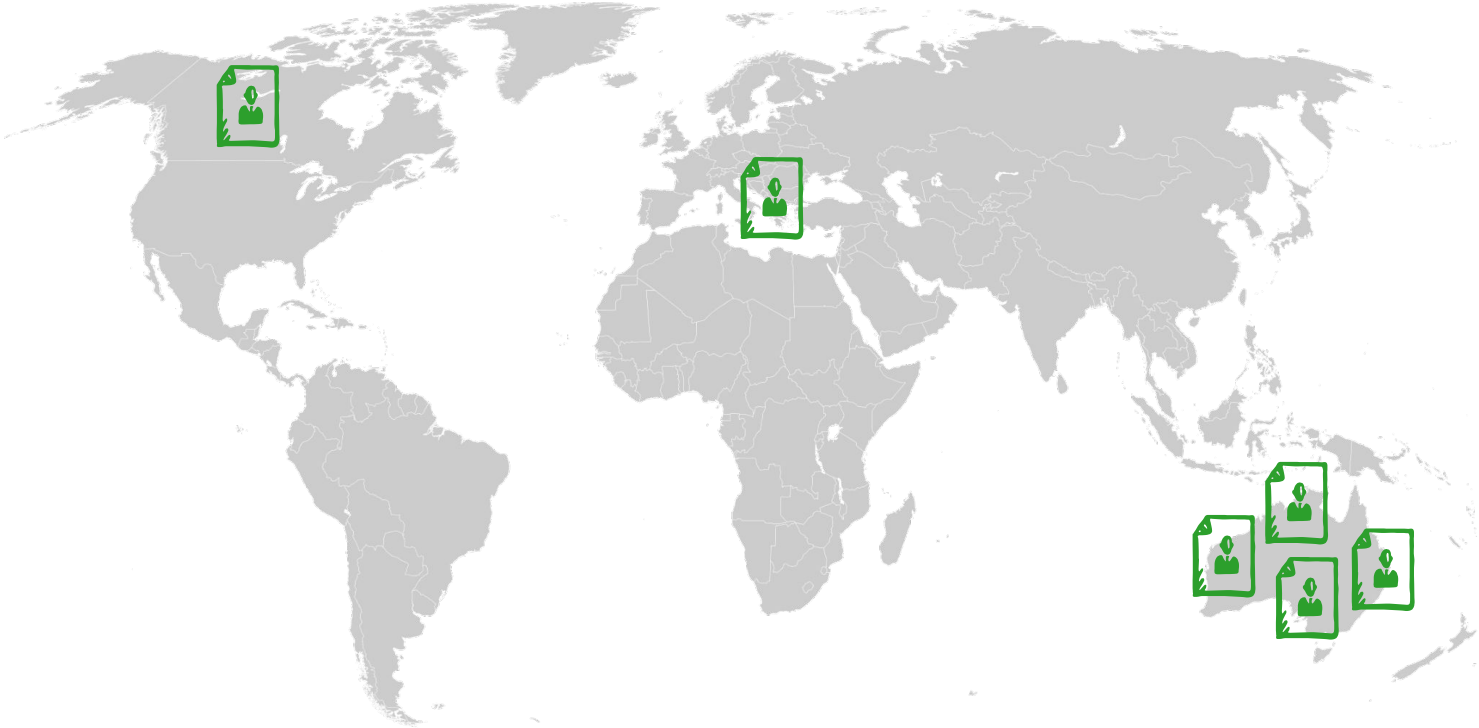
chair dash possible april simple cabin
similar pave swear alien spin elder media
ensure bachelor cage project dress spike
want stereo rural element permit

③ Codewords — Encrypts the **key shard** contents. Can optionally be cut off and stored separately from the **key shard**.

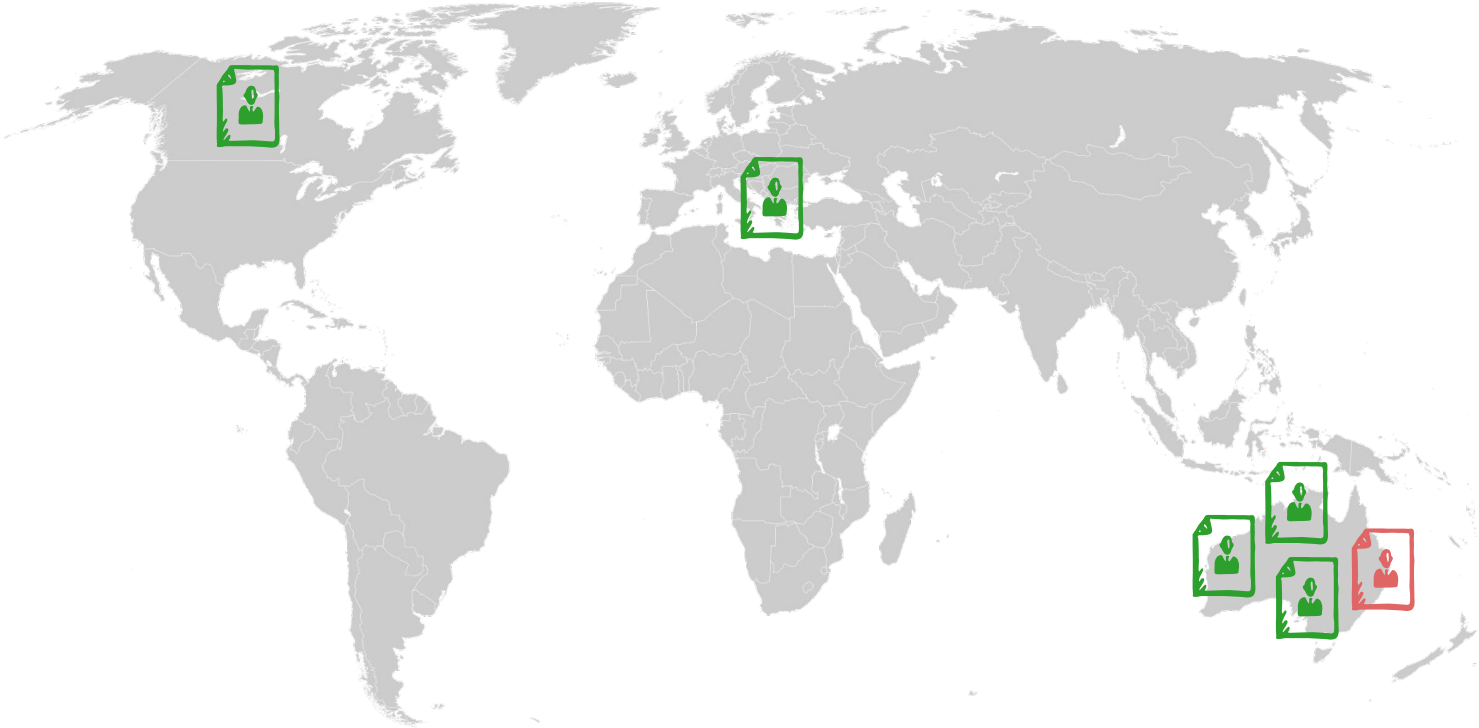
Shard
hxdisc4y
Document
w78qo39h

chair dash possible april simple cabin
similar pave swear alien spin elder media
ensure bachelor cage project dress spike
want stereo rural element permit

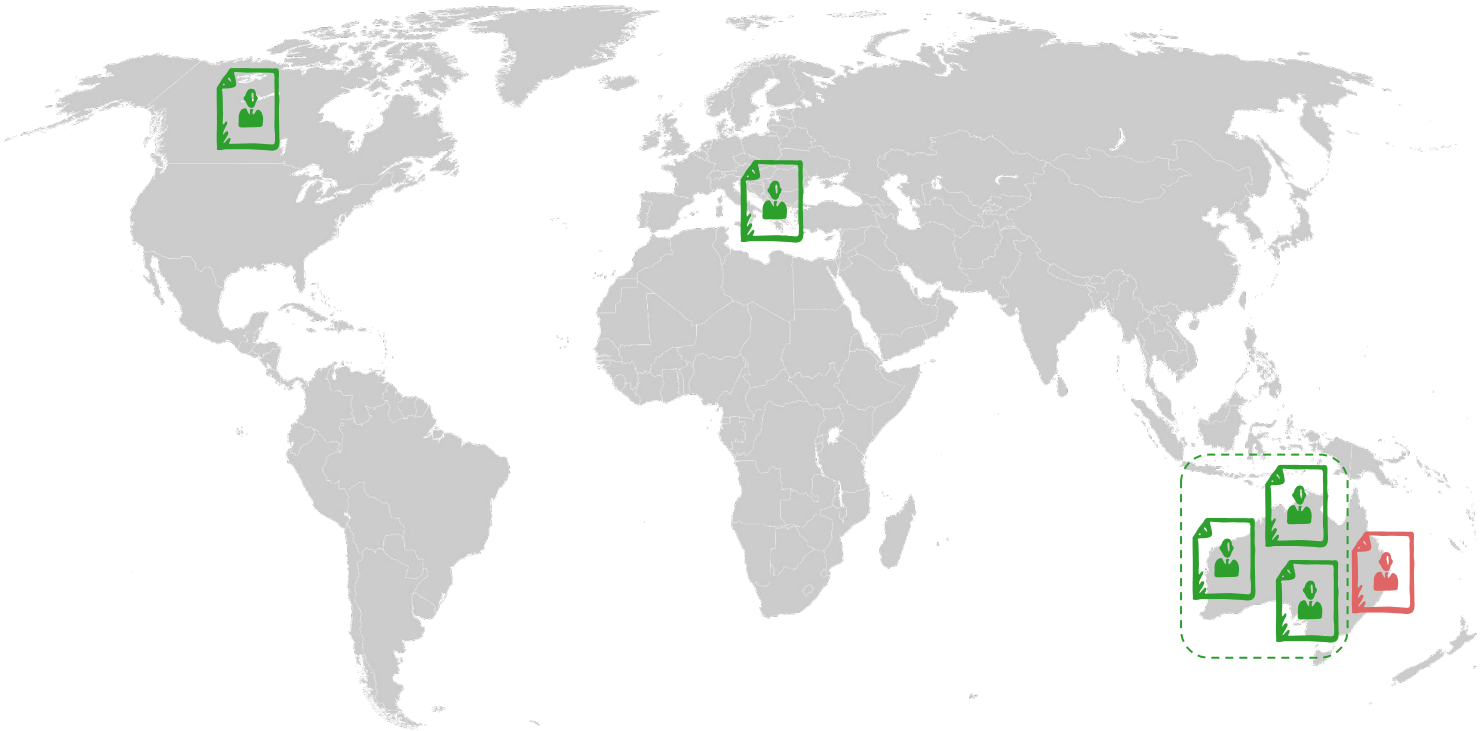
Quorum Expansion



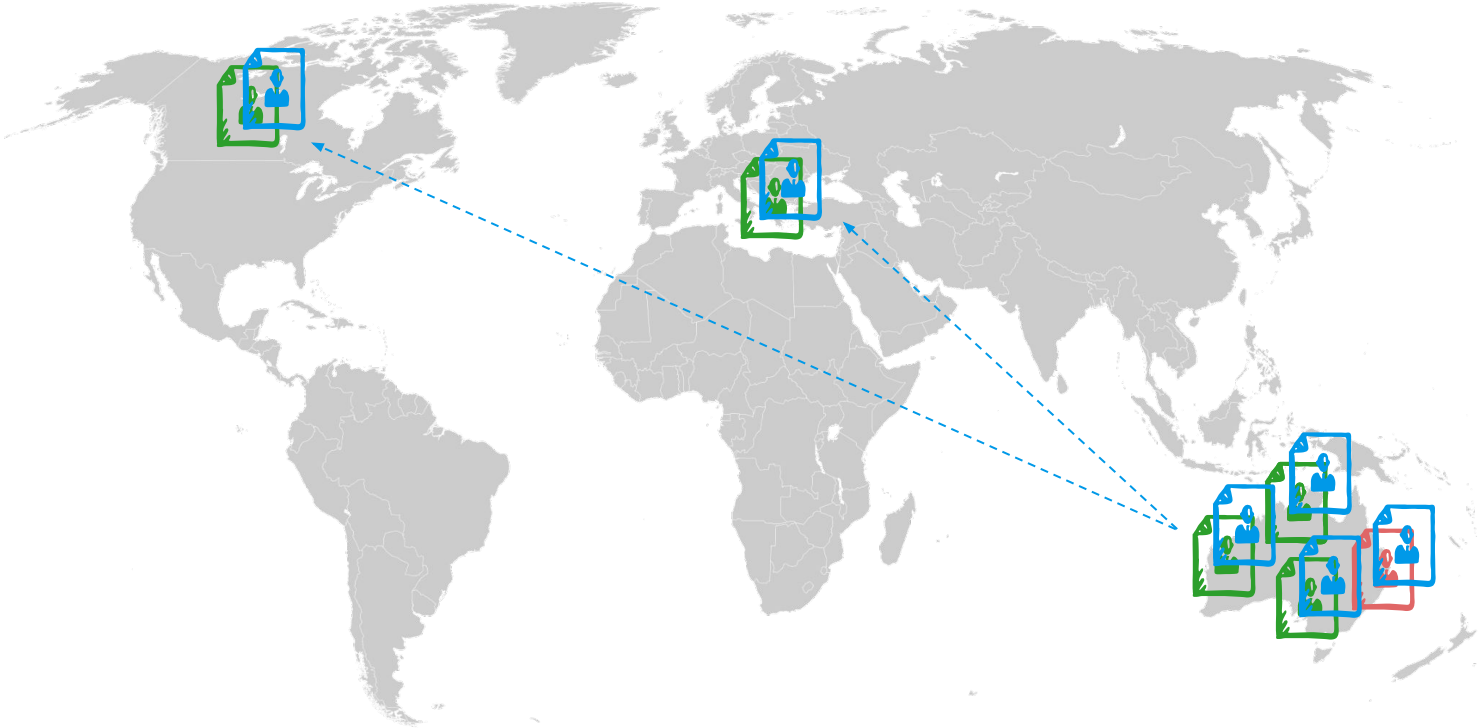
Quorum Expansion



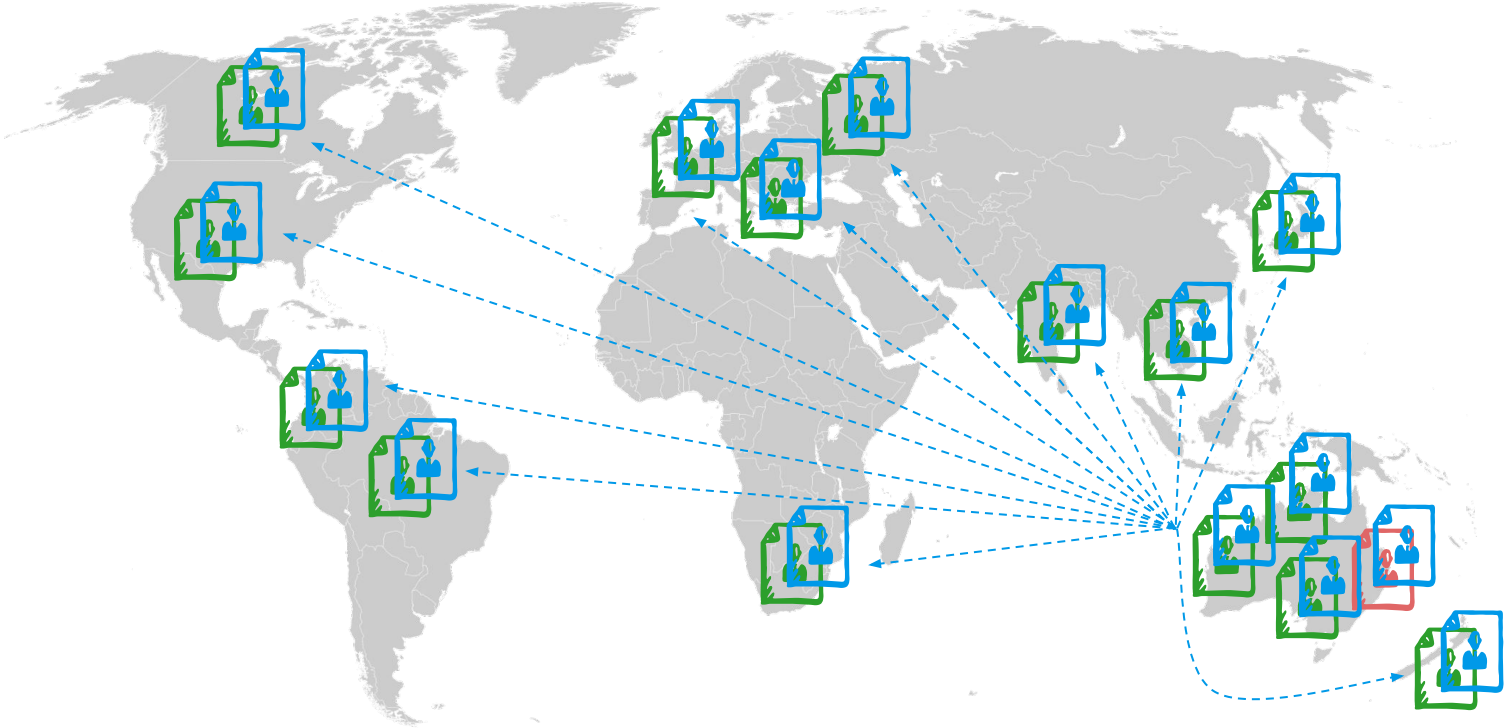
Quorum Expansion



Quorum Expansion (Naive Method)



Quorum Expansion (Naive Method)



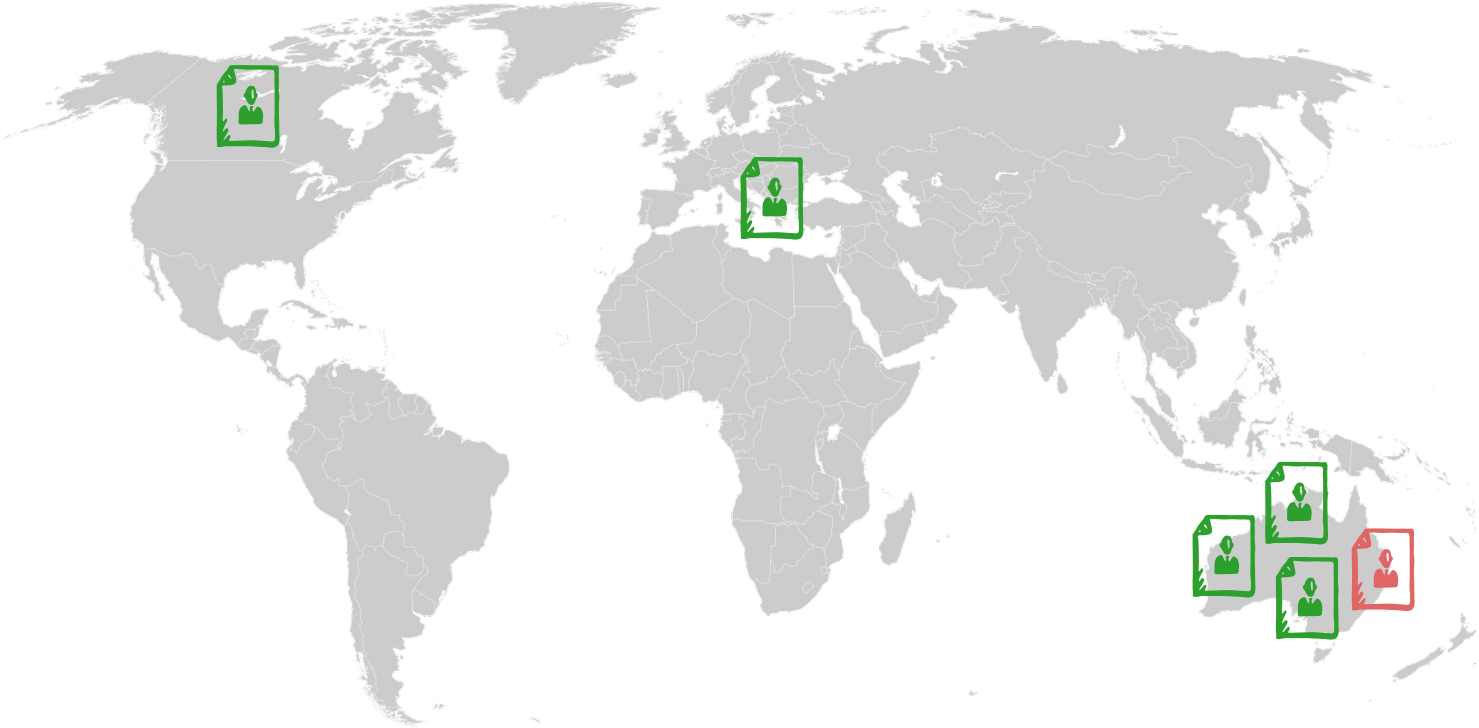
*Quorum Expansion (**Very** Naive Method)*

Recovering the Polynomial

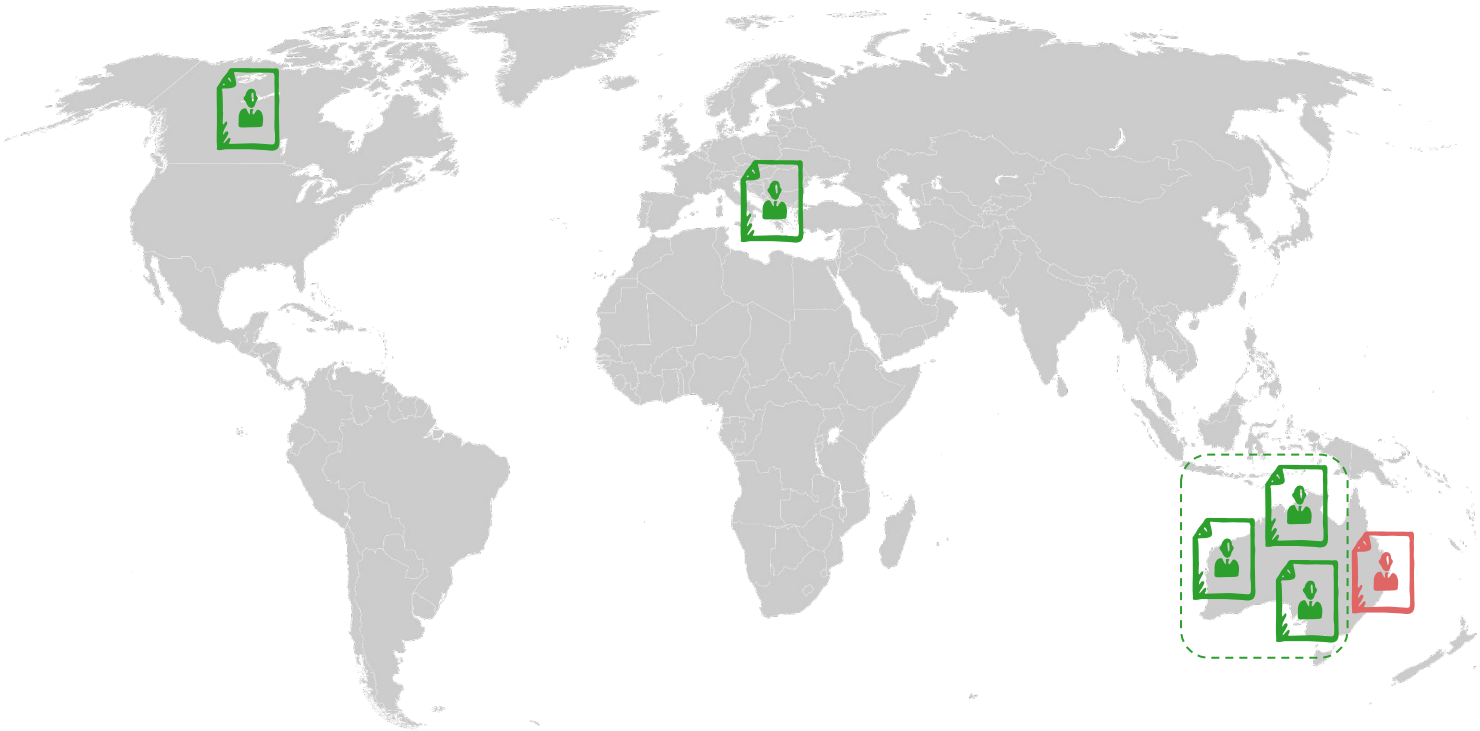
Lagrange interpolation will let you compute any *x* value (not just *x*=0). Thus if we have **N** shards, we can make new shards by just using the same interpolation to evaluate new *x* values.

$$\ell_j(x) := \prod_{\substack{0 \le m \le k \\ m \ne j}} \frac{x - x_m}{x_j - x_m} = \frac{(x - x_0)}{(x_j - x_0)} \cdots \frac{(x - x_{j-1})}{(x_j - x_{j-1})} \frac{(x - x_{j+1})}{(x_j - x_{j+1})} \cdots \frac{(x - x_k)}{(x_j - x_k)}$$

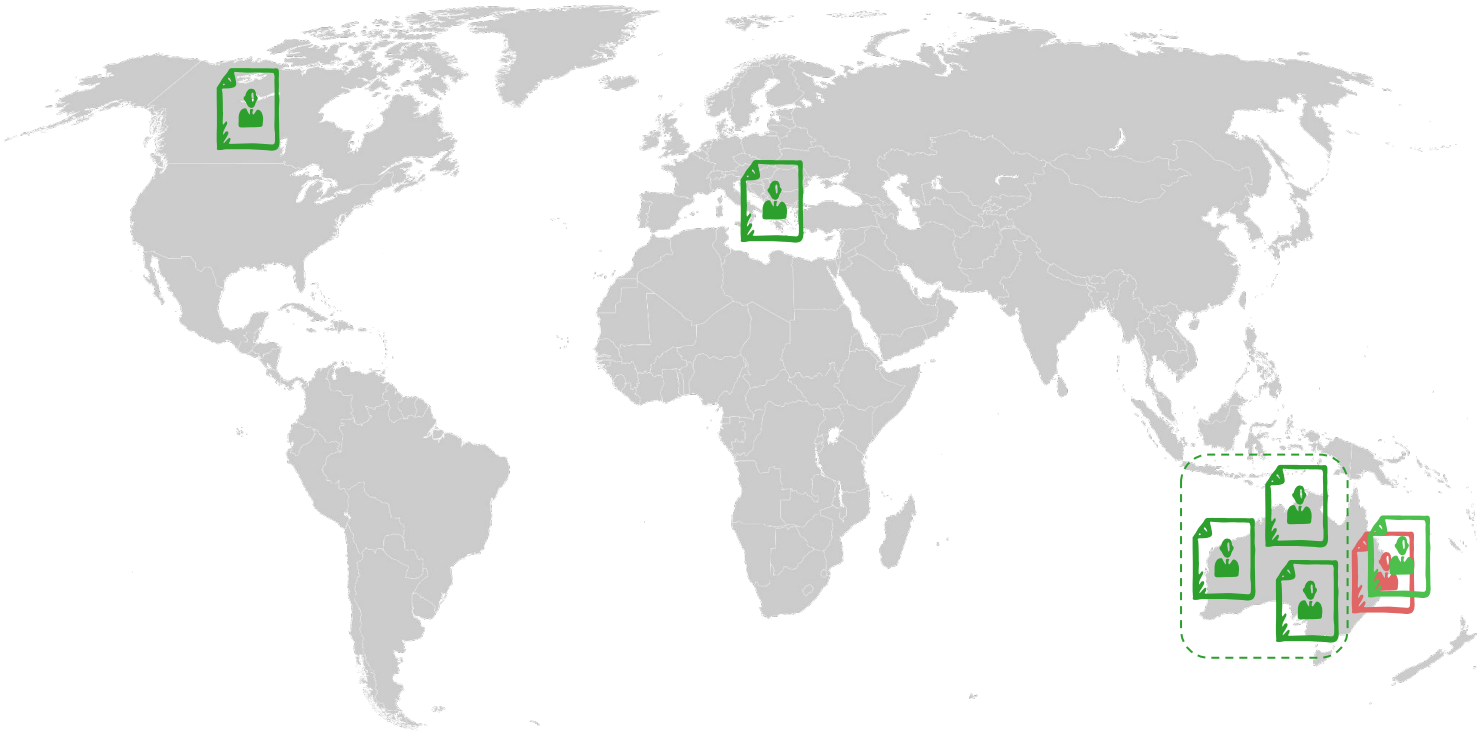
$$L(x) := \sum_{j=0}^{k} y_j \ell_j(x)$$



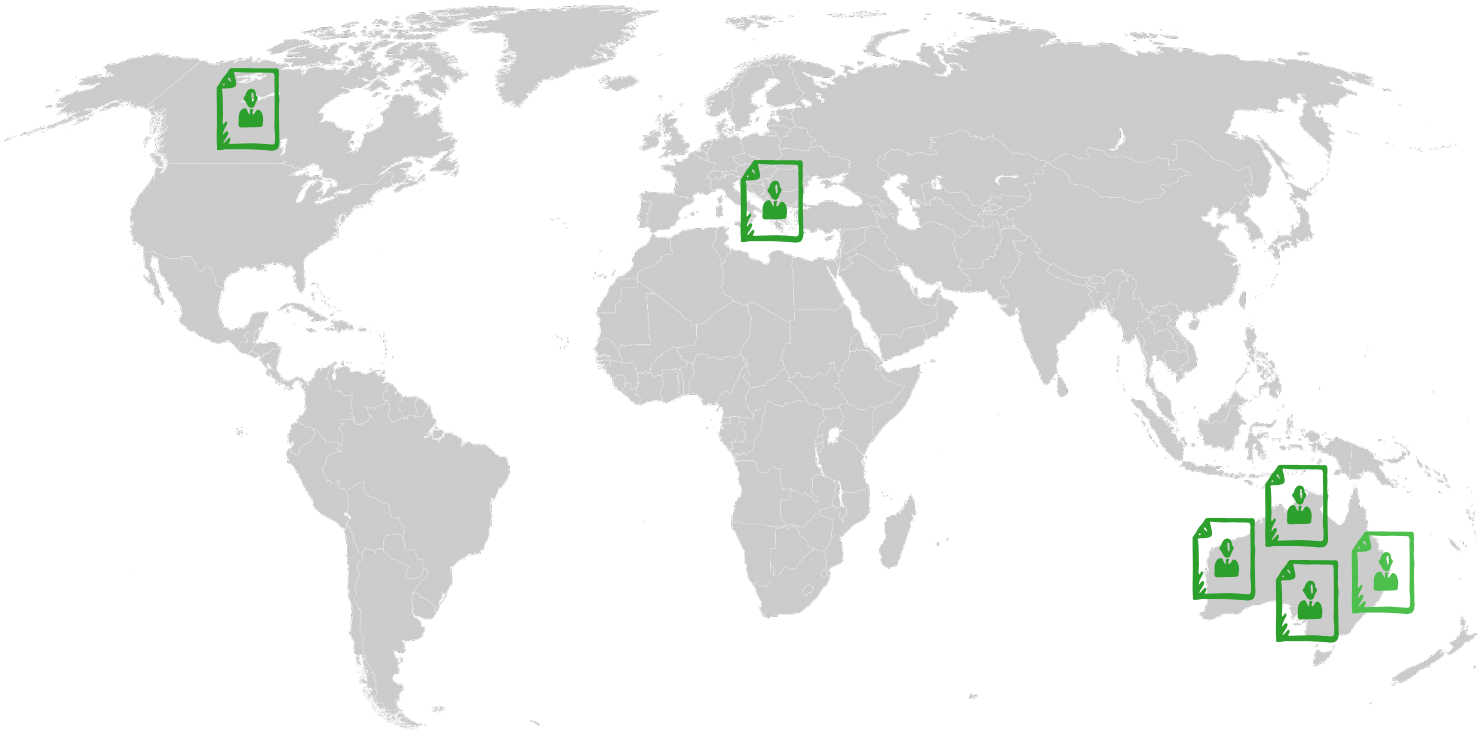
Quorum Expansion (Paperback)



Quorum Expansion (Paperback)



Quorum Expansion (Paperback)



Quorum Expansion (Paperback)

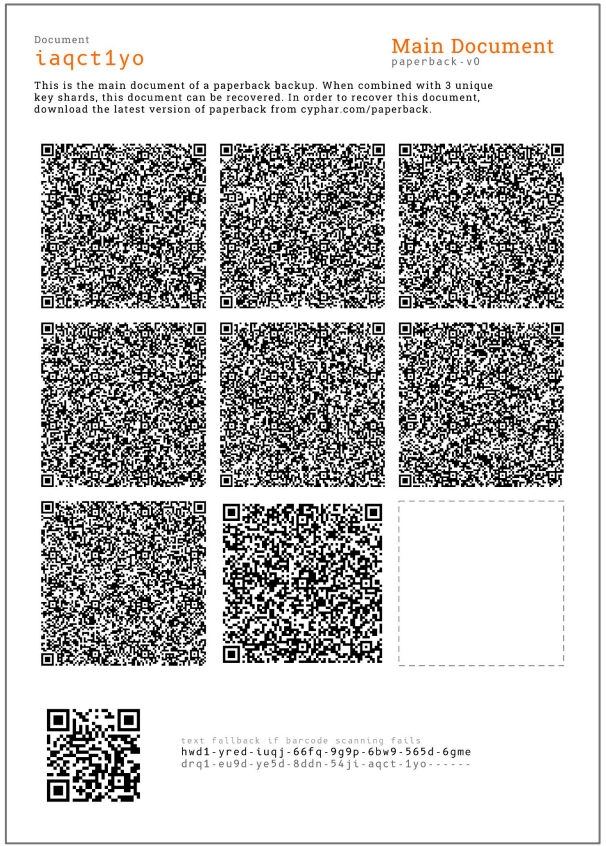
Additional Features

All documents include document checksum and signature from Ed25519 signing key to detect forgeries.

While paperback can construct new shards after the fact, this can be made “tamper evident” if the document is “sealed”.

Use zbase32 to make it harder to mistake characters (might look at emojisum to use more bytes in document identifier).

Current Status

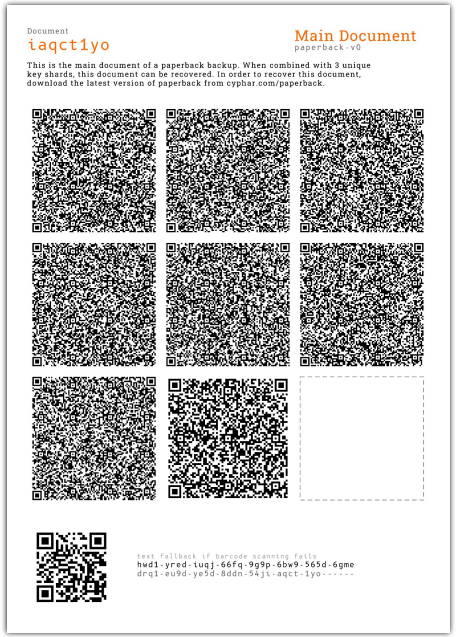


Paperback Documents

Current Status

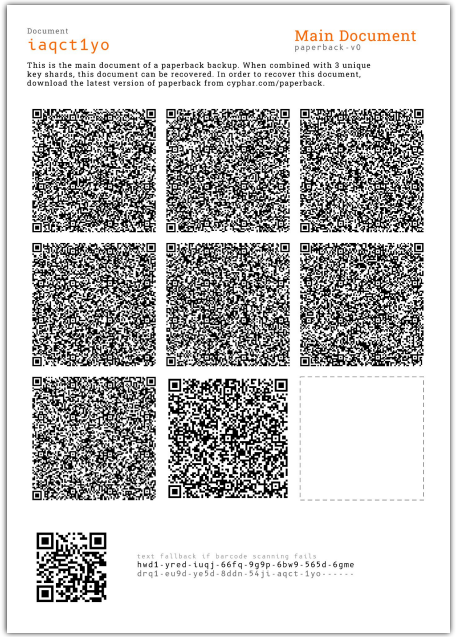
- All paperback operations functional (command-line).
  - Including generation of paperback documents (PDF).
  - Styling doesn't yet match the mockups.

Next Steps



- Handle larger documents.
  - Is it safe to compress the contents?
  - Currently an all-or-nothing single-page approach.
- Mobile or web-based applications.
  - A graphical interface that my relatives could use.
  - Not to mention scanning of QR codes on desktop.
- Some kind of cryptographic review and audit.

Open Problems



- How to ensure documents are recoverable in 20/30/50/100 years?
  - Textual description of recovery algorithm?
  - Basic implementation in pseudo-code / python / javascript?
  - Can we assume QR codes will be understood in the future?
- Are there better-density codes we could use?
  - DataMatrix, HCCB, Jab Codes, HCC2D, etc...

Aside: Paper Storage

Paper Storage

- Print everything duplex (two-sided).
- Use archival-grade acid-free paper.
  - Regular copy paper degrades much faster.
- Avoid hot lamination.
  - Use encapsulation sleeves (made of an inert material).
- National libraries usually have good archival advice.
  - The Canadian Conservation Institute has lots of good information.

Questions?

cyphar.com/paperback