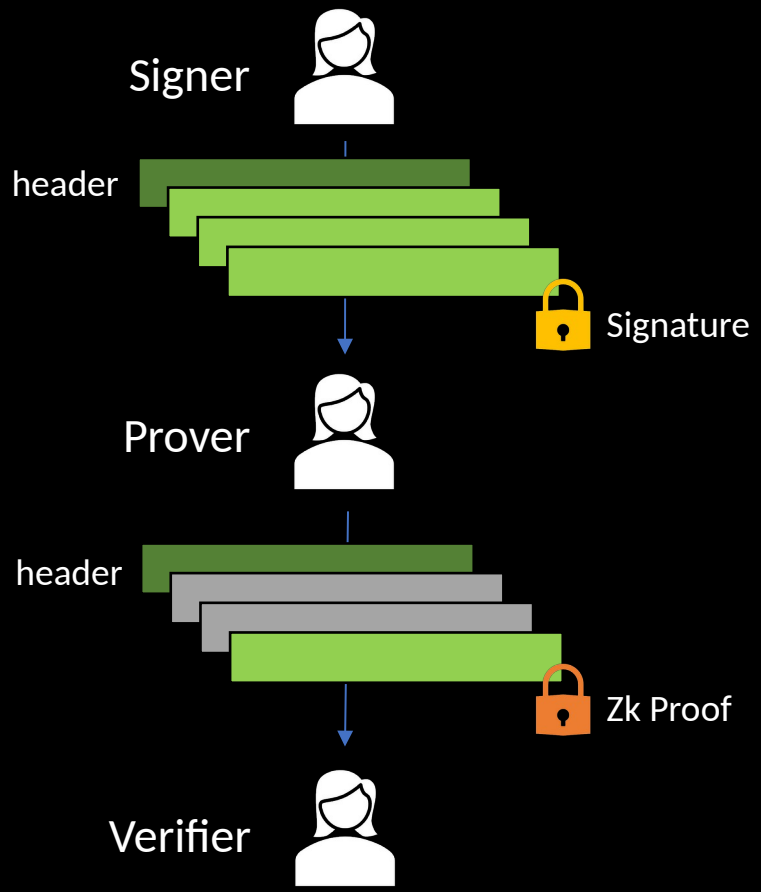
BBS Signatures

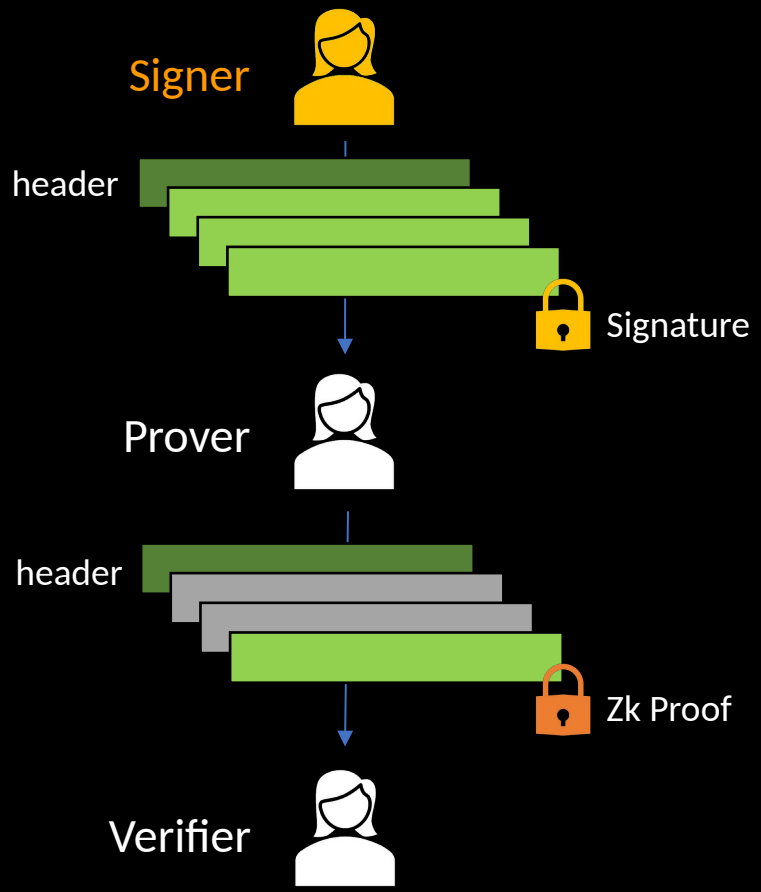
Tobias Looker, Vasilis Kalos, Andrew Whitehead, Mike Lodder

BBS Signatures Recap



A pairing based, multi message signature supporting selective disclosure and zero-knowledge proofs.

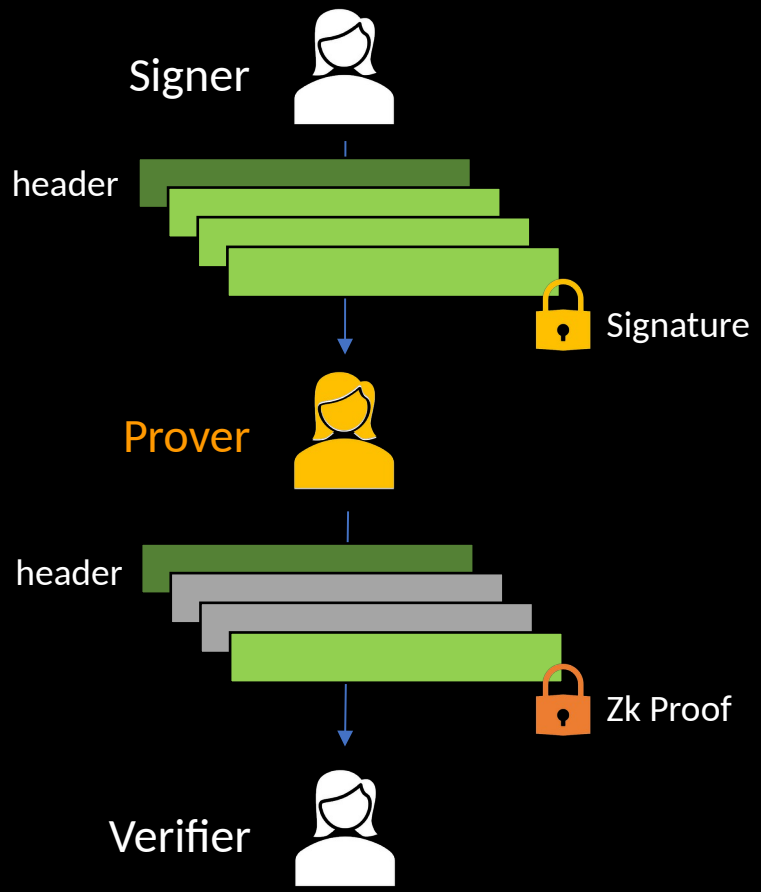
BBS Signatures Recap



Signer:

- Signs multiple messages.
- Can also choose a header.
- A header is a value that the Prover must always disclose.

BBS Signatures Recap



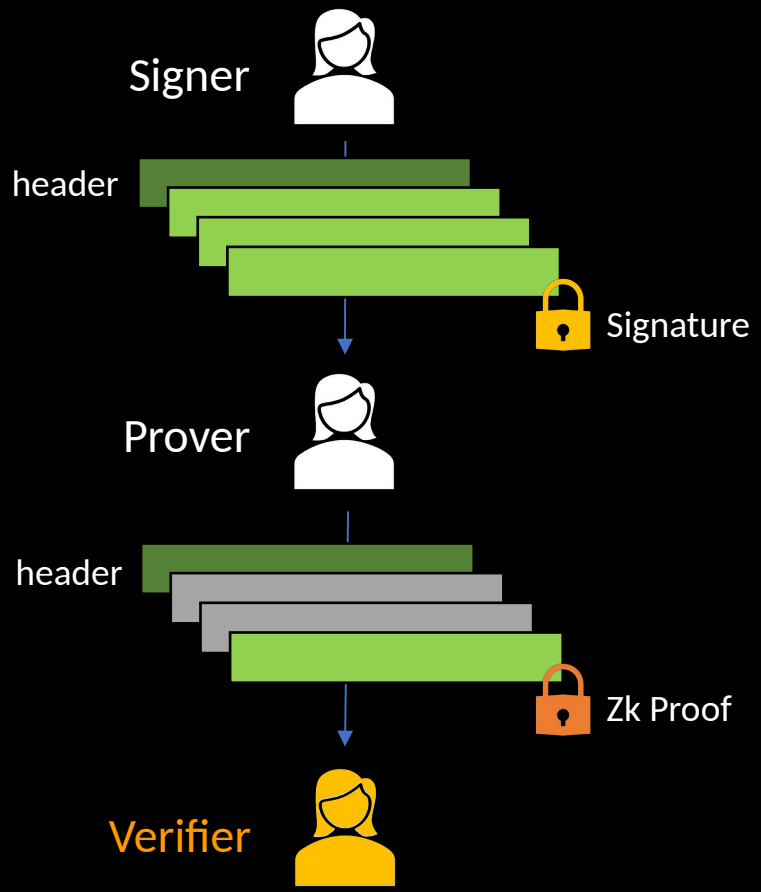
Signer:

- Signs multiple messages.
- Can also choose a header.
- A header is a value that the Prover must always disclose.

Prover:

- Chooses the messages to be disclosed.
- Create a zk proof of knowledge of a signature.

BBS Signatures Recap



Signer:

- Signs multiple messages.
- Can also choose a header.
- A header is a value that the Prover must always disclose.

Prover:

- Chooses the messages to be disclosed.
- Create a zk proof of knowledge of a signature.

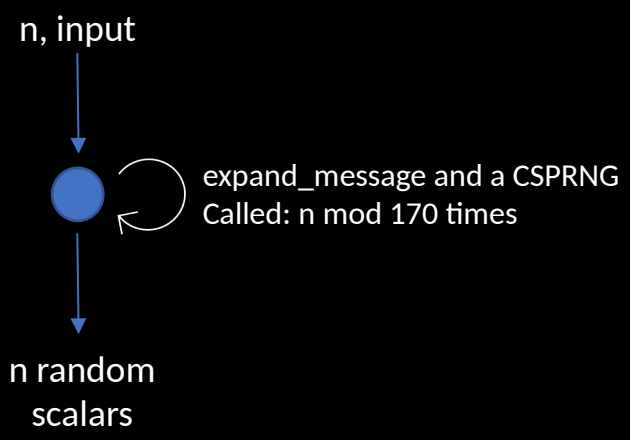
Verifier:

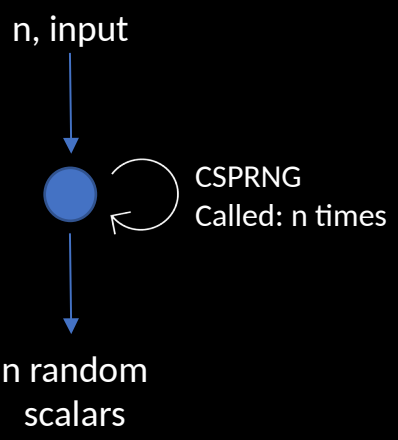
- Validates the proof using only the revealed messages and the Signer's Pk.
- Learns nothing other than that the Prover has a signature on a superset of the revealed messages.

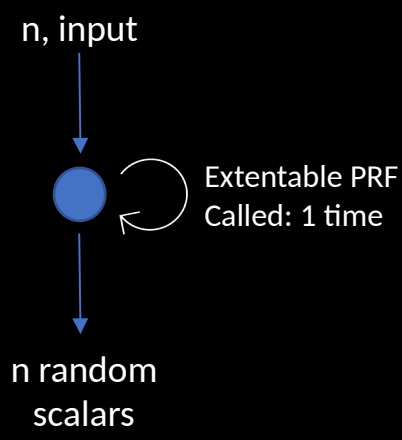
Updates: Random Scalars

The Issue?

- We used `expand_message` to extend a single seed to many pseudo random bytes.
- This imposed a limit of 170 signed messages.







The solution:

- Different approaches: use `expand_message` in a loop, use a CSPRNG in a loop or a combination of both.
- Using a CSPRNG in a loop was the fastest option, that does not introduce a new dependency.

Updates: Proof Test Vectors

The situation last IETF

Updates: Proof Test Vectors

- Signature Fixtures ✓
- Map to Scalar Fixtures ✓
- Generators Fixtures ✓
- Proof Fixtures ✓

The situation now!

Updates: Proof Test Vectors

What was the issue?

- The proof algorithm is randomized, needing multiple random scalars.
- Describing how to mock those random scalars in a way that would not be abused was not that easy.

The solution:

- Use a single PRF with extendable output.
- Expand a single seed to all the random scalars for the test vectors.

Updates: Proof Test Vectors

- We used `expand_message` from the Hash-to-Curve draft, since we already have a dependency on it.
- Even if someone uses this in production, as long as the seed is random, security holds
- New fixtures are cross validated by independent implementations.

Other Updates

Removed "total_number_of_messages" from a required input to ProofVerify.

Switched to use a CSPRNG instead of a PRF.

New academic paper on BBS will be presented in this year's EUROCRYPT. Includes simplifications and performance improvements.

Already made some changes in anticipation of that paper.

Needing Feedback

Negative test cases — We describe test cases that operations should fail.

- Complicate the document. Implementations ignore them.
- ? Should we remove them?

Needing Feedback

Negative test cases — We describe test cases that operations should fail.

- Complicate the document. Implementations ignore them.
- ? Should we remove them?

Map to Scalar — We use hash-to-scalar, to map messages (octets) to scalars.

- Not possible to combine with other protocols (i.e., range proofs or accumulators for revocation).
- ? Should an application or only a ciphersuite be able to define Map to Scalar operations?

  Is there any use for context aware encoding (i.e., dates encoded differently than IDs)?

Needing Feedback

Negative test cases — We describe test cases that operations should fail.

- Complicate the document. Implementations ignore them.
- ? Should we remove them?

Map to Scalar — We use hash-to-scalar, to map messages (octets) to scalars.

- Not possible to combine with other protocols (i.e., range proofs or accumulators for revocation).
- ? Should an application or only a ciphersuite be able to define Map to Scalar operations?

  Is there any use for context aware encoding (i.e., dates encoded differently than IDs)?

Point Encoding — We use the ZCash compact encoding of Bls12381 points.

- Curve specific.
- ? Should we use I-D.ietf-lwig-curve-representations compressed or uncompressed format?

Questions?