libZK: a zero knowledge proof library

(The Longfellow ZK scheme)

Matteo Frigo
abhi shelat
Tim Geoghegan
David Cook

CFRG - IETF 125 - Shenzhen
March 19 2026

Identifying the problem

∃ Desire for ZK schemes for “Small” Identity theorems

7.4.3.5.3 Zero-Knowledge Proofs

“Attestation Provider linkability cannot be fully eliminated when using attestation formats based on salted hashes. The only viable mitigation is to adopt Zero-Knowledge Proofs (ZKPs) as a verification mechanism instead of relying on salted-attribute hashes.”

…

“One key area of development is age verification, where the European Commission is actively exploring and testing ZKP-based solutions. The outcomes of this initiative could pave the way for the adoption of ZKPs within the EUDI Wallet ecosystem, further strengthening privacy protections in future implementations.”

https://eu-digital-identity-wallet.github.io/eudi-doc-architecture-and-reference-framework/2.4.0/architecture-and-reference-framework-main/#74353-zero-knowledge-proofs

7.1 Zero-Knowledge Proofs

A next version of the Technical Specifications for Age Verification Solutions will include as an experimental feature the Zero-Knowledge Proof (ZKP) solution described in the paper *"Matteo Frigo and abhi shelat, Anonymous credentials from ECDSA, Cryptology ePrint Archive, Paper 2024/2010, 2024, available at https://eprint.iacr.org/2024/2010"*. This zkSNARK-based solution was selected through a process where several alternatives were evaluated. This solution requires no changes to the attestation issuance process or to the structure of the Proof of Age credential itself. This backward compatibility allows AVIs to gracefully fall back to traditional protocols in environments where ZKPs are not supported.

https://ageverification.dev/Technical%20Specification/architecture-and-technical-specifications/#71-zero-knowledge-proofs

∃ Desire for ZK schemes in IETF

- SD-JWT/Oauth

“anecdotally I do think there is (or would be) general interest in [ZK] techniques [that are pq-secure] from a nontrivial number of folks in the community. So yes, I'd be supportive of such a statement as long as it was properly qualified as such.”

∃ Desire for PQ ZK schemes

Feedback from both the EU and organizations (e.g., Apple, Google) indicates a desire to *avoid* deploying new ECC-based schemes.

The preference is to choose PQ schemes when possible.

Other requirements such as *device-binding* and *legacy support* rule out schemes such as the BBS family.

There is an urgent market need to specify PQ ZK schemes for these applications.

Longfellow ZK scheme

- Follows the oldest ZK "recipe" from **BGGHKMR88**

Run an IP ⟶ Transcript.
Commit to Transcript ⟶ Com.
ZK for "Com contains T and IP-verifier(T)=1"

IP: Sumcheck (Prover runs in O(C) time.)

Commit/ZK: Ligero (IOP, only uses SHA256.)

Longfellow ZK scheme

- eprint/2024/2010
- Open source
- Only relies on SHA256 as hardness assumption.
- Best proof/work for 10k–50m gate theorems

What happened since 124

External interest

- ISRG has finished a 2nd implementation
  - Bit compatible

- EU Age verification
- EUDI wallets (wwWallet, Multipaz)

Security Reviews

3 security reviews have been completed.
(Trail of Bits, Ligero, ISRG)

No issues raised wrt to the ZK scheme.

Performance improvements

The scheme’s performance has improved by >20% due to algorithmic improvements in Reed-Solomon encoding and Sumcheck.

Protocol remains the same, but in comparison with other alternatives.

Example:
Proof of Possession of ECDSA signature

“Exists (r,s) such that Verify(pk, H(m), r, s) = 1.”

BM_ECDSAZKProver/1	16.7 ms	42
BM_ECDSAZKProver/2	26.5 ms	27
BM_ECDSAZKProver/3	38.3 ms	18
BM_ECDSAZKVerifier/1	10.3 ms	67
BM_ECDSAZKVerifier/2	16.0 ms	44
BM_ECDSAZKVerifier/3	23.4 ms	31

(Mac M4). These times are close to BBS signature proof of possession.

Example:
Proof of Possession of ML-DSA-44 Sig

```
“Exists (r,s) such that
Verify_Internal( pk=(Ahat,t1,tr) sig, mu) = 1.”
```

```
BM_MLDSA44SigProver/1          858 ms                 8
```

(Mac M4).

Example:
PQ Bitcoin Address

“Exists (x) such that Ripemd(sha256(compress(G * x))) = A.”

BM_BitaddrProver/1	87 ms	8
BM_BitaddrVerifier/1	59 ms	12

(Mac M4).

Comparison with other recent techniques

Vega: Uses folding, which requires a homomorphic commitment (specialized ECC curve that is not pq-secure), and its security reduction offers no extraction soundness for P256 due to losses from multi-round rewinding.

Appendix

Slides to describe the scheme.

Code Complexity

Ligero + Merkle

Language	files	comment	code
Header	6	252	851
C++	2	11	275
SUM:	8	263	1126

Sumcheck

Language	files	comment	code
Header	8	159	795
C++	2	45	342
SUM:	10	204	1137

ZK

Language	comment	code
Header	232	806
C++	27	292
SUM:	259	1098

bls12_381: 11,299 code lines, 1401 comments
Whir + merkle: 3385 lines; whir/sumcheck: 916 lines