# Kohaku Wallet: Post-Quantum Smart Accounts on Ethereum Today

**Simon Masson**, Renaud Dubois
ZKNox

Privacy and Compliance Summit

November 18th, 2025 – Buenos Aires, DevConnect

# ZKNOX Casting

**Nicolas Bacca, "Chief"**
$20^+$ years experience ($10^+$y web3)
Security and hardware specialist
Prev. Ledger cofounder/CTO

**Renaud Dubois, "Agent Smith"**
$20^+$ years experience ($3^+$y web3)
Cryptographer
Prev. Ledger, Thales

**Simon Masson, "Bizut"**
$8^+$ years experience ($4^+$y web3)
Cryptographer
Prev. Heliax, Thales

# ZKNOX Casting

**Nicolas Bacca, "Chief"**
$20^+$ years experience ($10^+$y web3)
Security and hardware specialist
Prev. Ledger cofounder/CTO

**Renaud Dubois, "Agent Smith"**
$20^+$ years experience ($3^+$y web3)
Cryptographer
Prev. Ledger, Thales

**Simon Masson, "Bizut"**
$8^+$ years experience ($4^+$y web3)
Cryptographer
Prev. Heliax, Thales

Expertise and innovation to every challenge on the whole security chain:

- user end
  (secure enclaves,
  hardware wallets),
- back end
  (TEE, HSMs),
- on-chain
  (smart contracts).

# ZKNOX Casting

**Nicolas Bacca, "Chief"**
$20^+$ years experience ($10^+$y web3)
Security and hardware specialist
Prev. Ledger cofounder/CTO

**Renaud Dubois, "Agent Smith"**
$20^+$ years experience ($3^+$y web3)
Cryptographer
Prev. Ledger, Thales

**Simon Masson, "Bizut"**
$8^+$ years experience ($4^+$y web3)
Cryptographer
Prev. Heliax, Thales

Expertise and innovation to every challenge on the whole security chain:

- user end
  (secure enclaves,
  hardware wallets),
- back end
  (TEE, HSMs),
- on-chain
  (smart contracts).

`https://zknox.eth.limo/`

`https://github.com/zknoxhq/`

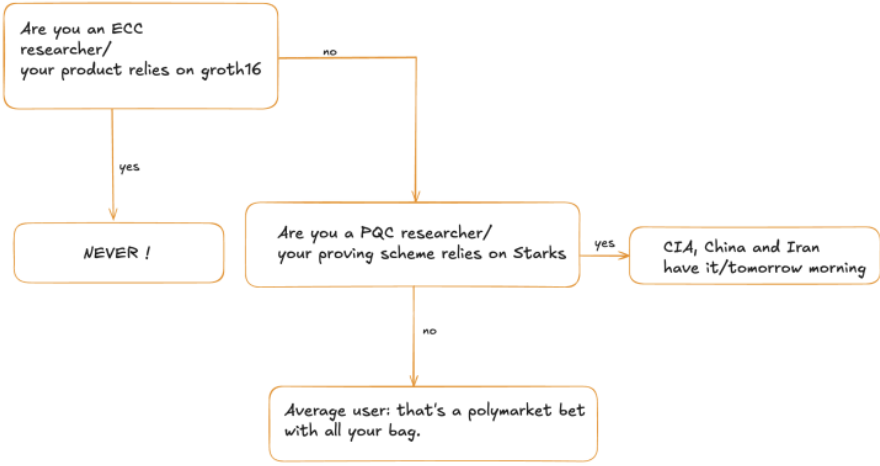
# Summary

# Introduction: Quantum Apocalypse

Shor algorithm solves factorization and discrete logarithm problems. All current authentication systems are cooked as soon as Quantum computer rises.

# When ?

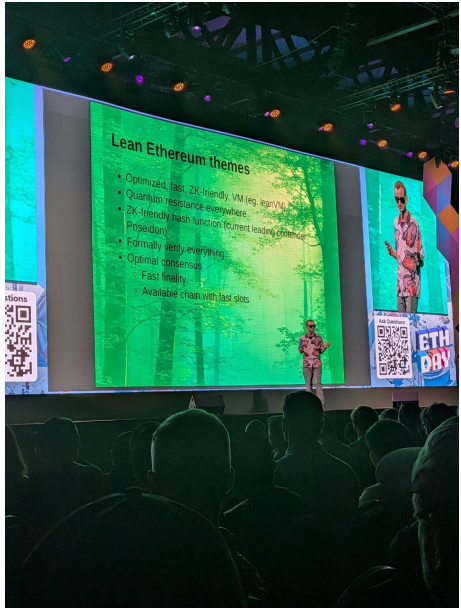
# When ?


Are you an ECC researcher/ your product relies on groth16
no
yes
NEVER !
Are you a PQC researcher/ your proving scheme relies on Starks
yes
CIA, China and Iran have it/tomorrow morning
no
Average user: that's a polymarket bet with all your bag.

# When ?

Federal agencies:
NIST: 2035 ANSSI/BSI: 2030
Blockchains might not care about feds, but for stablecoins regulation, those might be hard deadlines.

# EF got your back



# Candidates

| Complexity metric | FALCON512 | DILITHIUM2 | SPHINCS+ | GemSS |
|---|---|---|---|---|
| Public Key size | 897 | 1312 | 32 | 33 |
| Private Key size | 7553 | 2528 | 32 | 64 |
| Signature size | 666 | 2420 | 17088 | 21952 |

Current proposals:

- FALCON: EIP8052/EIP7619
- Dilithium: EIP8051

Difference for 8052/7619: zk friendly hash (separate core from hashing to domain), 7932 specification

8051 and 8052 comes with contracts, signers, and hardware signer (8051 only). (Wait for next talk for the onchain demonstration).

# Advertisement

All is delivered as public good, experiment, integrate, give feedback, hack this week end (ETHGLOBAL).


A PQ-vault, staking ETH (gas cost is high, better suited for high amount with few movements)

# Remarks

- Authentication might be solved a bit later, (but we shall on the shelf solution).
- Confidentiality shall be solved NOW.

Ethereum components at risk:

1. EoA private keys (notably using ECDSA)
2. Private Payments (Privacy Pool, TC, sRAILGUN)
3. BLS signatures in consensus (lean CL)
4. Data Availability Sampling (leveraging KZG commitments)

ZKNOX provides solution for first point, and experimentations results for 2 and 3.

# Progressive Roadmap

Verifiers:

- Step1: use Account Abstraction (EIP-7702+7579/4337) with full solidity to experiment
- Step2: benchmark in nodes (validate gas hypothesis)
- Step3: EIP accepted
- remove eoA (EIP-7701/EIP-7560)

Signers:

- Step1: Software signers
- Step2: hardware signers
- remove eoA (EIP-7701/EIP-7560)

# Verifiers: implementation results

| Function | Description | Gas Cost |
|---|---|---|
| TETRATION ethfalcon.verify | EVM Friendly* | 24M |
| ZKNOX falcon.verify | NIST | 7M |
| ZKNOX ethfalcon.verify | EVM Friendly | 1.8 M |
| ZKNOX epervier.verify | Recover EVM friendly | 1.9M |

| Function | Description | Gas Cost |
|---|---|---|
| ZKNOX dilithium.verify | NIST | 13.3M |
| ZKNOX ethfalcon.verify | EVM Friendly | 6 M |

# Signers implementation

| Scheme | Operation | RAM Consumption | Code size |
|---|---|---|---|
| FALCON | Keygen | 40 Kb | 30-50 Kb |
| | Signing | 40 Kb | 30-50 |
| Dilithium | Keygen | 7 Kb | 30-50 Kb |
| | Signing | 7 Kb | 30-50 |

Low Ram implementation of Dilithium is possible, falcon reach the limit for a SE implementation.

Next experiment: solve RAM limitation with ORAM (extend RAM with external ciphering on host). Part of another ZKNOX implication in Kohaku: universal Ethereum Application.

# Proving FALCON and DILITHIUM

Need

- Allow FALCON/DILITHIUM integration into ZKEVM
- Allow PQ-Private Payments (PQ-Railgun)
- Solution for BLS replacement is Batching with snarks/starks.

Second one could use a dedicated scheme (LABRADOR, Lean CL team work).

- FALCON and DILITHIUM operates on non native fields (Babybear, STwo)
- Non Native Fields require x30 more constraints (source: Bandersnatch vs P256 in Gnark, https://eprint.iacr.org/2025/933)
- Solution for BLS replacement is Batching with snarks/starks.

# ZK friendly FALCON/DILITHIUM

Obvious tweaks

- Replace Hash function by a ZK friendly one
- Provide hints (see epervier definition) for easy batch inversion
- Accumulate several NTT steps, reduce once (Credit to Zhenfei)

Harder tweaks

- Starks fields are going shorter (no accumulation trick) Replacing fields is less trivial than for ECC (pick prime order curve and twist)
- FALCON is prone to overstretch attacks for ZK fields
- Estimator for S2/Babybear on dilithium provides 20factor (WIP)

# Results

| Function | Description | gas cost | Max per block |
|---|---|---|---|
| ECDSA | Starkware built in | - | 3000 |
| ETH_FALCON | ZKnox Use Keccak-ctr | 102M | 10 |
| ETH_FALCON | Starkware+ ZKNOX Keccak-ctr | 340M 3 | |
| FALZKON Use | Blake2s-ctr | 41M | 25 |

Result for a high level Cairo, dedicated AIR shall dramatically reduce this factor.

Dilithium requires less hashing and more easily tuned : better zk candidate.

https://github.com/ZKNoxHQ/falzkon

# Conclusion

Easy

- Easy solutions for EOA migration (use AA+ 8051/8052)
- 8051 and 8052 proposals, help us with ACD
- PQ ciphering (view keys) is easy (just more storage cost)

FALCON is greater for plain constraints, less for ZK/MPC/signers constraints.

Hard

- no succinct replacement of pairings (consensus + private Payments systems)
- almost probably no constant size PQ replacement

Addiction to 0(1) bandwidth will strongly strike those part of the protocol.