One ECC library to rule them all

ETHcc 2024

Renaud Dubois

Smoo.th

Make digital ownership accessible

July, 9, 2024

Smoo.th

Nicolas Bacca
20+ years' experience (10y+ web3)

Security & Hardware specialist
@Ledger Co-founder/CTO

Renaud Dubois
20+ years' experience (2y+ web3)

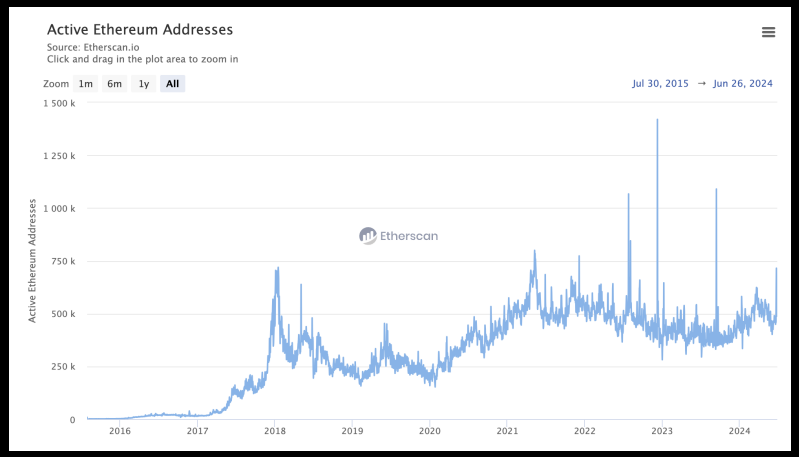
Cryptographer
@Ledger @Thales

Jonathan Giamporcaro (CEO)
7+ years' experience (7y web3)

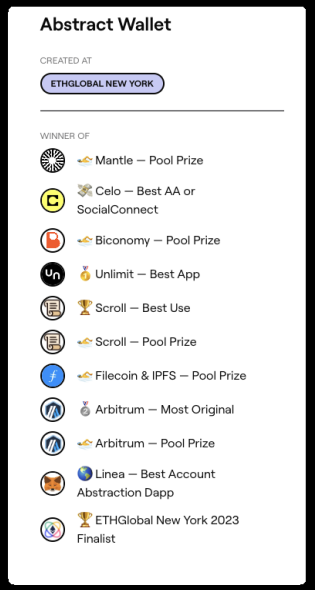
Full-Stack Software Engineer
@Ledger

Alexandre Lemarchand
25+ years' experience (6y+ web3)

Sales, Partnership and Business Dev
@Ledger @Microsoft



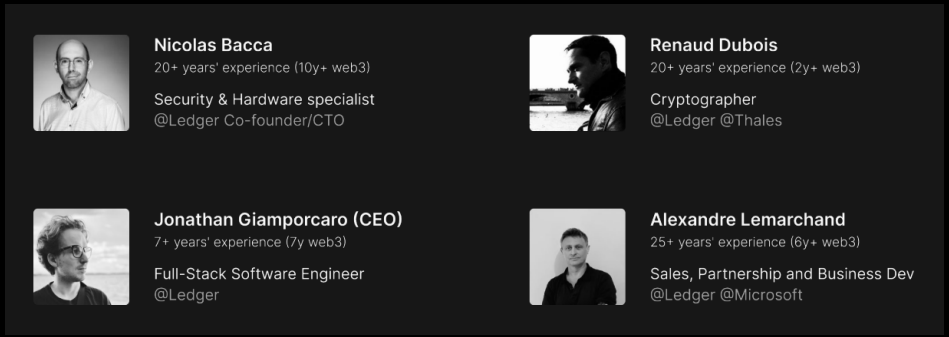
Smoo.th



Last year we delivered FCL, the fastest Cairo0 & **solidity** implementation of Passkeys. (From 1M to 60/220K).

- CoinBase SmartWallet
- Safe passkeys module
- Cometh Kit

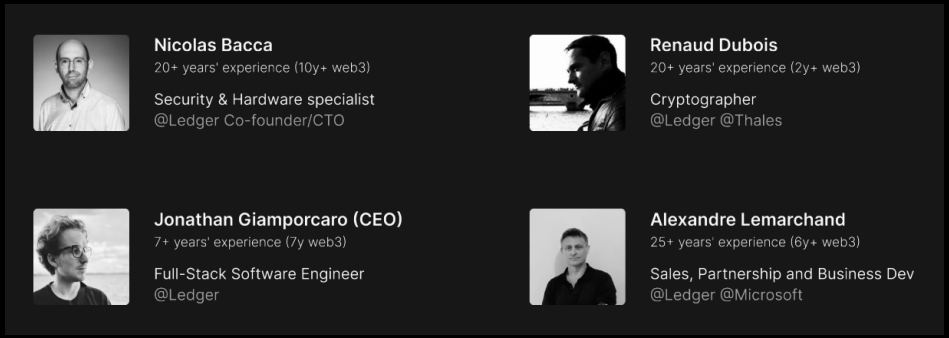
Smoo.th

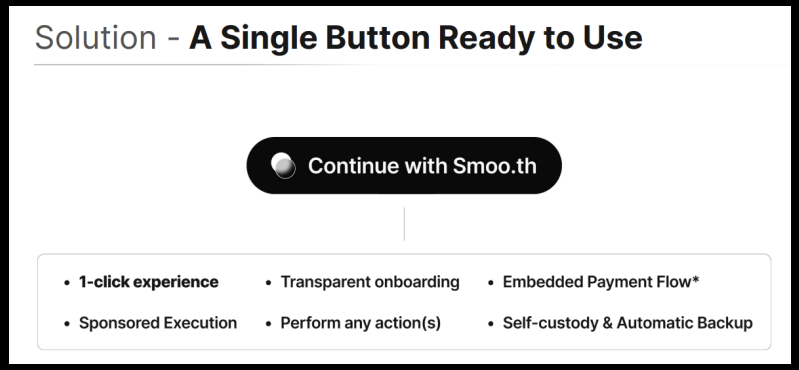


Stan's Granma doesn't want to know what is gas, or even ETH.
Today we deliver

- SCL : even faster and **generic** ECC and propose RIP7696.
- Smoo.th : The paypal of Web3

Smoo.th






```
import { Smooth } from "@smooth/kit";

<Smooth
  business-id="6pRNASCoBOKtIshFeQd4XMuh"
  action={{ type: "execute", chainId: "0x", target: "0x", calldata: "0x" }}
/>
```

Smoo.th, 5 lines to integrate in your App DX : no blockchain knowledge required.

ECC implementation

ECC

- Elliptic curve cryptography (ECC) enables signing and key exchange in modern communications.
- ECDSA over P256 secures our daily lives with our TLS exchanges, some Passports; Intel SGX, SSH and passkeys (TouchID/FaceID).
- there are several other curves and applications.
- RIP7212/current implementations only cover ECDSA over P256. This is far from what ECC can bring.

P256 is not the only curve of FIDO2/Passkey specification.
ECDSA sucks, bad legacy (covert channel, misuse weak, MPC/ZK hard).

Use cases

SGX 2FA settlement

Use 2 proofs to advance the on-chain zk-rollup state root:

1. cryptographic proof (STARK/SNARK)
2. 2FA: Additional SGX proof

Intel SGX **switched** from a custom BN curves to P256. Live on scroll and Taiko.

Use cases

SGX 2FA settlement

Use 2 proofs to advance the on-chain zk-rollup state root:

1. cryptographic proof (STARK/SNARK)
2. 2FA: Additional SGX proof

Intel SGX **switched** from a custom BN curves to P256. Live on scroll and Taiko.

(Dudes, come see us for Precomp. version, save 60% of settlement cost)

Use cases

Schnorr Use Cases

- Strong and easy MPC/TSS (Musig2/Frost)
- Lightning/plasma
- Stealth signatures
- ZK friendly (Jujub)
- Half aggregation
- EDdsa is deterministic by definition

Ed25519 is schnorr and part of FIDO/Passkeys spec.

ECC in EIPs

Zoo

- secp256r1 (RIP/EIP7212)
- ed25519 (EIP665), ed25519 $>$ secp256r1
  - not NIST, faster, schnorr (MPC/ZK friendly)
  - farcaster, SGX, IBC
- BN254 (EIP1962)
- BLS12381-G1 (EIP 2537)
- BLS12377 (EIP2539)
- Palla/Vesta
- Jujub

ECC in EIPs

Zoo

- secp256r1 (RIP/EIP7212)
- ed25519 (EIP665), ed25519 $>$ secp256r1
  - not NIST, faster, schnorr (MPC/ZK friendly)
  - farcaster, SGX, IBC
- BN254 (EIP1962)
- BLS12381-G1 (EIP 2537)
- BLS12377 (EIP2539)
- Palla/Vesta
- Jujub

RIP7696: Go for Generic Double Scalar Multiplication (95% of computations).

ECC in EIPs

Zoo

- secp256r1 (RIP/EIP7212)
- ed25519 (EIP665), ed25519 $>$ secp256r1
  - not NIST, faster, schnorr (MPC/ZK friendly)
  - farcaster, SGX, IBC
- BN254 (EIP1962)
- BLS12381-G1 (EIP 2537)
- BLS12377 (EIP2539)
- Palla/Vesta
- Jujub

What is the overcost of genericity ?

Solidity implementations

Library	ecaddN (gas)	ecDbl (gas)	ecmulmul (gas)	Prec. Bytes
orbs-network	2250	1750	1.06M	0
Androlo	2073	1229	866K	0
Maxrobot	1949	1502	760K	0
Numerology	1973	1003	422K	0
alembich-tech	2250	1750	335K	3.2MB
itsobvioustech	946	578	290K	0
Ours(1)	566	522	**202K**	0M
Ours(3)			**61.6 K**	3.2MB

New Results : performances

SCL/RIP7696

Two functions/opcodes:

- takes curves parameters as input (genericity)
- opcode 1 : 2MSM+windowing
- opcode 2 : 4MSM

Library	Number of bases	ecdbl	ecadd	full ecdsa
FCL	2	256	192	221 K
	8	64	64	81 K
SCL	2	256	128	202 K
	4	128	128	180 K

By adding a single point in verification API, even faster than FCL. The asymptotic gain, which ZK and nodes (RIP) implementations are closer to 33%.

New Results : implementation insight

- Double always, Add when ‘1’

  5P (b101) : P->2P->4P+P

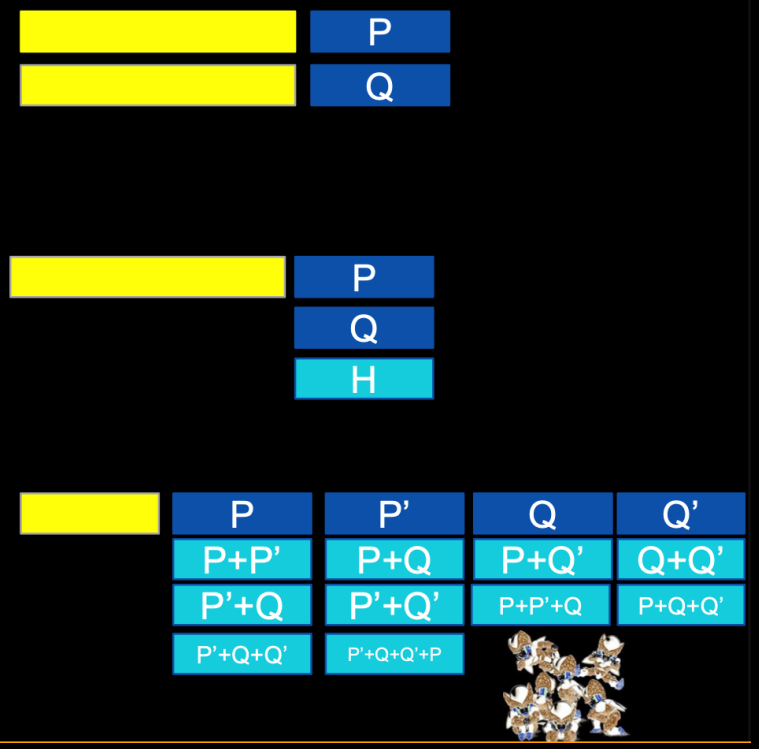
  7Q : (b111): Q->2Q+Q=3Q, 3Q+Q=7Q

- Strauss Shamir : mutualize doubling, compute H=P+Q

  101 , (P+Q), 2(P+Q)+Q, 2(2P+3Q))+P+Q

  111

- Higher dimension : choose $2^{(n/2)}P$ and $2^{(n/2)}Q$ as extra points



New Results (genericity)

Integrated

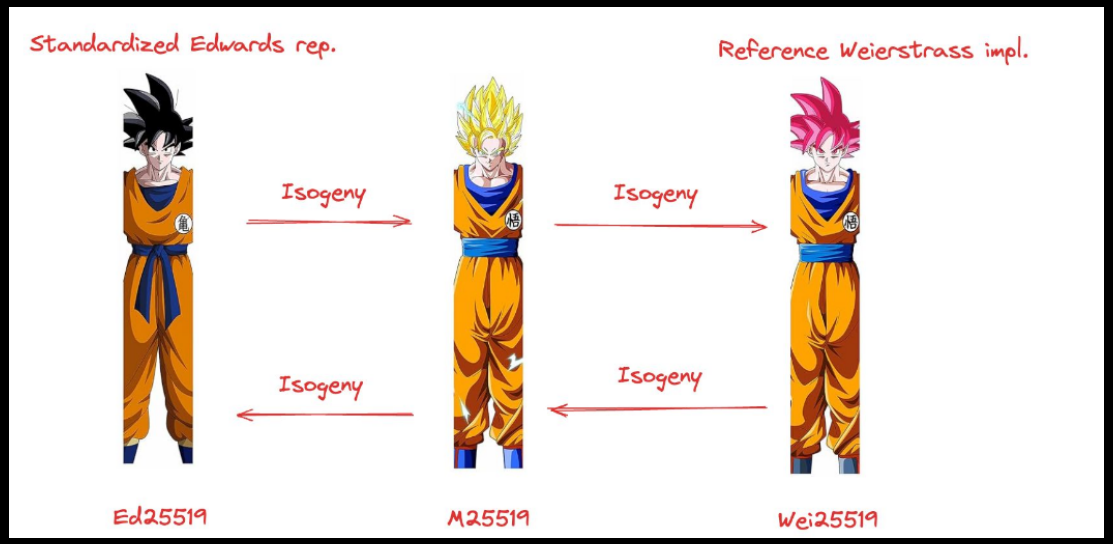
- P256
- Ed25519, using isogenies
- Jujub (missing isogeny rn)

Ongoing

Two functions/opcodes:

- **Starkcurve**
- Palla, Vesta

New Results (genericity)



Isogenies

- An isogeny between two elliptic curves and is a morphism of curves that sends the origin of E1 to the origin of E2.
- Convert point from edwards from/to Weierstrass (Jujub/ed25519)
- Negligible cost compared to whole Multiplication.

Conclusions

- SCL/RIP7696 provides a future proof/agile implementation for same/better cost
- Going through two audits

- Happy to help to integrate our CryptoLib to ZKEVM or Client
- Defy Dapps, come to us for instant onboarding, chain agnostic.

Questions ?

SCL

Telegram

Choisir c'est mourir un peu.

-André Gide

(To choose is die a little.)