RIP7696: Generic Double Scalar Multiplication (DSM)

RollCall, Armistice Day, 2024

Renaud Dubois

Smoo.th Innovation Lab

Context

RIP7212

Use cases

- passkeys over P256 (RIP7212)
- SGX

RIP7212 Limitations

Restricted to **non recovery** version over P256

- Non transparent curve
- Covert Channel
- Not MPC/ZK friendly as Schnorr
- No ecrecover-like hacky mul possible.
- Slower than k1 (gas cost issues)

Context

RIP7696

Generic Double Scalar Multiplication

- All RIP7212 (7696 emulates 7212)
- Ed25519 : transparent, no covert channel, MPC friendly
- General Protocols: Ring (Monero Like), Stealth, Pala, Vesta, Babyjujub
- Bridges with L2s Starknet, Cosmos, Solana

Impact

Only require a generic Montgomery Multiplier/Modular Operator (mulmod). Impact in term of

- Code complexity,
- Performances (Solidity, General Pseudo Code).

Specfication

Two opcodes proposed

- `ecmulmuladd(p,a,b,n, Gx,Gy,Qx,Qy)` is classical DSM (2 bases MSM)
- `ecmulmuladdB4(p,a,b,n, Gx,Gy,Qx,Qy,G128x,G128y,Q128x,Q128y)` is optimized DSM (4 bases MSM)

where offchain computation $P128 = 2^128.P, Q128 = 2^128.Q$ enables GLV like performances.

Those optimizations are independant of language.

Solidity Implementation

2023 Results

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
Daimo			330K	0
FCL(1)	566	522	**202K**	0M
FCL(3)			**61.6 K**	3.2MB

Table: 2023 Results

On chain benchmarks:

https://goerli.basescan.org/address/0x936632cC3B9BC47ad23D41dC7cc200015c447f71

Solidity Implementation

2024 Results
(Asset code of RIP7696)

Library	ecaddN (gas)	ecDbl (gas)	ecmulmul (gas)	Prec. Bytes
FCL(1)	566	522	**202K**	0M
FCL(3)			**61.6 K**	3.2MB
SCL(a)			**185 K**	0M
SCL(b)			**201 K**	0M
SCL(c)			**160 K**	128

Table: 2023 Results

(a) new RIP7212 compatible specific by SCL
(b) generic (p,a,b) as input (*)
(c) Same + P' and Q' as calldata (*)

(*) This work is partially granted by EF Grant (independantly of RIP)

Insight for Circuit

RIP7696 requires a Modular multiplier (`mulmod`).
In practice, Nodes and ZKVM implements Montgomery multiplication toMontgomery MontgoMul FromMontgomery
Implemented in gnark-crypto, openSSL, but specific code used instead for each curve. Second opcode restores comparable performances as using k1.

Conclusion

Choisir c'est mourir un peu.
-André Gide

(To choose is die a little.)