Evolution of recursive snarks

Oleg Taraskin, Waves

tog.postquant@gmail.com

Zero Knowledge Proofs

- Completeness
- Soundness
- Zero-knowledge

zkSNARK

- zero knowledge Succinct Non-Interactive Argument of Knowledge

$C(x, w) = 0$

$h = SHA\text{-}256(m)$

The most widely used snarks

- Groth16
- STARK
- Plonk (and its modifications)
- Halo2

Pairing Friendly Curves

$E: \ y^2 = x^3 + ax + b$ over field F_p

q $-$ order of prime subgroup of E

Embedding degree with respect to q : minimal k : q divides $p^d - 1$

Pairing-friendly if d is small:

Example: curve BLS12-381 has degree = 12

Pairings

Bilinear map e : $e(G_{1,}\ G_2) \rightarrow G_t$

1. Bilinearity:

for any S from $G_{1,}$ T from G_2 and integers a and b

$$e(aS, bT) = e(S, T)^{ab}$$

2. Non-degeneracy:

for any S from $G_{1,}$ $e(S, T) = 1$ iff $T = 0$

for any T from $G_{2,}$ $e(S, T) = 1$ iff $S = 0$,

where O – point at infinity

Pairing Friendly Curves security

Depends both on

1. hardness of solving of ECDLP:

$$A = xB$$

2. hardness of DLP problem in the field $GF(p^k)$:

$$a^x = b$$

for known a, b from $GF(p^k)$ and unknown natural x

k is embedding degree

Best known method of breaking DLP is called Number Field Sieve (NFS)

Chains of elliptic curves

$E_1: y^2 = x^3 + ax + b$ over F_p

q - order of prime subgroup G_1 of E_1

$E_2: y^2 = x^3 + a'x + b'$ over F_r

p - order of prime subgroup G_2 of E_2

E_1 and E_2 are curves of chain of length 2

Next curve must have order of G equal to field of previous curve

Cycles of elliptic curves

$E_1 : y^2 = x^3 + ax + b$ over F_p

q - order of prime subgroup G_1 of E_1

$E_2 : y^2 = x^3 + a'x + b'$ over F_q

p - order of prime subgroup G_2 of E_2

E_1 and E_2 are curves of cycle of length 2

Some cyclic pairing-friendly curves

- Curves MNT 4 and MNT 6 form a cycle
- Length of field characteristic is 753 bits !!
- Solving down ~ 10 times ☹

Recursive proof

Circuit C :

$$C(w, x) = 0$$

w – witness, x – public

vk – verification key

1. “Internal” Prover:

proves $C(w, x) = 0$: creates proof π_{inner}, so all can check it by running

$$Verify_{int}(vk, x, \pi_{int}) = 1$$

2. “External” Prover:

proves circuit $Verify_{int}(vk, x, \pi_{int}) - 1 = 0$

using π_{int} as witness

so all can check it by running

$$Verify_{ext}(vk, x, \pi_{ext}) = 1$$

Use cases of snark recursion

- Compression of proof
- zkRollups
- IVC incremental verifiable computing

Verification in Groth16

$$e(A, B) = e(\alpha G, \beta H) * e\left(\sum_{j=0}^{t} a_j S_j, \gamma H\right) * e(C, \gamma H)$$

proof verification key public inputs

Plonk

- Uses KZG commitment that need pairing-friendly curves
- As result Plonk has the same problems as groth16 with recursion

Solution

Use another polynomial commitments, such that don't use pairings:

FRI (Fast Reed-Solomon Interactive oracle proofs)

Inner Product Argument

Cycle curves

- Pasta curves (Pallas and Vesta)

$y^2 = x^3 + 5$ over F_p

Pallas curve:

$p =$ 0x40000000000000000000000000000000224698F
C094CF91B992D30ED00000001

Vesta curve:

$p =$ 0x40000000000000000000000000000000224698F
C0994A8DD8C46EB2100000001

Inner Product Argument

Prover

Verifier

C = Commitment(p(x))

C

x

calc. v = p(x)

and proof π

v, π

check v, π

and know is p(x) = v ?

Pedersen commitment

G – vector of n group generators

U – generator

p – vector of n coeff. of p(x)

C = < **G, p** > - commitment of p(x)

Proof of v = f(x) :

$$\pi = \{\mathbf{L}, \mathbf{R}, G^{(0)}, p^{(0)}\}$$

L, **R –** vectors of length k = $log_2 n$

Inner Product Argument

Prover

Verifier

b – vector $\{1, x, \dots, x^{n-1}\}$

v = <**p, b**>

$C^{(k)} = C + \text{vU}$

Round k

u_k

$\boldsymbol{p}^{(k-1)} = u_k \boldsymbol{p}_{lo}^{k} + u_k^{-1} \boldsymbol{p}_{hi}^{k}$

$\boldsymbol{b}^{(k-1)} = u_k^{-1} \boldsymbol{b}_{lo}^{k} + u_k \boldsymbol{b}_{hi}^{k}$

$\boldsymbol{G}^{(k-1)} = u_k^{-1} \boldsymbol{G}_{lo}^{k} + u_k \boldsymbol{G}_{hi}^{k}$

Inner Product Argument

$$C^{(k)} = C + \text{vU}$$

↓

$$C^{(k-1)} = < \boldsymbol{p}^{(k-1)}, \boldsymbol{G}^{(k-1)} > + < \boldsymbol{p}^{(k-1)}, \boldsymbol{b}^{(k-1)} > \text{U}$$

$$C^{(k-2)} = < \boldsymbol{p}^{(k-2)}, \boldsymbol{G}^{(k-2)} > + < \boldsymbol{p}^{(k-2)}, \boldsymbol{b}^{(k-2)} > \text{U}$$

$$C^{(k-2)} = C^{(k-1)} + u_{k-2} L^{(k-1)} + u_{k-2}^{-2} R^{(k-1)}$$

Inner Product Argument

Verifier :

$b_0 = < \boldsymbol{b} , \boldsymbol{s} >$

Check $\boldsymbol{G}^{(0)} = < \boldsymbol{s}, \boldsymbol{G} >$

$$
\mathbf{s} = \begin{pmatrix}
u_1^{-1} & u_2^{-1} & \cdots & u_k^{-1}, \\
u_1 & u_2^{-1} & \cdots & u_k^{-1}, \\
u_1^{-1} & u_2 & \cdots & u_k^{-1}, \\
u_1 & u_2 & \cdots & u_k^{-1}, \\
 & & \vdots & \\
u_1 & u_2 & \cdots & u_k
\end{pmatrix}
$$