PLONK

→ Groth16 gives proof of constant size (~ 200 bytes)

→ The IOP assumes that the crypto compiler uses bilinear pairings. (Non Modular) - Groth16

→ Plonk is Modular ⇒ you can mix and match the IOP with various other crypto compilers.

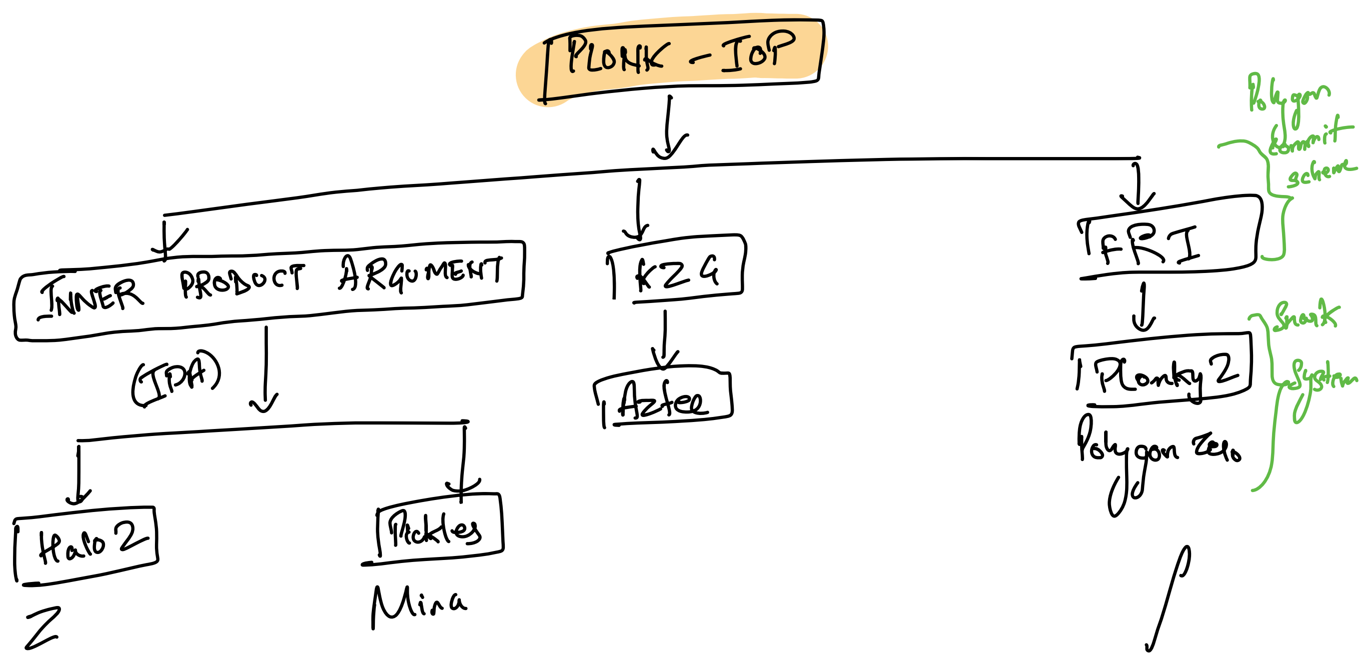
→ Arithmetization

  → Uses RAP (Randomized PAIR ⇒ Preprocessed Algebraic Intermediate Representation)

  → uses custom gates (non linear also possible)

Modularity

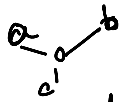
Plonk in practice is an abstract IOP and can be combined with various PCS schemes



Plonk Arithmetization

Plonk Arithmetization → Local constraints
Plonk Arithmetization → Global constraints

$a + b = c$ $\qquad$ $a * b = c$



Selector values

$\boxed{q_L, q_R, q_O, q_M, q_C}$

$$(q_L)a + (q_R)b + (q_O)c + (q_M)a \cdot b + q_C = 0$$

$a + b - c = 0 \Rightarrow (1)a + (1)b + (-1)c + (0)a \cdot b + (0) = 0$

$a \times b - c = 0 \Rightarrow (0)a + (0)b + (-1)c + (1)a \cdot b + (0) = 0$

$a = 42 \Rightarrow (1)a + (0)b + (0)c + (0)a \cdot b + (-42) = 0$

For Ex $a_1 + b_1 = c_1$

$a_2 \times b_2 = c_2$

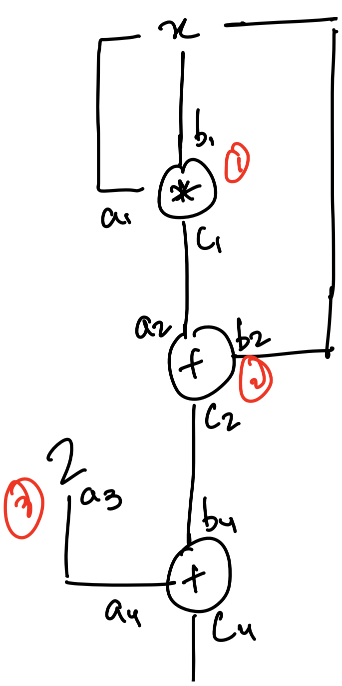
and we want $a_2 = c_1$

Here we are not assigning: this as we cannot link them without using global variables

★ Even constants have a gate

PEN & PAPER EX

$x^2 + x + 2 = 32$ $\quad$ $(x = 5)$



	q_L	q_R	q_O	q_M	q_C
$\overset{5}{a_1} * \overset{5}{b_1} = \overset{25}{c_1}$:	0	0	-1	1	0
$\overset{25}{a_2} + \overset{5}{b_2} = \overset{30}{c_2}$:	1	1	-1	0	0
$\overset{2}{a_3} = 2$:	1	0	0	0	-2
$\overset{2}{a_4} + \overset{30}{b_4} = 32$:	1	1	-1	0	-32
	$q_L(x)$	$q_R(x)$	$q_O(x)$	$q_M(x)$	$q_C(x)$

$\vec{a} = [5, 25, 2, 2]$

$\vec{b} = [5, 5, 0, 30]$

$\vec{c} = [25, 30, 0, 0]$

|

32

$c = [25, 30, 0, 0]$

$\vec{a} = [5, 25, 2, 2]$

$\vec{b} = [5, 5, 0, 30]$

$\vec{c} = [25, 30, 0, 0]$

q_L	q_R	q_O	q_M	q_C
0	0	-1	1	0
1	1	-1	0	0
1	0	0	0	-2
1	1	-1	0	-32

$f_a(x)$

$f_b(x)$

$f_c(x)$

$q_L(x)$

$q_R(x)$

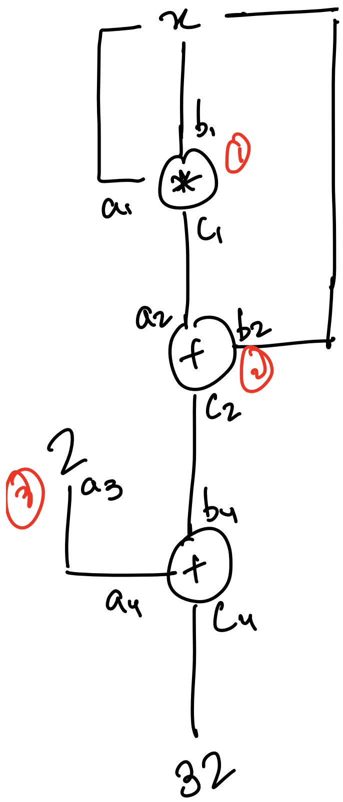
$q_M(x)$

$q_C(x)$

Domain	Value
1	a_1
2	a_2
3	a_3
4	a_4

Poly $f_a(x)$

n



$a_1 = b_1$

$c_1 = a_2$

$c_2 = b_4$

$a_3 = a_4$

	A	B
a_1	3	5
a_2	12	12
a_3	4	9
a_4	9	4
a_5	5	3

Accum: $acc(0) = 1$ $\quad acc_i(\ ) = acc(i-1) \times \frac{A_i}{B_i}$

$$1 \times \frac{\cancel{3}}{\cancel{5}} \times \frac{\cancel{12}}{\cancel{12}} \times \frac{\cancel{4}}{\cancel{9}} \times \frac{\cancel{9}}{\cancel{4}} \times \frac{\cancel{5}}{\cancel{3}} = 1$$

1 5 3

1 15 1

$\frac{\cancel{4}}{\cancel{8}} \quad \frac{1}{1} \times \frac{\cancel{5}}{\cancel{15}} \times \frac{\cancel{3}}{\cancel{1}}$

$$acc(i) = acc(i-1) * \frac{r + A_i}{r + B_i}$$

$$\boxed{p(x) = \prod_{i=1}^{n} \frac{r + a(i)}{r + b(i)}} = 1$$

These will be roots of unity

	a_1	a_2	a_3	a_4	a_5
$A =$	3	12	4	9	5

$A = [3, 12, 4, 9, 5] \to a_1 = a_5$

$B = [5, 12, 9, 4, 3]$

$a_2 = a_2$

$a_3 = a_4$

A's index $\leftarrow i = [1, 2, 3, 4, 5]$

$\sigma(i) = [5, 2, 4, 3, 1]$

permutation index OR B's index

$$(A, i) \quad , (B, \sigma(i))$$

(3,1)	(5,5)
(12,2)	(12,2)
(4,3)	(9,4)
(9,4)	(4,3)
(5,5)	(3,1)

$$acc_i = acc_{i-1} * \left(\frac{r + A_i + \overset{\text{const.}}{\beta} \cdot i}{r + B_i + \beta \cdot \sigma(i)} \right)$$

$$= \frac{r + 3 + \beta 1}{r + 5 \times \beta 5} \times \frac{r + 5 + \beta 5}{}$$

$a_1 = b_1$

$a_2 = c_1$

$a_3 = a_4$

$b_1 = b_2$

$b_2 = a_1$

$b_3 = b_3$

$b_4 = c_2$

$c_1 = a_2$

$c_2 = b_4$

LHS	RHS	Domain LHS	Domain RHS	
a_1	b_1	1	2	$S\sigma_1$
a_2	c_1	4	3	
a_3	a_4	16	13	
a_4	a_3	13	16	
b_1	b_2	2	8	$S\sigma_2$
b_2	a_1	8	1	
b_3	b_3	15	15	
b_4	c_2	9	12	
c_1	a_2	3	4	

$c_3 = c_3$
$c_4 = c_4$

c_2	b_4	12	9
c_3	c_3	14	14
c_4	c_4	15	15

$\Big\} S_{\sigma_3}$

Assignments

$\vec{a} = [5, 25, 2, 2]$

COSETS

$k_1 = 2$
$k_2 = 3$

$H = \{1, 4, 16, 13\}$

$k_1 H = \{2, 8, 15, 9\}$

$k_2 H = \{3, 12, 14, 15\}$

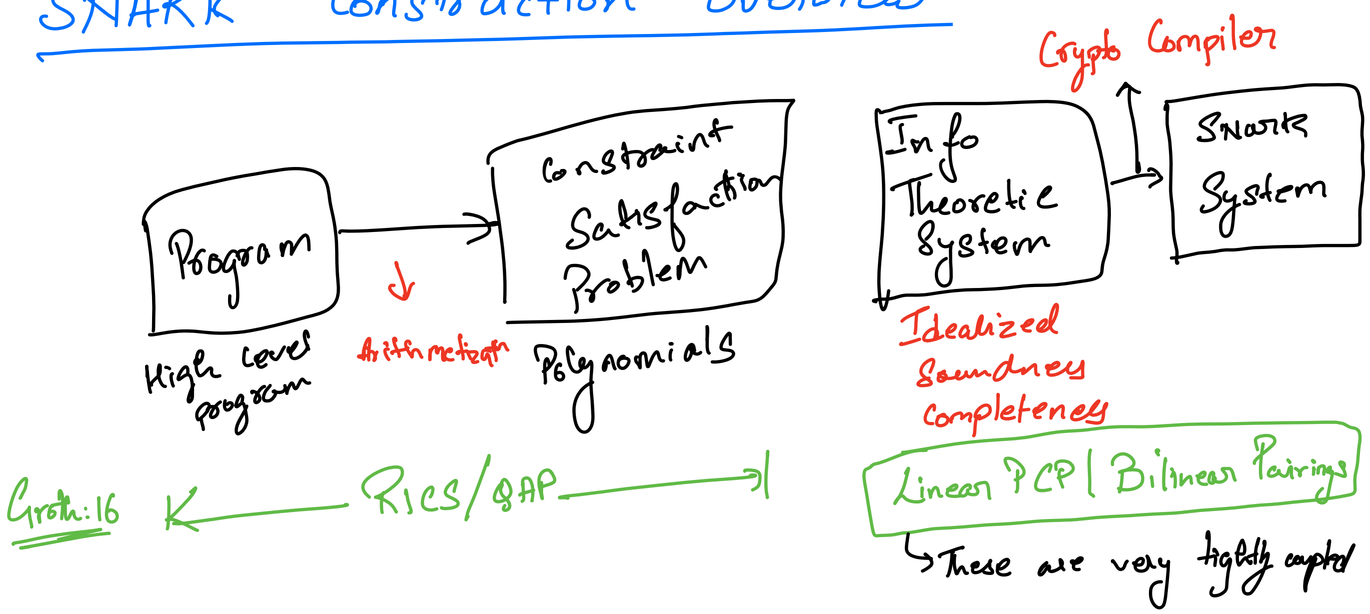
$S_{\sigma_1} = ((1,2), (4,3), (16,13), (13,16))$

$$\underbrace{f_a(x) * f_b(x) * f_c(x)}_{f(x)} = S_{\sigma_1}(x) * S_{\sigma_2}(x) * S_{\sigma_3}(x)$$

$$\prod \frac{f(x)}{S_\sigma(x)} = 1$$

Lec 2

SNARK Construction Overview



Plonk $\xleftarrow{\hphantom{xxxxxx}}$ RAP (PAR) $\xrightarrow{\hphantom{xxxxxxxx}}$ $\boxed{\text{PLONK IOP}}$ Poly. commit scheme $\boxed{\text{PCS}}$

KZG, FRI

Randomized Arith Prg.

Plonkish Arith

Commitment Scheme

Two Proper → Hiding
→ Binding

Optionaly homomorphic properties → (HASH vs Generators)

API

$\boxed{\text{SETUP}}$ → Pulic Params used for the commit (p_k, v_k)

$\boxed{\text{Commit}}$ → (pk, M) → Commit $= \boxed{M}$ Add random (r) for hiding

$\boxed{\text{OPEN}}$ → (v_K, C, M) → $\{0, 1\}$

Polynomial Commitment Scheme

Message M is a Polynomial ⇒ Com $(P(x))$

Open – verifier can choose 'Z' randomly and ask to $P(z)$

$$P \xleftarrow{\quad z \quad} V$$

$\xrightarrow{\text{comm}(p(z)), \pi}$

$\boxed{\text{comm}(p(x)), \pi}$ are short and easy to verify

PCS

	Used By	Proof Size	Proof time	Uses	Setup
KZG	SNARKS	Tiny — $O(1)$	High	EC pairings	Trusted
FRI	STARKS	Large — n	Highest	Hash func. → used Merkle Trees	Transparent
IPA	SNARKS	Small — $O(\log n)$	High	EC	Transparent

IPA ↓
Inner Product
used in Halo2
Bulletproofs

KZG Commitment Scheme

Setup: SRS (p_k, v_k)

$G_1 \times G_2$ $\quad \mathbb{F}_p$

Secret point $= s$

proving key $= s^0G_1, \; sG_1, \; s^2G_1 \; \ldots \; s^dG_1$

OR

$[s^0]_1G_1, \; [s]_1G_1, \; [s^2]_1G_1 \; \ldots \; [s^d]_1$

Prover

$p(x) = x^2 - 3x + 2$

$e(p(s)) = s^2 - 3s + 2$

$= [s^2]_1 - 3[s]_1 + 2[1]_1$

commit $= [p(s)]_1$

$\boxed{\text{verification key} = s^0G_2 \quad s.G_2}$

OPEN 1) Verifier sends a random value "z"

2) $p(z) = (a) \Rightarrow q(x) = \underline{p(x) - a}$

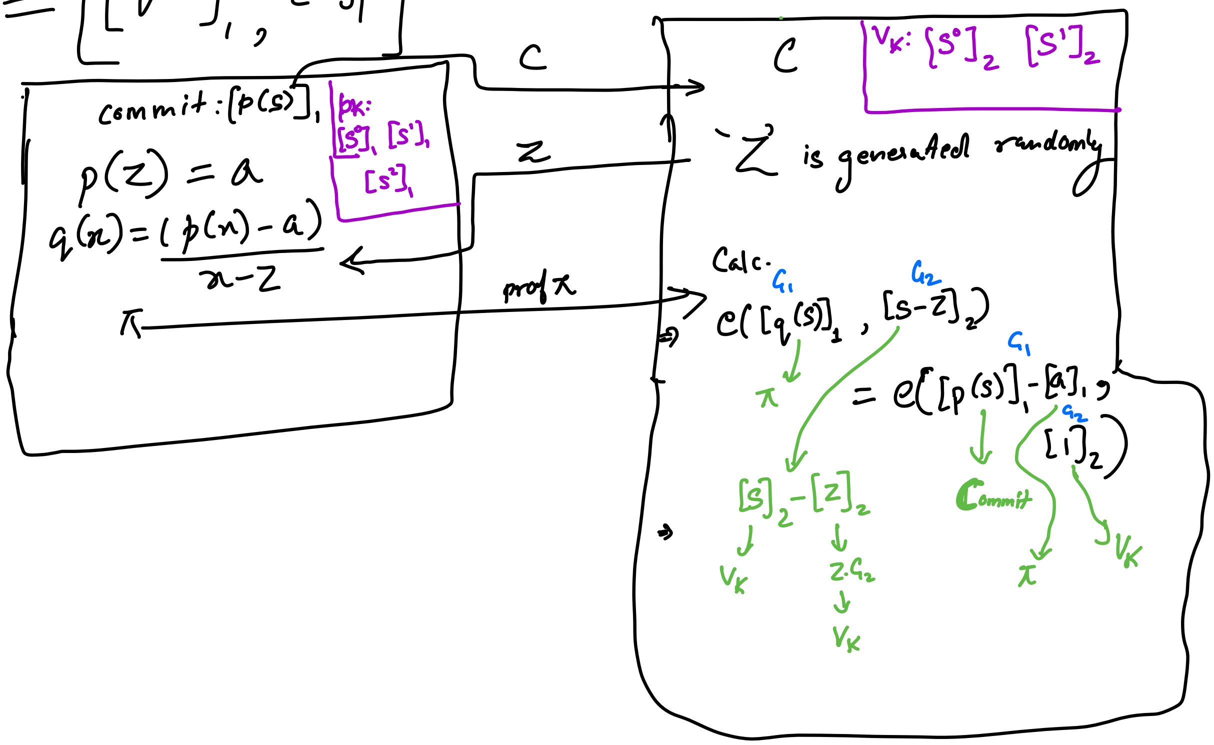
$x - z$

The prover doesn't want to send $[p(s)]_1$

$p(z) = a$

$p(x) = q(x)(x-z) + a \Rightarrow \frac{p(x)-a}{x-z} = q(x)$

$\pi = \left[[q(s)]_1 , [a]_1 \right]$



Pen and Paper Example KZG

$\mathbb{F}_{17}$

$x^2 - 3x + 2 = P(x) = (x-1)(x-2)$

$s = 10$

proving key: $s^0 G, s^1 G, s^2 G \Rightarrow G_1, 10G_1, 100G_1$

verific. key: $s^0 G, s^1 G \Rightarrow G_2, 10G_2$

Commit $P(S)$ [S is not given] [proving key is given]

$$[S^2]_1 - 3[S]_1 + 2[1]_1 = [100 - 30 + 2]_1$$
$$= [72]_1$$
$$= 72\, g_1 \bmod 17$$
$$= 4 g_1$$

Open $z = 5 \leftarrow$ verifier sends this

$$\frac{3 \pm \sqrt{9+40}}{2}$$
$$\frac{3 \pm 7}{2} = 5, -2$$

$$P(5) = 25 - 3 \times 5 + 2 = 12$$

$$Q(\cancel{x}) = \frac{P(\cancel{x}) - P(\cancel{z})}{x - z} = \frac{\cancel{x}^2 - 3\cancel{x} + 2 - 12}{\cancel{x} -}$$

$$= \frac{\cancel{x}^2 - 3\cancel{x} - 10}{x - 5}$$

$$Q(S) = S + 2$$

$$\pi = \left[[Q(S)]_1, [a]_1 \right]$$

$$= \left[[12]_1, [12]_1 \right]$$

$$Q(x) = \frac{x^2 - 3x - 10}{x - 5} = x + 2$$

At prover end the he only has 's' to evaluate so it sends the evaluated polynomial at S

At verifiers end

$$q(s) = \frac{p(s) - a}{s - z}$$

$$q(s) \cdot (s - z) = p(s) - a$$

$$e([q(s)]_1, [(s - z)]_2) = e([p(s) - a]_1, [1]_2)$$

$$e([12]_1, [s]_2 - [z]_2) = e([p(s)]_1 - [a]_1, [1]_2)$$

$$e([12]_1, [10]_2 - [5]_2) = e([4]_1 - [12]_1, [1]_2)$$

$$e([12]_1, [5]_2) = e([4+5]_1, [1]_2)$$

$$G_T^{60 \bmod 17} = G_T^{9}$$

$$G_T^a = G_T^a$$