2K Bootcamp

GROUP SET & BINARY OPERATORS

- → Closed
- → Associative
- → Identity
- → Inverse

→ Properties

eg of groups.

$G(\mathbb{Z}, +)$ → Group

$G(\mathbb{Z}, \times)$ → Not a group

$\mathbb{R}$ → Real

$\mathbb{C}$ - Complex

$\mathbb{Q}$ - Rational.

$\mathbb{Z}$ - Integers

→ Only closed & associative

↳ Semigroup

→ Only closed, associative & Identity.

↳ Monoid

→ only closed

↳ Magma ; eg:

uint256, Keccak256

Homomorphism

→ Homo - Same

→ morp - Shape.

→ There is a homomorphism from A to B ~~where~~ if there exists a function $\phi : a \to b \quad a \in A \quad b \in B$ and $\phi(a \oplus b) = \phi(a) + \phi(b)$

eg: A: all string under concat

B: all non-negative integer under addition

$$\phi(a \oplus b) = \phi(a) \oplus \phi(b)$$

H/w

A: $G(z, +)$

B: $G(b^2, \times)$

ELLIPTIC CURVES

→ An elliptic curve group is the set of (x, y) points that satisfies the eq: $y^2 = x^3 + b$

→ $(x, y) \oplus (x', y') = (x'', y'')$

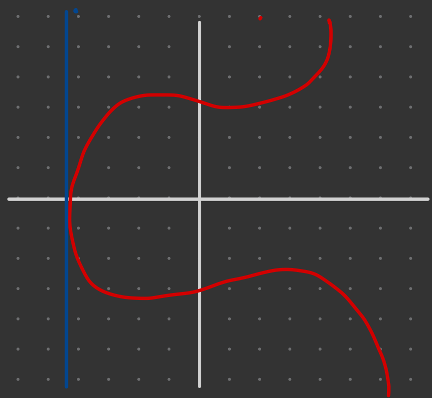
closed property

THEOREM

→ If a non-vertical line intersects 2 points on an elliptic curve, it will always intersect a third point.

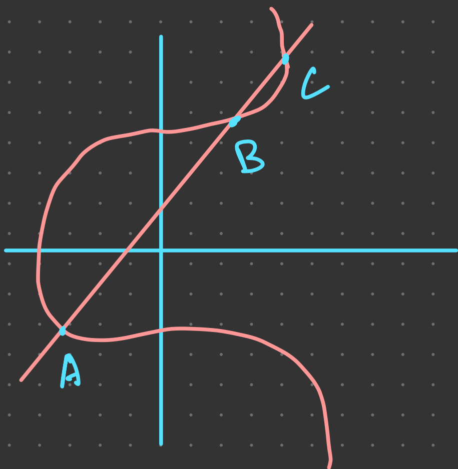
→ If a line is vertical and crosses 2 points, it will not intersect a third.

→ A elliptic curve group is the set of (x, y) points that satisfy $y^2 = x^3 + b$ Union "Point at Infinity."



P.O.I is the 3rd intersection of a vertical line.

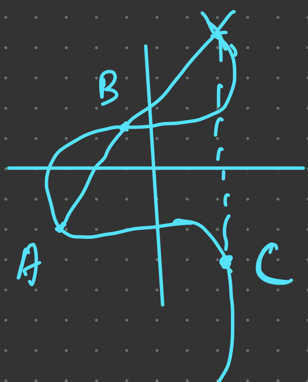
→ The P.O.I is the identity element of the set.



$$A \oplus B = C$$
$$A \oplus C = B$$
$$B \oplus C = A$$

Contradiction

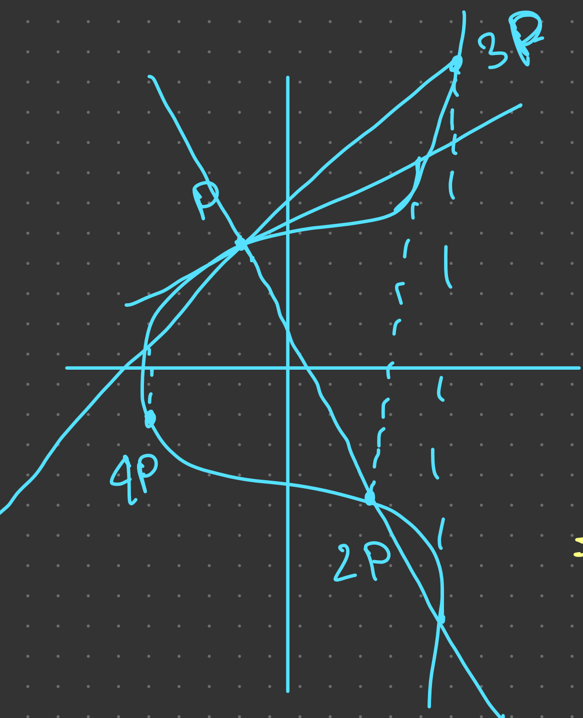
$$(B \oplus C) \oplus B = C$$
$$\Rightarrow 2B \oplus C = C$$
$$\Rightarrow \cancel{B^{-1} + B} + C + B = B^{-1} + C$$
$$\Rightarrow C + B = B^{-1} + C$$
$$\Rightarrow \cancel{C^{-1} + C} + B = B^{-1} + \cancel{C} + \cancel{C^{-1}}$$
$$\Rightarrow B = B^{-1} \quad ???$$



→ To avoid the contradiction $A \oplus B$ is mirrored wrt the x-axis

→ The binary operator is to take both points, draw a straight line, determine the third intersection point, then flip over the x-axis.

if $P_1 x == P_2 x$; return I



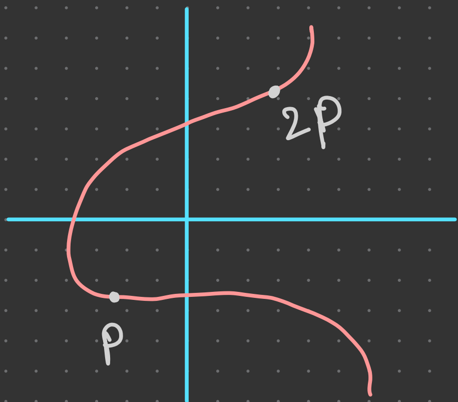
$\Rightarrow 2P \oplus 2P = 4P$

$\Rightarrow 100P \Rightarrow$ 2P 32P
4P 64P
8P
16P

$\Rightarrow 100P \Rightarrow 64P + 32P + 4P$

worst case : $2 \log n$

DISCRETE LOG ON ELLIPTICAL CURVES



$y^2 = x^3 + 7 \pmod n$

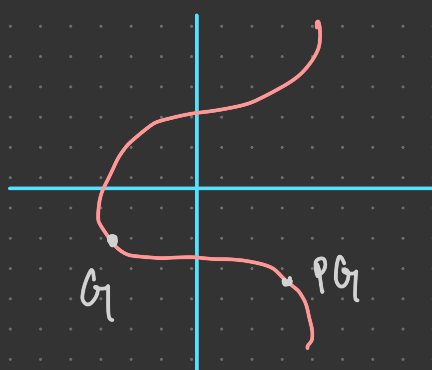
→ Given a point P and a scalar z, it is easy to compute zP. But given 2 points P & zP, its extremely difficult to compute z.

ECDSA

→ Private key = $\sim 2^{256}$ bit scalar.

→ Public key = PG → (x, y)

G is a publicly accepted starting point.



→ Eth address = hash(x_{pub}, y_{pub})

Signing with ECDSA

random number K (r, s, v)

$\Rightarrow R = KG$

$\Rightarrow R = (x_r, y_r) \rightarrow x_r \rightarrow r$

$\Rightarrow h = \text{hash}(msg)$

$\Rightarrow s = K^{-1}(h + r \times priv)$

(Pub, msg, r, s)

Verification

$R' = s^{-1}(hG + r\,Pub)$

$R' = K(h + r \times priv)^{-1}(hG + r\,Pub)$

$R' = K\cancel{(h + r \times priv)^{-1}}\cancel{(h + r \times priv)}G$

$R' = KG = r$

→ Scalar multiplication is associative.

$$\boxed{b(aG) = (ba)G}$$

→ Addition is commutative.

$$\boxed{aG + bG = (a+b)G}$$

$$aG + I = aG$$

$$aG + (-aG) = I.$$

<u>Claim</u>: I have 2 numbers x & y such that they add upto 15.

$x = mul(G, 5)$

$y = mul(G, 10)$

→ with $(x, y, 15)$ the verifier can verify my statement

Verifier =>

$$add(X, Y) == mul(G, 15)$$

$\rightarrow$ $mul(G, a) == mul(G, a + curve\_order)$

$\rightarrow$ $neg(mul(G, a)) = mul(G, curve\_order - a)$

MATRIX MULTIPLY

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} aG \\ bG \end{bmatrix} = \begin{bmatrix} aG + 2bG \\ 3aG + 4bG \end{bmatrix}$$

System of equations.

$$\Rightarrow \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \times \begin{bmatrix} aG \\ bG \end{bmatrix} = \begin{bmatrix} 4 \\ 10 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 4 \\ 10 \end{bmatrix}$$

$$\Rightarrow \quad x + 2y = 4 \qquad \equiv \qquad aG + 2bG = 4G$$

$$\Rightarrow \quad 3x + 4y = 10 \qquad \qquad 3aG + 4bG = 10G$$

→ To prove the solution to a system of equation, we can provide the values of aG & bG without revealing the solutions a & b.

→ Math ops in finite fields can be done without any precision issues.

→ Precompiles

BN128 addition - address(6)

BN128 multiply - address(7)

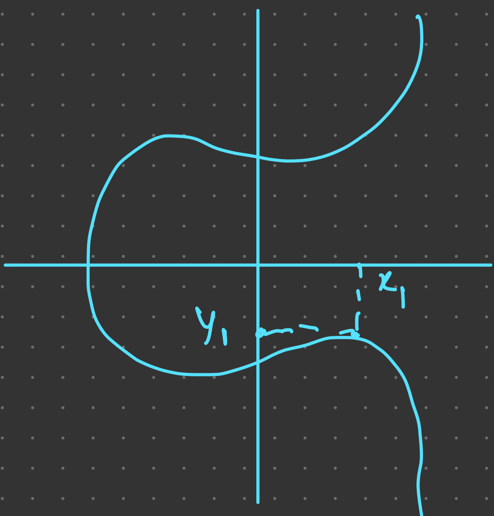
Add

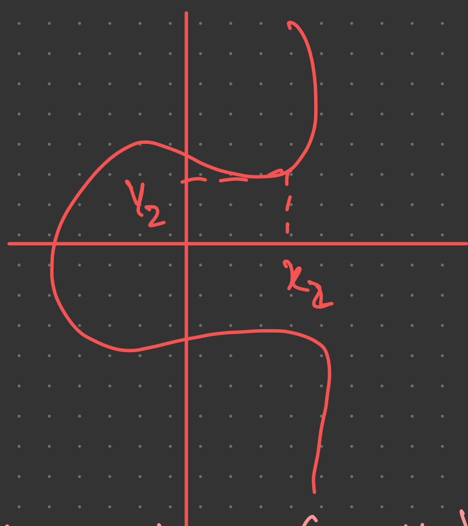
```
(bool ok, bytes memory result)
    = address(6).staticcall(abi.encode
                    (x1, y1, x2, y2));
require(ok, "add failed");
(x, y) = abi.decode(result, (uint256,
                    uint256));
```

Point Multiplication

$$y^2 = x^3 + b$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}^2 = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^3 + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$





The set of points (x_1, y_1) & (x_2, y_2) is considered one point since it's a solution to

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}^2 = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^3 + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

→ These points in the curve form a group.

→ When you are using a 4-D ECC, we call the point $G_2 \rightarrow ((x_2, x_1), (y_2, y_1))$

$$aG_2 + bG_2 = (a+b)G_2$$

$$a(bG_2) = (ab)G_2$$

→ The above properties hold for G_2.

→ We want to perform symmetric pairing.

$$aG_1 \cdot bG_1 = (ab)G_1$$

but this is very hard to compute.

→ But we can perform asymmetric pairing efficiently.

$$\boxed{aG_1 \cdot bG_2 = abG_{12}}$$

→ Pairing $(G_2, G_1) \rightarrow G_{12}$

→ Precompile address (8) performs pairing.

→ In ZKP, we typically look for something like, is

$$AB = CD + EF + GH$$

where A, C, E, G can be G_1 points & B, D, F, H can be G_2 points.

→ The solidity precompile does not return the G_{12} point. It only checks if $AB = CD + EF + GH$ holds.

Adding G12 Points.

→ final_exponentiate ($AG_{12} \times BG_{12}$)

→ You can only perform pairing with G_1 & G_2 points.

→ if $x = 7$; how do we compute x^3

$a = x^2 \Rightarrow$ pairing(mul(G_2, 7), mul(G_1, 7))

$x^3 = a \times x \Rightarrow$

Pairing(mul(G_2, 49), mul(G_1, 7))

Homework - 4

$$0 = -A_1 B_2 + \alpha_1 \beta_2 + X_1 \gamma_2 + C_1 \delta_2$$

$$X_1 = x_1 G1 + x_2 G_2 + x_3 G1$$

$$-(8 \times 4) + 5 \times 2 + 6 \times 3 + 1 \times 4$$

$A_1 = -8$	$\alpha_1 = 5$	$x_1 = 3$	$C_1 = 1$
$B_2 = 4$	$\beta_2 = 2$	$x_2 = 2$	$\delta = 4$
		$x_3 = 1$	
		$\gamma = 3$	

ZK Goal → to prove we carried out an algorithm.

eg: 1 → Sudoku

1	2	3	4
3	4	1	2
4	1	2	3
2	3	4	1

→ Phase 1 : Solve

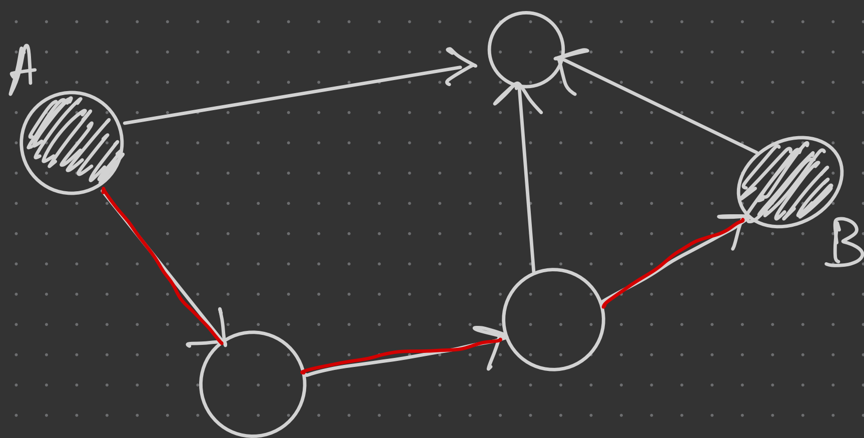
Phase 2 : verify solution.

ZK is only concerned with verifying the solution.

eg: Sort a list → Quick sort.

verify a sorted list → iterate.

eg ③ : Graph.



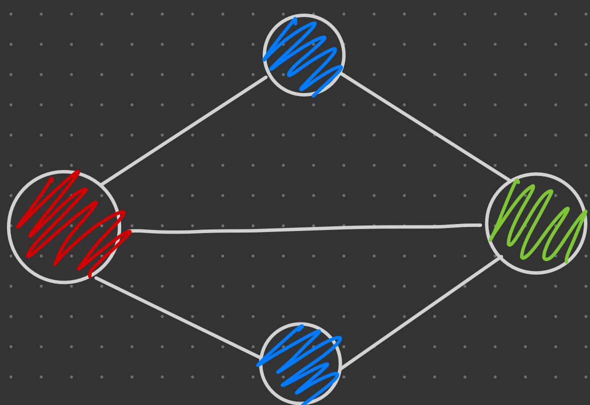
→ Solve it with BFS

→ verify by traversing the path.

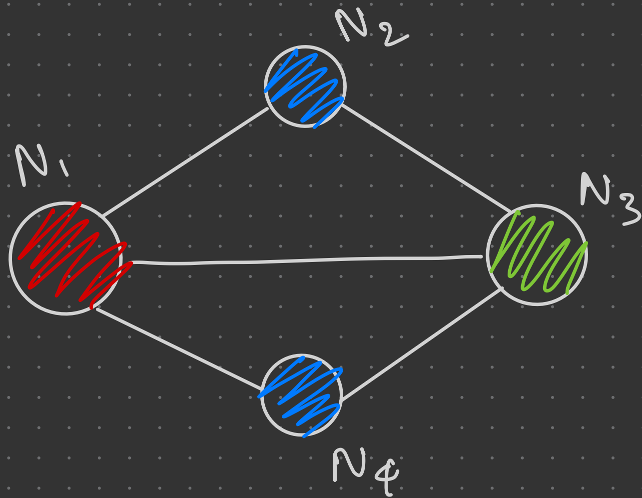
→ We need to express these problems in terms of addition & multiplication cuz that's what we can do in 2k.

eg: 4 - Three colouring.

→ Colour each node of a graph such that none of the neighbouring nodes share the same colour.



→ How to express this graph only in terms of addition & multiplication.



Red = 1

Blue = 2

Green = 3

N can only be 1 or 2 or 3

$$\Rightarrow (N_1-1)(N_1-2)(N_1-3)=0$$

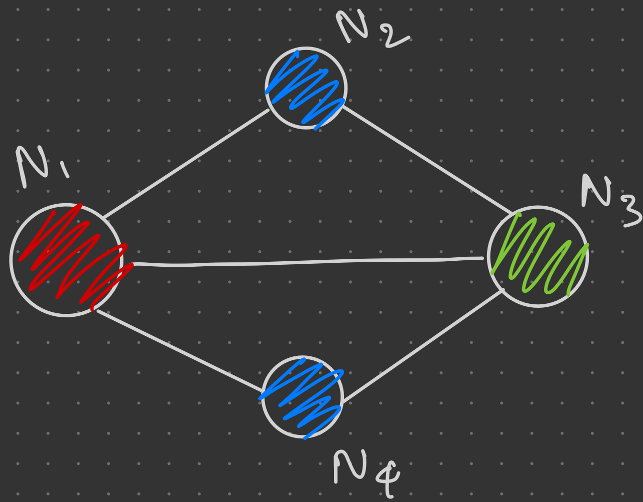
$$(N_2-1)(N_2-2)(N_2-3)=0$$

$$\vdots$$

$$(N_4-1)(N_4-2)(N_4-3)=0$$

> These equations constrain the nodes from having values other than 1, 2 or 3

Neighbouring nodes not sharing the same colour.



N_1	N_2	P
1	1	1
1	2	2
1	3	3
2	1	2
2	2	4
2	3	6
3	1	3
3	2	6
3	3	9

$$(N_1N_2-2)(N_1N_2-3)(N_1N_2-9) = 0$$

This can only be true if N_1 & N_2 are different values.

→ The same applies for N_2N_3, N_1N_3, N_1N_4, N_4N_3

→ We have seen how to do ZK system of equations.

→ Given these constrains, if the equations are balanced, then we can prove yo know the solution to system of eqns..

$x < 16 \implies$ Arithmetic Circuits.

Represent 15 in bits.

$$x = 2^3 b_3 + 2^2 b_2 + 2 b_1 + b_0$$

$$x = 10 = 8 \cdot 1 + 4 \cdot 0 + 2 \cdot 1 + 0$$

$$x = 13 = 8 \cdot 1 + 4 \cdot 1 + 2 \cdot 0 + 1$$

$$x = 15 = 8 \cdot 1 + 4 \cdot 1 + 2 \cdot 1 + 1$$

$$(b_0 - 1) b_0 = 0$$

$$(b_1 - 1) b_1 = 0$$

$$\vdots$$

$$(b_3 - 1) b_3 = 0$$

With constrains

$$x = 8b_3 + 4b_2 + 2b_1 + b_0$$

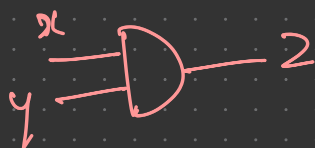
$$(b_0 - 1)\,b_0 = 0 \qquad (b_2 - 1)\,b_2 = 0$$

$$(b_1 - 1)\,b_1 = 0 \qquad (b_3 - 1)\,b_3 = 0$$

the x value cannot be greater than 15.

LOGIC GATES ARITHMETIZATION

AND GATE



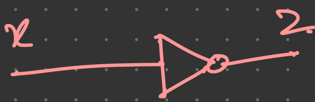
$$z = x \cdot y \qquad \begin{aligned} x(x-1) &= 0 \\ y(y-1) &= 0 \end{aligned}$$

OR GATE

$$x(x-1) = 0$$

$$z = x + y - xy \qquad y(y-1) = 0$$

NOT GATE



$$z = (1-x) \quad \Big| \quad x(x-1) = 0$$

XOR GATE

$$z = x + y - 2xy$$

or

$$z = (x-y)(x-y)$$

$$x(x-1) = 0$$
$$y(y-1) = 0$$

NAND GATE

$$z = 1 - xy$$

$$x(x-1) = 0$$
$$y(y-1) = 0$$

Bitwise AND

$z = x \& y$

$x = 0010$
$y = 1011$
$z = 0010$

Step 1 : Decompose the nums to bits.

$x = 8b_3 + 4b_2 + 2b_1 + b_0$ | $b_0(b_0-1) = 0$ $\vdots$ $b_n(b_n-1) = 0$

$y = 8c_3 + 4c_2 + 2c_1 + c_0$ | $c_0(c_0-1) = 0$ $\vdots$ $c_n(c_n-1) = 0$

$z = 8d_3 + 4d_2 + 2d_1 + d_0$ | $d_0(d_0-1) = 0$ $\vdots$ $d_n(d_n-1) = 0$

$$(b_3c_3 - d_3)(b_2c_2 - d_2)(b_1c_1 - d_1)(b_0c_0 - d_0) = 0$$

Homework - 5

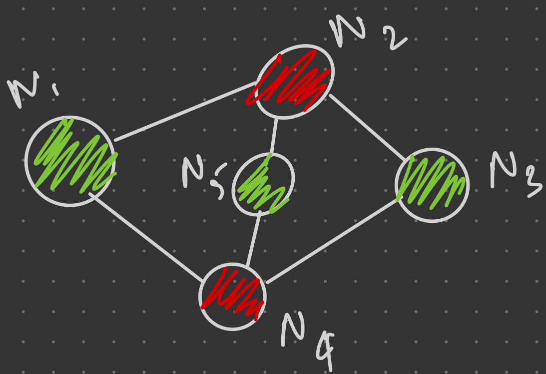
Q1. x_1 to x_n such that at least one signal is 0.

$$x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdots x_n = 0$$

Q_2. x_1 to x_n such that all signals are 1

$$x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdots x_n - 1 = 0$$

Q_3. Bipartite Graph.



$$N_1(N_1 - 1) = 0$$
$$\vdots$$
$$N_5(N_5 - 1) = 0$$

N_1	N_2	
1	1	1
1	2	2
2	1	2
2	2	4

$(N_1N_2 - 2) = 0$ $(N_3N_4 - 2) = 0$

$(N_2N_3 - 2) = 0$ ..... $(N_4N_5 - 2) = 0$

Q4) $K = \max(x, y, z)$

$V = \max(x, y)$

$x = n$ bits

$y = n$ bits

$\max(x, y)$

$= 2^{n-1} + (x - y)$

$= 2^n b_n + 2^{n-1} b_{n-1} \ldots b_0$

If the MSB of $2^{n-1} + (x-y) = 1$; then $x \geq y$. If MSB $= 0$; then $y > x$

$V = b_n x + (1 - b_n) y$

$V = \max(x, y) = b_n x + (1 - b_n) y$

$K = \max(v, 2)$

Q5) Signals $x_1, x_2 \ldots x_n$ such that at least one of the signal is 1.

$$(1-x_1)(1-x_2)(1-x_3)\ldots(1-x_n) = 0$$

Q6) Signal v such that v is a power of 2.

$$v = 2^n b_n + 2^{n-1} b_{n-1} + 2^{n-2} b_{n-2} \ldots b_0$$

$$b_n + b_{n-1} + b_{n-2} + \ldots b_0 - 1 = 0$$

$$b_n(b_n - 1) = 0$$

$$\vdots$$

$$b_0(b_0 - 1) = 0$$

Q7) Arithmetization of the covering set problem.

$$S = \{1, 2, 3, 4 \dots 10\}$$

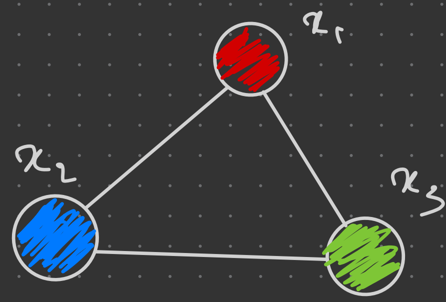
Embed subsets = $\{0, 0, 1, 1, 0\}$

Bitwise OR of embedded subset.

$$S_1 = \{a_1, a_2, a_3 \dots a_n\}$$
$$S_2 = \{b_1, b_2, b_3 \dots b_n\}$$
$$\vdots$$
$$S_k = \{z_1, z_2, z_3 \dots z_n)$$

$$(1-a_1)(1-b_1) \dots (1-z_1) = 0$$
$$(1-a_2)(1-b_2) \dots (1-z_2) = 0$$
$$\vdots$$
$$(1-a_n)(1-b_n) \dots (1-z_n) = 0$$

RANK-1 CONSTRAIN SYSTEM



ARITHMETIC CIRCUITS

$$(x_1-1)(x_1-2)(x_1-3)=0$$

$$(x_2-1)(x_2-2)(x_2-3)=0$$

$$(x_3-1)(x_3-2)(x_3-3)=0$$

$$(x_1x_2-2)(x_1x_2-3)(x_1x_2-6)=0$$

$$(x_2x_3-2)(x_2x_3-3)(x_2x_3-6)=0$$

$$(x_1x_3-2)(x_2x_3-3)(x_1x_3-6)=0$$

→ If the x_1, x_2 & x_3 are G1 points then we can't perform multiplication of G1 points. R1CS is used to fix this.

System of eqn

$$x \cdot y = z + b + 5x$$

$\underbrace{x \cdot y}$ → Quadratic Constrain

How to do $xyz = v$ with elliptical curves??

$$xyz = v$$

$$\boxed{\begin{aligned} w &= xy \\ v &= wz \end{aligned}}$$

eg: $x_1 x_2 x_3 + x_2 x_4 + x_3^2 = 5x_1 + 6 - x_3 x_4 x_1$

$$2x_3 x_4 + 3 = 4x_2^2$$

$v_1 = x_1 x_2$ $\quad$ $v_3 = x^3$

$v_2 = x_2 x_4$ $\quad$ $v_4 = x_3 x_4$

$\Rightarrow$ $v_1 x_3 + v_2 + v_3 = 5x_1 + 6 - v_4 x_1$

$$2v_4 + 3 = 4x_2^2$$

$$V_5 = V_1 x_3$$

$$\Rightarrow \quad V_5 + V_2 + V_3 = 5x_1 + 6 - V_4 x_1$$

$$2V_4 + 3 = 4x_2^2$$

$$\Rightarrow \quad \left.\begin{aligned}
V_1 &= x_1 x_2 \\
V_2 &= x_2 x_4 \\
V_3 &= x_3^2 \\
V_4 &= x_3 x_4 \\
V_5 &= V_1 x \\
V_5 + V_2 + V_3 - 5x_1 - 6 &= -V_4 x_1 \\
2V_4 + 3 &= 4x_2 x_2
\end{aligned}\right\} \text{R1CS Circuits}$$

↳ convert this to matrix

$\Rightarrow \quad 2v_4 + 3 = 4x_2^2$

$$
\left.\begin{aligned}
v_1 &= x_1 x_2 \\
v_2 &= x_2 x_4 \\
v_3 &= x_3^2 \\
v_4 &= x_3 x_4 \\
v_5 &= v_1 x \\
v_5 + v_2 + v_3 - 5x_1 - 6 &= -v_4 x_1 \\
2v_4 + 3 &= 4x_2 x_2
\end{aligned}\right]
$$

R1CS Circuits

convert this to matrix

$$
\begin{array}{cccccccccc}
1 & x_1 & x_2 & x_3 & x_4 & v_1 & v_2 & v_3 & v_4 & v_5
\end{array}
$$

$$
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
-6 & -3 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\
& & & & & & & & &
\end{bmatrix}
\begin{bmatrix}
1 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5
\end{bmatrix}
=
\begin{bmatrix}
v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_5 + v_2 + v_3 - 5x_1 - 6 \\ 2v_4 + 3 \\ \\
\end{bmatrix}
$$

$$\Rightarrow \boxed{Cs \circ Rs = Os}$$

→ s is the witness vector.

→ We take the Hadamard product of Cs & Rs

Hadamard product is element wise multiplication

eg. $\begin{bmatrix} 1 \\ 2 \end{bmatrix} \circ \begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 3 \\ 8 \end{bmatrix}$

eg: 2 $\quad x_1 x_2 x_3 = x_4$

$$6x_2 + x_1 x_3 = x_5 + 1$$

$$x_1 x_2 = v_1$$

$$v_1 x_3 = x_4$$

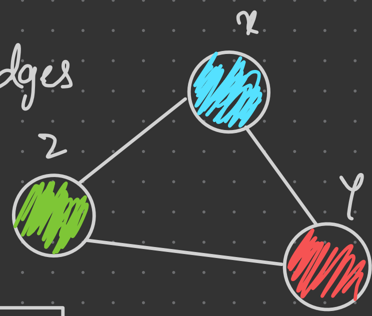
$$x_1 x_3 = x_5 + 1 - 6x_2$$

$$\begin{array}{c} \begin{matrix} 1 & x_1 & x_2 & x_3 & x_4 & x_5 & v_1 \end{matrix} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \end{array} \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ v_1 \end{bmatrix} \circ \begin{array}{c} \begin{matrix} 1 & x_1 & x_2 & x_3 & x_4 & x_5 & v_1 \end{matrix} \\ \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \end{array} \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ v_1 \end{bmatrix}$$

$$= \begin{array}{c} \begin{matrix} 1 & x_1 & x_2 & x_3 & x_4 & x_5 & v_1 \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & -6 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array} \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ v_1 \end{bmatrix}$$

Homework - 6

Q1. Graph with 3 nodes & 3 edges



Constrains

$$(x-1)(x-2)(x-3)=0$$
$$(y-1)(y-2)(y-3)=0$$
$$(z-1)(z-2)(z-3)=0$$

$$(xy-2)(xy-3)(xy-6)=0$$
$$(yz-2)(yz-3)(yz-6)=0$$
$$(xz-2)(xz-3)(yz-6)=0$$

x	y	z
1	1	1
1	2	2
1	3	3
2	1	2
2	2	4
2	3	6
3	1	3
3	2	6
3	3	9

$$\Rightarrow \quad x^3 - 6x^2 + 11x - 6 = 0$$
$$y^3 - 6y^2 + 11y - 6 = 0$$
$$z^3 - 6z^2 + 11z - 6 = 0$$

$$x^3y^3 - 11x^2y^2 + 42xy - 36 = 0$$
$$y^3z^3 - 11y^2z^2 + 42yz - 36 = 0$$
$$z^3x^3 - 11z^2x^2 + 42zx - 36 = 0$$

$$x^2 = V_1$$
$$y^2 = V_2$$
$$z^2 = V_3$$
$$V_1V_2 = V_4 = x^2y^2$$
$$V_2V_3 = V_5 = y^2z^2$$
$$V_3V_1 = V_6 = z^2x^2$$
$$xy = V_7$$
$$yz = V_8$$
$$zx = V_9$$

$$\Rightarrow \quad V_1 x - 6V_1 + 11x - 6 = 0$$
$$V_2 y - 6V_2 + 11y - 6 = 0$$
$$V_3 z - 6V_3 + 11z - 6 = 0$$
$$V_4V_7 - 11V_4 + 36V_7 - 36 = 0$$
$$V_5V_8 - 11V_5 + 36V_8 - 36 = 0$$
$$V_6V_9 - 11V_6 + 36V_9 - 36 = 0$$

$$x^2 = v_1$$

$$y^2 = v_2$$

$$z^2 = v_3$$

$$v_1 v_2 = v_4$$

$$v_2 v_3 = v_5$$

$$v_3 v_1 = v_6$$

$$xy = v_7$$

$$yz = v_8$$

$$zx = v_9$$

$$v_1 x = 6v_1 - 11x + 6$$

$$v_2 y = 6v_2 - 11y + 6$$

$$v_3 z = 6v_3 - 11z + 6$$

$$v_4 v_7 = 11v_4 - 36v_7 + 36$$

$$v_5 v_8 = 11v_5 - 36v_8 + 36$$

$$v_6 v_9 = 11v_6 - 36v_9 + 36$$

$$
\begin{array}{ccccccccccccc}
1 & x & y & z & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 & v_9
\end{array}
$$

$$
\begin{bmatrix}
0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0
\end{bmatrix}
\times
\begin{bmatrix}
1\\ x\\ y\\ z\\ v_1\\ v_2\\ v_3\\ v_4\\ v_5\\ v_6\\ v_7\\ v_8\\ v_9
\end{bmatrix}
$$

$$
\begin{array}{ccccccccccccc}
1 & x & y & z & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 & v_9
\end{array}
$$

$$
\begin{bmatrix}
0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1
\end{bmatrix}
\times
\begin{bmatrix}
1\\ x\\ y\\ z\\ v_1\\ v_2\\ v_3\\ v_4\\ v_5\\ v_6\\ v_7\\ v_8\\ v_9
\end{bmatrix}
$$

$$
= \begin{array}{c}
\begin{array}{ccccccccccccc} 1 & x & y & z & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 & v_9 \end{array} \\
\begin{bmatrix}
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
6 & -11 & 0 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
6 & 0 & -11 & 0 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
6 & 0 & 0 & -11 & 0 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 0 \\
36 & 0 & 0 & 0 & 0 & 0 & 0 & 11 & 0 & 0 & -36 & 0 & 0 \\
36 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 11 & 0 & 0 & -36 & 0 \\
36 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 11 & 0 & 0 & -36
\end{bmatrix}
\end{array}
\times
\begin{bmatrix} 1 \\ x \\ y \\ z \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \\ v_9 \end{bmatrix}
$$

Q4) why only 2 multiplication in R1CS?

We can only perform bilinear pairing with a G1 & G2 points. Therefore we can only perform it once.

Quadratic Arithmetic Program

→ R1CS is not succinct.

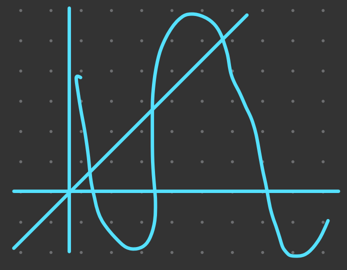
→ we need n-rows to check equality

$\boxed{\text{R1CS} \rightarrow \text{QAP}}$

→ So we turn the R1CS form to a polynomial form.

Schwartz - Zippel lemma

→ two polynomial intersects at most d points where d is the degree of the larger polynomial.



Verify if $P(x) = q(x)$

VERIFIER | **PROVER**

$\xleftarrow{\text{Polynomial Commitment}}$ $P(z)$, $q(z)$

τ $\xrightarrow{\text{random point } \tau}$

$\xleftarrow{\text{Evaluate } P(\tau) \text{ \& } q(\tau)}$ $P(\tau)$, $q(\tau)$

$\xleftarrow{\text{Proof of evaluation}}$ π

→ The prover first commits the polynomial.

→ The verifier gives a random point τ to the prover.

→ The prover evaluates $P(\tau)$ & $q(\tau)$ and returns them to the verifier along with the proof (π).

→ If $p(\tau) = q(\tau)$ then $p(x) = q(x)$

→ This is due to schwattz - zippel lemma.

→ If the degree of polynomial is 3 then both the polynomials will intersect atmost 3 points.

→ In a 256-bit field the probability of choosing a τ where $p(\tau) = q(\tau)$ but $p(x) \neq q(x)$ is $3/2^{256}$.

RING

→ Ring is a set with two binary operators + & X

under +: abelian group (group + distributive)

under x: monoid (no inverse)

$\mathbb{Z}$ under + & x is a ring.

Polynomials under + & × are rings

Under +

→ Closed
→ associative
→ Identity (0)
→ inverse (-polynomial)
→ distributive

Under ×

→ Closed
→ Associative
→ Identity (1)

Column Vectors under + & Hadamard product are rings

It's a monoid because the inverse does not exist for every element.

$$\begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix} \cdot \begin{bmatrix} 1/2 \\ 1/3 \\ 1/4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \checkmark$$

$$\begin{bmatrix} 2 \\ 0 \\ 4 \end{bmatrix} \cdot \begin{bmatrix} 1/2 \\ ? \\ 4 \end{bmatrix} \times$$

Inverse for 0 does not exist.

A RICS can be expressed as column vectors under + & Hadamard product ($\circ$)

eg:

$$\begin{bmatrix} 2 & 3 \\ 4 & 9 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \circ \begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$$

$$\Rightarrow \left(\begin{bmatrix} 2 \\ 4 \end{bmatrix} \circ \begin{bmatrix} a_1 \\ a_1 \end{bmatrix} + \begin{bmatrix} 3 \\ 9 \end{bmatrix} \circ \begin{bmatrix} a_2 \\ a_2 \end{bmatrix} \right) \circ$$

$$\left(\begin{bmatrix} 3 \\ 0 \end{bmatrix} \circ \begin{bmatrix} a_1 \\ a_1 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix} \begin{bmatrix} a_2 \\ a_2 \end{bmatrix} \right)$$

$$= \begin{bmatrix} 1 \\ 2 \end{bmatrix} \circ \begin{bmatrix} a_1 \\ a_1 \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \end{bmatrix} \circ \begin{bmatrix} a_2 \\ a_2 \end{bmatrix}$$

⇒ Column Vectors & polynomials are rings.

⇒ There exists a homomorphism b/w them : CV $\rightleftarrows$ polynomial.

⇒ A polynomial can be thought of as a vector with all the points that satisfy the equation.

eq: $2x^2 - 3x + 5 \Rightarrow \begin{bmatrix} -1 & 10 \\ 0 & 5 \\ 1 & 4 \\ 2 & 7 \\ \vdots & \vdots \end{bmatrix}$

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix}$$

$$\Downarrow \qquad\qquad \Downarrow \qquad\qquad \Downarrow$$

$f(x) = x \qquad f(x) = 1 \qquad f(x) = x+1$

$\Rightarrow \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$ can be written as a polynomial $f(x) = x$.

$\Rightarrow \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$ can be written as a polynomial $f(x) = 1$

$\Rightarrow x + 1 = \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix} = f(x) = x + 1$

$\Rightarrow$ Use Lagrange Interpolation to convert points to polynomial.

$\Rightarrow$ given a set of (x, y) pairs Lagrange interpolation will return the lowest degree polynomial that passes through those points.

$\Rightarrow$ we turn column vectors to polynomials using Lagrange interpolation. This makes it succinct.

Multiplication of Column Vectors

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 4 \end{bmatrix} = \begin{bmatrix} 2 \\ 8 \end{bmatrix}$$

$\Downarrow$ $\quad$ $\Downarrow$ $\quad$ $\Downarrow$ Lagrange interpolation

$x \qquad 2x \qquad 6x-4$

$\Rightarrow$ The degrees on LHS & RHS is not balanced. (LHS $\rightarrow$ degree 2 ; RHS $\rightarrow$ degree 1)

$\Rightarrow$ So we need to add a degree 2 polynomial to RHS to balance the equation.

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 4 \end{bmatrix} = \begin{bmatrix} 2 \\ 8 \end{bmatrix} + P_2$$

$\Rightarrow$ Polynomial P_2 should have degree 2 and be equal to $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$

$\Rightarrow$ if we have 2 polynomials P_1 & P_2 with roots $\{r_1\}$ & $\{r_2\}$ then $P_1 \times P_2$ will have the roots $\{r_1\} \cup \{r_2\}$.

eg: $P_1 = (x-3)(x+5) \Rightarrow 3, -5$

$P_2 = (2x+1)(3x) \Rightarrow -\frac{1}{2}, 0$

$P_1 P_2 = (x-3)(x+5)(2x+1)(3x)$

$\Rightarrow 3, -5, -\frac{1}{2}, 0$

$P_1 \cdot P_2 = P_3 \cdot b$] $\rightarrow$ balancing polynomial.

$$P_1 \cdot P_2 = P_3 \cdot \underset{\downarrow \; (x-1)(x-2) \Rightarrow \begin{bmatrix} 0 \\ 0 \end{bmatrix}}{t} \cdot h$$

$$\frac{P_1 \cdot P_2 - P_3}{t} = h$$

$\Rightarrow$ in this case h's degree is 0.

$$\left(\underbrace{\begin{bmatrix}2\\4\end{bmatrix}}_{u_1}\cdot\begin{bmatrix}a_1\\a_1\end{bmatrix}+\underbrace{\begin{bmatrix}3\\9\end{bmatrix}}_{u_2}\cdot\begin{bmatrix}a_2\\a_2\end{bmatrix}\right)\cdot\left(\underbrace{\begin{bmatrix}3\\0\end{bmatrix}}_{v_1}\cdot\begin{bmatrix}a_1\\a_1\end{bmatrix}+\underbrace{\begin{bmatrix}1\\2\end{bmatrix}}_{v_2}\begin{bmatrix}a_2\\a_2\end{bmatrix}\right)$$

$$=\underbrace{\begin{bmatrix}1\\2\end{bmatrix}}_{w_1}\cdot\begin{bmatrix}a_1\\a_1\end{bmatrix}+\underbrace{\begin{bmatrix}2\\1\end{bmatrix}}_{w_2}\begin{bmatrix}a_2\\a_2\end{bmatrix}$$

$$\Rightarrow \sum_{i=1}^{m} a_i u_i(x)\cdot\sum_{i=1}^{m} a_i v_i(x) = \sum_{i=1}^{m} a_i w_i(x) + h(x)\cdot t(x)$$

$\Rightarrow$ This is how ZK-SNARKS are succinct.

$\Rightarrow$ check if the LHS = RHS using schwartz-zippel lemma.

$\rightarrow$ Everything is done in a finite field with order being the curve order.

Homework - 7 QAP

Q2.

$$\begin{bmatrix} 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ out \\ x \\ y \\ v_1 \\ v_2 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} 3 \\ 0 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ x \\ x \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_1 \\ v_1 \end{bmatrix}$$

$$\Rightarrow x(3a^2 - 12a + 12) + v_1(-1a^2 + 4a - 3)$$

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ out \\ x \\ y \\ v_1 \\ v_2 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ x \\ x \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 5 \end{bmatrix} \begin{bmatrix} y \\ y \\ y \end{bmatrix}$$

$$\Rightarrow x\left(\frac{1}{2}a^2 - \frac{5}{2}a + 3\right) + y\left(\frac{3}{2}a^2 - \frac{7}{2}a + 2\right)$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -3 & 1 & 1 & 2 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ out \\ x \\ y \\ v_1 \\ v_2 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} 0 \\ 0 \\ -3 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} out \\ out \\ out \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ x \\ x \end{bmatrix}$$

$$+ \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix} \cdot \begin{bmatrix} y \\ y \\ y \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_1 \\ v_1 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix} \begin{bmatrix} v_2 \\ v_2 \\ v_2 \end{bmatrix}$$

$$\Rightarrow 1\left(\frac{-3}{2}a^2 + \frac{9}{5}a - 3\right) + out\left(\frac{1}{2}a^2 - \frac{3}{2}a + 1\right)$$

$$+ x\left(\frac{1}{2}a^2 - \frac{3}{2}a + 1\right) + y\left(1a^2 - 3a + 2\right)$$

$$+ v_1\left(\frac{1}{2}a^2 - \frac{3}{5}a + 3\right) + v_2\left(-\frac{1}{2}a^2 + \frac{3}{2}a - 1a\right)$$

$$+ h(a)\, t(a)$$

$$h(a) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = (a-1)(a-2)(a-3)$$

$$\Rightarrow \left(3a^2x - 12ax + 12x - 1a^2v_1 + 4av_1 - 3v_1\right) x$$

$$\left(\frac{1}{2}a^2x - \frac{5}{2}ax + 3x + \frac{3}{2}a^2y - \frac{7}{2}ay + 2y\right)$$

$$-\left(\frac{1}{2}a^2x - \frac{3}{2}ax + x\right) - \left(1a^2y - 3ay + 2y\right)$$

$$-\left(\frac{1}{2}a^2v_1 - \frac{3}{5}av_1 + 3v_1\right)$$

$$\Rightarrow \frac{3}{2}a^4x^2 - \frac{15}{2}a^3x^2 + 9a^2x^2 + \frac{9}{2}a^4xy - \frac{21}{2}a^3xy$$

$$+ 6a^2xy - 6a^3x^2 + 30a^2x^2 - 36a^2x^2$$

$$- 18a^3xy + 42a^2xy - 24axy + 6a^2x^2$$

$$- 30ax^2 + 36x^2 + 18a^2xy - 42axy$$

$$+ 24xy - \frac{a^4v_1}{2} + \frac{5}{2}a^3xv_1 - 3a^2xv_1 - \frac{3}{2}a^4v_1y$$

$$+ \frac{7}{2}a^3v_1y - 2a^2v_1y + 2a^3xv_1 - 10a^2v_1x$$

$$+ 12axv_1 + 6a^3v_1y - 14a^2v_1y + 8av_1y$$

$$\Rightarrow \left(3a^2x - 12ax + 12x - 1a^2v_1 + 4av_1 - 3v_1\right) \times$$

$$\left(\frac{1}{2}a^2x - \frac{5}{2}ax + 3x + \frac{3}{2}a^2y - \frac{7}{2}ay + 2y\right)$$

$$-\frac{3}{2}a^2xv_1 + \frac{15}{2}axv_1 - 9xv_1 - \frac{9}{2}a^2v_1y + \frac{21}{2}av_1y$$

$$-6yv_1$$

Polynomials on Elliptic Curves.

$$f(x) = x^3 + x^2 + 2$$

$\Rightarrow$ How to evaluate this if x is a point on a elliptic curve?

$\Rightarrow f(3G_1) = ?$

$\Rightarrow$ This is a <u>polynomial commitment</u>

$$f(x) = 2x^3 + 3x^2 + 7x + 12$$
$$= 2x^3 + 3x^2 + 7x^1 + 12x^0$$

inner product

$$= \langle [2, 3, 7, 12], [x^3, x^2, x, 1] \rangle$$

$$\Rightarrow \langle [2, 3, 7, 12], [x^3G_1, x^2G_1, xG_1, G_1] \rangle$$

$$f(x)G_1 = 2x^3G_1 + 3x^2G_1 + 7xG_1 + 12G_1$$

$\Rightarrow$ You can evaluate a polynomial without knowing the input.

$$\underbrace{[x^3 G_1, x^2 G_1, x G_1, G_1]}$$

→ Structured Reference string

→ Groth-16 → trusted setup.

<u>Trusted</u> <u>Setup</u> <u>Vector</u>

$$\underbrace{[\ldots, \tau^3 G_1, \tau^2 G_1, \tau G_1, G_1]}$$

→ powers of Tau.

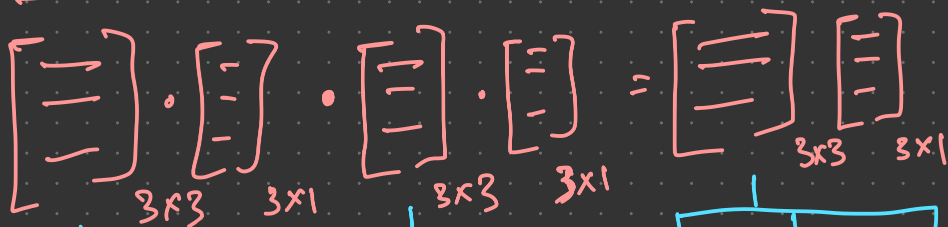
⇒ τ is a toxic waste. It needs to be discarded.

⇒ If the prover knows τ, then they can create equations that satisfy at τ.

$$\sum_{i=1}^{m} a_i u_i(x) \cdot \sum_{i=1}^{m} a_i v_i(x) = \sum_{i=1}^{m} a_i w_i(x) + h(x)\cdot t(x)$$

$$\boxed{[\tau^2 G_1, \tau G_1, G_1], [\tau^2 G_2, \tau G_2, G_2], [\tau t(\tau) G_1, t(\tau) G_1]}$$

<u>R1CS form</u>



$u_1(x) \quad u_2(x) \quad u_3(x)$

$$\underbrace{a_1[u_1(\tau)]_1, a_2[u_2(\tau)]_1, a_3[u_3(\tau)]_1}_{[A]_1 \text{ (} G_1 \text{ Point)}}$$

$v_1(x) \quad v_2(x) \quad v_3(x)$

$$\underbrace{a_1[v_1(\tau)]_2\, a_2[v_2(\tau)]_2\, a_3[v_3(\tau)]_2}_{[B]_2 \text{ (} G_2 \text{ Point)}}$$

$w_1(x) \quad w_2(x) \quad w_3(x)$

$$\underbrace{a_1[w_1(\tau)]_1, a_2[w_2(\tau)]_1, a_3[w_3(\tau)]_1}_{[C]_1 \text{ (} G_1 \text{ Point)}}$$

$\Rightarrow$ we can evaluate all these polynomials at τ using $[\tau^2 G_1, \tau G_1, G_1]$ & $[\tau^2 G_2, \tau G_2, G_2]$

$$\Rightarrow [A]_1 \cdot [B]_2 = [C]_1 \cdot G_2$$

$\Rightarrow$ we still have to solve for $\boxed{h(x)t(x)}$

$h(x)\,t(x)$ at $x = \tau$

$\Rightarrow t(x) = (x-1)(x-2)(x-3)$

$= x^3 - 6x^2 + 11x - 6$

$h(x) = 2x + 1$

$h(x)\,t(x) = \underline{(2x+1)(x^3 - 6x^2 + 11x - 6)}$

$\Rightarrow$ Since $t(x)$ is known before hand, the trusted setup can include $[t(\tau)G_1]$

$\Rightarrow$ $h(x)\,t(x) = (2x+1)\,(t(\tau)G_1)$

$\Rightarrow$ This multiplication can be rewritten to a inner product.

eg: $(2x+1)(6) = \langle [2, 1], [x6, 6] \rangle$

$\Rightarrow$ $h(x)\,t(x) = \langle [2, 1], [\tau t(\tau), t(\tau)] \rangle$

at $x = \tau$

$$\Rightarrow [A]_1 = a_1[u_1(\tau)]_1 + a_2[u_2(\tau)]_1 + a_3[u_3(\tau)]_1$$

$$\Rightarrow [B]_2 = a_1[v_1(\tau)]_2 + a_2[v_2(\tau)]_2 + a_3[v_3(\tau)]_2$$

$$= [C]_1 = a_1[w_1(\tau)]_1 + a_2[w_2(\tau)]_1 + a_3[w_3(\tau)]_1 + h(\tau)t(\tau)$$

$\Rightarrow$ Now our ZKP is succinct with only 3 elliptical curve point A_1, B_2, C_1

GROTH16

⇒ The prover can currently send any $[A]_1$, $[B]_2$ & $[C]_1$ points to the verifier.

⇒ If we add a G_{12} point with unknown discrete logarithm to $[A]_1 \cdot [B]_2 = [C]_1 \cdot G_2$ then it will be impossible to judge the proof.

$$[A]_1 \cdot [B]_2 = \underbrace{[\alpha]_1 [\beta]_2}_{} + [C]_1 \cdot G_2$$

$[\alpha]_1, [\beta]_2$ are generated during trusted setup. α & β are destroyed.

$$\sum_{i=1}^{m} a_i u_i(x) \cdot \sum_{i=1}^{m} a_i v_i(x) = \sum_{i=1}^{m} a_i w_i(x) + h(x)t(x)$$

$[\,l_i \;,\; \phi\,] \rightarrow$ Random points

$$\left(\sum_{i=1}^{m} a_i u_i(x) + l_i\right)\left(\sum_{i=1}^{m} a_i v_i(x) + \phi\right)$$

$$\Rightarrow \sum_{i=1}^{m} a_i u_i(x) \sum_{i=1}^{m} a_i v_i(x) + l_i \sum_{i=1}^{m} a_i v_i(x)$$

$$+ \phi \sum_{i=1}^{m} a_i u_i(x) + l_i \phi$$

$$\therefore \sum_{i=1}^{m} a_i u_i(x) \cdot \sum_{i=1}^{m} a_i v_i(x) = \sum_{i=1}^{m} a_i w_i(x) + h(x)t(x)$$

$$\Rightarrow \sum_{i=1}^{m} a_i w_i(x) + h(x)t(x) + l_i \sum_{i=1}^{m} a_i v_i(x)$$

$$+ \phi \sum_{i=1}^{m} a_i u_i(x) + l_i \phi$$

$$\sum_{i=1}^{m} f(x) + \sum_{i=1}^{m} g(x) = \sum_{i=1}^{m} f(x) + g(x)$$

$$\Rightarrow \sum_{i=1}^{m} \left(a_i w_i(x) + \xi a_i v_i(x) + \phi a_i u_i(x)\right) + \xi\phi + h(x)t(x)$$

$$\Rightarrow \overbrace{\boxed{\left(\sum_{i=1}^{m} a_i u_i(x) + \xi\right)}}^{[A]_1} \overbrace{\boxed{\left(\sum_{i=1}^{m} a_i v_i(x) + \phi\right)}}^{[B]_2}$$

$$= \underbrace{\boxed{\sum_{i=1}^{m} \left(a_i w_i(x) + \xi a_i v_i(x) + \phi a_i u_i(x)\right) \quad h(x)\,t(x)}}_{[C]_1} + \underbrace{\boxed{\xi\phi}}_{\alpha\beta}$$

$$\Rightarrow [A]_1 \cdot [B]_2 = [C]_1 G_2 + [\alpha]_1 [\beta]_2$$

$$\sum_{i=1}^{m} \left(a_i w_i(x) + \xi a_i v_i(x) + \phi a_i u_i(x) \right)$$

$$\Rightarrow \sum_{i=1}^{m} a_i \left(w_i(x) + \xi v_i(x) + \phi u_i(x) \right)$$

$$\psi_i = \left(w_i(x) + \xi v_i(x) + \phi u_i(x) \right) G_1$$

$$[\ldots\ \psi_3, \psi_2, \psi_1, \psi_0] = \psi_n$$

$\Rightarrow$ ψ_n is generated during trusted setup.

$$\Rightarrow [C]_1 = \sum_{i=0}^{m} a_i \psi_i$$

$$\Rightarrow \boxed{\left(\sum_{i=0}^{m} a_i u_i(x) + \alpha \right) \left(\sum_{i=0}^{m} a_i v_i(x) + \beta \right) = \sum_{i=0}^{m} a_i \psi_i + \alpha\beta}$$

⇒ So far we have acheived zero knowledgeness, succinctness, non-interactivity, & argument of knowledge.

⇒ But our SNARK does not have the the ability to hold public signals that the verifier can see.

⇒ All inputs are Elliptic curve points.

$$\sum_{i=1}^{m} a_i u_i(x) \cdot \sum_{i=1}^{m} a_i v_i(x) = \sum_{i=1}^{m} a_i w_i(x) + h(x)t(x)$$

$$\Rightarrow \sum_{i=1}^{m} a_i w_i(x) = \underbrace{\sum_{i=1}^{l} a_i w_i(x)}_{} + \sum_{i=l+1}^{m} a_i w_i(x)$$

we can make the a_i values here public.

$$\sum_{i=1}^{m} a_i w_i(x) = \sum_{i=1}^{m} a_i \psi_i$$

$$\Rightarrow a_1 \psi_1 + a_2 \psi_2 + a_3 \psi_3 \dots a_m \psi_m$$

If $\ell = 2$, ie; the public signals are 2

$$\Rightarrow \underbrace{a_1 \psi_1 + a_2 \psi_2}_{\text{evaluated by the verifier.}} + \underbrace{a_3 \psi_3 \dots a_m \psi_m}_{\text{evaluated by the prover.}}$$

Verification

$$A \cdot B \stackrel{?}{=} [\alpha]_1 [\beta]_2 + \underbrace{C \cdot G_2}_{\sum_{i=\ell+1}^{m} a_i \psi_i} + \underbrace{X \cdot G_2}_{\sum_{i=1}^{\ell} a_i \psi_i}$$

$$\Rightarrow a_1\psi_1 + a_2\psi_2 + a_3\psi_3 + \underset{\displaystyle \Big\downarrow \atop \displaystyle a_4'\psi_2}{\boxed{a_4\psi_4}} + a_5\psi_5$$

→ The prover might provide an invalid witness paired with a verification key ψ that may produce a valid proof depending on the problem.

→ We introduce 2 unknown points $[\delta]_2$ & $[\gamma]_2$ that are generated during the trusted setup process.

$$A\cdot B = [\alpha]_1[\beta]_2 + C[\delta]_2 + X[\gamma]_2$$

→ On computing ψ_i during trusted setup we divide ψ_i with δ or γ.

$$\psi_i = \frac{\alpha v_i(\tau) + \beta u_i(\tau) + w_i(\tau)}{\gamma} \quad \text{if } i \leq l$$

$$\psi_i = \frac{\alpha v_i(\tau) + \beta u_i(\tau) + w_i(\tau)}{\delta} \quad \text{if } i > l$$

→ To prevent attackers from guessing our witness (its possible when the witness options are less), we add a random salt to [A] [B] & [C]

$$[A]_1 = [\alpha]_1 + \sum_{i=1}^{m} a_i u_i(\tau) + r[\delta]_1$$

$$[B]_2 = [\beta]_2 + \sum_{i=1}^{m} a_i v_i(\tau) + s[\delta]_2$$

$$[B]_1 = [\beta]_1 + \sum_{i=1}^{m} a_i v_i(\tau) + s[\delta]_1$$

$$[C]_1 = \sum_{i=0}^{m} a_i [\beta u_i(\tau) + \alpha v_i(\tau) + w_i(\tau)]_1 + s[A]_1 + r[B]_1 - rs[\delta]_1$$

$$\text{proof} = ([A]_1, [B]_2, [C]_1)$$

Summary

Step 1 :

Write a given problem in terms of a set of constrained equation with only addition & multiplication.

Step 2 :

Convert the set of constrained equations to R1CS form (matrix) such that you only have a maximum of 1 multiplication in the equations. It is done through substitution.

Step 3 :

Convert the R1CS form to a QAP (polynomial) form.

Step 4 :

Evaluate the resulting QAP at τ, which is provided by the trusted setup.

Step 5 :

Setup proof and verification based on the algorithm. ie; groth16, plonky, etc.