zero-knowledge of Succinct Non-interactive ARguments of Knowledgs

- [GGPR13] @ Eurocrypt'13 proposed a Quadratic Arithmetic Program (QAP) and Quadratic Span Program (QSP).
  - translation of computations into polynomials
  - a basis for SNARKs construction
  - https://eprint.iacr.org/2012/215.pdf
- 주어진 입/출력에 대한 연산의 중간 과정을 압축하여 증명
- 함수 F 에 대해 입력값을 넣으면 출력값이 나오고, 이렇게 출력값이 나오게 하는 중간값도 알고 있음을 증명

Overview of Constructing SNARK

- Computation (Equation) $\Rightarrow$ Circuits $\Rightarrow$ QAP $\Rightarrow$ SNARK

Def 1. (Arithmetic and Boolean Circuit)

- *An arithmetic circuit* consists of wires that carry values from $\mathbb{F}_p$ and connect to addition and multiplication gate.
- *Boolean circuits* operate over bits, with bitwise gates for `AND`, `OR`, `XOR`, etc.

Def 2. (Quadratic Arithmetic Program)

A QAP Q over a field $\mathbb{F}$ contains three sets of $m+1$ polynomials $\mathcal{V}=\{v_k(x)\}$, $\mathcal{W}=\{w_k(x)\}$, $\mathcal{Y}=\{y_k(x)\}$ for $k\in\{0,\dots,m\}$ and a target polynomial $t(x)$ and define

$$p(x)=\left(v_0(x)+\sum c_k v_k(x)\right)\left(w_0(x)+\sum c_k w_k(x)\right)-\left(y_0(x)+\sum c_k y_k(x)\right)$$

Suppose F is a function that takes as input n elements of $\mathbb{F}$ and outputs n' elements and let $N=n+n'$. Then we say that Q *computes* F if: $(c_1,\dots,c_N)$ is a valid assignment of F's inputs and outputs if and only if there exists coefficients $(C_{N+1},\dots,c_m)$ such that $t(x)$ divides $p(x)$.

Constructing a QAP Q for an arithmetic circuit C

1. For each multiplication gate g in C, pick an aribtrary root $r_g\in\mathbb{F}$.
2. Define the target polynomial $T(x):=\prod(x-r_g)$.
3. Label an index $k\in\{1,\dots,m\}$ to each input of the circuit and each output from a multiplication gate.
4. Interpolate the polynomials in $\mathcal{V},\mathcal{W}$ and $\mathcal{Y}$ using Lagrange interpolation technique
   - $\mathcal{V}$: the set of polynomials encoding the left input into each gate such that
     - $v_k(x)=1$, if k-th wire is a left input to gate g
     - $v_k(x)=0$, otherwise

- $\mathcal{W}$: the set of polynomials encoding the right input into each gate such that
    - $w_k(x) = 1$, if k-th wire is a right input to gate g
    - $w_k(x) = 0$, oterwise
- $\mathcal{Y}$: the set of polynomials encoding the output from each gate such that
    - $y_k(x) = 1$, if k-th wire is output from gate g
    - $y_k(x) = 0$, otherwise

5. Define $V(x) := \sum c_k v_k(x)$, $W(x) = \sum c_k w_k(x)$, and $Y(x) = \sum c_k y_k(x)$, where $(c_1, \ldots, c_m)$ is an assignment of C.
6. Define $P(x) := V(x) \cdot W(x) - Y(x)$
    - Then, $T(x)$ divides $P(x)$, that is, there exists $H(x)$ such that $P(x) = H(x)T(x)$

Example.

Suppose Alice watns to prove to Bob she knows $c_1, c_2, c_3 \in \mathbb{Z}_p^*$ s.t. $c_1 \cdot c_2 \cdot (c_1 + c_3) = 7$

For this circuit, a legal assignment is of the form:

$$(c_1, \ldots, c_5) \text{ where } c_4 = c_1 \cdot c_2 \text{ and } c_5 = c_4 \cdot (c_1 + c_2)$$

Therefore, what Alice wants to prove is that <u>she knows a legal assignment $(c_1, \ldots, c_5)$ s.t. $c_5 = 7$</u>

0. Express $c_1 \cdot c_2 \cdot (c_3 + c_4) = 7$ as an arithmetic circuit
1. Suppose g_1 is associated with $1 \in \mathbb{F}_p$ and g_2 with $2 \in \mathbb{F}_p$
2. A target polynomial is defined by $t(x) = (x-1)(x-2)$
3. Label each input and output of the multiplication gate
4. By definition of $\mathcal{V} = \{v_k(x)\}$, $\mathcal{W} = \{w_k(x)\}$, $\mathcal{Y} = \{y_k(x)\}$,

$v_1(g_1) = 1$	$v_1(g_2) = 0$	$w_1(g_1) = 0$	$w_1(g_2) = 1$	$y_1(g_1) = 0$	$y_1(g_2) = 0$
$v_2(g_1) = 0$	$v_2(g_2) = 0$	$w_2(g_1) = 1$	$w_2(g_2) = 0$	$y_2(g_1) = 0$	$y_2(g_2) = 0$
$v_3(g_1) = 0$	$v_3(g_2) = 0$	$w_3(g_1) = 0$	$w_3(g_2) = 1$	$y_3(g_1) = 0$	$y_3(g_2) = 0$
$v_4(g_1) = 0$	$v_4(g_2) = 1$	$w_4(g_1) = 0$	$w_4(g_2) = 0$	$y_4(g_1) = 1$	$y_4(g_2) = 0$
$v_5(g_1) = 0$	$v_5(g_2) = 0$	$w_5(g_1) = 0$	$w_5(g_2) = 0$	$y_5(g_1) = 0$	$y_5(g_2) = 1$

- $v_1(x) = w_2(x) = y_4(x) = 2 - x$
    - $v_1(1) = w_2(1) = y_4(1) = 1$
    - $v_1(2) = w_2(2) = y_4(2) = 0$
- $v_4(x) = w_1(x) = w_3(x) = y_5(x) = x - 1$
    - $v_4(1) = w_1(1) = w_3(1) = y_5(1) = 0$
    - $v_4(2) = w_1(2) = w_3(2) = y_5(2) = 1$

5. Given fixed assignment $(c_1, \ldots, c_5)$,
    - $V(x) = \sum_{i=1}^{5} c_i v_i(x)$
    - $W(x) = \sum_{i=1}^{5} c_i w_i(x)$

- $Y(x) = \sum_{i=1}^{5} c_i y_i(x)$
    - $P(x) = V(x) \cdot W(x) - Y(x)$
6. $(c_1, \ldots, c_5)$ is a legal assignment if and only if $P(1) = P(2) = 0$
    - $P(1) = V(1) \cdot W(1) - Y(1) = c_1 \cdot c_2 - c_4 = 0$
    - $P(2) = V(2) \cdot W(2) - Y(2) = c_4 \cdot (c_1 + c_3) - c_5 = 0$
    - $T(x)$ divides $P(x)$, that is, there exists $H(x)$ s.t. $P(x) = H(x)T(x)$.
    - For an invalid assignment, $T(x)$ does not divide $P(x)$

Pinocchio Protocol

- [PG13] @ S&P'13
- https://eprint.iacr.org/2013/279

Suppose Alice wants to prove to Bob she knows $c_1, c_2, c_3 \in \mathbb{F}$ s.t. $c_1 \cdot c_2 \cdot (c_1 + c_3) = 7$.

Let $\textsf{Com}$ be a (Pedersen) commitment.

1. Alice computes polynomials $V(x), W(x), Y(x)$ and $H(x)$.
2. Bob chooses a random point $s \in \mathbb{F}_p$ and computes $\textsf{Com}(T(s))$
3. Alice computes $\textsf{Com}(V(s)), \textsf{Com}(W(s)), \textsf{Com}(Y(s))$ and $\textsf{Com}(H(s))$ and sends it to Bob
4. Bob checks $\textsf{Com}(V(s) \cdot W(s) - Y(s)) = \textsf{Com}(T(s) \cdot H(s))$
    - If Alice does not have a satisfying assignment, she cannot find $V(x), W(x), Y(x)$ and $H(x)$ s.t. $V(x) \cdot W(x) - Y(x) = H(x) \cdot T(x)$
    - Step 4 does not hold.

Question.

- In Step 3, how to compute $\textsf{Com(T(s))}$ without knowing $T(x)$?
    - Only Alice knows the target polynomial $T(x)$
- In Step 4, how to compute $\textsf{Com}(V(s)), \textsf{Com}(W(s)), \textsf{Com}(Y(s))$ and $\textsf{Com}(H(s))$?
    - Only Bob knows the random point s

Blind Evaulation of a Polynomial

Suppose Alice has a polynomial $P(x) = a_0 + a_1 x + \cdots a_d x^d$ and Bob has a point $s \in \mathbb{F}_p$.

Alice and Bob wish to compute $\textsf{Com(P(s))}$.

- naive method;
    - Alice sends $P(x)$ to Bob and he computes $\textsf{Com}(P(s))$ (and sends it to Alice).
        - Bob *can learn* $p(x)$.
    - Bob sends s to Alice and she computes $\textsf{Com}(P(s))$ (and sends it to Bob).
        - Alice *can learn* s.

Remark. Since Pedersen commitment supports addition,

$$\mathsf{Com}(\mathsf{x})^{\mathsf{a}} \cdot \mathsf{Com}(\mathsf{y})^{\mathsf{b}} = \mathsf{Com}(\mathsf{ax} + \mathsf{by})$$

and thus, Pedersen commitment also supports *linear combinations*

- using Pedersen commitment;
    1. Bob sends to Alice $\mathsf{Com}(1)$, $\mathsf{Com}(s)$, $\ldots$, $\mathsf{Com}(s^d)$
    2. Alice computes

$$\mathsf{Com}(P(s)) = \mathsf{Com}(1)^{a_0} \mathsf{Com}(s)^{a_1} \cdots \mathsf{Com}(s^d)^{a^d}$$

$\Rightarrow$ Alice *cannot* learn s and Bob *cannot* learn $P(x)$, neither.

The Remaining Problem

1. making sure Alice computes her polynomials according to an assignment
    - resolved by introducing "verifiable" blind polynomial evaluation
2. hiding the assignment
    - resolved by masking the polynomials
3. computing multiplications from two commitments
    - resolved by the use of pairing of elliptic curves
4. non-interactive

Boolean Circuits and QSPs

- Quadratic Span Programs (QSPs) are very similar to QAPs.
- QSPs use only two sets of polynomials $\mathcal{V}$ and $\mathcal{W}$ since they only supports Boolean wire values.
- The divisibility check is updated to

$$P(x) = V(x) \cdot W(x)$$