Recall that the flow of constructing SNARK is

- Computation (Equation) $\Rightarrow$ Circuits $\Rightarrow$ QAP $\Rightarrow$ Verifiable Computation $\Rightarrow$ SNARK
  - [GGPR13] @ Eurocrypt'13: construct QAP for a circuit
  - [PG13] @ S&P'13 (Pinocchio): construct a verifiable computation protocol
    - [CFH+15] @ S&P'15 (Geppetto)
    - [DFK+16] @ S&P'16 (Cinderella)
  - [BCH+13] @ TCC'13: construct SNARK

Pinocchio Protocol

- 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 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 $\mathsf{Com}(T(s))$
3. Alice computes $\mathsf{Com}(V(s)), \mathsf{Com}(W(s)), \mathsf{Com}(Y(s))$ and $\mathsf{Com}(H(s))$ and sends it to Bob
4. Bob checks $\mathsf{Com}(V(s) \cdot W(s) - Y(s)) = \mathsf{Com}(T(s) \cdot H(s))$
   - 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 $\mathsf{Com}(\mathsf{T(s)})$ without knowing $T(x)$?
  - Only Alice knows the target polynomial $T(x)$
- In Step 4, how to compute $\mathsf{Com}(V(s)), \mathsf{Com}(W(s)), \mathsf{Com}(Y(s))$ and $\mathsf{Com}(H(s))$?
  - 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 $\mathsf{Com}(\mathsf{P(s)})$.

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

  - Bob sends s to Alice and she computes $\mathsf{Com}(P(s))$ (and sends it to Bob).
    - Alice *can learn* s.

Remark. Since Pedersen commitment supports addition,

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

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

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