0.1 Probabilistically Checkable Proofs

Before going further we should get acquainted with one more concept from the computational complexity theory, that have an important application in zk-SNARK and provides the theoretical backbone.

A Probabilistically Checkable Proof (PCP) is a type of proof system where the verifier can efficiently check the correctness of a proof by examining only a small, random portion of it, rather than verifying it entirely.

Definition 0.1. A language $\mathcal{L} \subseteq \Sigma^*$ (for some given alphabet Σ) is in the class $\mathsf{PCP}(r, q)$ (**probabilistically checkable proofs**), where r is the *randomness complexity* and q is the *query complexity*, if for a given pair of algorithms $(\mathcal{P}, \mathcal{V})$:

- *Syntax:* $\mathcal{P}$ calculates a proof (bit string) $\pi \in \Sigma^*$ in polynomial time $\mathsf{poly}(|x|)$ of the common input x. The prover $\mathcal{P}$ and verifier $\mathcal{V}$ interact, where the verifier has an oracle access to π (meaning, he queries it at any position).
- *Complexity:* $\mathcal{V}$ uses at most r random bits to decide which part of the proof to query and the verifier queries at most q bits of the proof.

Such pair of algorithms $(\mathcal{P}, \mathcal{V})$ should satisfy the following properties (with a security parameter $\lambda \in \mathbb{N}$):

- **Completeness**: If $x \in \mathcal{L}$, then $\Pr[\mathcal{V}^{\pi}(x) = 1] = 1$.
- **Soundness**: If $x \notin \mathcal{L}$, then for any possible (malicious) proof π^*,

$$\Pr\left[\mathcal{V}^{\pi^*}(x) = 1\right] = \mathsf{negl}(\lambda).$$

This allows a verification of huge statements with high confidence while using limited computational resources.

Theorem 0.2. PCP theorem (PCP characterization theorem)
Any decision problem in NP has a PCP verifier that uses logarithmic randomness $O(\log n)$ and a constant number of queries $O(1)$, independent of n.

$$\mathsf{NP} = \mathsf{PCP}(O(\log n), O(1))$$

0.1.1 PCP application in QAP

When constructing a Quadratic Arithmetic Program (QAP) for a circuit C, we represented the whole circuit's computation using the following relation:

$$L(X)R(X) - O(X) = Z_{\Omega}(X)H(X),$$

where by $L(X)$, $R(X)$, $O(X)$ we denote the polynomials that represent the left, right and output wires of the circuit, respectively. $Z_{\Omega}(X)$ is the target polynomial, while $H(X) := M(X)/Z_{\Omega}(X)$ for master polynomial $M(X) = L(X)R(X) - O(X)$ is the quotient polynomial.

We effectively managed to transform all the circuit's constraints, and computations in the short form. It still allows one to verify that each computational step is preserved by verifying the polynomial evaluation in specific (random) points, instead of recomputing everything. However, it is not quite clear why such a check is safe and how it can be used in a PCP. In other words,

why checking that $L(s)R(s) - O(s) = Z_\Omega(s)H(s)$ for randomly selected s is enough to verify the circuit $\mathcal{C}$?

Soundness justification. Why is it safe to use such a check? As it was said early, we perform all the computations in some finite field $\mathbb{F}$. The polynomials $L(X)$, $R(X)$ and $O(X)$ are interpolated polynomials using $|\mathcal{C}|$ (number of gates) points, so

$$\deg(L) \leq |\mathcal{C}|\,, \quad \deg(R) \leq |\mathcal{C}|\,, \quad \deg(O) \leq |\mathcal{C}|$$

Thus, using properties of polynomials' degrees, we can estimate the degree of polynomial $M(X) = L(X)R(X) - O(X)$.

$$\deg(M) \leq \max\{\deg(A) + \deg(B), \deg(C)\} = \max\{2\,|\mathcal{C}|\,, |\mathcal{C}|\} = 2\,|\mathcal{C}|$$

Now, using the Schwartz-Zippel Lemma **??**, we can deduce that if an adversary $\mathcal{A}$ does not know a valid witness $\mathbf{w}$, resolving the circuit $\mathcal{C}$, he can compute a polynomial $\widetilde{M}(X) \leftarrow \mathcal{A}(\cdot)$ that satisfies a verifier $\mathcal{V}_\mathcal{C}$ with probability less then $2\,|\mathcal{C}|\,/|\mathbb{F}|$. To put it formally, we can write:

$$\Pr_{s \xleftarrow{R} \mathbb{F}}[\widetilde{M}(s) = M(s)] \leq \frac{2\,|\mathcal{C}|}{|\mathbb{F}|}$$

This probability becomes negligible as $|\mathbb{F}|$ grows large (which is typically the case), giving us soundness. In the same time, the verifier accepts the $M(X)$ generated using a valid witness with probability 1 giving us the completeness, so, we can categorize QAP as PCP.

We will modify the form of our proof with the next modifications, but still preserve the PCP properties.

0.1.2 Encrypted Verification

Now, assume we have the cyclic group $\mathbb{G}$ of prime order r with a generator g. Typically, this is the group of points on an elliptic curve. Assume that $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a symmetric pairing function, where $\mathbb{G}_T$ is a target group.

Now, suppose during the setup phase, we have a trusted party that generated a random value τ and public parameters $g^\tau, g^{\tau^2}, \ldots, g^{\tau^d}$ for $d = 2|\mathcal{C}|$ — maximum degree of used polynomials (later we will use notation $\{g^{\tau^i}\}_{i\in[d]}$ for brevity). This way, we can find the KZG commitment for each polynomial. Thus, for example,

$$g^{L(\tau)} = g^{\sum_{i=0}^{d} L_i \tau^i} = \prod_{i=0}^{d} (g^{\tau^i})^{L_i},$$

and the same goes for $g^{R(\tau)}, g^{O(\tau)}, g^{H(\tau)}, g^{Z_\Omega(\tau)}$. Now, given these give points, how can we verify that the polynomial $M(X) = L(X)R(X) - O(X)$ is correct? Well, first notice that the check is equivalent to

$$L(\tau)R(\tau) = Z_\Omega(\tau)H(\tau) + O(\tau).$$

Notice that we transferred $O(\tau)$ to the right side of the equation to further avoid finding the inverse. Now, we can check this equality using encrypted values as follows:

$$e(g^{L(\tau)}, g^{R(\tau)}) = e(g^{Z_\Omega(\tau)}, g^{H(\tau)}) \cdot e(g^{O(\tau)}, g),$$