Modern Zero Knowledge Cryptography - MIT IAP, 2023.1

# Solutions 5: Commitment Schemes

*Lecturer: Ying Tong*

1. The $\mathsf{Setup}$ phase of the $\mathsf{KZG}$ polynomial commitment scheme involves computing commitments to powers of a secret evaluation point $\tau$. This is called the “trusted setup" and is often generated in a multi-party computation known as the “Powers of Tau" ceremony. One day, you find the value of $\tau$ on a slip of paper. How can you use it to make a fake $\mathsf{KZG}$ opening proof?

   Recall that the $\mathsf{KZG}$ commitment to a polynomial $p(X)$ is

   $$C = \mathsf{Commit}(p(X)) = [p(\tau)]_1.$$

   An opening proof for the claim “$C = \mathsf{Commit}(p(X)), p(x) = y$" is a group element $\pi \in \mathbb{G}_1$. Let's write this as $\pi = [\pi_{\mathsf{inner}}]_1$. The verifier checks this proof using the following equation:

   $$\begin{aligned} e(\pi, [\tau - x]_2) &\stackrel{?}{=} e(C - [y]_1, G_2) \\ \implies e([\pi_{\mathsf{inner}}]_1, [\tau - x]_2) &= e([p(\tau)]_1 - [y]_1, G2). \end{aligned}$$

   The equation that we check “in the exponent" is:

   $$\begin{aligned} \pi_{\mathsf{inner}} \cdot (\tau - x) &\stackrel{?}{=} p(\tau) - y \\ \implies \pi_{\mathsf{inner}} &= \frac{p(\tau) - y}{\tau - x}. \end{aligned}$$

   For some invalid $y' \neq p(x)$, we now replace $\pi_{\mathsf{inner}}$ with a fake

   $$\begin{aligned} \pi^{\mathsf{Fake}}_{\mathsf{inner}} &= \frac{p(\tau) - y'}{\tau - x} \\ \implies \pi^{\mathsf{Fake}} &= \left[\frac{p(\tau) - y'}{\tau - x}\right]_1 \\ &= [p(\tau) - y']_1^{\frac{1}{\tau - x}} \\ &= (C - [y']_1)^{\frac{1}{s - z}}. \end{aligned}$$

2. Construct a **vector commitment scheme** from the $\mathsf{KZG}$ polynomial commitment scheme. (Hint: For a vector $\vec{m} = (m_1, \ldots, m_q)$, is there an "interpolation polynomial" $I(X)$ such that $I(i) = m[i]$?)

   A vector $\vec{m} = (m_1, \ldots, m_q)$ can be viewed as the evaluations of a polynomial at points $1, dots, m$. We can construct a Lagrange interpolation polynomial $I(X)$ such that

   $$I(X) = \sum_i m_i \cdot \mathcal{L}_i(X) = \begin{cases} m_i \text{ if } X = 1, \\ 0 \text{ otherwise.} \end{cases}$$

   (Here, the Lagrange basis polynomial $\mathcal{L}_i(X) = \prod_{j, j \neq i} \frac{X-j}{i-j}$ is $1$ if $X = i$, and $0$ otherwise.) The $\mathsf{KZG}$ commitment to $I(X)$ can be queried at the vector positions $x \in [m]$ to give a vector commitment.

   **Fun fact:** *The Verkle tree [1] is a Merkle tree that uses a* ***vector commitment*** *instead of a hash function. Using the* $\mathsf{KZG}$ *vector commitment scheme, can you see why a Verkle tree is more efficient?*

   In a depth $n$ $k$-ary tree, the Merkle tree construction would give an authentication path of size $k \log n$ ($k$ child nodes at each of the $\log n$ levels). In contrast, a Verkle tree authentication path consists of only $\log n$ $\mathsf{KZG}$ opening proofs: at each parent node, we commit to a vector of child nodes, and provide a constant-size opening proof. We can further reduce the proof size to a **single** $\mathsf{KZG}$ proof by cleverly capturing the parent-child relationships in a single polynomial. See *Verkle trees* (Vitalik Buterin, 2018) for the construction.

3. The KZG polynomial commitment scheme makes an opening proof $\pi$ for the relation $p(x) = y$. Can you extend the scheme to produce a multiproof $\pi$, that convinces us of $p(x_i) = y_i$ for a list of points and evaluations $(x_i, y_i)$? (Hint: assume that you have an interpolation polynomial $I(X)$ such that $I(x_i) = y_i$.)

   The KZG opening proof is $\pi = [q(\tau)]_1$, where the quotient polynomial

   $$q(X) := \frac{p(X) - y}{X - x}.$$

   This is a well-constructed polynomial if and only if $x$ is a root of $p(X) - y$; in other words, if $p(x) - y = 0$. To extend this proof to multiple points and evaluations $\{(x_i, y_i)\}$, we set

   $$q_{\mathsf{multiproof}}(X) := \frac{p(X) - I(X)}{\prod(X - x_i)}.$$

   Now, this checks that every $x_i$ is a root of $p(X) - I(X)$; in other words, that

   $$p(x_i) - I(x_i) = p(x_i) - y_i = 0 \text{ at every } x_i.$$

# References

[1] J. Kuszmaul. Verkle trees. https://math.mit.edu/research/highschool/primes/materials/2018/Kuszmaul.pdf, 2019.