Understanding Hiding KZG10

Hiding KZG10 is a variant of the KZG10 protocol that produces polynomial commitments with a random blinding factor, thus possessing the property of Perfect Hiding. This means that even if an attacker has unlimited computational power, they cannot reverse-engineer any information about the polynomial from the commitment. While Hiding KZG10 is not common, it is an important component in constructing zkSNARKs with Zero-knowledge properties or other secure protocols.

This article introduces two different Hiding KZG10 schemes. The first scheme is from [KT23], and its main technique is a simplified version of multivariate polynomial commitment from [PST13], [ZGKPP17], and [XZZPS19]. The second scheme is from [CHMMVW19], which is an improvement on the original KZG10 protocol paper [KZG10].

None-hiding KZG10

Let's first recall the basic KZG10 protocol. KZG10 is based on a pre-setup (Universal Trusted Setup) SRS:

$$SRS = ([1]_1, [\tau]_1, [\tau^2]_1, [\tau^3]_1, \ldots, [\tau^D]_1, [1]_2, [\tau]_2) \tag{1}$$

Here, τ is a secret value that needs to be forgotten after the Setup phase, otherwise, any party knowing τ could launch an attack. We use bracket notation $[a]_1$ to represent scalar multiplication (Scalar Multiplication) $a \cdot G$ on an elliptic curve group element, where $G \in \mathbb{G}_1$ is the generator of the group. The SRS consists of group elements from $\mathbb{G}_1$ and $\mathbb{G}_2$, which we call Base elements, as subsequent commitments to polynomials are based on linear operations of these Base elements.

Since division operations on elliptic curve group elements are a difficult problem, if our computing power is limited, we cannot calculate a from $[a]_1$. This can be seen as, once we multiply a value $a \in \mathbb{F}_r$ by a Base element, a is hidden.

KZG10 requires a pairing friendly curve, meaning there exists another elliptic curve group $\mathbb{G}_2$ (with generator G'), where each element is represented as $[b]_2$, i.e., $b \cdot G'$. And there exists a bilinear pairing operation that satisfies the following bilinearity and non-degeneracy properties:

$$e(a \cdot G, b \cdot G') = (ab) \cdot e(G, G') \tag{2}$$

Now assume we have a univariate polynomial $f(X) \in \mathbb{F}_r[X]$

$$f(X) = f_0 + f_1 X + f_2 X^2 + \cdots + f_d X^d \tag{3}$$

Here, the Degree d of the polynomial needs to satisfy $d < D$. Then the commitment C_f of the polynomial is calculated as follows:

$$C_f = \mathsf{Commit}(f(X)) = f_0 \cdot [1]_1 + f_1 \cdot [\tau]_1 + f_2 \cdot [\tau^2]_1 + \cdots + f_d \cdot [\tau^d]_1 \tag{4}$$

After derivation, it's not hard to find that the following equation holds

$$C_f = [f(\tau)] \tag{5}$$

Evaluation proof

For any polynomial $f(X)$ we choose, which is an element of the polynomial ring $\mathbb{F}_r[X]$, it satisfies the following division formula:

$$f(X) = q(X) \cdot g(X) + r(X) \tag{6}$$

where $g(X)$ is the divisor polynomial, $r(X)$ is the remainder polynomial. Obviously, $\deg(r) < \deg(g)$.

If we let $g(X) = X - z$, then clearly $r(X)$ is a constant polynomial, so the above division decomposition formula can be written as:

$$f(X) = q(X) \cdot (X - z) + r \tag{7}$$

Furthermore, substituting $X = z$ into the above equation, we can get $f(z) = r$. So, the above division decomposition formula can be rewritten as:

$$f(X) = q(X) \cdot (X - z) + f(z) \tag{8}$$

This formula is the core formula of the KZG10 protocol. That is, if we want to prove $f(z) = r$, we only need to prove that $f(X) - f(z)$ can be divided by $(X - z)$. Or in other words, there exists a quotient polynomial $q(X)$ satisfying:

$$q(X) = \frac{f(X) - f(z)}{X - z} \tag{9}$$

If the value of $f(X)$ at $X = z$ is not equal to r, then $q(X)$ is not a polynomial, but a Rational Function. And for any Rational Function whose denominator is not a constant polynomial, we cannot use the above SRS to calculate its commitment.

Therefore, the Prover only needs to send the commitment of $q(X)$ to prove the existence of $q(X)$ to the Verifier, which is equivalent to proving that the evaluation of $f(X)$ is correct.

$$\pi_{eval} = q_0 \cdot [1]_1 + q_1 \cdot [\tau]_1 + \cdots + q_{d-1} \cdot [\tau^{d-1}]_1 = [q(\tau)]_1 \tag{10}$$

Then the Verifier uses the Base elements provided by SRS to check the correctness of the decomposition formula. For the Verifier, $f(z)$ and z are public, so the Verifier can check the decomposition formula through the following formula:

$$e\Big([f(\tau)]_1 - f(z) \cdot [1]_1,\ [1]_2\Big) = e\Big([q(\tau)]_1,\ [\tau] - z \cdot [1]_2\Big) \tag{11}$$

The red parts in the above formula are provided by the Prover and do not expose the values inside $[\cdot]_1$.

The Polynomial Evaluation proof of the KZG10 protocol only contains one element $[q(\tau)]_1$ of $\mathbb{G}_1$, with a size of $O(1)$. And the verification algorithm of the Verifier is also $O(1)$. It should be mentioned that the Verifier needs to complete two Pairing calculations, which, although of $O(1)$ complexity, are quite expensive.

Degree Bound proof

KZG10 also supports proving that the Degree of a polynomial $f(X) \in \mathbb{F}_r[X]$ is less than or equal to d.

The Prover's proof method is very straightforward, which is to construct a new polynomial $\hat{f}(X)$,

$$\hat{f}(X) = X^{D-n} \cdot f(X) \tag{12}$$

Obviously, the Degree of $\hat{f}(X)$ is less than or equal to D. And because the highest power of τ in the Base elements contained in the SRS is $[\tau^D]_1$, theoretically, anyone (who doesn't know the value of τ) cannot construct the commitment of any polynomial with a degree greater than or equal to D.

Therefore, the Prover can construct the Degree Bound proof π as:

$$\pi_{deg} = f_0 \cdot [\tau^{D-d}]_1 + f_1 \cdot [\tau^{D-d+1}]_1 + \cdots + f_d \cdot [\tau^D]_1 = [\tau^{D-d} \cdot f(\tau)]_1 \tag{13}$$

The Verifier's verification method is also straightforward, checking whether $\hat{f}(X)$ is obtained by multiplying $f(X)$ with X^{D-d}.

$$e\Big([\tau^{D-d} \cdot f(\tau)]_1,\ [1]_2\Big) = e\Big([f(\tau)]_1,\ [\tau^{D-d}]_2\Big) \tag{14}$$

Here, $[\tau^{D-d}]_2$ should also be a Base element in the SRS. This requires the SRS of KZG10 to include more Base elements of $\mathbb{G}_2$:

$$SRS = \begin{pmatrix} [1]_1, & [\tau]_1, & [\tau^2]_1, & [\tau^3]_1, & \ldots, & [\tau^D]_1 \\ [1]_2, & [\tau]_2, & [\tau^2]_2, & [\tau^3]_2, & \ldots, & [\tau^D]_2 \end{pmatrix} \tag{15}$$

Continuing to think, if the Prover needs to prove both the Degree Bound and Evaluation of $f(X)$ at the same time, then when generating the commitment of $f(X)$, he needs to produce two elements of $\mathbb{G}_1$, $([\hat{f}(\tau)]_1, [\tau^{D-d}\hat{f}(\tau)]_1)$. Then the Verifier needs to complete 4 Pairing calculations to check both the Evaluation and Degree Bound proofs simultaneously.

Perfect Hiding

If we care more about privacy protection during protocol interaction, then KZG10 needs to enhance the protection of the polynomial $f(X)$. In the above KZG10 protocol, we assume that the attacker's computational power is limited, meaning the attacker cannot reverse-engineer any information about $f(X)$ through $[f(\tau)]_1$.

In comparison, the traditional Pedersen Commitment has the property of Perfect Hiding, because its commitment carries a random number as a blinding factor, so even if the attacker has infinite computing power, they cannot reverse-engineer any information about $f(X)$ through $[f(\tau)]_1$.

$$\mathsf{Pedersen.Commit}(f(X), r) = f_0 \cdot G_0 + f_1 \cdot G_1 + \cdots + f_d \cdot G_d + r \cdot G' \tag{16}$$

Where $\{G_i\}_{i=0}^{d} \in \mathbb{G}_1^{d+1}$ and $G' \in \mathbb{G}_1$ are the public parameters of Pedersen Commitment. And $r \in \mathbb{F}_r$ is the so-called blinding factor.

We will now explain how to convert the KZG10 protocol into a Perfect Hiding protocol. This scheme is from [KT23], and its basic idea comes from [ZGKP17] and [PST13].

First, we can "try" to consider adding a random number as a blinding factor when KZG10 commits to $f(X)$, for example:

$$\mathsf{KZG.Commit}(f(X), r) = f_0 \cdot [1]_1 + f_1 \cdot [\tau]_1 + \cdots + f_d \cdot [\tau^d]_1 + r \cdot [1]_1 \tag{17}$$

But such a commitment would have security issues. Because in Pedersen Commitment, the element G' used to commit r is a special Base element, and its relationship with other G_i is unknown (i.e., independent). Therefore, the introduction of r does not affect the commitment of the constant term f_0 of $f(X)$.

Therefore, we need to expand the SRS and introduce an additional preset random value γ, specifically used to commit the blinding factor r:

$$SRS = ([1]_1, [\tau]_1, [\tau^2]_1, [\tau^3]_1, \ldots, [\tau^D]_1, [\gamma]_1, [1]_2, [\tau]_2, [\gamma]_2) \tag{18}$$

Then the commitment of $f(X)$ is defined as:

$$\mathsf{KZG.Commit}(f(X), r) = f_0 \cdot [1]_1 + f_1 \cdot [\tau]_1 + \cdots + f_d \cdot [\tau^d]_1 + r \cdot [\gamma]_1 \tag{19}$$

We will use a shorter symbol $\mathsf{cm}(f)$ below to represent the commitment of $f(X)$.

Evaluation proof of Hiding-KZG10

Although we add a Blinding Factor to the Commitment, the Evaluation proof of $f(X)$ may still expose information about $f(X)$.

Imagine if the Prover wants to prove $f(z) = v$ to the Verifier, he needs to calculate the quotient polynomial $q(X)$, compute and send its commitment $[q(\tau)]_1$. If $[q(\tau)]_1$ is sent directly, this would break the Perfect Hiding property we want, because an attacker with "infinite computing power" could reverse-engineer $q(X)$ from $[q(\tau)]_1$, and then continue to calculate $f(X)$.

Therefore, we also need to add another different blinding factor to $[q(\tau)]_1$, denoted as s:

$$\begin{aligned} \mathsf{KZG.Commit}(q(X), s) &= q_0 \cdot [1]_1 + q_1 \cdot [\tau]_1 + \cdots + q_d \cdot [\tau^{d-1}]_1 + s \cdot [\gamma]_1 \\ &= [q(\tau) + s \cdot \gamma]_1 \end{aligned} \tag{20}$$

We denote the commitment of $q(X)$ with the blinding factor added as the short symbol $\mathsf{cm}(q)$.

Continuing to recall, the Verifier of Non-hiding KZG10 needs to check the following equation to verify the commitment of $q(X)$:

$$e\Big([f(\tau)]_1 - f(z) \cdot [1]_1,\ [1]_2\Big) = e\Big([q(\tau)]_1,\ [\tau] - z \cdot [1]_2\Big) \tag{21}$$

However, in Hiding-KZG10, since both the polynomial commitment $\mathsf{cm}(f)$ and the quotient polynomial commitment $\mathsf{cm}(q)$ have blinding factors, the Verifier can no longer complete the verification according to the above Pairing equation:

$$e\Big([f(\tau) + r \cdot \gamma]_1 - f(z) \cdot [1]_1,\ [1]_2\Big) \neq e\Big([q(\tau) + s \cdot \gamma]_1,\ [\tau] - z \cdot [1]_2\Big) \tag{22}$$

Let's reason why the above equation doesn't hold. First, look at the left side of the equation, which is equivalent to calculating

$$\begin{aligned} lhs &= f(\tau) + r \cdot \gamma - f(z) \\ &= f(\tau) - f(z) + r \cdot \gamma \end{aligned} \tag{23}$$

The right side of the equation is equivalent to calculating

$$\begin{aligned} rhs &= (q(\tau) + s \cdot \gamma) \cdot (\tau - z) \\ &= q(\tau) \cdot (\tau - z) + s \cdot (\tau - z) \cdot \gamma \\ &= f(\tau) - f(z) + s \cdot (\tau - z) \cdot \gamma \end{aligned} \tag{24}$$

The difference between the left and right sides is

$$\begin{aligned} lhs - rhs &= r\cdot\gamma - s\cdot(\tau - z)\cdot\gamma \\ &= (r - s\cdot(\tau - z))\cdot\gamma \end{aligned} \tag{25}$$

To allow the Verifier to verify, we need to introduce an additional "group element" to balance the Pairing verification formula:

$$E = r\cdot[1]_1 - s\cdot[\tau]_1 + (s\cdot z)\cdot[1]_1 \tag{26}$$

Thus, the Verifier can verify through the following formula:

$$e\Big([f(\tau) + r\cdot\gamma] - f(z)\cdot[1]_1,\ [1]_2\Big) = e\Big([q(\tau) + s\cdot\gamma],\ [\tau] - z\cdot[1]_2\Big) + e\Big(E,\ [\gamma]_2\Big) \tag{27}$$

Or written as:

$$e\Big(\mathsf{cm}(f) - f(z)\cdot[1]_1,\ [1]_2\Big) = e\Big(\mathsf{cm}(q),\ [\tau] - z\cdot[1]_2\Big) + e\Big(E,\ [\gamma]_2\Big) \tag{28}$$

Where the red parts are provided by the Prover, and the blue parts are public values.

Degree Bound proof of Hiding-KZG10

To prove the Degree Bound of $f(X)$, we also need to add a Blinding Factor to the polynomial $\hat{f}(X)$, then calculate its commitment as the Degree Bound proof of $f(X)$:

$$\mathsf{cm}(\hat{f}) = [\tau^{D-d}\cdot f(\tau)]_1 + \eta\cdot[\gamma]_1 \tag{29}$$

At the same time, an additional element $E\in\mathbb{G}_1$ is needed for balancing,

$$E = \rho\cdot[\tau^{D-d}]_1 - \eta\cdot[1]_1 \tag{30}$$

This way, the Verifier can verify the Degree Bound proof of $f(X)$ through the following equation:

$$e\Big(\mathsf{cm}(f),\ [\tau^{D-d}]_2\Big) = e\Big(\mathsf{cm}(\hat{f}),\ [1]_2\Big) + e\Big(E,\ [\gamma]_2\Big) \tag{31}$$

Readers can verify for themselves why the above equation holds.

Evaluation-and-degree-bound proof of Hiding KZG10

Suppose for the same Polynomial $f(X)$, the Prover needs to prove both the Evaluation and Degree Bound of $f(X)$ simultaneously. If we use the above Evaluation and Degree Bound proof protocols separately, the Prover would need to send two $\mathbb{G}_1$ elements, and then the Verifier would need to complete 4 Pairing calculations. In fact, we can combine these two proof steps into one: the Prover only sends two $\mathbb{G}_1$ elements, and the Verifier only needs to use two Pairings to complete the verification.

The Prover needs to construct two $\mathbb{G}_1$ elements,

$$\mathsf{cm}(q) = [\tau^{D-d}\cdot q(\tau)]_1 + \eta\cdot[\gamma]_1 \tag{32}$$

Another element E is defined as:

$$E = \rho\cdot[\tau^{D-d}]_1 - \eta\cdot[\tau]_1 + (\eta\cdot z)\cdot[1]_1 \tag{33}$$

The Prover sends the proof

$$\pi = (\mathsf{cm}(q), E) \tag{34}$$

And the Verifier needs to verify the following equation:

$$e\Big(\mathsf{cm}(f) - f(z)\cdot[1]_1,\ [\tau^{D-d}]_2\Big) = e\Big(\mathsf{cm}(q),\ [\tau] - z\cdot[1]_2\Big) + e\Big(E,\ [\gamma]_2\Big) \tag{35}$$

Another construction of Hiding KZG10

In the original [KZG10] paper, a scheme for achieving Perfect Hiding was also provided. We can compare these two different styles of Hiding KZG10 variants.

The idea of this scheme is to add a random polynomial $r(X)$ when committing to $f(X)$, rather than just a single random blinding factor. Here, $f(X)$ and $r(X)$ are defined as follows:

$$f(X) = \sum_{i=0}^{d} f_i \cdot X^i \qquad r(X) = \sum_{i=0}^{d} r_i \cdot X^i \tag{36}$$

Note that here, the Degree of the blinding polynomial $r(X)$ is consistent with the Degree of $f(X)$. To support the blinding polynomial (Blinding Polynomial), the SRS produced in the initial Setup phase needs to introduce a random number γ to isolate the blinding factor from the normal message to be committed. So the SRS is expanded to:

$$SRS = \begin{pmatrix} [1]_1, & [\tau]_1, & [\tau^2]_1, & [\tau^3]_1, & \ldots, & [\tau^D]_1 \\ [\gamma]_1, & [\gamma\tau]_1, & [\gamma\tau^2]_1, & [\gamma\tau^3]_1, & \ldots, & [\gamma\tau^D]_1 \\ [1]_2, & [\tau]_2, & [\tau^2]_2, & [\tau^3]_2, & \ldots, & [\tau^D]_2 \end{pmatrix} \tag{37}$$

Below we define the calculation formula for $\mathsf{cm}(f)$:

$$\begin{aligned} \mathsf{KZG10.Commit}(f(X), r(X)) &= \sum_{i=0}^{d} f_i \cdot [\tau^i]_1 + \sum_{i=0}^{d} r_i \cdot [\gamma\tau^i]_1 \\ &= [f(\tau) + \gamma \cdot r(\tau)]_1 \end{aligned} \tag{38}$$

Essentially, the commitment to the polynomial $f(X)$ is actually a commitment to $\bar{f}(X) = f(X) + \gamma \cdot r(X)$.

$$\mathsf{cm}(f) = [f(\tau) + \gamma \cdot r(\tau)]_1 = [\bar{f}(\tau)]_1 \tag{39}$$

When the Prover needs to prove $f(z) = v$, he not only needs to send the commitment of the quotient polynomial $q(X)$, but also needs to calculate the value of $r(X)$ at $X = z$.

$$\pi = (\mathsf{cm}(q), r(z)) \tag{40}$$

Where the polynomial $\bar{q}(X)$ is the quotient polynomial after dividing $\bar{f}(X)$ with blinding polynomial by $(X - z)$:

$$\bar{q}(X) = q(X) + \gamma \cdot q'(X) = \frac{f(X) - f(z)}{X - z} + \gamma \cdot \frac{r(X) - r(z)}{X - z} \tag{41}$$

When the Verifier receives $\pi_{eval} = (\mathsf{cm}(\bar{q}), r(z))$, he can verify the following equation:

$$e\Big(\mathsf{cm}(\bar{f}) - f(z)\cdot[1]_1 - r(z)\cdot[\gamma]_1,\ [1]_2\Big) = e\Big(\mathsf{cm}(\bar{q}),\ [\tau] - z\cdot[1]_2\Big) \tag{42}$$

Intuitively, although the Prover sent the value of $r(X)$ at $r(z)$, as long as the Degree of $r(X)$ is greater than or equal to 1, the attacker cannot reverse-engineer $r(X)$ through the value of $r(z)$ alone, so there is at least one random factor still protecting $f(X)$.

In fact, if we know that $f(X)$ will be opened at most $k < d$ times throughout its lifecycle, then we don't need to force the Degree of $r(X)$ to be d, but it can be a polynomial of Degree k. Because the k-degree blinding factor polynomial consists of $k+1$ random factors, when $r(X)$ is calculated k times, there is still one random factor protecting the commitment of $f(X)$.

Take an extreme example where the Degree of $r(X)$ is 1, then when the Prover proves the value at a different point again, say $f(z') = v'$, the Verifier would have the ability to recover $r(X)$, thus breaking the Perfect Hiding property of the commitment to $f(X)$.

Evaluation-with-degree-bound proof

The next question is, in this Hiding-KZG10 scheme, can we prove $f(z) = v$ and $\deg f \leq d$ simultaneously like in the first scheme? The paper [CHMMVW19] provided a scheme, which is different from the first scheme. This scheme requires an interactive process (or using Fiat-Shamir transformation) when proving Evaluation with degree bound, that is, the Verifier needs to provide a public random challenge number.

Commit

Assuming $f(X)$ is opened at most e times, then the Degree of the blinding polynomial $r(X)$ only needs to be equal to e.

$$\begin{aligned} C_f = \mathsf{Commit}(f(X), r(X)) &= \Big(\sum_{i=0}^{d} f_i \cdot [\tau^i]_1\Big) + \Big(\sum_{i=0}^{e} r_i \cdot [{\color{red}\gamma}\tau^i]_1\Big) \\ &= [f(\tau) + {\color{red}\gamma} \cdot r(\tau)]_1 \end{aligned} \tag{43}$$

To prove the Degree Bound, we also need to commit to $X^{D-d} \cdot f(X)$:

$$\begin{aligned} C_{xf} = \mathsf{Commit}(X^{D-d} \cdot f(X), s(X)) &= \Big(\sum_{i=0}^{d} f_i \cdot [\tau^{D-d+i}]_1\Big) + \Big(\sum_{i=0}^{d} s_i \cdot [{\color{red}\gamma} \cdot \tau^i]_1\Big) \\ &= [\tau^{D-d} \cdot f(\tau) + {\color{red}\gamma} \cdot s(\tau)]_1 \end{aligned} \tag{44}$$

So overall, the commitment $\mathsf{cm}(f)$ of $f(X)$ is defined as:

$$\mathsf{cm}(f) = (C_f, C_{xf}) \tag{45}$$

Evaluation with degree bound protocol

Public inputs:

1. Commitment C_f of polynomial $f(X)$
2. Commitment C_{xf} of polynomial $X^{D-d} \cdot f(X)$
3. Evaluation point of polynomial $f(X)$, $X = z$
4. Evaluation result of polynomial: $f(z) = v$

Witness:

1. Blinding polynomial $r(X)$ of polynomial $f(X)$

2. Blinding polynomial $s(X)$ of polynomial $X^{D-d} \cdot f(X)$

Step 1: Verifier sends random number $\alpha \leftarrow \mathbb{F}_r$,

Step 2: Prover follows these steps

1. Prover calculates quotient polynomial $q(X)$:

$$q(X) = \frac{f(X) - f(z)}{X - z} \tag{46}$$

3. Prover calculates aggregated blinding polynomial $t(X)$, obviously $\deg(t) \leq d$

$$t(X) = r(X) + \alpha \cdot s(X) \tag{47}$$

4. Prover calculates quotient polynomial $q_t(X)$

$$q_t(X) = \frac{t(X) - t(z)}{X - z} \tag{48}$$

5. Prover introduces an auxiliary polynomial $f^*(X)$, which takes value 0 at $X = z$, i.e., $f^*(z) = 0$

$$f^*(X) = X^{D-d} \cdot f(X) - X^{D-d} \cdot f(z) \tag{49}$$

6. Prover calculates the quotient polynomial $q^*(X)$ of $f^*(X)$ divided by $(X - z)$,

$$\begin{aligned} q^*(X) &= \frac{f^*(X) - f^*(z)}{X - z} \\ &= \frac{\big(X^{D-d} \cdot f(X) - X^{D-d} \cdot f(z)\big) - 0}{X - z} \\ &= X^{D-d} \cdot q(X) \end{aligned} \tag{50}$$

6. Prover commits to quotient polynomial $q(X)$, without adding any blinding factor

$$Q = \sum_{i=0}^{d-1} q_i \cdot [\tau^i]_1 = [q(\tau)]_1 \tag{51}$$

7. Prover commits to quotient polynomial $q^*(X)$, without adding any blinding factor

$$Q^* = \sum_{i=0}^{d-1} q_i \cdot [\tau^{D-d+i}]_1 = [q^*(\tau)]_1 \tag{52}$$

8. Prover commits to quotient polynomial $q_t(X)$ of blinding polynomial

$$\begin{aligned} Q_t &= \sum_{i=0}^{d-1} q_{t,i} \cdot [\gamma \tau^i]_1 \\ &= [\gamma \cdot q_t(\tau)]_1 \end{aligned} \tag{53}$$

9. Prover calculates merged commitment Q

$$\begin{aligned} Q &= Q + \alpha \cdot Q^* + Q_t \\ &= [q(\tau)]_1 + \alpha \cdot [q^*(\tau)]_1 + [\gamma \cdot q_t(\tau)]_1 \end{aligned} \tag{54}$$

10. Prover outputs proof $\pi = \big(Q, t(z)\big)$

The principle of this protocol can actually be understood from another perspective. The construction process can be decomposed into: Batch of evaluations of two polynomials at the same point (using random number α). One is to prove that the polynomial $f(X)$ takes value $f(z)$ at $X = z$, and the other is to prove that $f^*(X)$ takes value 0 at $X = z$. We can introduce an auxiliary polynomial $g(X)$ to represent the random linear combination of these two polynomials about α:

$$g(X) = f(X) + \alpha \cdot (X^{D-d} \cdot f(X) - X^{D-d} \cdot f(z)) \tag{55}$$

And the quotient polynomial $q_g(X)$ of this aggregated polynomial $g(X)$ divided by $(X - z)$ can be expressed as:

$$q_g(X) = \frac{g(X) - g(z)}{X - z} = q(X) + \alpha \cdot q^*(X) \tag{56}$$

Finally, the commitment Q calculated by the Prover is exactly equal to the commitment $[q_g(\tau)]$ of the quotient polynomial plus the commitment of the random polynomial $[\gamma \cdot q_t(\tau)]$.

Therefore, this proof idea is actually consistent with the idea of Evaluation proof.

Verification

The proof received by the Verifier is $\pi = \big(Q, t(z)\big)$, then verify according to the following steps:

1. Calculate the commitment of $g(X) + t(X)$, denoted as C_{g+t}:

$$C_{g+t} = C_f + \alpha \cdot (C_{xf} - f(z) \cdot [\tau^{D-d}]_1) \tag{57}$$

2. Calculate the commitment of the value of $g(X) + t(X)$ at $X = z$, denoted as V_{g+t}:

$$V_{g+t} = f(z) \cdot [1]_1 + t(z) \cdot [\gamma]_1 \tag{58}$$

3. Verify the correctness of C_{g+t}:

$$e\Big(C_{g+t} - V_{g+t},\ [1]_2\Big) = e\Big(Q,\ [\tau] - z \cdot [1]_2\Big) \tag{59}$$

Comparison

In the first scheme, the Prover doesn't need to care about how many times the polynomial will be opened in the future when committing, and only needs to add one random factor to achieve Perfect Hiding. The second scheme requires the Prover to add enough random factors (in the form of random polynomials) at once, and ensure that the number of times the polynomial is opened in the future will not exceed this random factor.

An advantage brought by the second scheme is that in each proof of Evaluation, the proof only includes one $\mathbb{G}_1$ element, plus one $\mathbb{F}_r$ element; while the first scheme requires two $\mathbb{G}_1$ elements.

Furthermore, the first advantage brought by the second scheme is that the Verifier only needs to calculate two Pairings, while the first scheme requires three Pairings.

References

- [KZG10] Kate, Aniket, Gregory M. Zaverucha, and Ian Goldberg. "Constant-size commitments to polynomials and their applications." Advances in Cryptology-ASIACRYPT 2010: 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings 16. Springer Berlin Heidelberg, 2010.
- [KT23] Kohrita, Tohru, and Patrick Towa. "Zeromorph: Zero-knowledge multilinear-evaluation proofs from homomorphic univariate commitments." Cryptology ePrint Archive (2023). https://eprint.iacr.org/2023/917
- [PST13] Papamanthou, Charalampos, Elaine Shi, and Roberto Tamassia. "Signatures of correct computation." Theory of Cryptography Conference. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. https://eprint.iacr.org/2011/587
- [ZGKPP17] "A Zero-Knowledge Version of vSQL." Cryptology ePrint Archive (2023). https://eprint.iacr.org/2017/1146
- [XZZPS19] Tiancheng Xie, Jiaheng Zhang, Yupeng Zhang, Charalampos Papamanthou, and Dawn Song. "Libra: Succinct Zero-Knowledge Proofs with Optimal Prover Computation." https://eprint.iacr.org/2019/317
- [CHMMVW19] Alessandro Chiesa, Yuncong Hu, Mary Maller, Pratyush Mishra, Psi Vesely, and Nicholas Ward. "Marlin: Preprocessing zkSNARKs with Universal and Updatable SRS." https://eprint.iacr.org/2019/1047