缺失的协议 PH23-PCS（二）

本文给出 PH23-KZG10 的完整的优化协议。

1. 协议框架与优化

首先回顾下 PH23+KZG10 协议的 Evaluation Argument 的简单流程，然后我们看看有哪些可以优化的地方。

P: 发送 $c(X)$ 的承诺 C_c V: 发送随机数 α 用来聚合多个多项式的约束等式 P: 计算公开的多项式集合 $\{s_i(X)\}$ P: 计算聚合的约束多项式 $h(X)$

$$h(X) = G(c(X), s_0(X), s_1(X), \ldots, s_{n-1}(X), z(X), z(\omega^{-1}X), X) \tag{1}$$

P: 计算商多项式 $t(X)$ 的承诺 C_t, $z(X)$ 的承诺 C_z

$$t(X) = \frac{h(X)}{v_H(X)} \tag{2}$$

V: 发送随机的求值点 ζ

P: 计算 $c(\zeta\cdot\omega), c(\zeta\cdot\omega^2), c(\zeta\cdot\omega^4), \ldots, c(\zeta\cdot\omega^{2^{n-1}}), c(\zeta)$，还有 $z(\zeta), z(\omega^{-1}\cdot\zeta)$，$t(\zeta), a(\zeta)$；发送上述多项式求值的 KZG10 Evaluation Arguments

V: 验证所有的 KZG10 Evaluation Arguments，然后验证下面的等式：

$$h(\zeta) \overset{?}{=} t(\zeta)\cdot v_H(\zeta) \tag{3}$$

$c^*(X)$ 在多点求值的证明优化

在证明中，Prover 需要证明 $c(X)$ 多项式在 $n+1$ 个点上的 Evaluation，即

$$c(\omega\cdot\zeta), c(\omega^2\cdot\zeta), c(\omega^4\cdot\zeta), \ldots, c(\omega^{2^{n-1}}\cdot\zeta), c(\zeta) \tag{4}$$

利用 [BDFG20] 论文中的技术，如果一个 $f(X)$ 在 m 个点 $D=(z_0, z_1, \ldots, z_{m-1})$ 上的 Evaluation 为 $\vec{v}=(v_0, v_1, \ldots, v_{m-1})$，定义 $f^*(X)$ 是 $\vec{v}$ 在 D 上的插值多项式，即 $\deg(f^*(X)) = m-1$，并且有 $f^*(z_i) = f(z_i), \forall i\in[0,m)$

$$v_D(X) = \prod_{i=0}^{m-1}(X - z_i) \tag{5}$$

那么 $f(X)$ 满足下面的等式：

$$f(X) - f^*(X) = q(X)\cdot(X-z_0)(X-z_1)\cdots(X-z_{m-1}) \tag{6}$$

上面等式这个很容易验证，因为当 $X=z_i$ 的时候，等式左边等于零，那么 $f(X)-f^*(X)$ 可以被 $(X-z_i)$ 整除。那么对于所有的 $i=0,1,\ldots,m-1$，$f(X)-f^*(X)$ 可以被 $v_D(X)$ 整除，

$$v_D(X) = \prod_{i=0}^{m-1}(X - z_i) \tag{7}$$

这样一来，Prover 只要向 Verifier 证明存在 $q(X)$，使得 $f(X)-f^*(X)=q(X)\cdot v_D(X)$，那么 $f(X)$ 在 D 上的 Evaluation 就等于 $\vec{v}$。而这个等式又可以通过 Verifier 提供一个随机挑战点 $X=\xi$ 来验证，其中 $v_D(\xi)$ 与 $f^*(\xi)$ 可以由 Verifier 自行计算，而 $f(\xi)$ 与 $q(\xi)$ 可以通过 KZG10 的 Evaluation Argument 来证明。

$c^*(X)$ 多项式计算的优化

Prover 可以构造多项式 $c^*(X)$，它是下面向量在 ζD 上的插值多项式。这样做的优势是可以让 Prover 一次证明 $c(X)$ 的多个不同点的 Evaluation，记为 $\vec{c^*}$：

$$c(\omega\cdot\zeta), c(\omega^2\cdot\zeta), c(\omega^4\cdot\zeta), \ldots, c(\omega^{2^{n-1}}\cdot\zeta), c(\zeta) \tag{8}$$

我们引入 D 满足 $|D|=n+1$，其定义为

$$D=\big(\omega,\ \omega^2,\ \omega^4,\ \ldots,\ \omega^{2^{n-1}}, \omega^{2^n}=1\big) \tag{9}$$

那么 $c^*(X)$ 的 Evaluation 的 Domain 就可以表示为 ζD,

$$D'=D\zeta=\big(\omega\cdot\zeta,\ \omega^2\cdot\zeta,\ \omega^4\cdot\zeta,\ \ldots,\ \omega^{2^{n-1}}\cdot\zeta,\ \zeta\big) \tag{10}$$

其 Vanishing 多项式 $v_{D'}(X)$ 定义如下：

$$v_{D'}(X)=(X-\omega\zeta)(X-\omega^2\zeta)(X-\omega^4\zeta)\cdots(X-\omega^{2^n}\zeta) \tag{11}$$

对于 D' 上的 Lagrange 多项式 可以定义如下：

$$L_j^{D'}(X)=\hat{d}_j\cdot\frac{v_{D'}(X)}{X-\omega^{2^j}\zeta}, \qquad j=0,1,\ldots,n \tag{12}$$

其中 $\hat{d}_j$ 是 D' 上的 Bary-Centric Weights，定义为

$$\hat{d}_j=\prod_{l\neq j}\frac{1}{\zeta\cdot\omega^{2^j}-\zeta\cdot\omega^{2^l}}=\frac{1}{\zeta^n}\cdot\prod_{l\neq j}\frac{1}{\omega^{2^j}-\omega^{2^l}}=\frac{1}{\zeta^n}\cdot\hat{w}_j \tag{13}$$

这里 $\hat{w}_j$ 是 D 上的 Bary-Centric Weights，并且它的定义只与 D 相关，和 ζ 无关。因此，我们可以事先预计算 $\hat{w}_j$，然后利用 $\hat{w}_j$ 来计算 $c^*(X)$：

$$c^*(X)=c_0^*\cdot L_0^{D'}(X)+c_1^*\cdot L_1^{D'}(X)+\cdots+c_n^*\cdot L_n^{D'}(X) \tag{14}$$

上面的等式可以进一步优化，等式右边除以一个常数项多项式 $g(X)=1$

$$g(X)=1\cdot L_0^{D'}(X)+1\cdot L_1^{D'}(X)+\cdots+1\cdot L_n^{D'}(X) \tag{15}$$

可以得到：

$$c^*(X)=\frac{c^*(X)}{g(X)}=\frac{c_0^*\cdot L_0^{D'}(X)+c_1^*\cdot L_1^{D'}(X)+\cdots+c_n^*\cdot L_n^{D'}(X)}{g(X)} \tag{16}$$

展开 $g(X)$ 与 $L_i^{D'}(X)$，可以得到：

$$c^*(X)=\frac{c_0^*\cdot\hat{d}_0\cdot\frac{z_{D'}(X)}{X-\omega\zeta}+c_1^*\cdot\hat{d}_1\cdot\frac{z_{D'}(X)}{X-\omega^2\zeta}+\cdots+c_n^*\cdot\hat{d}_n\cdot\frac{z_{D'}(X)}{X-\omega^{2^n}\zeta}}{1\cdot\hat{d}_0\cdot\frac{z_{D'}(X)}{X-\omega\zeta}+1\cdot\hat{d}_1\cdot\frac{z_{D'}(X)}{X-\omega^2\zeta}+\cdots+1\cdot\hat{d}_n\cdot\frac{z_{D'}(X)}{X-\omega^{2^n}\zeta}} \tag{17}$$

分子分母同时消去 $z_{D'}(X)$，可以得到

$$c^*(X)=\frac{c_0^*\cdot\hat{d}_0\cdot\frac{1}{X-\omega\zeta}+c_1^*\cdot\hat{d}_1\cdot\frac{1}{X-\omega^2\zeta}+\cdots+c_n^*\cdot\hat{d}_n\cdot\frac{1}{X-\omega^{2^n}\zeta}}{1\cdot\hat{d}_0\cdot\frac{1}{X-\omega\zeta}+1\cdot\hat{d}_1\cdot\frac{1}{X-\omega^2\zeta}+\cdots+1\cdot\hat{d}_n\cdot\frac{1}{X-\omega^{2^n}\zeta}} \tag{18}$$

再展开 $\hat{d}_i$ 的定义，并且分子分母同时消去 $\frac{1}{\zeta^n}$ ，可以得到

$$c^*(X)=\frac{c_0^*\cdot\frac{\hat{w}_0}{X-\omega\zeta}+c_1^*\cdot\frac{\hat{w}_1}{X-\omega^2\zeta}+\cdots+c_n^*\cdot\frac{\hat{w}_n}{X-\omega^{2^n}\zeta}}{\frac{\hat{w}_0}{X-\omega\zeta}+\frac{\hat{w}_1}{X-\omega^2\zeta}+\cdots+\frac{\hat{w}_n}{X-\omega^{2^n}\zeta}} \tag{19}$$

Prover 可以利用事先预计算的 D 上的Bary-Centric Weights $\{\hat{w}_i\}$ 来快速计算 $c^*(X)$，如果 n 是固定的。尽管如此，$c^*(X)$ 的计算复杂度仍为 $O(n\log^2(n))$。不过考虑到 $n=\log(N)$，所以 $c^*(X)$ 的计算复杂度是对数级别的。

$$c^*(X)=\sum_{j=0}^{n-1}\frac{\hat{w}_j}{\zeta^n}\cdot\frac{z_{D_\zeta}(X)}{X-\zeta\cdot\omega^{2^j}} \tag{20}$$

预计算的 $\hat{w}_j$ 的定义为

$$\hat{w}_j=\prod_{l\neq j}\frac{1}{\omega^{2^j}-\omega^{2^l}} \tag{21}$$

不仅如此，Verifier 需要计算 $c^*(X)$ 在某个挑战点上的取值，比如 $X=\xi$， Verifier 可以通过上面的等式，以 $O(\log N)$ 时间复杂度根据 Prover 提供的 $\vec{c^*}$ 来计算 $c^*(\xi)$。

2. PH23+KZG10 协议（优化版）

对于 KZG10 协议，因为其 Commitment 具有加法同态性。

Precomputation

1. 预计算 $s_0(X),\ldots,s_{n-1}(X)$ and $v_H(X)$

$$v_H(X)=X^N-1 \tag{22}$$

$$s_i(X)=\frac{v_H(X)}{v_{H_i}(X)}=\frac{X^N-1}{X^{2^i}-1} \tag{23}$$

2. 预计算 $D=(1,\omega,\omega^2,\ldots,\omega^{2^{n-1}})$ 上的 Bary-Centric Weights $\{\hat{w}_i\}$。这个可以加速

$$\hat{w}_j=\prod_{l\neq j}\frac{1}{\omega^{2^j}-\omega^{2^l}} \tag{24}$$

3. 预计算 Lagrange Basis 的 KZG10 SRS
   $A_0=[L_0(\tau)]_1,A_1=[L_1(\tau)]_1,A_2=[L_2(\tau)]_1,\ldots,A_{N-1}=[L_{2^{n-1}}(\tau)]_1$

Common inputs

1. $C_a=[\hat{f}(\tau)]_1$: the (uni-variate) commitment of $\tilde{f}(X_0,X_1,\ldots,X_{n-1})$
2. $\vec{u}=(u_0,u_1,\ldots,u_{n-1})$: 求值点

3. $v = \tilde{f}(u_0, u_1, \ldots, u_{n-1})$: MLE 多项式 $\tilde{f}$ 在 $\vec{X} = \vec{u}$ 处的运算值

Commit 计算过程

1. Prover 构造一元多项式 $a(X)$，使其 Evaluation form 等于 $\vec{a} = (a_0, a_1, \ldots, a_{N-1})$，其中 $a_i = \tilde{f}(\mathsf{bits}(i))$, 为 $\tilde{f}$ 在 Boolean Hypercube $\{0,1\}^n$ 上的取值。

$$a(X) = a_0 \cdot L_0(X) + a_1 \cdot L_1(X) + a_2 \cdot L_2(X) + \cdots + a_{N-1} \cdot L_{N-1}(X) \tag{25}$$

2. Prover 计算 $\hat{f}(X)$ 的承诺 C_a，并发送 C_a

$$C_a = a_0 \cdot A_0 + a_1 \cdot A_1 + a_2 \cdot A_2 + \cdots + a_{N-1} \cdot A_{N-1} = [\hat{f}(\tau)]_1 \tag{26}$$

其中 $A_0 = [L_0(\tau)]_1, A_1 = [L_1(\tau)]_1, A_2 = [L_2(\tau)]_1, \ldots, A_{N-1} = [L_{2^{n-1}}(\tau)]_1$ ，在预计算过程中已经得到。

Evaluation 证明协议

回忆下证明的多项式运算的约束：

$$\tilde{f}(u_0, u_1, u_2, \ldots, u_{n-1}) = v \tag{27}$$

这里 $\vec{u} = (u_0, u_1, u_2, \ldots, u_{n-1})$ 是一个公开的挑战点。

Round 1.

Prover:

1. 计算向量 $\vec{c}$，其中每个元素 $c_i = \widetilde{eq}(\mathsf{bits}(i), \vec{u})$
2. 构造多项式 $c(X)$，其在 H 上的运算结果恰好是 $\vec{c}$ 。

$$c(X) = \sum_{i=0}^{N-1} c_i \cdot L_i(X) \tag{28}$$

3. 计算 $c(X)$ 的承诺 $C_c = [c(\tau)]_1$，并发送 C_c

$$C_c = \mathsf{KZG10.Commit}(\vec{c}) = [c(\tau)]_1 \tag{29}$$

Round 2.

Verifier: 发送挑战数 $\alpha \leftarrow_{\$} \mathbb{F}_p$

Prover:

1. 构造关于 $\vec{c}$ 的约束多项式 $p_0(X), \ldots, p_n(X)$

$$\begin{aligned} p_0(X) &= s_0(X) \cdot \Big(c(X) - (1-u_0)(1-u_1)\ldots(1-u_{n-1})\Big) \\ p_k(X) &= s_{k-1}(X) \cdot \Big(u_{n-k} \cdot c(X) - (1-u_{n-k}) \cdot c(\omega^{2^{n-k}} \cdot X)\Big), \quad k = 1 \ldots n \end{aligned} \tag{30}$$

2. 把 $\{p_i(X)\}$ 聚合为一个多项式 $p(X)$

$$p(X) = p_0(X) + \alpha \cdot p_1(X) + \alpha^2 \cdot p_2(X) + \cdots + \alpha^n \cdot p_n(X) \tag{31}$$

3. 构造累加多项式 $z(X)$，满足

$$
\begin{aligned}
z(1) &= a_0\cdot c_0 \\
z(\omega_i)-z(\omega_{i-1}) &= a(\omega_i)\cdot c(\omega_i), \quad i=1,\ldots,N-1 \\
z(\omega^{N-1}) &= v
\end{aligned}
\tag{32}
$$

4. 构造约束多项式 $h_0(X),h_1(X),h_2(X)$，满足

$$
\begin{aligned}
h_0(X) &= L_0(X)\cdot\big(z(X)-c_0\cdot a(X)\big) \\
h_1(X) &= (X-1)\cdot\big(z(X)-z(\omega^{-1}\cdot X)-a(X)\cdot c(X)\big) \\
h_2(X) &= L_{N-1}(X)\cdot\big(z(X)-v\big)
\end{aligned}
\tag{33}
$$

5. 把 $p(X)$ 和 $h_0(X),h_1(X),h_2(X)$ 聚合为一个多项式 $h(X)$，满足

$$
h(X)=p(X)+\alpha^{n+1}\cdot h_0(X)+\alpha^{n+2}\cdot h_1(X)+\alpha^{n+3}\cdot h_2(X) \tag{34}
$$

6. 计算 Quotient 多项式 $t(X)$，满足

$$
h(X)=t(X)\cdot v_H(X) \tag{35}
$$

7. 计算 $C_t=[t(\tau)]_1$， $C_z=[z(\tau)]_1$，并发送 C_t 和 C_z

$$
\begin{aligned}
C_t &= \mathsf{KZG10.Commit}(t(X))=[t(\tau)]_1 \\
C_z &= \mathsf{KZG10.Commit}(z(X))=[z(\tau)]_1
\end{aligned}
\tag{36}
$$

Round 3.

Verifier: 发送随机求值点 $\zeta\leftarrow_{\$}\mathbb{F}_p$

Prover:

1. 计算 $s_i(X)$ 在 ζ 处的取值：

$$
s_0(\zeta),s_1(\zeta),\ldots,s_{n-1}(\zeta) \tag{37}
$$

这里 Prover 可以高效计算 $s_i(\zeta)$ ，由 $s_i(X)$ 的公式得

$$
\begin{aligned}
s_i(\zeta) &= \frac{\zeta^N-1}{\zeta^{2^i}-1} \\
&= \frac{(\zeta^N-1)(\zeta^{2^i}+1)}{(\zeta^{2^i}-1)(\zeta^{2^i}+1)} \\
&= \frac{\zeta^N-1}{\zeta^{2^{i+1}}-1}\cdot(\zeta^{2^i}+1) \\
&= s_{i+1}(\zeta)\cdot(\zeta^{2^i}+1)
\end{aligned}
\tag{38}
$$

因此 $s_i(\zeta)$ 的值可以通过 $s_{i+1}(\zeta)$ 计算得到，而

$$
s_{n-1}(\zeta)=\frac{\zeta^N-1}{\zeta^{2^{n-1}}-1}=\zeta^{2^{n-1}}+1 \tag{39}
$$

因此可以得到一个 $O(n)$ 的算法来计算 $s_i(\zeta)$ ，并且这里不含除法运算。计算过程是：
$s_{n-1}(\zeta)\to s_{n-2}(\zeta)\to\cdots\to s_0(\zeta)$ 。

2. 定义求值 Domain D'，包含 $n+1$ 个元素：

$$D' = D\zeta = \{\zeta, \omega\zeta, \omega^2\zeta, \omega^4\zeta, \ldots, \omega^{2^{n-1}}\zeta\} \tag{40}$$

3. 计算并发送 $c(X)$ 在 D' 上的取值

$$c(\zeta), c(\zeta\cdot\omega), c(\zeta\cdot\omega^2), c(\zeta\cdot\omega^4), \ldots, c(\zeta\cdot\omega^{2^{n-1}}) \tag{41}$$

4. 计算并发送 $z(\omega^{-1}\cdot\zeta)$
5. 计算 Linearized Polynomial $l_\zeta(X)$

$$\begin{aligned}
l_\zeta(X) =& \Big(s_0(\zeta)\cdot(c(\zeta)-c_0) \\
&+ \alpha\cdot s_0(\zeta)\cdot(u_{n-1}\cdot c(\zeta) - (1-u_{n-1})\cdot c(\omega^{2^{n-1}}\cdot\zeta)) \\
&+ \alpha^2\cdot s_1(\zeta)\cdot(u_{n-2}\cdot c(\zeta) - (1-u_{n-2})\cdot c(\omega^{2^{n-2}}\cdot\zeta)) \\
&+ \cdots \\
&+ \alpha^{n-1}\cdot s_{n-2}(\zeta)\cdot(u_1\cdot c(\zeta) - (1-u_1)\cdot c(\omega^2\cdot\zeta)) \\
&+ \alpha^n\cdot s_{n-1}(\zeta)\cdot(u_0\cdot c(\zeta) - (1-u_0)\cdot c(\omega\cdot\zeta)) \\
&+ \alpha^{n+1}\cdot(L_0(\zeta)\cdot\big(z(X) - c_0\cdot a(X)) \\
&+ \alpha^{n+2}\cdot(\zeta-1)\cdot\big(z(X) - z(\omega^{-1}\cdot\zeta) - c(\zeta)\cdot a(X)) \\
&+ \alpha^{n+3}\cdot L_{N-1}(\zeta)\cdot(z(X) - v) \\
&- v_H(\zeta)\cdot t(X)\Big)
\end{aligned} \tag{42}$$

显然，$l_\zeta(\zeta) = 0$，因此这个运算值不需要发给 Verifier，并且 $[l_\zeta(\tau)]_1$ 可以由 Verifier 自行构造。

6. 构造多项式 $c^*(X)$，它是下面向量在 $D\zeta$ 上的插值多项式

$$\vec{c^*} = \Big(c(\omega\cdot\zeta), c(\omega^2\cdot\zeta), c(\omega^4\cdot\zeta), \ldots, c(\omega^{2^{n-1}}\cdot\zeta), c(\zeta)\Big) \tag{43}$$

Prover 可以利用事先预计算的 D 上的Bary-Centric Weights $\{\hat{w}_i\}$ 来快速计算 $c^*(X)$，

$$c^*(X) = \frac{c^*_0\cdot\frac{\hat{w}_0}{X-\omega\zeta} + c^*_1\cdot\frac{\hat{w}_1}{X-\omega^2\zeta} + \cdots + c^*_n\cdot\frac{\hat{w}_n}{X-\omega^{2^n}\zeta}}{\frac{\hat{w}_0}{X-\omega\zeta} + \frac{\hat{w}_1}{X-\omega^2\zeta} + \cdots + \frac{\hat{w}_n}{X-\omega^{2^n}\zeta}} \tag{44}$$

这里 $\hat{w}_j$ 为预计算的值：

$$\hat{w}_j = \prod_{l\neq j}\frac{1}{\omega^{2^j} - \omega^{2^l}} \tag{45}$$

7. 因为 $l_\zeta(\zeta) = 0$，所以存在 Quotient 多项式 $q_\zeta(X)$ 满足

$$q_\zeta(X) = \frac{1}{X-\zeta}\cdot l_\zeta(X) \tag{46}$$

8. 构造 $D\zeta$ 上的消失多项式 $z_{D_\zeta}(X)$

$$z_{D_\zeta}(X) = (X-\zeta\omega)\cdots(X-\zeta\omega^{2^{n-1}})(X-\zeta) \tag{47}$$

9. 构造 Quotient 多项式 $q_c(X)$:

$$q_c(X) = \frac{(c(X) - c^*(X))}{(X-\zeta)(X-\omega\zeta)(X-\omega^2\zeta)\cdots(X-\omega^{2^{n-1}}\zeta)} \tag{48}$$

10. 构造 Quotient 多项式 $q_{\omega\zeta}(X)$

$$q_{\omega\zeta}(X) = \frac{z(X) - z(\omega^{-1}\cdot\zeta)}{X - \omega^{-1}\cdot\zeta} \tag{49}$$

11. 发送 $\big(Q_c = [q_c(\tau)]_1, Q_\zeta = [q_\zeta(\tau)]_1, Q_{\omega\zeta} = [q_{\omega\zeta}(\tau)]_1,\big)$

Round 4.

1. Verifier 发送第二个随机挑战点 $\xi \leftarrow_{\$} \mathbb{F}_p$
2. Prover 构造第三个 Quotient 多项式 $q_\xi(X)$

$$q_\xi(X) = \frac{c(X) - c^*(\xi) - z_{D_\zeta}(\xi)\cdot q_c(X)}{X - \xi} \tag{50}$$

3. Prover 计算并发送 Q_ξ

$$Q_\xi = \mathsf{KZG10.Commit}(q_\xi(X)) = [q_\xi(\tau)]_1 \tag{51}$$

证明表示

$7\cdot\mathbb{G}_1$, $(n+1)\cdot\mathbb{F}$

$$\begin{split}\pi_{eval} = \big(&z(\omega^{-1}\cdot\zeta), c(\zeta),\ c(\omega\cdot\zeta), c(\omega^2\cdot\zeta), c(\omega^4\cdot\zeta), \ldots, c(\omega^{2^{n-1}}\cdot\zeta),\\ &C_c, C_t, C_z, Q_c, Q_\zeta, Q_\xi, Q_{\omega\zeta}\big)\end{split} \tag{52}$$

验证过程

1. Verifier 计算 $c^*(\xi)$ 使用预计算的 Barycentric Weights $\{\hat{w}_i\}$

$$c^*(\xi) = \frac{\sum_i c_i \frac{w_i}{\xi - x_i}}{\sum_i \frac{w_i}{\xi - x_i}} \tag{53}$$

2. Verifier 计算 $v_H(\zeta), L_0(\zeta), L_{N-1}(\zeta)$

$$v_H(\zeta) = \zeta^N - 1 \tag{54}$$

$$L_0(\zeta) = \frac{1}{N}\cdot\frac{z_H(\zeta)}{\zeta - 1} \tag{55}$$

$$L_{N-1}(\zeta) = \frac{\omega^{N-1}}{N}\cdot\frac{z_H(\zeta)}{\zeta - \omega^{N-1}} \tag{56}$$

3. Verifier 计算 $s_0(\zeta),\ldots,s_{n-1}(\zeta)$，其计算方法可以采用前文提到的递推方式进行计算。
4. Verifier 计算线性化多项式的承诺 C_l

$$
\begin{aligned}
C_l = \Big(& (c(\zeta) - c_0) s_0(\zeta) \\
& + \alpha \cdot (u_{n-1} \cdot c(\zeta) - (1 - u_{n-1}) \cdot c(\omega^{2^{n-1}} \cdot \zeta)) \cdot s_0(\zeta) \\
& + \alpha^2 \cdot (u_{n-2} \cdot c(\zeta) - (1 - u_{n-2}) \cdot c(\omega^{2^{n-2}} \cdot \zeta)) \cdot s_1(\zeta) \\
& + \cdots \\
& + \alpha^{n-1} \cdot (u_1 \cdot c(\zeta) - (1 - u_1) \cdot c(\omega^2 \cdot \zeta)) \cdot s_{n-2}(\zeta) \\
& + \alpha^n \cdot (u_0 \cdot c(\zeta) - (1 - u_0) \cdot c(\omega \cdot \zeta)) \cdot s_{n-1}(\zeta) \\
& + \alpha^{n+1} \cdot L_0(\zeta) \cdot (C_z - c_0 \cdot C_a) \\
& + \alpha^{n+2} \cdot (\zeta - 1) \cdot \big(C_z - z(\omega^{-1} \cdot \zeta) - c(\zeta) \cdot C_a\big) \\
& + \alpha^{n+3} \cdot L_{N-1}(\zeta) \cdot (C_z - v) \\
& - v_H(\zeta) \cdot C_t \Big)
\end{aligned} \tag{57}
$$

5. Verifier 产生随机数 η 来合并下面的 Pairing 验证：

$$
\begin{aligned}
e(C_l + \zeta \cdot Q_\zeta, [1]_2) &\stackrel{?}{=} e(Q_\zeta, [\tau]_2) \\
e(C - C^*(\xi) - z_{D_\zeta}(\xi) \cdot Q_c + \xi \cdot Q_\xi, [1]_2) &\stackrel{?}{=} e(Q_\xi, [\tau]_2) \\
e(C_z + \zeta \cdot Q_{\omega\zeta} - z(\omega^{-1} \cdot \zeta) \cdot [1]_1, [1]_2) &\stackrel{?}{=} e(Q_{\omega\zeta}, [\tau]_2)
\end{aligned} \tag{58}
$$

合并后的验证只需要两个 Pairing 运算。

$$
\begin{aligned}
P = & \Big(C_l + \zeta \cdot Q_\zeta\Big) \\
& + \eta \cdot \Big(C - C^* - z_{D_\zeta}(\xi) \cdot Q_c + \xi \cdot Q_\xi\Big) \\
& + \eta^2 \cdot \Big(C_z + \zeta \cdot Q_{\omega\zeta} - z(\omega^{-1} \cdot \zeta) \cdot [1]_1\Big)
\end{aligned} \tag{59}
$$

$$
e\Big(P, [1]_2\Big) \stackrel{?}{=} e\Big(Q_\zeta + \eta \cdot Q_\xi + \eta^2 \cdot Q_{\omega\zeta}, [\tau]_2\Big) \tag{60}
$$

3. 优化性能分析

Proof size: $7\ \mathbb{G}_1 + (n+1)\ \mathbb{F}$

Prover's cost

- Commit 阶段：$O(N \log N)\ \mathbb{F}$ + $\mathbb{G}_1$
- Evaluation 阶段：$O(N \log N)\ \mathbb{F}$ + $7\ \mathbb{G}_1$

Verifier's cost: $4\ \mathbb{F} + O(n)\ \mathbb{F} + 3\ \mathbb{G}_1 + 2\ P$

References

- [BDFG20] Dan Boneh, Justin Drake, Ben Fisch, and Ariel Gabizon. "Efficient polynomial commitment schemes for multiple points and polynomials". Cryptology {ePrint} Archive, Paper 2020/081. https://eprint.iacr.org/2020/081.