理解 Hiding KZG10

Hiding KZG10 是 KZG10 协议的变种，其产生的多项式承诺带有随机的盲化因子（Blinding Factor），从而具备 Perfect Hiding 的性质。即假设攻击者的计算能力无限，并且攻击者不能通过承诺来逆向计算出多项式的任何信息。Hiding KZG10 并不常见，但它是构造具有 Zero-knowledge 性质的 zkSNARK 或者其它安全协议的重要组件。

本文介绍两种不同的 Hiding KZG10，第一种方案出自 [KT23]，其主要的技术是多元多项式承诺一个简化版本 [PST13]，[ZGKPP17]，与 [XZZPS19]。第二种方案出自 [CHMMVW19]，其主要的技术是对原始 KZG10 协议论文 [KZG10] 的改进。

None-hiding KZG10

先回忆一下基本的 KZG10 协议。首先 KZG10 基于一个事先经过预设置（Universal Trusted Setup）的 SRS：

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

其中 τ 是秘密值，需要在 Setup 阶段后被遗忘，否则任何知道 τ 的一方都可以发起攻击。我们用中括号记号 $[a]_1$ 来表示一个椭圆曲线群元素的上的标量乘法（Scalar Multiplication） $a \cdot G$，这里 $G \in \mathbb{G}_1$ 是群中的生成元。SRS 正是由 $\mathbb{G}_1$ 和 $\mathbb{G}_2$ 的群元素构成，我们把这些元素称为 Base 元素，因为后续对多项式的承诺正是基于这些 Base 元素的线性运算。

由于椭圆曲线群上的元素的除法运算是一个困难问题，所以如果我们的算力有限，那么就无法通过 $[a]_1$ 中计算得到 a。这可以看成是，一旦我们把一个 $a \in \mathbb{F}_r$ 的值乘上一个 Base 元素，那么 a 就被隐藏了起来。

KZG10 需要一个双线性配对运算友好的曲线，即存在另一个椭圆曲线群 $\mathbb{G}_2$（生成元为 G'），其中的每个元素表示为 $[b]_2$，即 $b \cdot G'$。并且存在一个双线性配对操作，满足下面的双线性性质与 Non-degeneracy 性质：

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

现在假设有一个一元多项式 $f(X) \in \mathbb{F}_r[X]$

$$f(X) = f_0 + f_1X + f_2X^2 + \cdots + f_dX^d \tag{3}$$

这里多项式的 Degree d 需要满足 $d < D$。然后多项式的承诺 C_f 的计算方式如下：

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

经过推导，不难发现下面的等式成立

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

Evaluation 证明

对于我们任意取的多项式 $f(X)$ 是多项式环 $\mathbb{F}_r[X]$ 中的元素，它满足下面的除法公式：

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

其中 $g(X)$ 是除数多项式，$r(X)$ 是余数多项式。显然，$\deg(r) < \deg(g)$。

如果我们令 $g(X) = X - z$，那么显然 $r(X)$ 是一个常数多项式，于是上面的除法分解公式可以写为：

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

进一步，将 $X=z$ 代入到上面的等式，我们可以得到 $f(z)=r$。于是，上面的除法分解公式可以改写为：

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

这个公式正是 KZG10 协议的核心公式。即如果我们要证明 $f(z)=r$，那么我们只要证明 $f(X)-f(z)$ 能被 $(X-z)$ 整除。或者换句话说，存在一个商多项式 $q(X)$，满足：

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

如果 $f(X)$ 在 $X=z$ 处的取值不等于 r，那么 $q(X)$ 就不是一个多项式，而是一个 Rational Function。而对于任意分母不为常数多项式的 Rational Function，我们无法利用上面的 SRS 来计算它的承诺。

因此，Prover 只要发送 $q(X)$ 的承诺，向 Verifier 证明 $q(X)$ 的存在性，即相当于证明了 $f(X)$ 的求值是正确的。

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

然后 Verifier 利用 SRS 提供的 Base 元素来检查分解公式的正确性。对于 Verifier， $f(z)$ 与 z 是公开的，所以 Verifier 可以通过下面的公式来检查分解公式：

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

上面公式中红色的部分由 Prover 提供，并且不暴露 $[\cdot]_1$ 中的值。

KZG10 协议的 Polynomial Evaluation 证明仅仅包含一个 $\mathbb{G}_1$ 的元素 $[q(\tau)]_1$，尺寸为 $O(1)$。而 Verifier 的验证算法也是 $O(1)$ 的。需要提及的是，Verifier 需要完成两次 Pairing 计算，这个计算虽然是 $O(1)$ 复杂度的，但是比较昂贵。

Degree Bound 证明

KZG10 还支持证明一个多项式 $f(X)\in\mathbb{F}_r[X]$ 的 Degree 小于等于 d。

Prover 证明的方式非常直接，即构造一个新的多项式 $\hat{f}(X)$，

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

显然 $\hat{f}(X)$ 的 Degree 小于等于 D。而因为 SRS 中所包含的关于 τ 的最高次幂的 Base 元素是 $[\tau^D]_1$，理论上任何人（不知道 τ 的值）都不能构造任何一个次数大于等于 D 的多项式的承诺。

因此，Prover 可以构造 Degree Bound 证明 π 为：

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

而 Verifier 的验证方法也比较直接，检查 $\hat{f}(X)$ 是否由 $f(X)$ 与 X^{D-d} 相乘得到的。

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

其中 $[\tau^{D-d}]_2$ 也应该是 SRS 中的 Base 元素。这要求 KZG10 的 SRS 中要包含更多的 $\mathbb{G}_2$ 的 Base 元素：

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

继续思考，如果 Prover 要同时证明 $f(X)$ 的 Degree Bound 和 Evaluation，那么他在产生 $f(X)$ 的承诺时，要产生两个 $\mathbb{G}_1$ 的元素，$([\hat{f}(\tau)]_1, [\tau^{D-d}\hat{f}(\tau)]_1)$。然后 Verifier 需要完成 4 个 Pairing 计算来同时检查 Evaluation 和 Degree Bound 证明

Perfect Hiding

如果我们比较在意协议交互过程中的隐私保护问题，那么 KZG10 需要增强对 $f(X)$ 多项式的保护。在上面的 KZG10 协议中，我们假设攻击者的计算能力有限，即攻击者不能通过 $[f(\tau)]_1$ 来逆向计算出 $f(X)$ 的任何信息。

相比之下，传统的 Pedersen Commitment 则具备 Perfect Hiding 的性质，因为它的承诺带有一个随机数作为盲化因子（Blinding Factor），所以即使攻击者有无限的算力，也不能通过 $[f(\tau)]_1$ 来逆向得到 $f(X)$ 的任何信息。

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

其中 $\{G_i\}_{i=0}^{d} \in \mathbb{G}_1^{d+1}$ 与 $G' \in \mathbb{G}_1$ 是 Pedersen Commitment 的公共参数。并且 $r \in \mathbb{F}_r$ 正是所谓的盲化因子。

我们接下来解释如何将 KZG10 协议转换为 Perfect Hiding 的协议。这个方案出自[KT23]，其基本思想来自 [ZGKP17] 与 [PST13]。

首先，我们可以「尝试」考虑让 KZG10 在承诺 $f(X)$ 的同时中加入一个随机数作为盲化因子，比如：

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

但是这样承诺会有安全问题。因为在 Pedersen Commitment 中，用来承诺 r 的元素 G' 是一个专门的 Base 元素，它与其他的 G_i 之间的关系未知（即独立）。所以，r 的引入并不影响 $f(X)$ 的常数项 f_0 的承诺。

因此我们需要扩充 SRS，并引入一个额外的预设置随机值 γ，专门用以承诺盲化因子 r：

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

那么 $f(X)$ 的承诺定义为：

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

我们下面用一个更短的符号 $\mathsf{cm}(f)$ 来表示 $f(X)$ 的承诺。

Hiding-KZG10 的 Evaluation 证明

虽然我们在 Commitment 加上 Blinding Factor，但是 $f(X)$ 的 Evaluation 证明仍然可能会暴露 $f(X)$ 的信息。

设想如果 Prover 要向 Verifier 证明 $f(z) = v$，那么他需要计算得到商多项式 $q(X)$，计算并发送它的承诺 $[q(\tau)]_1$。如果直接发送 $[q(\tau)]_1$，这会破坏我们想要的 Perfect Hiding 的性质，因为一个「算力无限」的攻击者可以从 $[q(\tau)]_1$ 中逆向计算出 $q(X)$，然后从而继续计算出 $f(X)$。

因此，我们也需要为 $[q(\tau)]_1$ 也加上另一个不同的盲化因子，记为 ${\color{green}s}$：

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

我们把加上盲化因子的 $q(X)$ 的承诺记为短符号 $\mathsf{cm}(q)$。

继续回忆下，Non-hiding KZG10 的 Verifier 需要检查下面的等式来验证 $q(X)$ 的承诺：

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

不过在 Hiding-KZG10 中，由于多项式承诺 $\mathsf{cm}(f)$ 和商多项式承诺 $\mathsf{cm}(q)$ 都带上了盲化因子，所以 Verifier 就不再能按照上面的 Pairing 等式完成验证了：

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

我们来推理下为什么上面的等式不成立。先看看等式左边相当于在计算

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

等式右边相当于在计算

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

左右两边相差的项为

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

为了让 Verifier 能够验证，我们需要引入一个额外的「群元素」来配平 Pairing 验证公式：

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

于是，Verifier 可以通过下面的公式来验证：

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

或者写为：

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

其中红色部分由 Prover 提供，蓝色的部分是公开值。

Hiding-KZG10 的 Degree Bound 证明

为了证明 $f(X)$ 的 Degree Bound，我们需要给多项式 $\hat{f}(X)$ 也加上 Blinding Factor，然后计算其承诺，作为 $f(X)$ 的 Degree Bound 证明：

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

同时还要附加上一个用来配平的元素 $E\in\mathbb{G}_1$，

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

这样 Verifier 可以用过下面的等式来验证 $f(X)$ 的 Degree Bound 证明：

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

读者可以自行验证下，上面等式为何成立。

Hiding KZG10 的 Evaluation-and-degree-bound 证明

假如对于同一个 Polynomial $f(X)$，Prover 需要同时对 $f(X)$ 的 Evaluation 和 Degree Bound 进行证明。如果我们分别使用上面的 Evaluation 和 Degree Bound 证明协议，那么 Prover 需要发送两个 $\mathbb{G}_1$ 的元素，然后 Verifier 需要完成 4 个 Pairing 计算。事实上，我们可以把这两个证明步骤合并为一步：Prover 仅发送两个一个 $\mathbb{G}_1$ 元素，而 Verifier 仅使用两次 Pairing 即可完成验证。

Prover 需要构造两个 $\mathbb{G}_1$ 的元素，

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

另一个元素 E 定义为：

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

Prover 发送证明

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

而 Verifier 需要验证下面的等式：

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

另一种 Hiding KZG10 的构造

在原始的 [KZG10] 论文中，也提供了实现 Perfect Hiding 的构造方案。我们可以对比下两种不同风格的 Hiding KZG10 变种。

这种方案的想法是在 Commit $f(X)$ 的时候，补上一个随机的多项式 $r(X)$，而不仅仅是单个的随机盲化因子。这里 $f(X)$ 与 $r(X)$ 的定义如下：

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

注意这里，盲化多项式 $r(X)$的 Degree 与 $f(X)$ 的 Degree 一致。为了支持盲化多项式（Blinding Polynomial），最初 Setup 阶段产生的 SRS 需要引入一个随机数 γ 来隔离盲化因子与正常要 Commit 的消息。于是 SRS 被扩充为：

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

下面我们定义下 $\mathsf{cm}(f)$ 的计算公式：

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

本质上，对 $f(X)$ 多项式的承诺实际上是对 $\bar{f}(X) = f(X) + \gamma \cdot r(X)$ 的承诺。

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

当 Prover 要证明 $f(z) = v$ 时，他不仅需要发送商多项式的 $q(X)$ 的承诺，还需要计算 $r(X)$ 在 $X = z$ 处的取值。

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

其中多项式 $\bar{q}(X)$ 是带有盲化多项式的 $\bar{f}(X)$ 除以 $(X-z)$ 后的商多项式：

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

当 Verifier 接收到 $\pi_{eval} = (\mathsf{cm}(\bar{q}), r(z))$ 后，他可以验证下面的等式：

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

直觉上，虽然 Prover 发送了 $r(X)$ 在 $r(z)$ 处的取值，只要 $r(X)$ 的 Degree 大于等于 1，那么仅通过 $r(z)$ 的取值，攻击者并不能逆向计算出 $r(X)$，因而至少还有一个随机因子在保护 $f(X)$。

实际上如果我们知道 $f(X)$ 在整个生命周期内最多只会被打开 $k<d$ 次，那么我们就没必要强制 $r(X)$ 的 Degree 为 d，而可以是一个 Degree 为 k 的多项式。因为 k 次盲化因子多项式由 $k+1$ 个随机因子构成，当 $r(X)$ 被计算 k 后，仍然还有一个随机因子在保护 $f(X)$ 的承诺。

举一个极端的例子 $r(X)$ 的 Degree 为 1，那么当 Prover 再次证明一个不同点的取值，比如 $f(z')=v'$ 时，Verifier 就有能力恢复出 $r(X)$，这样就破坏了 $f(X)$ 承诺的 Perfect Hiding 性质。

Evaluation-with-degree-bound 证明

接下来的问题是，在这个 Hiding-KZG10 方案中，能否像第一种方案一样同时证明 $f(z)=v$ 和 $\deg f \leq d$。论文 [CHMMVW19] 中给出了一个方案，与第一种方案不同的是。这个方案在证明 Evaluation with degree bound 时，需要一个交互过程（或者利用 Fiat-Shamir 转换），即 Verifier 需要提供一个公开的随机挑战数。

Commit

假设 $f(X)$ 最多只被打开 e 次，那么盲化多项式 $r(X)$ 的 Degree 只需要等于 e 即可。

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

为了证明 Degree Bound, 我们还需要承诺 $X^{D-d}\cdot f(X)$ ：

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

所以整体上，$f(X)$ 的承诺 $\mathsf{cm}(f)$ 定义为：

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

Evaluation with degree bound 协议

公共输入：

1. 多项式 $f(X)$ 的承诺 C_f
2. 多项式 $X^{D-d}\cdot f(X)$ 的承诺 C_{xf}
3. 多项式 $f(X)$ 的求值点，$X=z$

4. 多项式求值结果：$f(z) = v$

Witness：

1. 多项式 $f(X)$ 的盲化多项式 $r(X)$
2. 多项式 $X^{D-d} \cdot f(X)$ 的盲化多项式 $s(X)$

第一步：Verifier 发送随机数 $\alpha \leftarrow \mathbb{F}_r$，

第二步：Prover 按照下面的步骤

1. Prover 计算商多项式 $q(X)$:

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

3. Prover 计算聚合的盲化多项式 $t(X)$，显然 $\deg(t) \leq d$

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

4. Prover 计算商多项式 $q_t(X)$

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

5. Prover 引入一个辅助多项式 $f^*(X)$，它在 $X = z$ 处取值为 0，即 $f^*(z) = 0$

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

6. Prover 计算 $f^*(X)$ 除以 $(X - z)$ 的商多项式 $q^*(X)$，

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

6. Prover 承诺商多项式 $q(X)$，不加任何盲化因子

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

7. Prover 承诺商多项式 $q^*(X)$，不加任何盲化因子

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

8. Prover 承诺盲化多项式的商多项式 $q_t(X)$

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

9. Prover 计算合并的承诺 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 输出证明 $\pi = \big(Q, t(z)\big)$

实际上这个协议原理可以换个角度来理解。构造过程可以分解为：两个多项式在同一个点的求值的 Batch（利用 α 随机数）。其中一个是证明多项式 $f(X)$ 在 $X = z$ 处取值为 $f(z)$, 另一个是证明 $f^*(X)$ 在 $X = z$ 处取值为 0。我们可以引入一个辅助理解的多项式 $g(X)$ 来表示这两个多项式的关于 α 的随机线性组合：

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

而这个聚合后的多项式 $g(X)$ 除以 $(X - z)$ 的商多项式 $q_g(X)$ 可以表示为：

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

最后 Prover 计算的承诺 Q 恰好等于是商多项式的承诺 $[q_g(\tau)]$ 附加上随机的多项式 $[\gamma \cdot q_t(\tau)]$ 的承诺。

因此这个证明思路其实和 Evaluation 的证明思路基本一致。

Verification

Verifier 接收到的证明为 $\pi = \big(Q, t(z)\big)$，然后按下面的步骤验证：

1. 计算 $g(X) + t(X)$ 的承诺，记为 C_{g+t}：

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

2. 计算 $g(X) + t(X)$ 在 $X = z$ 处的取值的承诺，记为 V_{g+t}：

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

3. 验证 C_{g+t} 的正确性：

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

对比

第一种方案 Prover 在 Commit 的时候无需关心多项式以后会被打开几次，而只需要加一个随机因子即可实现 Perfect Hiding。而第二种方案则要求 Prover 一次性地加上足够的随机因子（以随机多项式的形式），并且保证多项式以后被打开的次数不会超过这个随机因子。

第二种方案因此带来的一个优势时，在每次证明 Evalutation 时，证明只包含一个 $\mathbb{G}_1$ 元素，加上一个 $\mathbb{F}_r$ 元素；而第一种方案则需要两个 $\mathbb{G}_1$ 元素。

进一步，第二种方案带来的第一个优势是 Verifier 只需要计算两个 Pairing，而第一种方案则需要计算三个 Pairing。

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