Gemini-PCS (Part II)

- Tianyu ZHENG tian-yu.zheng@connect.polyu.hk

在第一部分中， 我们介绍了 Gemini [BCH+22] 中实现多元多项式求值证明的 Tensor product 检查协议，并简要说明了如何将它应用到实际证明系统中来实现多元多项式到一元多项式的转换。 这一部分，我们主要关注 Tensor product 检查协议安全性，并在 Gemini 的基础上提出了一些优化。

回顾

方便阅读，我们先回顾一下 tensor product 检查协议的流程：

【Tensor-product 检查协议】

目标关系：$\langle \vec{f}, \otimes_{j=0}^{n-1}(1, \rho_j)\rangle = u$

证明者输入：公共参数，实例 $x = (\rho_0, \ldots, \rho_{n-1}, u)$, 秘密 $w = \vec{f}$

验证者输入：公共参数，实例 $x = (\rho_0, \ldots, \rho_{n-1}, u)$

1. 证明者构造一元多项式 $f^{(0)}(X) = f(X)$。
2. 对 $j \in 1, \ldots, n$，证明者计算

$$f^{(j)}(X) = f_e^{(j-1)}(X) + \rho_{j-1} \cdot f_o^{(j-1)}(X) \tag{1}$$

其中 $f_e^{(j-1)}, f_o^{(j-1)}$ 分别为 $f^{(j-1)}$ 的偶数阶项和奇数阶项构成的多项式，满足 $f^{(j-1)}(X) = f_e^{(j-1)}(X^2) + X \cdot f_o^{(j-1)}(X^2)$。

1. 证明者向验证者发送 $f^{(0)}, f^{(1)}, \ldots, f^{(n-1)}$ 的 Oracles。
2. 验证者随机选取挑战值 $\beta \leftarrow \mathbb{F}$ 并对 Oracles 进行以下查询：

$$e^{(j-1)} := f^{(j-1)}(\beta), \bar{e}^{(j-1)} := f^{(j-1)}(-\beta), \hat{e}^{(j-1)} := f^{(j)}(\beta^2) \tag{2}$$

其中 $j = 1, \ldots, n$, 当 $j = n$ 时，忽略查询 $f^{(n)}(\beta^2)$ ，并直接令 $\hat{e}^{(n-1)} := u$。

1. 对 $j = 0, \ldots, n-1$，验证者检查

$$\hat{e}^{(j)} = \frac{e^{(j)} + \bar{e}^{(j)}}{2} + \rho_j \cdot \frac{e^{(j)} - \bar{e}^{(j)}}{2\beta} \tag{3}$$

安全性分析

【Completeness】

给定一个多项式系数向量满足 $\langle \vec{f}, \otimes_{j=0}^{2}(1, \rho_j)\rangle = u$， 一个诚实执行上述协议的证明者一定能够通过验证。根据我们在【Split-and-fold 检查协议】中的讨论，当证明者正确地执行 split 得到 $f_e^{(j-1)}(X), f_o^{(j-1)}(X)$ 后，其 fold 得到的 $f^{(j)}(X)$ 一定能够通过验证，并且 $f^{(j)}(X)$ 满足如下表达式

$$f^{(j)}(X) = \sum_{i_0, \ldots, i_{n-1} \in \{0,1\}} f_{i_0 \cdots i_{n-1}} \cdot \rho_0^{i_0} \cdots \rho_{j-1}^{i_{j-1}} \cdot X^{\langle \vec{i}_{[j:n-1]}, \vec{2}^{n-1-j} \rangle} \tag{4}$$

其中 $\vec{2}^{n-1-j} = (2^0, \ldots, 2^{n-1-j})$。显然当 $j = n$ 时，$f^{(n)}(X) = \tilde{f}(\vec{\rho}) = u$。

【Soundness】

这里我们采用和 Gemini 原文不同，但更加容易理解的证明方法。假设 $\langle \vec{f}, \otimes_{j=0}^{2}(1, \rho_j) \rangle \neq u$，对于一个恶意的证明者，如果它能够输出一系列的交互数据，其中包括多项式 $f^{(j)}, j = 0, \ldots, n-1$ 的 Oracles，并通过验证。那么一定至少存在一对 Oracles，它们所包含的多项式不满足 split-and-fold 关系。即存在 j，使得

$$p_j(X) = f^{(j)}(X^2) - \frac{f^{(j-1)}(X) + f^{(j-1)}(-X)}{2} - \rho_{j-1}\frac{f^{(j-1)}(X) - f^{(j-1)}(-X)}{2X} \tag{5}$$

是非零多项式。注意到 $p_j(X)$ 的最高阶数为 $2^{n-(j-1)} - 1$。令事件 E_j 表示 $p_j(X)$ 为非零多项式且 $p_j(\beta) = 0$，根据 Schwartz-Zippel 引理，$Pr(E_j) \leq deg(p_j)/|F|$。那么，对于所有 $p_1(X), \ldots, p_n(X)$，存在非零多项式且在 β 点上刚好为 0 的概率可以用 union bound 来约束

$$Pr(E_1 \vee \cdots \vee E_n) \leq Pr(E_1) + \cdots + Pr(E_n) = \sum_{j=1}^{n} \frac{deg(p_j)}{|F|} = \frac{2N}{|F|} \tag{6}$$

【关于 degree bound】

注意在 Gemini 原论文中，验证者需要检查每个 $f^{(j)}$ 的阶数都严格小于等于 $2^{n-j} - 1$。这一点在上述证明中也有体现：即假设 $p_j(X)$ 的最高阶数为 $2^{n-(j-1)} - 1$。

但经过研究我们会发现 degree check 并不是必需的：即使恶意的证明者被允许在每一轮构造一个非法的 $f^{(j')}$，其阶数大于合法 $f^{(j)}$ 但小于等于 N，我们仍然能够得到一个可以忽略的 soundness error（尽管比原来的略大一些）。具体来说，对于任意 E_j，有 $Pr(E_j) \leq N/|F|$，可得 $Pr(E_1 \vee \cdots \vee E_n) \leq N \log N/|F|$（当 $N < D$ 时，为 $D \log N/|F|$）。

因此，第一部分中基于 KZG10 实现的 Tensor-product 检查协议中的 degree bound 检查可以忽略，以减少证明中的 $\log N$ 个 $\mathbb{G}_1$ 元素。

实现 Zero-Knowledge

Gemini 当中并没有讨论如何实现 tensor product check 的 ZK 性质，这里我们给出两种可行的方案。

【方案一】

采用与论文 [CFS17] 中实现 zk sumcheck 类似的思想，我们可以直接在原多项式 $f(X)$ 上加一个相同大小的盲化多项式 $g(X)$，即对于 $f(X)$ 中每一个系数不为零的单项式，$g(X)$ 都包含对应的一个单项式，且系数为随机值。令 $\langle \vec{g}, \otimes_{j=0}^{n-1}(1, \rho_j) \rangle = v$。

接下来，证明者只需要额外对 $g(X)$ 进行承诺，并将承诺值 $cm(g(X))$ 和 v 发送给验证者。验证者则随机选取一个挑战者 c 将 f, g 的 tensor product 关系合并为

$$u + c \cdot v = \langle \vec{f} + c \cdot \vec{g}, \otimes_{j=0}^{n-1}(1, \rho_j) \rangle \tag{7}$$

接下来，证明者和验证者共同完成对上述关系式的 Tensor product check 协议即可。我们不过多赘述这个方案的具体构造。

【方案二】

上述方法十分简单直接，但缺点是证明者需要额外增加一个和 $f(X)$ 一样大（N 长度）的随机多项式。参考Libra [XZZPS19] 中对 [CFS17] 中 zk sumcheck 的优化方案，我们同样可以提出一个实现 zk tensor product 协议的优化方案，能够显著减少盲化多项式的大小。

该优化方案的思路是：由于证明者在 tensor product check 中一共只发送了 $3n$ 个点值，因此盲化多项式至少需要包含 $3n$ 个随机系数才能够保证协议的零知识性。

具体地，令盲化多项式为：

$$g(X) = a_{0,1}X + a_{0,2}X^2 + \sum_{i=1}^{n-1}\left(a_{i,1}X^{3i} + a_{i,2}X^{3i+1} + a_{i,3}X^{3i+2}\right) + a_n \tag{8}$$

【Zero-Knowledge Tensor Product 检查协议】

为了证明 $\langle \vec{f}, \otimes_{j=0}^{n-1}(1, \rho_j)\rangle = u$

1. 证明者构造盲化多项式 $g(X)$ 并将其系数向量用零补全长度为 N
2. 证明者计算并发送下面数据给验证者

$$\begin{aligned} v &= \langle \vec{g}, \otimes_{j=0}^{n-1}(1, \rho_j)\rangle \\ C_g &= cm(g(X)) \end{aligned} \tag{9}$$

1. 验证者随机选取 $c \in F^*$ 发送给证明者，并计算 $u + c\cdot v, C_f + c \cdot C_g$。
2. 证明者和验证者运行 tensor product 检查协议证明如下关系

$$u + c\cdot v = \langle \vec{f} + c\cdot \vec{g}, \otimes_{j=0}^{n-1}(1, \rho_j)\rangle \tag{10}$$

方便起见，我们令 $h(X) = f(X) + c\cdot g(X)$。上述构造满足零知识性的证明如下：

【证明】

首先，模拟器 S 可以按照下面步骤构造：

1. S 首先输入一个随机挑战值 $c \neq 0$
2. S 均匀随机生成向量 $\vec{h}$ 并计算多项式 $h(X)$，以及 $C_h = \mathsf{Commit}(h(X)), w = \langle \vec{h}, \otimes_{j=0}^{n-1}(1, \rho_j)\rangle$
3. S 计算 $v = (w-u)/c$ 和承诺 $C_g = (C_h/C_f)^{1/c}$
4. S 与 V^* 运行一个 tensor product 检查协议，其证明关系为 $w = \langle \vec{h}, \otimes_{j=0}^{n-1}(1, \rho_j)\rangle$.

显然，第1，3步中的消息和一个诚实证明者 P 发送的消息是概率上不可区分的。

接下来，我们只需要说明第4步中运行的 S 与 V^* 运行 tensor product 检查协议同样满足这一性质即可。具体来说，因为协议满足 soundness，对于每一个 oracle $h^{(j)}, j = 0, \ldots, n-1$，其满足

$$h^{(j)}(X^2) = \frac{h^{(j-1)}(X) + f^{(j-1)}(-X)}{2} + \rho \cdot \frac{h^{(j-1)}(X) - h^{(j-1)}(-X)}{2X} \tag{11}$$

注意到，对于等式右边的 $h^{(j-1)}(X)$ 它对应的 oracle 同样满足一个与 $h^{(j-2)}(X)$ 相关的等式。因此我们总能够将任意 $h^{(j)}(X^2)$ 所满足的右式展开为一个只包含 $h^{(0)}(X), h^{(0)}(-X), h^{(0)}(X^2)$ 形式。因此 V^* 对 oracle $h^{(j)}$ 在任意点 β 上进行问询所得到的回复，一定是一个关于 $\vec{h}$ 的一个线性独立的约束。

总的来说，在进行 tensor product 检查协议后，V^* 会得到 $3 \cdot (n-1)$，2 个 $h^{(0)}(X)$ 上的取值，和一个 $h^{(n)}(X)$ 的值，即 $u + c \cdot v$。因为 h 中包含了一个大小为 $3n$ 的盲化多项式，验证者无法通过插值求出盲化多项式的所有系数，所以该协议与诚实验证者执行的协议是不可区分的。

参考文献

[BCH+22] Bootle, Jonathan, Alessandro Chiesa, Yuncong Hu, **et al. "Gemini: Elastic SNARKs for Diverse Environments." *Cryptology ePrint Archive* (2022). https://eprint.iacr.org/2022/420

[CFS17] Chiesa, Alessandro, Michael A. Forbes, and Nicholas Spooner. "A zero knowledge sumcheck and its applications." *arXiv preprint arXiv:1704.02086* (2017).

[XZZPS19] Xie, T., Zhang, J., Zhang, Y., Papamanthou, C., & Song, D. “Libra: Succinct zero-knowledge proofs with optimal prover computation.” https://eprint.iacr.org/2019/317