Notes on Virgo-PCS

Virgo is a zkSNARK proof system based on the GKR protocol. Unlike Libra, Virgo adopts a different polynomial commitment scheme, referred to as zkVPD (Verifiable Polynomial Delegation) in the paper. Virgo-zkVPD is based on the FRI (Fast Reed-Solomon IOP) protocol derived from the STARK system, making it a proof system that doesn't require a trusted setup. Its security assumptions are based on information theory and the collision resistance of hash functions.

This article introduces the protocol principles of Virgo-PCS, which differ slightly from the original paper. The PCS system in the original paper supports arbitrary multivariate polynomials, while this article only considers MLE polynomials.

1. Protocol Principles

For any MLE polynomial, $\tilde{f}(X_0, X_1, \cdots, X_{n-1})$, it can be expressed in the following coefficient form (or Monomial form):

$$\tilde{f}(X_0, X_1, \cdots, X_{n-1}) = \sum_{i=0}^{2^n-1} c_i \cdot X_0^{i_0} X_1^{i_1} \cdots X_{n-1}^{i_{n-1}} \tag{1}$$

Where $\vec{c} = (c_0, c_1, \cdots, c_{2^n-1})$ is the coefficient vector, and $\textsf{bits}(i) = (i_0, i_1, \cdots, i_{n-1})$ is the binary representation of i (Little Endian).

If we consider how to prove the evaluation of this polynomial at a given point $\vec{u} = (u_0, u_1, \cdots, u_{n-1})$, i.e., $\tilde{f}(\vec{u}) = v$, we only need to prove the following "sum":

$$\sum_{i=0}^{2^n-1} c_i \cdot u_0^{i_0} u_1^{i_1} \cdots u_{n-1}^{i_{n-1}} = v \tag{2}$$

The approach of Virgo-PCS is similar to PH23-PCS, using a Univariate Sumcheck protocol to prove the above sum. The Prover first commits to $\vec{c}$, then proves through the following Univariate Sumcheck formula:

$$c(X) \cdot m(X) = h(X) \cdot v_{\mathbb{H}}(X) + X \cdot g(X) + \frac{v}{N} \tag{3}$$

Here, $\mathbb{H}$ is a multiplicative subgroup of the finite field $\mathbb{F}_p$, with a size of $N = |\mathbb{H}| = 2^n$. The polynomial $v_{\mathbb{H}} = \prod_{x\in\mathbb{H}}(X - x)$ is the vanishing polynomial of $\mathbb{H}$. The polynomial $m(X)$ encodes the vector $\vec{m} = (m_0, m_1, \cdots, m_{2^n-1})$:

$$m_i = u_0^{i_0} u_1^{i_1} \cdots u_{n-1}^{i_{n-1}} \tag{4}$$

Then the Prover calculates the FRI commitment of $h(X)$ and sends it to the Verifier. The Prover and Verifier then use the FRI protocol to prove the existence of the following Low Degree quotient polynomial $g(X)$:

$$g(X) = \frac{N \cdot c(X) \cdot m(X) - N \cdot h(X) \cdot v_{\mathbb{H}}(X) - v}{N \cdot X} \tag{5}$$

Clearly, if we can prove that $\deg(g) < N - 1$, then we have proven that $\sum_{i=0}^{2^n-1} c_i \cdot m_i = v$.

Here is an explanation of how to convert the proof of the above "sum" into proving $\deg(g) < N-1$. To prove $\sum_{i=0}^{2^n-1} c_i \cdot m_i = v$, we can first encode c_i and m_i as polynomials $c(X)$ and $m(X)$ over $\mathbb{H}$, and then convert it to proving

$$\sum_{x\in\mathbb{H}} c(x)\cdot m(x) = v \tag{6}$$

The degree of $c(X)\cdot m(X)$ is $N-1+N-1 = 2N-2$. By decomposing $c(X)\cdot m(X)$, we get

$$c(X)\cdot m(X) = g'(X) + h(X)\cdot v_{\mathbb{H}}(X) \tag{7}$$

where $\deg(h(X)) < 2N-1-N = N-1$ and $\deg(g'(X)) < N$. Therefore, the proof of the "sum" can be converted into proving

$$\sum_{x\in\mathbb{H}} c(x)\cdot m(x) = \sum_{x\in\mathbb{H}}(g'(x) + h(x)\cdot v_{\mathbb{H}}(x)) = \sum_{x\in\mathbb{H}} g'(x) = g'(0)\cdot N = v \tag{8}$$

The second-to-last equality above is obtained from the following lemma.

> **Lemma** ([BC99]) Let $\mathbb{H}$ be a multiplicative coset of $\mathbb{F}$, and $g(X)$ be a univariate polynomial over $\mathbb{F}$ with degree strictly less than $|\mathbb{H}|$. Then $\sum_{x\in\mathbb{H}} g(x) = g(0)\cdot|\mathbb{H}|$.

Now we only need to prove $\deg(g') < N$ and $g'(0) = v/N$. This can be converted into proving that the polynomial

$$\frac{g'(X)-g'(0)}{X-0} = \frac{g'(X)-v/N}{X-0} = \frac{N\cdot g'(X)-v}{N\cdot X} \tag{9}$$

has a degree strictly less than $N-1$. The above polynomial is

$$g(X) = \frac{N\cdot c(X)\cdot m(X) - N\cdot h(X)\cdot v_{\mathbb{H}}(X) - v}{N\cdot X} \tag{10}$$

Thus, proving the "sum" is converted into proving $\deg(g) < N-1$.

This approach is generally correct, but the Verifier needs $O(N)$ work because they need to calculate the values of $m(X)$ at κ sampling points. And $m(X)$ must be a publicly computable vector from $\vec{u}$.

The Virgo paper suggests that computing the value of $m(X)$ at a point can utilize a GKR protocol, delegating the Verifier's computation to the Prover while ensuring the correctness of the computation through the GKR protocol. This way, the Verifier only needs $O(\log^2(N))$ work.

This GKR circuit is divided into four parts:

1. Calculate the value of $\vec{m}$ based on $\vec{u}$
2. Calculate the coefficient vector of $m(X)$ using the IFFT algorithm based on $\vec{m}$
3. Calculate the values of $m(X)$ on the Extended Domain $\mathbb{L}$ based on the coefficient vector of $m(X)$
4. Filter out the values of $m(X)$ on $\{\mathbb{L}_i\}_{i\in Q}$ according to the FRI-Query index set Q provided by the Verifier

Since all calculations in this GKR protocol are based on public values, and the input length of the circuit is $n = \log(N)$, the depth of the circuit is $\log(N)$, and the width of the circuit is $|\mathbb{L}| = N/\rho$, the Verifier's computational complexity is only $O(\log^2(N))$ to complete the verification.

2. Simplified Protocol

Protocol Parameters

1. Domain $\mathbb{H}$ is a multiplicative subgroup of the prime field $\mathbb{F}_p$, with size $N = 2^n$.
2. Extended Domain $\mathbb{L} \subset \mathbb{F}_p$ is a multiplicative subgroup Coset of size $|\mathbb{L}| = \rho \cdot N$, where ρ represents the code rate.

Commitment Calculation

The Prover calculates the commitment value $C_{\tilde{f}}$ of $\tilde{f}(X_0, X_1, \cdots, X_{n-1})$ similar to the general FRI protocol, calculating the values of the corresponding Univariate polynomial $c(X)$ on $\mathbb{L}$.

$$C_{\tilde{f}} = \mathsf{MerkleTree.Commit}(\vec{c}) \tag{11}$$

Evaluation Proof

Public Input

1. $C_{\tilde{f}}$
2. $\vec{u}$
3. $v = \tilde{f}(\vec{u})$

Round 1.

1. Calculate $\vec{m}$, and construct $m(X)$, whose Evaluation is $\vec{m}$

$$m(X) = m_0 \cdot L_0(X) + m_1 \cdot L_1(X) + \cdots + m_{N-1} \cdot L_{N-1}(X) \tag{12}$$

1. Construct $h(X)$, and calculate its commitment C_h, where $h(X)$ satisfies the following equation:

$$h(X) = \frac{c(X) \cdot m(X) - X \cdot g(X) - v/N}{v_{\mathbb{H}}(X)} \tag{13}$$

$$C_h = \mathsf{MerkleTree.Commit}(h|_{\mathbb{L}}) \tag{14}$$

Round 2.

The Prover and Verifier use the FRI protocol to prove the existence of $g(X)$. In the Query phase of the protocol, the Verifier provides an index set Q, and the Prover calculates the values of $c(X)$ and $h(X)$ on $\{x_i\}_{i \in Q}$:

$$\{(c(x_i), \pi_c(x_i))\} \leftarrow \mathsf{MerkleTree.Open}(i, c(X)|_{\mathbb{L}}), \quad \forall i \in Q \tag{15}$$

$$\{(h(x_i), \pi_h(x_i))\} \leftarrow \mathsf{MerkleTree.Open}(i, h(X)|_{\mathbb{L}}), \quad \forall i \in Q \tag{16}$$

Here, all x_i are elements in $\mathbb{L}$.

Round 3.

The Verifier checks the correctness of $\{c(x_i), h(x_i)\}_{i \in Q}$.

$$\mathsf{MerkleTree.Verify}(C_f, i, c(x_i), \pi_c(x_i)) \overset{?}{=} 1, \quad \forall i \in Q \tag{17}$$

$$\mathsf{MerkleTree.Verify}(C_h, i, h(x_i), \pi_h(x_i)) \overset{?}{=} 1, \quad \forall i \in Q \tag{18}$$

Round 4.

The Prover and Verifier run the GKR protocol to calculate the values of $m|_{\mathbb{L}}$, and output the values of $\{m|_{x_i}\}_{i\in Q}$, where x_i is the i-th element in the multiplicative subgroup $\mathbb{L}$.

Round 5.

The Verifier verifies the correctness of each folding step in the FRI protocol using $\{c(x_i), h(x_i), m(x_i)\}_{i\in Q}$.

3. Supporting Zero-Knowledge

To support the Zero-Knowledge property, Virgo introduces random numbers in two places:

1. A Mask polynomial $r(X)$ is added to the commitment of $c(X)$
2. In the Univariate Sumcheck protocol, a random polynomial $s(X)$ is introduced. When verifying $\sum_{x_i\in\mathbb{H}} c(x_i)m(x_i) = v$, it simultaneously proves $\sum_{x_i\in\mathbb{H}} s(x_i) = v'$.

Commitment Calculation

The Prover samples a random polynomial $r(X)$ with Degree $\kappa - 1$, i.e., containing κ random numbers.

$$c^*(X) = c(X) + v_{\mathbb{H}}(X) \cdot r(X) \tag{19}$$

$$C_{\tilde{f}} = \mathsf{MerkleTree.Commit}(c^*(X)|_{\mathbb{L}}) \tag{20}$$

Clearly, $\deg(c^*(X)) = N + \kappa - 1$.

Evaluation Proof

Public Input

1. $C_{\tilde{f}}$
2. $\vec{u}$
3. $v = \tilde{f}(\vec{u})$

Witness

1. Coefficient vector $\vec{c}$ of the MLE polynomial $\tilde{f}(X_0, X_1, \cdots, X_{n-1})$
2. Random polynomial $r(X)$

Round 1.

1. The Prover calculates $\vec{m}$, and constructs $m(X)$, whose Evaluation is $\vec{m}$

$$m(X) = m_0 \cdot L_0(X) + m_1 \cdot L_1(X) + \cdots + m_{N-1} \cdot L_{N-1}(X) \tag{21}$$

2. The Prover samples a polynomial $s(X)$ with Degree $2N + \kappa - 1$, whose sum on $\mathbb{H}$ is v'

$$\sum_{a \in \mathbb{H}} s(a) = v' \tag{22}$$

3. The Prover calculates the commitment of $s(X)$

$$C_s = \mathsf{MerkleTree.\,Commit}(s|_{\mathbb{L}}) \tag{23}$$

Round 2.

1. The Verifier provides a random number α to aggregate the sums of two different Sumcheck protocols.

$$v^* = v + \alpha \cdot v' \tag{24}$$

2. The Prover constructs $h(X)$, and calculates its commitment C_h, where $h(X)$ satisfies the following equation:

$$h(X) = \frac{c^*(X) \cdot m(X) + \alpha \cdot s(X) - X \cdot g(X) - v^*/N}{v_{\mathbb{H}}(X)} \tag{25}$$

$$C_h = \mathsf{MerkleTree.\,Commit}(h|_{\mathbb{L}}) \tag{26}$$

Round 3.

The Prover and Verifier use the FRI protocol to prove that the Degree of $g(X)$ is less than $N - 1$. This includes $O(\log(N))$ rounds of Split-and-fold.

Round 4.

1. The Verifier samples κ random indices, Q, and requires the Prover to provide the values of $c^*(X)$, $s(X)$, and $h(X)$ on $\{a_i\}_{i \in Q}$. Here $a_i \in \mathbb{L}$ is the i-th element in $\mathbb{L}$.
2. The Prover sends the values of $c^*(X)$, $s(X)$, and $h(X)$ on $\{a_i\}_{i \in Q}$, along with the Merkle paths.

$$\{(c^*(a_i), \pi_{c^*}(a_i))\} \leftarrow \mathsf{MerkleTree.\,Open}(i, c^*(X)|_{\mathbb{L}}), \quad \forall i \in Q \tag{27}$$

$$\{(s(a_i), \pi_s(a_i))\} \leftarrow \mathsf{MerkleTree.\,Open}(i, s(X)|_{\mathbb{L}}), \quad \forall i \in Q \tag{28}$$

$$\{(h(a_i), \pi_h(a_i))\} \leftarrow \mathsf{MerkleTree.\,Open}(i, h(X)|_{\mathbb{L}}), \quad \forall i \in Q \tag{29}$$

Round 5.

The Prover and Verifier run the GKR protocol to calculate the values of $m|_{\mathbb{L}}$, and output the values of $\{m|_{a_i}\}_{i \in Q}$, where a_i is the i-th element in the multiplicative subgroup $\mathbb{L}$.

Verification

1. The Verifier checks the correctness of $\{c^*(a_i), s(a_i), h(a_i)\}_{i \in Q}$.

$$\mathsf{MerkleTree.\,Verify}(C_f, i, c^*(a_i), \pi_{c^*}(a_i)) \overset{?}{=} 1, \quad \forall i \in Q \tag{30}$$

$$\mathsf{MerkleTree.\,Verify}(C_s, i, s(a_i), \pi_s(a_i)) \overset{?}{=} 1, \quad \forall i \in Q \tag{31}$$

$$\mathsf{MerkleTree.\,Verify}(C_h, i, h(a_i), \pi_h(a_i)) \overset{?}{=} 1, \quad \forall i \in Q \tag{32}$$

2. The Verifier verifies the correctness of each folding step in the FRI protocol using $\{c^*(a_i), s(a_i), h(a_i), m(a_i)\}_{i\in Q}$.

4. Summary

Virgo-PCS is one of the earliest protocols to use the MLE-to-Univariate approach to construct polynomial commitments. It is also one of the earliest protocols to use Small Field, Mersenne-61 prime field to improve performance. Although the Virgo-PCS protocol requires the MLE polynomial to be given in Coefficient form, if we only consider the commitment of the MLE polynomial, we can directly use the Evaluation (Lagrange Basis) form of the MLE polynomial for proof without converting the MLE polynomial to Coefficient (Monomial Basis) form.

References

1. [ZXZS19] Jiaheng Zhang, Tiancheng Xie, Yupeng Zhang, and Dawn Song. "Transparent Polynomial Delegation and Its Applications to Zero Knowledge Proof". In 2020 IEEE Symposium on Security and Privacy (SP), pp. 859-876. IEEE, 2020. https://eprint.iacr.org/2019/1482.
2. [PH23] Papini, Shahar, and Ulrich Haböck. "Improving logarithmic derivative lookups using GKR." Cryptology ePrint Archive (2023). https://eprint.iacr.org/2023/1284.
3. [BCRSVW19] Eli Ben-Sasson, Alessandro Chiesa, Michael Riabzev, Nicholas Spooner, Madars Virza, and Nicholas P. Ward. "Aurora: Transparent succinct arguments for R1CS." Advances in Cryptology–EUROCRYPT 2019: 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19–23, 2019, Proceedings, Part I 38. Springer International Publishing, 2019. https://eprint.iacr.org/2018/828.
4. [BC99] Byott, Nigel P., and Robin J. Chapman. "Power sums over finite subspaces of a field." *Finite Fields and Their Applications* 5, no. 3 (1999): 254-265.