TRANSPARENT POLYNOMIAL COMMITMENT SCHEME WITH POLYLOGARITHMIC COMMUNICATION COMPLEXITY

A PREPRINT

Alexander Vlasov
Matter Labs
av@matterlabs.dev

Konstantin Panarin
Matter Labs
kp@matterlabs.dev

September 10, 2019

ABSTRACT

We introduce novel efficient and transparent construction of the polynomial commitment scheme. A polynomial commitment scheme allows one side (the prover) to commit to a polynomial of predefined degree d with a string that can be later used by another side (the verifier) to confirm claimed evaluations of the committed polynomial at specific points. Efficiency means that communication costs of interaction between prover and verifier during the protocol are very small compared to sending the whole committed polynomial itself, and is polylogarithmic in our case. Transparency means that our scheme doesn't require any preliminary trusted setup ceremony. We explicitly state that our polynomial commitment scheme is not hiding, although zero knowledge can be achieved at the application level in most of the cases.

***K*eywords** polynomial commitments · zero-knowledge proofs · proximity testing

1 Introduction

A polynomial commitment scheme is a cryptographic protocol that allows a prover to publish a value, called the commitment, which binds her to a particular polynomial without revealing it. Later, he may be asked to give the value of the committed polynomial at a specific point. Prover reveals the value alongside with some correctness proof that allows verifier to ensure that provided value is consistent with the committed polynomial.

Polynomial commitment and evaluation schemes are fundamental components of many novel succinct zero-knowledge protocols, e.g. Sonic ([1]), PLONK [2]. The role of polynomial evaluation schemes in such protocols is the following: secret witness is usually encoded as a univariate polynomial and verifier wish to ensure that such an encoding satisfy some polynomial relations. Prover commits to his secret witness and later verifier queries its values at some random point and checks if all relations are satisfied at those points. As those points were taken uniformly it will be highly likely that those polynomial relations are indeed satisfied at all points.

Our commitment scheme is based on IOP (Interactive Oracle Proof) and IOPP (IOP of Proximity) protocols, namely the FRI protocol by Eli Ben-Sasson et al. [3]. IOP is a model of proof system that captures the properties of both interactive and PCP proofs. Like interactive proofs they require several rounds of message exchange between the prover and verifier, and like PCP proofs the messages provided by prover are given in form of oracles (large tables) from which verifier is allowed to query only a few random bits (elements) at his disposal.

In this paper we consider IOPP as a black box that allows prover to convince a verifier that purported evaluations $f : D \to \mathbb{F}$ (for which the verifier has oracle access, either direct (by performing queries) or *simulated* (by performing oracle access to other functions and locally performing arithmetic operations over queried values)) came from the polynomial of degree at most d. Rephrased, in IOPP (and FRI in particularly) verifier is tasked with distinguishing between the "good" case that f is a polynomial of degree at most d and the "bad" case in which f is far in relative Hamming distance from all degree-d polynomials. We treat FRI protocol as the black box that only requires verifier to have some interaction with a prover and an oracle access to the purported evaluations $f : D \to \mathbb{F}$.

We also structure this paper in much more informal and educational style to allow interested readers to derive an intuition on the subject that is not yet well covered in many sources.

2 Definitions

In this section, we lay out the building blocks that are necessary to describe our constructions.

2.1 Notation

Through this paper we use the following notations:

- $\mathbb{F}$ is a field
- $D \subset \mathbb{F}$ is an evaluation domain for our RS code words.
- function f: $D \to \mathbb{F}$
- capital letter $F(x)$ is a polynomial
- symbol $|_D$ means evaluation on the domain D
- $\hat{f}$ is an oracle to the values of some function f on the domain D

2.2 Preliminaries

Reed-Solomon Codes For some subset D of a given field $\mathbb{F}$ and a rate parameter $\rho \in (0, 1]$, we denote by $\mathsf{RS}[\mathbb{F}, D, \rho]$ the set of all functions $f : D \to \mathbb{F}$ that are evaluations of polynomials of degree $d < \rho|D|$. A *binary additive RS code family* is a code family $\mathsf{RS}[\mathbb{F}, D, \rho]$ for which $\mathbb{F} = \mathbb{F}_{2^m}, m \in \mathbb{N}$. Moreover, the set D is required to be an *additive coset*- namely that it is an additive shift of some $\mathbb{F}_2$-linear space in $\mathbb{F}_{2^m}$. A *prime field RS code family* is a code family $\mathsf{RS}[\mathbb{F}, D, \rho]$ for which $\mathbb{F} = \mathbb{F}_q$, for prime q. In this case D is a multiplicative subgroup of F_q.

2.3 Interactive Oracle Proofs and IOPs of Proximity

IOP Interactive Oracle Proofs (IOPs) ([7]) are a proof system model $\mathsf{S} = (\mathsf{P}, \mathsf{V})$ consisting of an (untrusted) prover P and a verifier V. Similarly to Interactive Proofs (IPs), they permit several rounds of message exchanges between P and V. Just like PCP proofs, they allow for messages to not be given in their entity by the prover, but as black-box oracles from which the verifier is allowed to read several bits (at their choice). On some input with length n, we define the number of rounds $r(n)$ in the protocol as its *round complexity*. The query complexity $q(n)$ of an IOP is the total number of entries read by V, while the proof length $\ell(n)$ is the sum of all message lengths denoted in number of field elements.

IOPP An Interactive Oracle Proof of Proximity IOPP is an r-round interactive IOP for the following problem: given a field $\mathbb{F}$, $d \in \mathbb{N}$, $\delta > 0$ and domain $D \subset \mathbb{F}$, the prover is provided with the representation of some function f and the verifier is given oracle access to its evaluation on domain D (i.e. an oracle to $f(x)|_D$). The prover then needs to convince the verifier that $f|_D$ is in fact evaluations of some degree d-polynomial on this domain, namely that $f \in C$, where $C = \mathsf{RS}[\mathbb{F}, D, \rho = d/|D|]$ is a family of RS-codes of degree d for domain D. Let Δ be the corresponding distance measure (usually taken to be relative Hamming distance) between some f and C. An IOPP of proximity has the following properties:

1. First message format: the first prover message, denoted f^0, is a purported codeword (evaluation of $F(x)$ on the domain D)
2. Completeness: $Pr[\langle P \leftrightarrow V \rangle = accept \; |\Delta(f^0, C) = 0] = 1$
3. Soundness: For any $P^*, Pr[\langle P^* \leftrightarrow V \rangle = accept \; |\Delta(f^0, C) > \delta] \approx 0$

FRI We are interested in the problem of distinguishing between functions that are in the set $\mathsf{RS}[\mathbb{F}, D, \rho]$ and those that are at least δ-far in Hamming distance from the nearest such function. To this end, we use the constructions in [3] [6], which are state-of-the-art to the best of our knowledge. The formal statement of its performance is provided below for readers convenience.

Fix $\nu > 0, l$ (FRI inner parameters) and an RS code family $\mathsf{RS}[\mathbb{F}_q, D, \rho]$ for which $|D| = n$ and with rate $\rho = 2^{-R}$. For a given oracle $f : D \to \mathbb{F}_q$, there exists an IOPP with the following properties:

1. **Prover Complexity:** $O(n)$ arithmetic operations over $\mathbb{F}$.
2. **Verifier Complexity:** $O(\log n)$ arithmetic operations over $\mathbb{F}$.
3. **Completeness:** If $f \in \mathsf{RS}[\mathbb{F}_q, D, \rho]$ and the prover is honest, then the verifier always accepts.
4. **Soundness:** Suppose that $\Delta(f, \mathsf{RS}) = \delta > 0$, then soundness error is bounded above by:

$$\mathsf{err}(\delta) \leq \frac{3n}{|\mathbb{F}|} + \left(1 - min\left(\delta, \frac{1 - 3\rho - \frac{2^\nu}{\sqrt{n}}}{4}\right)\right)^l$$

Later this soundness bound was greatly improved in [5] and [6], where for small code rates our choice of $\delta_0 = (1-\rho)/2$ will prevail for the min function in the formula above.

List decoding, list size for Reed-Solomon codes For $u \in \mathbb{F}^D$, set $V = \mathsf{RS}[\mathbb{F}, D, \rho] \subset \mathbb{F}^D$ be our RS code words, and distance parameter $\delta \in [0, 1]$, let $List(u, V, \delta)$ be the set of elements in V that are at most δ-far from u in relative Hamming distance. The code V is said to be (δ, L)-list-decodable if $|List(u, V, \delta)| \leq L$ for all $u \in \mathbb{F}_q^D$.

For $D \subseteq \mathbb{F}$, we denote let $L_\delta = L(\mathbb{F}, D, d, \delta)$ be the maximum size of $List(u, V, \delta)$ taken over all $u \in \mathbb{F}^D$ for $V = RS[\mathbb{F}, D, \rho = d/|D|]$.

Unique decoding radius For Reed-Solomon codes there exists a unique decoding radius $\delta < \frac{1-\rho}{2}$ such that $L_\delta \leq 1$. We denote δ_0 the unique decoding radius through the rest of the paper.

3 Polynomial commitment schemes

In this section we provide formal definitions and properties of polynomial commitment schemes.

3.1 Definition

Polynomial commitment schemes can be formalized as a tuple of five algorithms:

$$\Pi = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{VerifyPoly}, \mathsf{Open}, \mathsf{Verify})$$

Definition 1. *A (transparent) polynomial commitment scheme Π is a tuple of algorithms with the following semantics:*

- *Setup: $(k, \mathbb{F}, d) \mapsto Params$. Given security parameter k, field F and maximal degree d of supported polynomials, generates scheme parameters.*
- *Commit: $(Params, \phi(x)) \mapsto \mathcal{C}$. Given a polynomial $\phi(x)$ of degree $\leq d$ and the generated public parameters, the prover outputs a commitment $\mathcal{C}$.*
- *VerifyPoly: $(Params, \phi(x), \mathcal{C})$. Given some polynomial, check if it is consistent with the commitment.*
- *Open: $(\phi(x), i, Params) \mapsto (\phi(i), w_i)$ Prover is asked to open evaluation of $\phi(x)$ at point i. He returns the opening alongside with some proof of correctness w_i.*
- *Verify: $(\mathcal{C}, Params, i, z, w_i) \mapsto$ (acc, rej) - given polynomial commitment, evaluation point, purported value $z = \phi(i)$ of committed polynomial on this point, verifier checks correctness proof and either accept or reject it.*

In our construction, we define a version of this scheme in which the open and verify algorithms are conducted through an IOP. *Secure* polynomial commitment scheme should posses additional properties:

Definition 2. *A polynomial commitment scheme Π is considered secure if the following holds:*

- ***Correctness.*** *If pp $\leftarrow$ $\mathsf{Setup}(1^\kappa, d)$ and $\mathcal{C} \leftarrow \mathsf{Commit}(pp, \phi(x))$, then $\forall \phi(x) \in \mathbb{F}_{<d}[X]$ and $\forall i \in \mathbb{F}$ we have that:*

$$\mathsf{Open}(pp, \phi(x), i) \mapsto (z, w_i) \Rightarrow \mathsf{Verify}(pp, \mathcal{C}, i, z, w_i) = 1 \cup z = \phi(i)$$

- ***Polynomial Binding.*** *For all adversaries $\mathcal{A}$:*

$$\Pr\left(\begin{array}{c|c} VerifyPoly(pp, \mathcal{C}, \phi(x)) = 1 & pp \leftarrow \mathsf{Setup}(1^\kappa, d) \\ VerifyPoly(pp, \mathcal{C}, \phi^{'}(x)) = 1 & (\mathcal{C}, \phi(x), \phi^{'}(x)) \leftarrow \mathcal{A}(pp) \\ \phi(x) \neq \phi^{'}(x) & \end{array}\right) \leq \epsilon(\kappa)$$

- ***Evaluation Binding.*** *For all adversaries $\mathcal{A}$:*

$$\Pr\left(\begin{array}{c|c} \textit{Verify}(pp, \mathcal{C}, i, z_1^i, w_1^i) = 1 & \\ \textit{Verify}(pp, \mathcal{C}, i, z_2^i, w_2^i) = 1 & pp \leftarrow \textsf{Setup}(1^\kappa, d) \\ z_1^i \neq z_2^i & (\mathcal{C}, \langle i, z_1^i, w_1^i \rangle, \langle i, z_2^i, w_2^i \rangle) \leftarrow \mathcal{A}(pp) \end{array}\right) \leq \epsilon(\kappa)$$

The informal explanation of this properties is the following:

- **polynomial binding**: the commitment should perfectly binds the polynomial, i.e. once prover publish a commitment data, it would be difficult for him to find another polynomial of degree $\leq d$ with the same commitment. Moreover, (and this is a very *strict* requirement!) publishing a commitment is indeed forces the prover to choose and fix one polynomial of degree less or equal than d. In other words published commitment means that the prover has particular polynomial in mind (and not just random garbage function).
- **evaluation binding:** If prover publishes z as an opening at point i, but $z \neq \phi(i)$, then this value of z will be with high probability rejected during the *Verify* step.

Remark 1: We do not require our scheme to be perfectly polynomial hiding. Kate commitment ([4]) has perfect hiding property: opening of committed polynomial at any point doesn't give the verifier any additional information on that polynomial (unless the verifier have collected $k + 1 \geq deg(\phi(x))$ openings from which the polynomial is completely derived via Lagrange interpolation). Often times hiding is achieved at the application level: in almost all zero-knowledge protocols some commit-reveal scheme is applied not to the witness polynomial directly (such witness incorporates prover's private data) but to some "masked" polynomial, generated from witness via addition of the masking coefficients.

Remark 2: We'll briefly show how Kate commitment forces prover to choose and fix some polynomial of degree $\leq d$. In Kate commitment precomputed CRS is a set of $\{g^{\alpha^k}\}_{k=0}^d$, where α - unknown parameter and $g \in \mathbb{G}$ - generator of some cyclic group $\mathbb{G}$ of size p. Prover's commitment is an element in the form g^b for some $b \in \mathbb{Z}_p$. Let us think how the prover may generate another commitment. If he doesn't have capabilities of breaking discrete-log problem, the only opportunity is to choose some product of elements from $\{g^{\alpha^k}\}_{k=0}^d$ (with repetitions). Assume g^{α^k} is used $b_k \in \mathbb{Z}_p$ times, so prover's commitment will be equal to commitment of polynomial $\sum_{i=0}^d b_k X^k \in \mathbb{Z}_p[X]$, and we assume it is the polynomial the prover bears in mind.
Remark 3: Another crucial property for us is *succinctness* of polynomial commitment scheme, where succinctness is measured in terms of communication complexity. Naively prover could output coefficients of the polynomial and verifier can perform the evaluation himself, but this scheme would require linear communication cost and linear verifier running time. We target communication complexity and verification time to be poly-logarithmic in d, where d is the degree of precommitted polynomial $\phi(x)$.

4 Transparent polynomial commitment scheme

Following Kate et al. ([4]) we base our scheme on the following simple observation: if $F(x)$ is a polynomial of degree d, then $F(i) = z$ iff $F(x) - z = (x - i)Q(x)$ where $Q(x)$ is some polynomial of degree $d - 1$.

4.1 Protocol

Roughly speaking, FRI protocol allows prover to convince a verifier that some oracle (commitment) $\hat{f}$ to arbitrary function f is δ-close in a Hamming weight to evaluation of some low degree polynomial $F(x)$, that is $F(x)|_D = f$, with high probability. For a chosen δ parameter soundness error $\epsilon(\delta)$ (that is f is not δ-close to any low-degree polynomial $F(x)$ on domain D, but verifier accepts) is constant, once we fix initial domain, internal parameters of the FRI protocol, purported degree d, rate parameter ρ and distance δ. Later we refer to such error as simply $\epsilon(\delta)$ as all other parameters are clear from the context.

- **Setup**: $1^k \mapsto F, D, \rho, \nu, l$ - given security parameter, choose field, evaluation domain, rate and FRI inner parameters.
- **Commit**: given polynomial $F(x)$ of degree d prover's commitment is evaluation $F(x)$ on domain D (i.e. prover gives an oracle $\hat{f}$ to *supposed* evaluations $F(x)|_D$). As a part of the commit phase prover and verifier are engaged in IOPP for $F(x)$, so that verifier would be convinced that $\hat{f}$ is indeed at least δ_0-close to an evaluation of some degree d polynomial and thus there exists a *unique* (we are inside unique decoding radius!) polynomial such that $|\hat{f}, F(x)|_D| < \delta_0$. If FRI verification fails the verifier aborts the protocol.

- **VerifyPoly**: Prover outputs coefficients of his polynomial $F(x)$ and verifier can independently recalculate $F(x)|_D$ and check that his evaluations are δ_0-close to $\hat{f}$.
- **Open**: prover is asked for opening of $F(x)$ at point i, $i \notin D$. Prover responds with some value $z \in F$. Define $q(x) = (F(x) - z)/(x - i)$. We assume $|D| \ll |F|$ and $i \notin D$. Prover and verifier are engaged in IOPP to prove that function $q(x) = \frac{F-z}{x-i}$ is δ_0 close to some degree $(d-1)$ polynomial. If IOPP passes than verifier is with high probability convinced that $F(i) = z$. Verifier simulates an oracle $\hat{q} = q(x)|_D$ to *supposed* evaluations of $q(x)$ through $\hat{f}$, i and z. By Lemma 5.3 from [6] function $q(x) = \frac{f-z}{x-i}$ is δ_0-close to some polynomial of degree $d-1$ iff $F(i) = z$. If FRI verification fails verifier outputs $reject$, and $accept$ otherwise.

Correctness of this scheme follows from the logic of the Kate commitment from above. Succintness stem from the fact that instead of giving d coefficients of polynomial $F(x)$ prover provides $O(log^2(d))$ commitments and openings during FRI.

Limitations. Such polynomial commitment scheme can only be used for field $\mathbb{F}$ that allow instantiating of the FRI protocol. For example, it requires a multiplicative subgroup of the proper size 2^k if $\mathbb{F} = \mathbb{F}_q$ is a prime field. That nevertheless allows a lot of practical applications. We once again emphasize that our scheme is not hiding because queries to committed values during the FRI protocol disclose information about the polynomial values at the point other than a queried point i. Nevertheless, the number of disclosed values is $O(log(d))$ for a polynomial of degree d.

4.2 Soundness

In our scheme prover provides an oracle $\hat{f}$ to *supposed* values of $F(x)$ on domain D. From verifier's point of view there is no structure in such an oracle. Without FRI (w.r.t to $F(x)$) there is no way to guarantee that function under the oracle is an evaluation of some low-degree polynomial. Even if prover has passed FRI verification on commitment step we may only guarantee that our initial commitment was δ-close to some RS-code. This situation is quite different from Kate commitment where verifier may be convinced that prover's commitment is indeed the commitment to some *unique* low-degree polynomial (otherwise prover has broken SDH-assumption, see remark 2 above).

Polynomial binding and evaluation binding properties of our scheme follow from the reasoning below with a slight limitation that verifier can only sample $i \in \mathbb{F}$, $i \notin D$.

In our case if prover passes FRI, then with high probability we may be sure that his commitment is at most δ_0-far from the space of RS codes. As δ_0 was chosen to be the unique decoding radius, there is in fact a *unique* polynomial $F(x)$ of degree $\leq d$ under the oracle $\hat{f}$, such that $\Delta(F(x)|_D, \hat{f}) \leq \delta_0$. Then analogously to our discussion of Kate commitment scheme we assume that the polynomial that prover bears in mind *is* $F(x)$ and this is the polynomial for which openings later will be provided. This is justified by the fact, that prover may easily obtain $F(x)$ from $\hat{f}$ in $\mathcal{O}(|D|^3)$ operations via *Berlekamp–Welch decoding* algorithm. Uniqueness of the polynomial gives the polynomial binding property.

Verifier ask a prover to open the committed polynomial at point i outside D, prover claims that corresponding opening is z. Then verifier and prover are engaged into FRI protocol with respect to $q(x) = \frac{f(x)-z}{x-i}$ of purported degree $d-1$. Note, that verifier may simulate values of q via oracles to f.

At the end of FRI protocol verifier is convinced (with except for $\epsilon(\delta_0)$ probability) that $q(x)$ is δ_0-close to an evaluation of some $d-1$ degree polynomial $Q(x)$. This means:

$$f(x) = z + Q(x)(x-i) \ \forall x \in D \text{ except at most } \delta_0 n \text{ points.}$$

This means that $H(x) = z + Q(x)(x-i)$ is a polynomial of degree at most d which is δ_0-far from $f(x)$. However from commit step we know that the only polynomial with such property is $F(x)$. Hence $F(x) = H(x)$ identically:

$$F(x) = z + Q(x)(x-i)$$

Substituting $x = i$ on both sides we arrive at $F(i) = z$. Uniqueness of the $Q(x)$ polynomial and a relationship $F(x) = z + Q(x)(x-i)$ gives evaluation binding property.

5 Optimizations

By the application of Lemma 5.3 from [6] we can eliminate the first FRI check on the *Commit* step. If we simulate an access to the function $q(x) = \frac{f-z}{x-i}$ using the existing oracle $\hat{f}$ and for some unique polynomial $Q(x)$ of degree

$d-1$ we are convinced that $|q(x), Q(x)|_D| < \delta_0$, then we can immediately reason that $|f, F(x)|_D| < \delta_0$ where $F(x) = Q(x)(x-i) + z$. In the original description of the protocol in a section 4.1 we require a FRI proof to be present at the commitment step *solely* for a purposes of identifying a *unique* polynomial early in the protocol - at the commitment step.

From the implementation perspective the protocol would be the following (leaving only the important steps):

- For a polynomial $F(x)$ of degree $\leq d$ prover provides an oracle access to *supposed* evaluations of $F(x)$ on domain D.
- Vefirier samples a point $i \notin D$, $i \in \mathbb{F}$ at which he asks a prover to provide a value of the polynomial $F(x)$.
- Prover outputs a purported values z along with a FRI proof that a function $q(x) = \frac{F-z}{x-i}$ is δ_0 close to some polynomial $Q(x)$ of degree $\leq d-1$.
- If FRI prove passes then verifier is convinced that under the commitment there was *unique* polynomial $F(x)$ of degree $\leq d$, such that $F(i) = z$.

6 Numeric estimates

Here, we give exact numbers for FRI-soundness derived from the formula above, more precisely for the case we are interested in: field size is 256-bits, ρ = 1/16, $d = 2^{28}, l = \log d = 28, \nu = 1, \delta = \delta_0 = (1-\rho)/2$. Then the formula gives us:

$$\mathsf{err} \leq \frac{3 \cdot 2^{32}}{2^{256}} + \left(1 - \frac{1 - 3/16 - 2/2^{16}}{4}\right)^{28} \approx 0.00173397200919$$

Using an improved bound from [5] where FRI soundness is instead determined by our δ_0 we get

$$\mathsf{err} \approx \left(1 - \frac{1 - 1/16}{2}\right)^{28} < 2^{-25}$$

In order to improve soundness we may repeat the FRI verification protocol as may times as needed and get multiplicative decrease in error.

References

[1] Mary Maller, Sean Bowe, Markulf Kohlweiss, Sarah Meiklejohn. Sonic: Zero-Knowledge SNARKs from Linear-Size Universal and Updateable Structured Reference Strings. In *Cryptology ePrint Archive, Report 2019/099*

[2] Ariel Gabizon, Zachary J. Williamson, Oana Ciobotaru. : Permutations over Lagrange-bases for Oecumenical Noninteractive arguments of Knowledge.

[3] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, Michael Riabzev. Fast Reed-Solomon Interactive Oracle Proofs of Proximity.

[4] Aniket Kate, Gregory M. Zaverucha, Ian Goldberg. Constant-Size Commitments to Polynomials and Their Applications.

[5] Eli Ben-Sasson, Swastik Kopparty, and Shubhangi Saraf. Worst-case to average case reductions for the distance to a code.

[6] Eli Ben-Sasson, Lior Goldberg, Swastik Kopparty, Shubhangi Saraf DEEP-FRI: Sampling Outside the Box Improves Soundness.

[7] Eli Ben-Sasson, Alessandro Chiesa, Nicholas Spooner Interactive Oracle Proofs.