Polynomial Commitments

Lecture 18
Hash-based Approaches
Polynomial Commitment

Prover wants to (succinctly) commit to a polynomial and later let the verifier (interactively) evaluate it on points of its choice.

Generally, a multi-variate polynomial with a known number of variables and known degree

e.g., a multi-linear polynomial in GKR. In some other applications, univariate polynomial of a known degree

Trivial solution: send the coefficients of the polynomial

But not succinct and evaluating the polynomial is expensive

Want verifier’s computation/communication to be sub-linear in the size of the polynomial

Non-trivial solutions: Using Merkle hashes and low-degree tests; from hardness of discrete logarithm; from bilinear pairings; using “IOPs”; ...
Today: Two approaches using collision-resistant hash functions

Involves committing to strings (implemented via Merkle trees) and proving that the committed strings are low degree polynomials

Approach 1: (“Ligero”-like)
- Faster prover, but longer proof

Approach 2: (“FRI”)
- Slower prover, but shorter proof
Polynomial Commitment

Ligero Approach

Write the n coefficients of the (univariate or multi-linear) polynomial P as a √n × √n matrix M, s.t. P(X) = u(X)^T M v(X), where u,v are vectors which do not depend on P

Univariate case: M_{ij} = coefficient of X^{i/\sqrt{n}+j} where i,j ∈ {0,...,\sqrt{n}-1}, v(X) = [ 0 X X^2 .. X^{\sqrt{n}-1} ]^T and u(X) = v(X^{\sqrt{n}})

Multi-linear case: Write the multi-linear extension of f in the Lagrange basis: P(X_1,..,X_n) = \sum_{\alpha \in \{0,1\}^n} f(\alpha) \Pi_{i: \alpha_i=0} (1-X_i) \Pi_{i: \alpha_i=1} X_i

Basis polynomial \chi_{\alpha}(X_1,..X_n) = \chi_a (X_1,..X_{n/2}) \chi_b (X_{n/2+1},..X_n), where \alpha \in \{0,1\}^n, a,b \in \{0,1\}^{n/2} such that \alpha = (a,b)

M_{a,b} = f(a,b)
Polynomial Commitment

Ligero Approach

- Write the $n$ coefficients of the (univariate or multi-linear) polynomial $P$ as a $\sqrt{n} \times \sqrt{n}$ matrix $M$, s.t. $P(X) = u(X)^T M v(X)$, where $u,v$ are vectors which do not depend on $P$

- Encode each row of $M$ into a longer vector, using a linear error correcting code

- $E = M G$, where $G$ is the generator matrix of the ECC

- Naïvely, $O(n^{1.5})$ time to compute $E$

- But using ECC with linear encoding time, can compute $E$ in time $O(n)$

- A single row can be computed in time $O(\sqrt{n})$
Polynomial Commitment

Ligero Approach

Write the $n$ coefficients of the (univariate or multi-linear) polynomial $P$ as a $\sqrt{n} \times \sqrt{n}$ matrix $M$, s.t. $P(X) = u(X)^T M v(X)$, where $u, v$ are vectors which do not depend on $P$.

Encode each row of $M$ into a longer vector, using a linear error correcting code.

Commitment: Commitment to all of $E$ using a Merkle hash, followed by a row & column check: Verifier asks for $m^T = r^T M$ for a random vector $r$, and encodes $m^T$ into $c^T = m^T G$; then it asks for opening of a few columns of $E$, and check that they match $c^T$.

Idea: Either most columns match a (unique) valid $E$, or w.h.p. $r^T E$ will be far from every codeword $c^T$. 
Polynomial Commitment

Ligero Approach

- Write the n coefficients of the (univariate or multi-linear) polynomial P as a √n × √n matrix M, s.t. P(X) = u(X)^T M v(X), where u,v are vectors which do not depend on P
- Encode each row of M into a longer vector, using a linear error correcting code
- Commitment: Commitment to all of E using a Merkle hash, followed by a row & column check: Verifier asks for m^T = r^T M for a random vector r, and encodes m^T into c^T = m^T G; then it asks for opening of a few columns of E, and check that they match c^T
- Opening: Need to evaluate u^T M v for known u,v. Verifier asks for m^T = u^T M, computes codeword c^T and checks it in a few committed columns, as in the commitment check step; it then evaluates m^T v
- Proof size is O(√n), from opening the columns and sending m^T
Polynomial Commitment

FRI Approach

To commit to a degree d univariate polynomial \( P(X) \), commit to \( \{ P(x) \mid x \in L_0 \} \) using a Merkle tree, where \( L_0 \) is a suitably chosen small subset.

To prove that \( P(\alpha) = v \) for \( \alpha \not\in L_0 \) and that \( P \) is a degree d polynomial.

Fact: If \( P(\alpha) = v \), then \( (X-\alpha) \) divides \( P(X) - v \).

i.e., if \( P(X) - v \) is a polynomial that has a root at \( \alpha \), it must have \( X-\alpha \) as a factor.

By considering remainder for the polynomial division of \( P(X) - v \) by the linear polynomial \( X-\alpha \): it is a constant \( c \) such that \( P(X) - v = (X-\alpha)Q(X) + c \). But evaluating at \( X=\alpha \), we get \( 0=c \).

Prover should prove (to be specified how) that the committed function \( f(X) \) is s.t. \( (f(X)-v) \cdot (X-\alpha)^{-1} \) corresponds to a degree d-1 polynomial \( Q(X) \).
Polynomial Commitment

FRI Approach

To commit to a degree d univariate polynomial \( P(X) \), commit to \( \{ P(x) \mid x \in L_0 \} \) using a Merkle tree, where \( L_0 \) is a suitably chosen small subset.

To prove that the committed function \( f(X) \) is s.t. \( (f(X)-v) \cdot (X-\alpha)^{-1} \) corresponds to (is close to) a degree \( d-1 \) polynomial \( Q(X) \) will be a recursive proof: At each level proving that \( G_i(X) \) (with evaluations on a set \( L_i \) committed) is close to a degree \( d_i \) polynomial.

Reduced to proving that evaluation of \( G_{i+1} \) in \( L_{i+1} \) is close to a degree \( d_{i+1} \) polynomial where \( d_{i+1} = \lfloor d_i/2 \rfloor \).

Base case is \( G_i^* \) is a constant. Instead of commitment send it directly.
Polynomial Commitment

FRI Approach

- At each level proving that \( G_i(X) \) (with evaluations on a set \( L_i \) committed) is close to a degree \( d_i \) polynomial
- Reduced to proving that evaluation of \( G_{i+1} \) in \( L_{i+1} \) is close to a degree \( d_{i+1} \) polynomial where \( d_{i+1} = \lfloor d_i / 2 \rfloor \)
- Sender commits to \( G_i(X) \) evaluated in \( L_i \); Verifier sends \( \alpha_i \leftarrow F \)
- Define \( G_{i+1}(Y) = B_i(\alpha_i, Y) \) where \( G_i(X) = B_i(X, X^2) \)
  - i.e., let \( B_i(X,Y) = E_i(Y) + X D_i(Y) \) where \( E_i(X^2) \) has the even degree terms of \( G_i(X) \) and \( X D_i(X^2) \) has the odd degree terms
- \( G_{i+1} \) has degree \( d_{i+1} = \lfloor d_i / 2 \rfloor \) \( L_{i+1} = \{ z^2 \mid z \in L_i \} \). For suitable \( F \) and \( L_0, |L_{i+1}| = |L_i| / 2, -1 \in L_i \) and \( L_i \) closed under multiplication
- Verifier repeats the following with several different \( s_0 \):
  - Pick \( s_0 \leftarrow L_0 \). For all \( i > 0 \), let \( s_{i+1} = s_1^2 \in L_{i+1} \).
  - Open \( G_i(s_i) = B_i(s_i, s_{i+1}) \) and \( G_i(-s_i) = B_i(-s_i, s_{i+1}) \). As \( B_i(X, s_{i+1}) \) is linear can compute \( B_i(\alpha_i, s_{i+1}) \) from \( B_i(\pm s_i, s_{i+1}) \). Check against \( G_{i+1}(s_{i+1}) \).
Polynomial Commitment

FRI Approach

At each level proving that $G_i(X)$ (with evaluations on a set $L_i$ committed) is close to a degree $d_i$ polynomial

Reduced to proving that evaluation of $G_{i+1}$ in $L_{i+1}$ is close to a degree $d_{i+1}$ polynomial where $d_{i+1} = \lceil d_i/2 \rceil$

Soundness idea: If the recursive check were perfect (i.e., $G_{i+1}$ is exactly a degree $d_{i+1}$ polynomial), what is the probability that $G_i$ not close to a degree $d_i$ polynomial is not caught?

$G_{i+1}(Y) = B_i(\alpha_i,Y) = B_i(0,Y) + \alpha_i [ B_i(1,Y)-B_i(0,Y) ]$

Case both $B_i(0,Y)$ and $B_i(1,Y)$ are close to degree $d_{i+1}$ polynomials $T_0(Y)$ and $T_1(Y)$: Can show $G_i(X)$ is close to the degree $d_i$ polynomial $T_0(X^2) + X [ T_1(X^2) - T_0(X^2) ]$

Case one of them is far from every degree $d_{i+1}$ polynomial: Can show $G_{i+1}$ is far from every such polynomial w.h.p.

Soundness amplified by repeating with different $s_0$ (but same $\alpha_i$)
Polynomial Commitment

FRI Approach: Summary

- To commit to a degree d univariate polynomial $P(X)$, commit to $\{ P(x) \mid x \in L_0 \}$ using a Merkle tree, where $L_0$ is a suitably chosen small subset.

- To prove that $P(\alpha) = v$ for $\alpha \not\in L_0$ and that $P$ is a degree $d$ polynomial:
  - Prove that the committed function $f(X)$ is s.t. $(f(X) - v) \cdot (X - \alpha)^{-1}$ corresponds to a degree $d-1$ polynomial $Q(X)$.
  - Recursively prove that committed function $G_i$ in $L_i$ is close to a degree $d_i$ polynomial, where $d_i = \lceil d_{i-1}/2 \rceil$.

- Write $G_i(X) = B_i(X, X^2)$ and define $G_{i+1}(Y) = B_i(\alpha_i, Y)$.

- Verifying the relation between $G_{i+1}$ and $G_i$ uses $G_i(\pm s_i)$ as two points on $B_i(X, s_{i+1})$ to reconstruct $G_{i+1}(s_{i+1})$.

- Relies on size of $F$ being suitable.

- FRI works for univariate polynomials. Extensions to multi-linear polynomials are less efficient.