FRI Algo

Position : **R&D Cryptography**

Mohamed Alaoui
[LinkedIn]

August 20, 2024

Agenda

Agenda Checkpoint I

FRI protocol overview

FRI protocol

Overview

Definition

- FR IOPP is for Fast Reed-Solomon Interactive Oracle Proofs of Proximity
- It's based on Reed-Solomon codes, which are error-correcting codes with important properties in coding theory and cryptography
- It's designed to be fast, with linear proof complexity and logarithmic verification complexity

Objective

- Goal : Proving that a committed polynomial is close to a low degree polynomial
- Goal transformation to reduce the committed domain and the polynomial degree

Context

- FRI is particularly useful for verifying computations over large datasets, making it valuable for blockchain and other distributed systems applications
- Key component in Zero-Knowledge Scalable Transparent ARguments of Knowledge (ZK-STARK) systems

FRI protocol
Quick high level description

Protocol

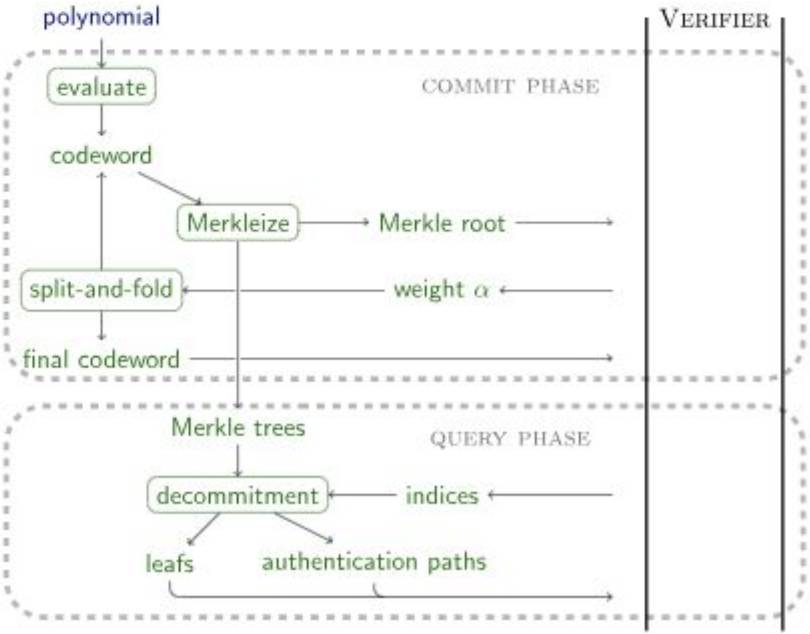
Input : The polynomial function f(x) is close to a Polynome P of a low degree (degree < to some D) :

Commit :

- Prover commit on some computation
  - Evaluate f(x) on a well chosen domain
  - Commit the Merkle root of the evaluation
- Reduce the problem size by using a transparent challenge (here alpha in the diagram) to allow him to control the computation at verification step
- Stop reduction when the final code word is computed. it converge as the polynomial function will reach the 0 degree

Decommit :

- Prover prepare the decommitment data layers according the the verifier query
- Verifier check some evaluation and stop when he is convinced



@https://aszepieniec.github.io/stark-anatomy/fri.html

Agenda Checkpoint II

Fundamental principles

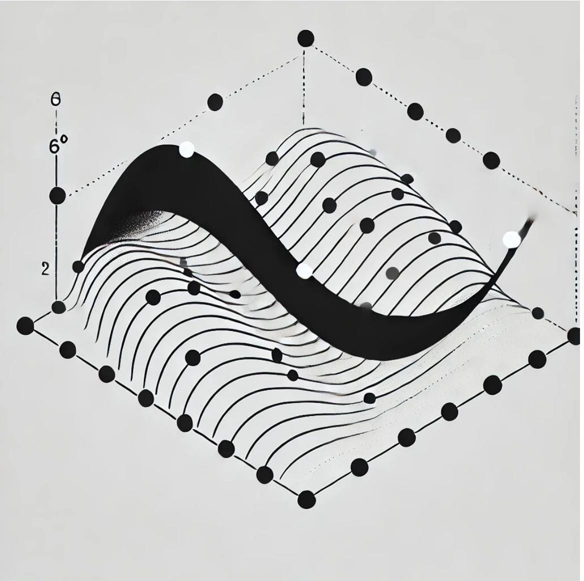
Fundamental principles

Proximity to Polynomials

Key points

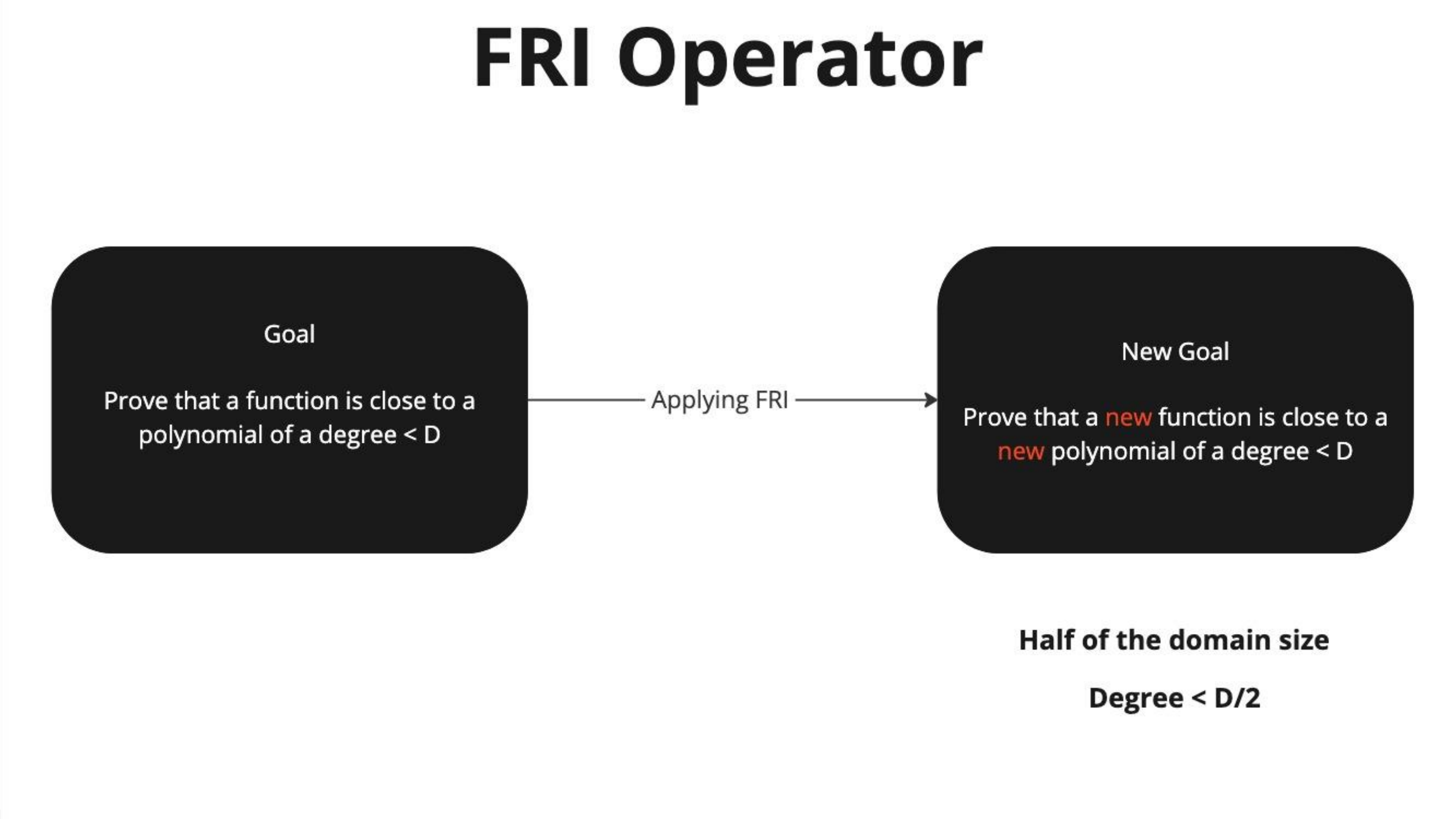
- Let consider **f:** $S \rightarrow F$ a function from S include in F a finite field F

- **f** is close to a polynomial **p** if the number of different evaluation points is **small.** **Small** means less to a specific value.



Fundamental principles

FRI operator reduction

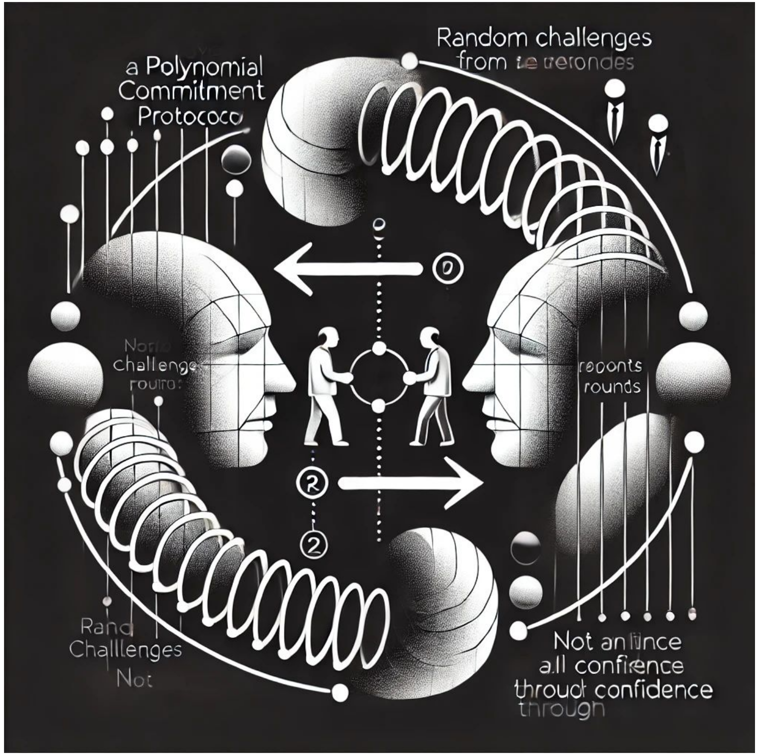


Fundamental principles

Prover/Verifier interaction

Key points

- **Prover**: The party that commits to a polynomial and wants to prove its close to a low degree polynomial.
- **Verifier**: The party that checks the proof and decides whether to accept or reject it.
- **Rounds of interaction**: The protocol involves multiple rounds where the verifier sends random challenges and the prover responds with specific polynomial evaluations.
- **Probabilistic verification**: The interactive nature allows for probabilistic verification, where the verifier can be convinced with high probability without checking every point of the polynomial.



Fundamental principles

Complexity

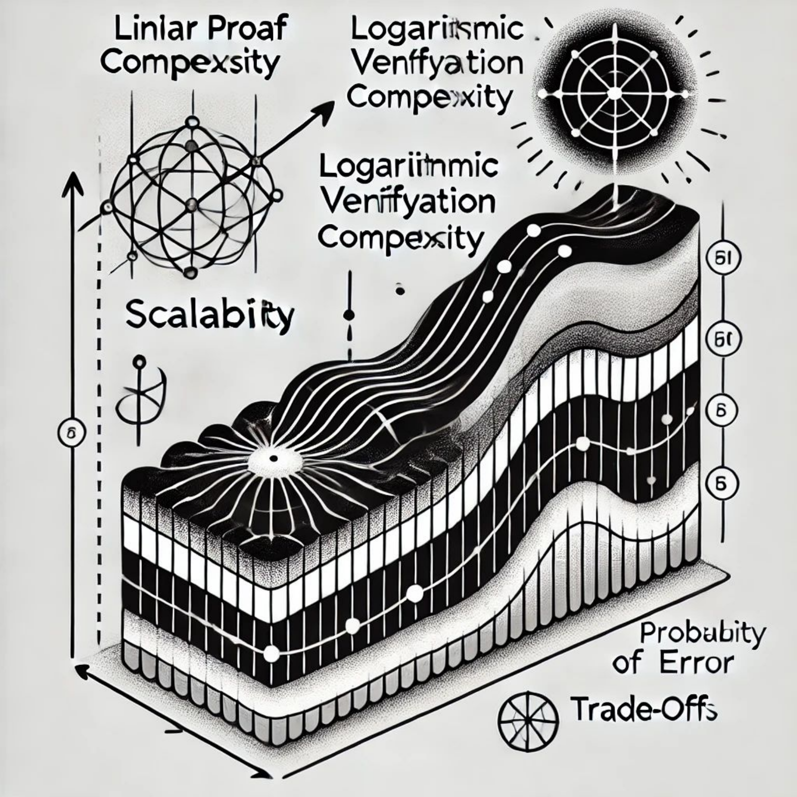
Key points

- **Linear proof complexity**: The size of the proof grows linearly with the degree of the polynomial being proven.

- **Logarithmic verification complexity**: The verifier's work grows logarithmically with the degree of the polynomial.

- **Scalability**: These complexity characteristics make FRI highly scalable, allowing it to handle proofs for very large computations efficiently.

- **Trade-off**: FRI achieves this efficiency by accepting a small probability of error, which can be made arbitrarily small by repeating the protocol.



Agenda Checkpoint III

MAIN STEP
Commit phase
Folding phase
Query phase

Commit

Process

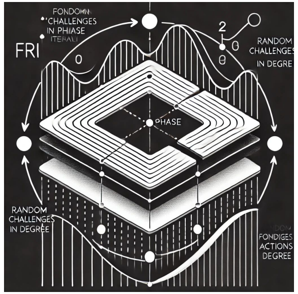
Key points

- **Initial commitment**: The prover starts by committing to the polynomial they claim has a bounded degree. This commitment fixes the polynomial and prevents the prover from changing it later.

- **Merkle tree usage**: The commitment is typically implemented using a Merkle tree. This data structure allows for efficient and secure commitments to large datasets.

- **Merkle root commitment**: Only the root of the Merkle tree is sent to the verifier. This root serves as a compact representation of the entire polynomial.

- **Efficiency**: Using a Merkle tree allows for succinct proofs and efficient verification later in the protocol.



Folding

Process

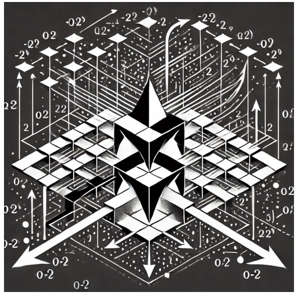
Key points

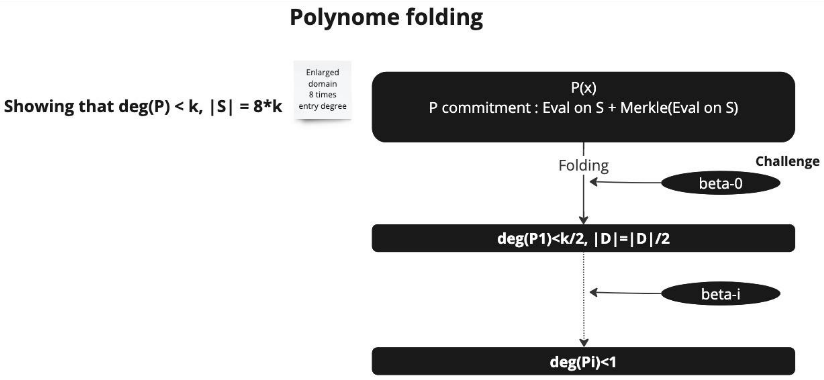
- **Degree reduction**: The core of FRI is an iterative process that reduces the degree of the polynomial at each step.

- **Verifier-provided randomness**: The verifier supplies random values that are used in the folding process. This randomness is crucial for the security of the protocol.

- **Layer creation**: Each iteration of the folding process creates a new "layer" of the proof, with each layer representing a polynomial of lower degree than the previous one.

- **Interactive nature**: This phase highlights the interactive aspect of FRI, with the verifier actively participating in the proof construction.





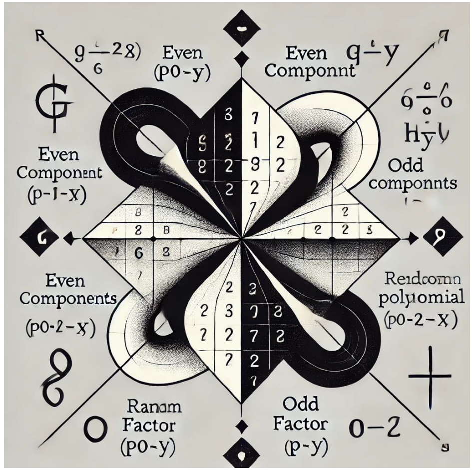
Folding

Mecanism

Key points

Split to Even and Odd Powers:

- The polynomial is represented as : $P_0(x) = g(x^2) + x \cdot h(x^2)$
- With beta we consider the new polynome : $P_1(y) = g(y) + \beta \cdot h(y)$
- This new polynomial is now of a lower degree than the original polynomial
- Eval on the new domain and commit the new merkle root
- As a final result the prover commit on several layers of evaluation and merkle roots



Query

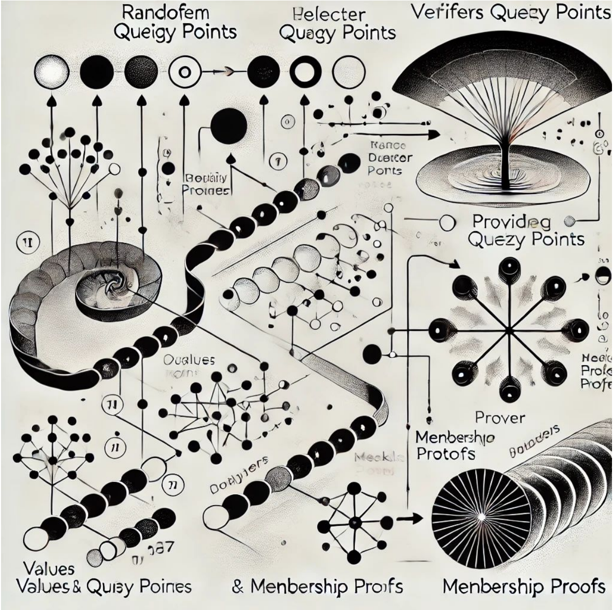
Process

Key points

I consider the FRI mandate ended when the prover prepare the value and the path proofs

Verifier selects query points

- **Random selection**: The verifier randomly chooses a set of query points. This randomness is crucial for the security of the protocol.
- **Number of queries**: The number of query points is typically logarithmic in the degree of the original polynomial. This contributes to the protocol's efficiency.
- **Query domain**: These points are selected from the domain of the polynomial at each layer of the FRI protocol.



Prover provides values and membership proofs

- **Polynomial evaluations**: For each query point, the prover must provide the value of the polynomial at that point for each layer of the FRI protocol.
- **Merkle proofs**: Along with each value, the prover must provide a Merkle proof. This proof demonstrates that the provided value was indeed part of the original commitment.
- **Efficiency**: The use of Merkle trees allows these proofs to be logarithmic in size relative to the polynomial degree.

Agenda Checkpoint IV

Integration & Application

Integration
ZK-STARK

Key points

Use in ZK-STARKs

FRI is a core component of ZK-STARKs (Zero-Knowledge Scalable Transparent Arguments of Knowledge).

- For verifying **computational integrity.**
- **Scalability** by enabling efficient proof generation and verification.
- Contributes to **the transparency** of ZK-STARKs, as it doesn't require a trusted setup (ZK-SNARKs does)

FRI offers several advantages compared to other proof systems:

- **Post-quantum security**: Unlike some other zero-knowledge proof systems, FRI is believed to be secure against quantum attacks.
- **Transparency**: FRI doesn't require a trusted setup, enhancing its security and reducing potential vulnerabilities.
- **Simplicity**: The underlying mathematics of FRI, while advanced, is relatively straightforward compared to some other proof systems.

Application
Trust digitalization

Key points

FRI and ZK-STARKs have several practical applications:

- **Blockchain scalability**: ZK-STARKs can be used to create layer-2 scaling solutions
- **Data privacy**: In financial applications, FRI can be used to prove the validity of transactions or computations without revealing sensitive information.
- **Verifiable computation**: FRI enables efficient verification of complex computations, which is useful in cloud computing and distributed systems.
- **Identity systems**: ZK-STARKs can be used to create privacy-preserving identity verification systems.
- **Supply chain management**: FRI can be used to prove the integrity of supply chain data without revealing sensitive business information.
- **Voting systems**: ZK-STARKs can enhance the privacy and verifiability of electronic voting systems.
- **Cryptocurrency mixers**: FRI can be used to create privacy-enhancing tools for cryptocurrencies (obfuscate transaction histories while proving the validity of transactions).

Agenda Checkpoint V

Conclusion

Conclusion

Notes

Key points

Soundness analysis

- **Probabilistic soundness**: FRI provides probabilistic soundness, meaning that a false proof will be accepted with only a very small probability.
- **Soundness error**: The protocol allows for tuning of the soundness error. By increasing the number of query points, the probability of accepting an invalid proof can be made arbitrarily small.
- **Security assumptions**: FRI's security relies on the hardness of finding collisions in the underlying hash function used for the Merkle tree commitments.
- **Post-quantum considerations**: Unlike some other proof systems, FRI is believed to be secure against quantum attacks, making it future-proof.

Proof and verification complexity

- **Proof size:** The proof size is O(log D), where D is the degree of the polynomial. This logarithmic growth in proof size is a significant advantage for large-scale applications. (The enlarged domain is a CONSTANT * D)
- **Prover complexity**: The prover's work is O(D log D), which is nearly linear in the degree of the polynomial. This efficiency allows for practical implementation even for very large polynomials.
- **Verifier complexity:** The verifier's work is O(log D), which is logarithmic in the degree of the polynomial. This extremely efficient verification process is one of FRI's most attractive features.

- **Trade-offs:**
  - FRI allows for various trade-offs between proof size, prover complexity, and soundness error.
  - These can be adjusted based on the specific requirements of the application.

Last Checkpoint

Thank You !