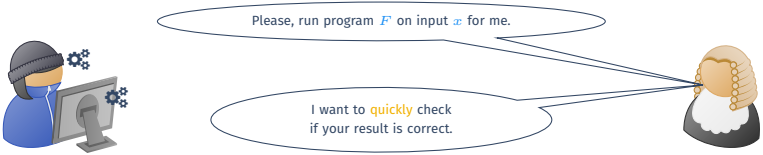
Efficient Proofs of Computational Integrity from Code-Based Interactive Oracle Proofs

Sarah Bordage

Project-team GRACE
LIX, Ecole Polytechnique, Institut Polytechnique de Paris
Inria Saclay Ile-de-France

GT Codes et Cryptographie
June 17, 2021

Verifiable Computing



Powerful Prover

On input (F, x), output result y
and proof of correctness π

y, π →

Weak Verifier

On input (F, x, y, π),
accept iff π is a valid proof
for statement "$y = F(x)$"

- **Completeness**: Verifier $\mathcal{V}$ always accepts honest proof
- **Soundness**: For any malicious $\tilde{P}$, $\Pr[\tilde{P}$ convinces $\mathcal{V}$ to accept incorrect statement$] =$ negl
- **Efficiency**: verification time, length of $\pi \ll$ time for computing $F(x)$

Remark: sublinear time($\mathcal{V}$) requires |description of the computation| $\ll$ | running time of the computation|, $\mathcal{V}$ should not "unroll" the computation.

A view of the “proofs-space” (by crypto assumptions)

	CRHF, ROM	DLOG	KoE/AGM/GGM (pairing-based)	Group of unknown order
2013			Pinocchio [PGHR]	
2014			[BCTV]	
2016	ZKBoo [GMO] SCI [BBC+]	[BCCGP]	[Groth16] [GM]	
2017	Ligero [AHIV]	Bulletproof [BBB+] Hyrax [WTS+]	(ZK) vSQL [ZGK+]	
2018	Stark [BBHR] Aurora [BCR+]		vRAM [ZGK+]	
2019	Fractal [COS] Succinct Aurora [BCG+] RedShift [KPV] Virgo [ZXZS]	Spartan [Setty] Halo [BGH]	Sonic [MBK+] Plonk [GWC] Marlin [CHM+] Libra [XZZ+]	Supersonic [BFS]
2020	Virgo++ [ZWZZ]		Mirage [KKPS]	

Some implementations of succinct non-interactive arguments for general computations

Outline

1. IOP-based succinct non-interactive arguments
2. STARK Arithmetization
3. Reed-Solomon Proximity Testing

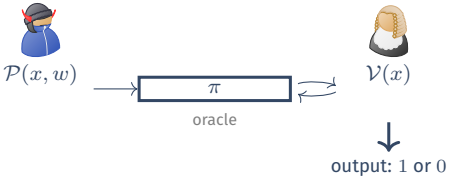
IOP-based succinct non-interactive arguments

Probabilistically Checkable Proofs

[Fortnow-Rompel-Sipser'88, Babai-Fortnow-Levin-Szegedy'91, Arora-Safra'92, Arora-Lund-Motwani-Sudan-Szegedy'98, ...]

Given a relation $\mathcal{R}$ of instance-witness pairs (x, w), denote $\mathcal{L}_{\mathcal{R}} = \{x \mid \exists w, (x, w) \in \mathcal{R}\}$.

Probabilistically checkable proof system (PCP)



Completeness: $\forall (x, w) \in \mathcal{R}, \exists \pi, \mathcal{V}^{\pi}(x) = 1$

Soundness: $\forall x \notin \mathcal{L}_{\mathcal{R}}, \forall \tilde{\pi}, \Pr[\mathcal{V}^{\pi}(x) = 0] > 1/2$

For verifiable computing, important measures are:

- time($\mathcal{P}$),
- time($\mathcal{V}$),
- query complexity q,
- proof length l.

PCP-based succinct interactive argument

$\mathbb{P}(x, w)$

computationally bounded

$\mathbb{V}(x)$

CRHF $H : \{0,1\}^* \to \{0,1\}^\kappa$ $\longleftarrow$

$\pi \leftarrow \mathcal{P}(x, w)$
root $=$ MTree$_H(\pi)$

$\xrightarrow{\text{root}}$

$\xleftarrow{Q}$

Sample unif. at random query set Q of size q

paths $= (\text{MProofs}_H(\pi[k]))_{k \in Q}$

$\xrightarrow{\pi_{|Q}, \text{ paths}}$

Check Merkle proofs

$\downarrow$

Output 1 iff $\mathcal{V}^\pi(x) = 1$

$q = |Q|, l = |\pi|$

From q-queries PCP of length l:

- Communication = $O_\kappa(q \log l)$,
- time$(\mathbb{P})$ = time$(\mathcal{P}) + O_\kappa(l)$,
- time$(\mathbb{V})$ = time$(\mathcal{V}) + O_\kappa(q \log l)$.

- [Kilian'92] PCP $(\mathcal{P}, \mathcal{V})$ + collision-resistant hashings
  $\rightsquigarrow$ interactive argument $(\mathbb{P}, \mathbb{V})$ for **NP** with sublinear communication.

PCP-based succinct non-interactive argument

computationally bounded

$\mathbb{P}(x, w)$

$\mathbb{V}(x)$

RO $H : \{0,1\}^* \to \{0,1\}^\kappa$ $\longleftarrow$

$q = |Q|, l = |\pi|$

$\pi \leftarrow \mathcal{P}(x, w)$
root $=$ MTree$_H(\pi)$

$Q \leftarrow H(x \| \text{root})$

paths
$=$ (MProofs$_H(\pi[k]))_{k \in Q}$

$\xrightarrow{\text{root, } \pi_{|Q}\text{, paths}}$

$Q \leftarrow H(x \| \text{root})$
Check Merkle proofs

$\downarrow$

Output 1 iff $\mathcal{V}^\pi(x) = 1$

From q-queries PCP of length l:

- Communication = $O_\kappa(q \log l)$,
- time$(\mathbb{P})$ = time$(\mathcal{P}) + O_\kappa(l)$,
- time$(\mathbb{V})$ = time$(\mathcal{V}) + O_\kappa(q \log l)$.

- [Kilian'92] PCP $(\mathcal{P}, \mathcal{V})$ + collision-resistant hashings
  $\rightsquigarrow$ interactive argument $(\mathbb{P}, \mathbb{V})$ for **NP** with sublinear communication.
- [Micali'00] Kilian's protocol + Fiat-Shamir paradigm (H random oracle; $Q \leftarrow H(x,$ root$)$)
  $\rightsquigarrow$ succinct non-interactive argument for **NP** in the Random Oracle Model.

Interactive Oracle Proofs

[Ben–Sasson-Chiesa-Spooner'16, Reingold-Rothblum-Rothblum'16]

IOP system

$(\mathcal{P}, \mathcal{V})$ is an IOP system for relation $\mathcal{R}$ with soundness error ε if

Completeness:

$$\forall (x, w) \in \mathcal{R}, \Pr[\langle P(x, w), V^{\pi_1, \ldots, \pi_r}(x) \rangle = 1] = 1.$$

Soundness:

$$\forall x \notin \mathcal{L}_{\mathcal{R}}, \forall \widetilde{P}, \Pr[\langle P(x, w), V^{\pi_1, \ldots, \pi_r}(x) \rangle = 1] \leq \varepsilon.$$

Proof length $l = \sum |\pi_i|$
Query complexity $q =$ number of queries to $\pi_1, \ldots, \pi_r$

$\mathcal{P}(x, w)$

$\mathcal{V}(x)$

m_0

π_1

m_1

π_2

$\vdots$

m_{r-1}

π_r

output: 1 or 0

IOP-based succinct non-interactive argument [Ben–Sasson-Chiesa-Spooner'16]

computationally bounded

$\mathbb{P}(x, w)$

$\mathbb{V}(x)$

CRHF $H : \{0,1\}^* \to \{0,1\}^\kappa$ $\longleftarrow$

For each round i

$\pi_i \leftarrow \mathcal{P}(x, w)$

$\text{root}_i = \text{MTree}_H(\pi_i)$

$\xrightarrow{\text{root}_i}$

Sample unif. at random query set Q_i

$\xleftarrow{Q_i}$

$\text{paths}_i = (\text{MProofs}_H(\pi_i[k]))_{k \in Q_i}$

$\xrightarrow{\pi_{i\,|\,Q_i},\ \text{paths}_i}$

Check Merkle proofs

$\downarrow$

Output 1 iff $\mathcal{V}^{\pi_1, \ldots, \pi_r}(x) = 1$

$q = \sum |Q_i|\,, l = \sum |\pi_i|$

From q-queries IOP with length l:

- Communication = $O_\kappa(q \log l)$,
- time($\mathbb{P}$) $= $ time($\mathcal{P}$) $+ O_\kappa(l)$,
- time($\mathbb{V}$) $= $ time($\mathcal{V}$) $+ O_\kappa(q \log l)$.

Interactive argument from Merkle Trees

$\underbrace{\text{IOP system } (\mathcal{P}, \mathcal{V})}_{\text{information theoretic}} + \underbrace{\text{hash functions}}_{\text{collision-resistant}} \implies \text{succinct interactive argument } (\mathbb{P}, \mathbb{V})$

IOP-based succinct non-interactive argument [Ben–Sasson-Chiesa-Spooner'16]

computationally bounded

$\mathbb{P}(x, w)$

$\mathbb{V}(x)$

CRHF $H : \{0,1\}^* \to \{0,1\}^\kappa$ $\longleftarrow$

For each round i

$\pi_i \leftarrow \mathcal{P}(x, w)$

$\text{root}_i = \text{MTree}_H(\pi_i)$

$\xrightarrow{\text{root}_i}$

Sample unif. at random query set Q_i

$\xleftarrow{Q_i}$

$\text{paths}_i = (\text{MProofs}_H(\pi_i[k]))_{k \in Q_i}$

$\xrightarrow{\pi_{i|Q_i},\ \text{paths}_i}$

Check Merkle proofs

$\downarrow$

Output 1 iff $\mathcal{V}^{\pi_1, \ldots, \pi_r}(x) = 1$

From q-queries IOP with length l:

- Communication = $O_\kappa(q \log l)$,
- time$(\mathbb{P})$ = time$(\mathcal{P}) + O_\kappa(l)$,
- time$(\mathbb{V})$ = time$(\mathcal{V}) + O_\kappa(q \log l)$.

Remove interaction: Q_i deduced from $RO(x||\text{root}_1||\ldots||\text{root}_{i-1}||Q_0||\ldots||Q_{i-1})$

Non-interactive argument from Merkle Trees + Fiat-Shamir

$\underbrace{\text{IOP system } (\mathcal{P}, \mathcal{V})}_{\text{information theoretic}}$ + $\underbrace{\text{hash functions}}_{\text{random oracle}}$ $\Longrightarrow$ succinct non-interactive argument $(\mathbb{P}, \mathbb{V})$

Computational integrity language $\mathcal{L}$:
Instances (F, x, y, T) such that running program F within T cycles on public input x and auxiliary (secret) witness w leads to output y.

STARK (**S**calable **T**ransparent **AR**gument of **K**nowledge) [Ben–Sasson-Bentov-Horesh-Riabzev'18]

$\rightsquigarrow$ non-interactive argument of knowledge for "machine computations", with or without zero-knowledge

Prover	Verifier	Communication complexity	Setup	Post-Quantum
$O_\kappa(T \log^2 T)$ H	$O_\kappa(\log^2 T)$ H	$O_\kappa(\log^2 T)$	transparent	yes

Construct an IOP system $(\mathcal{P}, \mathcal{V})$ for $\mathcal{L}$ with:

$$\text{time}(\mathcal{P}) = O(T \log^2 T) \qquad \text{time}(\mathcal{V}) = \widetilde{O}(n) + O(\log(T)), \qquad l = O(T \log T), \qquad q = O(\log T),$$

then apply [BCS'16] transformation.

Applications:
Allows verification of multiple programs in a single proof (StarkEx, Cairo).
One can build PQ signatures from ZK-STARKs (see Ziggy STARK).

STARK Arithmetization: From computational integrity to low-degree testing

STARK: Toy example

Toy program

- **Public input**: $x \in \mathbb{F}$, steps T ($T \ll |\mathbb{F}|$)
- **Private output**: $K \in \mathbb{F}^T$, $K = (K_0, K_1, \ldots, K_{T-1})$

Program $F(x, T, K)$:

For $i = 0$ to $T - 1$:

$x \leftarrow x^3 + K_i$

return x

Execution trace

x_0	K_0
x_1	K_1
x_2	K_2
...	...
x_{T-2}	K_{T-2}
x_{T-1}	K_{T-1}
x_T	0

Valid iff $\begin{cases} x_0 = x \\ x_T = y \\ x_{i+1} = x_i^3 + K_i \end{cases}$

Computational integrity statement for ZK-STARK:

"Given program F, public input (x, T) and public output y, I know secret K such that $F(x, T, K) = y$."

STARK: Toy example

Toy program

- **Public input**: $x \in \mathbb{F}$, steps T ($T \ll |\mathbb{F}|$)
- **Private output**: $K \in \mathbb{F}^T$, $K = (K_0, K_1, \ldots, K_{T-1})$

Program $F(x, T, K)$:

For $i = 0$ to $T - 1$:

$x \leftarrow x^3 + K_i$

return x

Execution trace

x_0	K_0
x_1	K_1
x_2	K_2
...	...
x_{T-2}	K_{T-2}
x_{T-1}	K_{T-1}
x_T	0

Valid iff $\begin{cases} x_0 = x \\ x_T = y \\ x_{i+1} = x_i^3 + K_i \end{cases}$

Computational integrity statement for ZK-STARK:

"Given program F, public input (x, T) and public output y, I know secret K such that $F(x, T, K) = y$."

... Today, we'll discuss only the computational integrity part.

$\rightsquigarrow$ "Given program F, public input (x, T) and public output y, there exists K such that $F(x, T, K) = y$."

Toy example: Arithmetization I

$\mathcal{P}$ and $\mathcal{V}$ agree on $\omega \in \mathbb{F}^\times$ of order $T+1$, which defines $G := \langle \omega \rangle$. They also define:

- "Contraint polynomial" $C(X_0, X_1, Y_0) = Y_0 - (X_0^3 + X_1)$
- "Boundaries" polynomial $B \in \mathbb{F}[X]_{\leq 1}$ such that $B(1) = x$ and $B(\omega^T) = y$
- Vanishing polynomial $Z(X) = \prod_{i=0}^{T-1}(X - \omega^i) = \frac{X^{T+1}-1}{X-\omega^T}$

$\mathcal{P}$ interpolates $P_0, P_1 \in \mathbb{F}[X]_{\leq T}$ such that for all $i \in \{0, ..., T\}$, $P_0(\omega^i) = x_i$ and $P_1(\omega^i) = K_i$.

Valid execution trace $\iff \begin{cases} x_0 = x, \\ x_T = y, \\ C(x_i, K_i, x_{i+1}) = 0, \quad \text{for } 0 \leq i < T \end{cases}$

Toy example: Arithmetization I

$\mathcal{P}$ and $\mathcal{V}$ agree on $\omega \in \mathbb{F}^\times$ of order $T+1$, which defines $G := \langle \omega \rangle$. They also define:

- "Contraint polynomial" $C(X_0, X_1, Y_0) = Y_0 - (X_0^3 + X_1)$
- "Boundaries" polynomial $B \in \mathbb{F}[X]_{\leq 1}$ such that $B(1) = x$ and $B(\omega^T) = y$
- Vanishing polynomial $Z(X) = \prod_{i=0}^{T-1}(X - \omega^i) = \frac{X^{T+1}-1}{X-\omega^T}$

$\mathcal{P}$ interpolates $P_0, P_1 \in \mathbb{F}[X]_{\leq T}$ such that for all $i \in \{0, ..., T\}$, $P_0(\omega^i) = x_i$ and $P_1(\omega^i) = K_i$.

Valid execution trace

$$\iff \begin{cases} x_0 = x, \\ x_T = y, \\ C(x_i, K_i, x_{i+1}) = 0, \quad \text{for } 0 \leq i < T \end{cases}$$

$$\iff \begin{cases} P_0(\omega^0) = B(\omega^0), \\ P_0(\omega^T) = B(\omega^T), \\ C(P_0(\omega^i), P_1(\omega^i), P_0(\omega^{i+1})) = 0, \quad \text{for } 0 \leq i < T \end{cases}$$

Toy example: Arithmetization I

$\mathcal{P}$ and $\mathcal{V}$ agree on $\omega \in \mathbb{F}^\times$ of order $T+1$, which defines $G := \langle \omega \rangle$. They also define:

- "Contraint polynomial" $C(X_0, X_1, Y_0) = Y_0 - (X_0^3 + X_1)$
- "Boundaries" polynomial $B \in \mathbb{F}[X]_{\leq 1}$ such that $B(1) = x$ and $B(\omega^T) = y$
- Vanishing polynomial $Z(X) = \prod_{i=0}^{T-1}(X - \omega^i) = \frac{X^{T+1}-1}{X-\omega^T}$

$\mathcal{P}$ interpolates $P_0, P_1 \in \mathbb{F}[X]_{\leq T}$ such that for all $i \in \{0, ..., T\}$, $P_0(\omega^i) = x_i$ and $P_1(\omega^i) = K_i$.

Valid execution trace

$$\iff \begin{cases} x_0 = x, \\ x_T = y, \\ C(x_i, K_i, x_{i+1}) = 0, \quad \text{for } 0 \leq i < T \end{cases}$$

$$\iff \begin{cases} P_0(\omega^0) = B(\omega^0), \\ P_0(\omega^T) = B(\omega^T), \\ C(P_0(\omega^i), P_1(\omega^i), P_0(\omega^{i+1})) = 0, \quad \text{for } 0 \leq i < T \end{cases}$$

$$\iff \begin{cases} (X-1)(X-\omega^T) \text{ divides } P_0(X) - B(X) \\ Z(X) \text{ divides } C(P_0(X), P_1(X), P_0(hX)) \end{cases}$$

Toy example: Arithmetization II

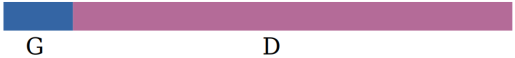
Valid execution trace $\iff \begin{cases} Q_0(X) = \frac{P_0(X) - B(X)}{(X-1)(X-\omega^T)} \in \mathbb{F}_q[X]_{\leq T-2} \\ Q_1(X) = \frac{C(P_0(X), P_1(X), P_0(hX))}{Z(X)} \in \mathbb{F}_q[X]_{\leq 2T} \end{cases}$

Toy example: Arithmetization II

Valid execution trace $\iff \begin{cases} Q_0(X) = \frac{P_0(X) - B(X)}{(X-1)(X-\omega^T)} \in \mathbb{F}_q[X]_{\leq T-2} \\ Q_1(X) = \frac{C(P_0(X), P_1(X), P_0(hX))}{Z(X)} \in \mathbb{F}_q[X]_{\leq 2T} \end{cases}$

Reed-Solomon encoding

RS$[\mathbb{F}, D, k] \subset \mathbb{F}^D$: evaluations on $D \subset \mathbb{F}$ of polynomials of degree $< k$.



We assume $|D| \geq 2(T+1)$, $|D| = \Theta(T)$, and $D \cap G = \emptyset$.

$\mathcal{P}$ sends oracle functions $f_0, f_1, g_0, g_1 \in \mathbb{F}^D$,
supposedly evaluations on D of $P_0(X), P_1(X), Q_0(X), Q_1(X)$, respectively.

Valid execution trace $\iff \begin{cases} f_0, f_1 \in \text{RS}[\mathbb{F}, D, T+1] \\ g_0 \in \text{RS}[\mathbb{F}, D, T-1] \\ g_1 \in \text{RS}[\mathbb{F}, D, 2T+1] \end{cases}$

Toy example: Arithmetization III

Assume $f_0, f_1 \in \mathsf{RS}[\mathbb{F}, D, T+1]$. Then,

- $(x_0 \neq x$ or $x_T \neq y) \implies \Delta(g_0, \mathsf{RS}[\mathbb{F}, D, T-1]) \geq 1 - \frac{T+1}{|D|}$
- $\exists i, x_{i+1} \neq x_i^3 + K_i \implies \Delta(g_1, \mathsf{RS}[\mathbb{F}, D, 2T+1]) \geq 1 - \frac{3T+1}{|D|}$

Δ relative Hamming distance

Arithmetization: $\begin{cases} \textbf{valid execution trace} \rightsquigarrow \textbf{RS codewords} \\ \textbf{incorrect execution trace} \rightsquigarrow \textbf{words far from any RS codewords} \end{cases}$

Toy example: Arithmetization III

Assume $f_0, f_1 \in \mathsf{RS}[\mathbb{F}, D, T+1]$. Then,

- $(x_0 \neq x$ or $x_T \neq y) \implies \Delta(g_0, \mathsf{RS}[\mathbb{F}, D, T-1]) \geq 1 - \frac{T+1}{|D|}$ $\quad$ Δ relative Hamming distance
- $\exists i, x_{i+1} \neq x_i^3 + K_i \implies \Delta(g_1, \mathsf{RS}[\mathbb{F}, D, 2T+1]) \geq 1 - \frac{3T+1}{|D|}$

Arithmetization: $\begin{cases} \textbf{valid execution trace} \rightsquigarrow \textbf{RS codewords} \\ \textbf{incorrect execution trace} \rightsquigarrow \textbf{words far from any RS codewords} \end{cases}$

Consistency at a random location

$\mathcal{V}$ samples $s \in D$ and requests $f_0(s), f_1(s), f_0(\omega \cdot s), g_0(s), g_1(s)$,

computes $B(s)$ and $Z(s)$ (in $O(\log T)$ $\mathbb{F}$-ops), then checks:

1. $g_0(s) \cdot (s-1) \cdot (s - \omega^T) \stackrel{?}{=} f_0(s) - B(s)$ $\quad$ soundness error $\leq \frac{T}{|D|}$
2. $g_1(s) \cdot Z(s) \stackrel{?}{=} C(f_0(s), f_1(s), f_0(\omega \cdot s))$ $\quad$ soundness error $\leq \frac{3T}{|D|}$

$\mathcal{V}$ rejects if one of the test fails.

Low-degree compliance

Instead of RS membership test: $\mathcal{V}$ must be able to test **proximity to RS code with $O(\log T)$ queries**.

Reed-Solomon Proximity Testing

Reed-Solomon Proximity Testing

Reed-Solomon proximity testing

Input:	a code RS[$\mathbb{F}, D, k$], a parameter δ
Input oracle:	$f : D \to \mathbb{F}$
Completeness:	If $f \in$ RS[$\mathbb{F}, D, k$], then the test always accepts.
Soundness:	If $\Delta\left(f, \text{RS}[\mathbb{F}, D, k]\right) > \delta$, then the test accepts with probability $\leq \text{err}(\delta)$.
	Δ relative Hamming distance

Reed-Solomon Proximity Testing

Reed-Solomon proximity testing

Input:	a code RS$[\mathbb{F}, D, k]$, a parameter δ
Input oracle:	$f : D \to \mathbb{F}$
Completeness:	If $f \in$ RS$[\mathbb{F}, D, k]$, then the test always accepts.
Soundness:	If $\Delta\left(f, \text{RS}[\mathbb{F}, D, k]\right) > \delta$, then the test accepts with probability $\leq \text{err}(\delta)$.
	Δ relative Hamming distance

Naive proximity test

1. Query $k+1$ entries of $f \in \mathbb{F}^D : f(x_0), \ldots, f(x_k)$,
2. Compute poly $P(X)$ of degree $< k$ by interpolating $\{(x_i, f(x_i)); 0 \leq i \leq k-1\}$
3. Tester accepts iff $P(x_k) = f(x_k)$.

   Soundness: $\mathcal{V}$ accepts with proba $\Pr_{x_k}[P(x_k) \neq f(x_k)] \leq 1 - \Delta(f, \text{RS}[\mathbb{F}, D, k])$.

   Problem: query complexity $k+1$ is **linear** in $|D|$ with $|D| = \Theta(k)$.

How to achieve **logarithmic** verification time? $\mathcal{V}$ will need some auxiliary proof from $\mathcal{P}$...
We also want a fast prover (linear in $|D|$) $\rightsquigarrow$ **IOP of Proximity (IOPP) for RS code.**

Reed-Solomon IOP of Proximity

RS IOP of Proximity

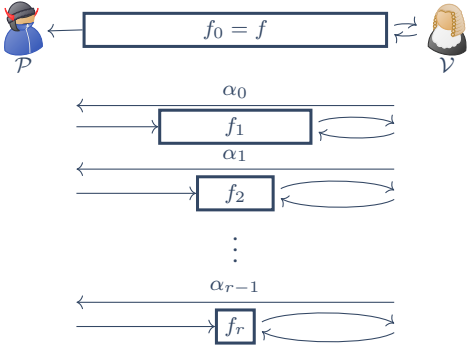
Input:	a code RS[$\mathbb{F}, D, k$], a parameter δ
Input oracle:	$f : D \to \mathbb{F}$
Completeness:	If $f \in$ RS[$\mathbb{F}, D, k$], then $\exists \mathcal{P}\ \Pr[\langle \mathcal{P}, \mathcal{V} \rangle = 1] = 1$
Soundness:	If $\Delta\left(f, \text{RS}[\mathbb{F}, D, k]\right) > \delta$, then $\forall \tilde{\mathcal{P}}\ \Pr[\langle \tilde{\mathcal{P}}, \mathcal{V} \rangle = 1] \leq \text{err}(\delta)$
	Δ relative Hamming distance

Fast Reed-Solomon IOPP:

FRI Protocol

[Ben–Sasson-Bentov-Horesh-Riabzev'18]

# rounds	$< \log \lvert D \rvert$
# queries	$< 2 \log \lvert D \rvert)$
prover time	$< 6 \lvert D \rvert$
verifier time	$< 21 \log \lvert D \rvert)$
proof length	$< \lvert D \rvert / 3$
soundness	" $\simeq 1 - \delta$"



Halving the size of the problem by folding

Assume there exists $\omega \in \mathbb{F}^\times$ of order a large power of 2.
Given $k = 2^r$ and evaluation domain $D = \langle \omega \rangle$ of size $n = |D|$ and closed under negation

How to check proximity of $f_0 : D \to \mathbb{F}$ to the code RS$[\mathbb{F}, D, k]$?

Define sequence of RS codes: RS$_i$:= RS$[\mathbb{F}, D_i, k/2^i]$ with $D_0 = D$ and $D_i := \langle \omega^{2^i} \rangle$
(same rate $\rho = k/n$)

Halving the size of the problem by folding

Assume there exists $\omega \in \mathbb{F}^\times$ of order a large power of 2.
Given $k = 2^r$ and evaluation domain $D = \langle \omega \rangle$ of size $n = |D|$ and closed under negation

How to check proximity of $f_0 : D \to \mathbb{F}$ to the code RS$[\mathbb{F}, D, k]$?

Define sequence of RS codes: $\mathsf{RS}_i := \mathsf{RS}[\mathbb{F}, D_i, k/2^i]$ with $D_0 = D$ and $D_i := \langle \omega^{2^i} \rangle$
(same rate $\rho = k/n$)

For each round i, reduce proximity to RS_i to RS_{i+1}:

- Split $f_i(X)$ into u_0, u_1, such that $f_i(X) = u_0(X^2) + X u_1(X^2)$
- For $\alpha \in \mathbb{F}$, define $\textsc{Fold}\,[f_i, \alpha] : D_{i+1} \to \mathbb{F}$ by $\textsc{Fold}\,[f_i, \alpha]\,(y) = u_0(y) + \alpha \cdot u_1(y)$

Completeness

For all $\alpha \in \mathbb{F}$, $\textsc{Fold}\,[\mathsf{RS}_i, \alpha] \subseteq \mathsf{RS}_{i+1}$

Halving the size of the problem by folding

Assume there exists $\omega \in \mathbb{F}^\times$ of order a large power of 2.
Given $k = 2^r$ and evaluation domain $D = \langle \omega \rangle$ of size $n = |D|$ and closed under negation

How to check proximity of $f_0 : D \to \mathbb{F}$ to the code $\mathsf{RS}[\mathbb{F}, D, k]$?

Define sequence of RS codes: $\mathsf{RS}_i := \mathsf{RS}[\mathbb{F}, D_i, k/2^i]$ with $D_0 = D$ and $D_i := \langle \omega^{2^i} \rangle$
(same rate $\rho = k/n$)

For each round i, reduce proximity to RS_i to RS_{i+1}:

- Split $f_i(X)$ into u_0, u_1, such that $f_i(X) = u_0(X^2) + X u_1(X^2)$
- For $\alpha \in \mathbb{F}$, define $\textsc{Fold}[f_i, \alpha] : D_{i+1} \to \mathbb{F}$ by $\textsc{Fold}[f_i, \alpha](y) = u_0(y) + \alpha \cdot u_1(y)$

Completeness

For all $\alpha \in \mathbb{F}$, $\textsc{Fold}[\mathsf{RS}_i, \alpha] \subseteq \mathsf{RS}_{i+1}$

Observe: for all $x \in D_i$,

$$\textsc{Fold}[f_i, \alpha](x^2) = \frac{f_i(x) + f_i(-x)}{2} + \alpha \frac{f_i(x) - f_i(-x)}{2x}.$$

Local computability

For any $y \in D_{i+1}$, compute $\textsc{Fold}[f_i, \alpha](y)$ with only **2 queries** to f_i.

Folding preserves distance to the code

Let $\varepsilon = \varepsilon(|\mathbb{F}|, n, \rho) = O_\rho\left(\frac{n^2}{|\mathbb{F}|}\right)$.

Distance preservation

Let $\delta < 1 - \sqrt{\rho}$. If $\Delta(f_i, \mathsf{RS}_i) > \delta$, then

$$\Pr_{\alpha \in \mathbb{F}}[\Delta(\textsc{Fold}[f_i, \alpha], \mathsf{RS}_{i+1}) < \delta] < \varepsilon$$

Proof relies on:

Theorem [Ben–Sasson-Carmon-Ishai-Kopparty-Saraf'20]

Let $C \subseteq \mathbb{F}^D$ be a RS code of rate ρ and let $u_0, u_1 : D \to \mathbb{F}$. For any $\delta < 1 - \sqrt{\rho}$, if

$$\Pr_{\alpha \in \mathbb{F}}[\Delta(u_0 + \alpha u_1, C) < \delta] > 1 - \varepsilon,$$

then, there exist $c_0, c_1 \in C$ and $T \subset D$ such that

- $|T| \geq (1 - \delta)|D|$
- $u_{0|T} = c_{0|T}$ and $u_{1|T} = c_{1|T}$

Folding preserves distance to the code - proof sketch

Distance preservation

Let $\delta < 1 - \sqrt{\rho}$. If $\Delta(f_i, \mathsf{RS}_i) > \delta$, then

$$\Pr_{\alpha \in \mathbb{F}}[\Delta(\textsc{Fold}[f_i, \alpha], \mathsf{RS}_{i+1}) < \delta] < \varepsilon$$

Proof sketch:

Idea: assume $\textsc{Fold}[f_i, \alpha]$ is δ-close of RS_{i+1} for many α's,

$\rightsquigarrow$ reconstruct a codeword $v \in \mathsf{RS}_i$ which is δ-close to f_i.

Folding preserves distance to the code - proof sketch

Distance preservation

Let $\delta < 1 - \sqrt{\rho}$. If $\Delta(f_i, \mathsf{RS}_i) > \delta$, then

$$\Pr_{\alpha \in \mathbb{F}}\left[\Delta(\textsc{Fold}\left[f_i, \alpha\right], \mathsf{RS}_{i+1}) < \delta\right] < \varepsilon$$

Proof sketch:

Idea: assume $\textsc{Fold}\left[f_i, \alpha\right]$ is δ-close of RS_{i+1} for many α's,
$\rightsquigarrow$ reconstruct a codeword $v \in \mathsf{RS}_i$ which is δ-close to f_i.

- Apply BCIKS'20 on $\textsc{Fold}\left[f_i, \alpha\right] = u_0 + \alpha u_1 \implies \exists v_0, v_1 \in \mathsf{RS}_{i+1}$ such that

$$T = \{y \in D_{i+1} \mid u_0(y) = v_0(y) \text{ AND } u_1(y) = v_1(y)\}$$

  is of size $|T| \geq (1-\delta)\,|D_{i+1}|$,
- Consider polynomial $r(X) = v_0(X^2) + X v_1(X^2)$ of degree $< k/2^i$,
- Then, the evaluation of $r(X)$ on D_i agrees with f on the set $\left\{x \in D_i \mid x^2 \in T\right\}$ of size $\geq (1-\delta)\,|D_i|$.

FRI Protocol: COMMIT phase

COMMIT Phase

Honest prover computes:

$\mathcal{P}$

$\mathcal{V}$

$\leftarrow \boxed{f_0 = f}$

$f_1 = \textsc{Fold}\,[f_0, \alpha_0]$

$\xleftarrow{\alpha_0}$

$\xrightarrow{} \boxed{f_1}$

$f_2 = \textsc{Fold}\,[f_1, \alpha_1]$

$\xleftarrow{\alpha_1}$

$\xrightarrow{} \boxed{f_2}$

$\vdots$

$\vdots$

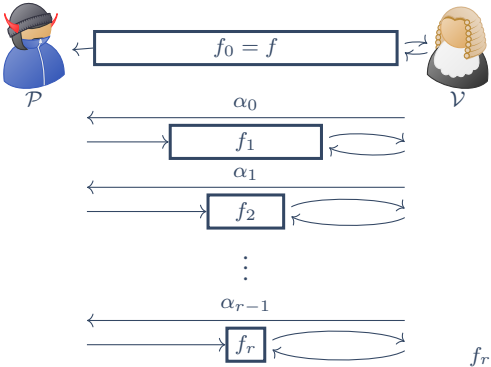
$f_r = \textsc{Fold}\,[f_{r-1}, \alpha_{r-1}]$

$\xleftarrow{\alpha_{r-1}}$

$\xrightarrow{} \boxed{f_r}$

FRI Protocol: QUERY phase

QUERY Phase



Global consistency test:

Sample $s \in D_0$ and check

$$f_1(s^2) \stackrel{?}{=} \textsc{Fold}\,[f_0, \alpha_0]\,(s^2)$$

$$f_2(s^4) \stackrel{?}{=} \textsc{Fold}\,[f_1, \alpha_1]\,(s^4)$$

$$\vdots$$

$$f_r(s^{2^r}) \stackrel{?}{=} \textsc{Fold}\,[f_{r-1}, \alpha_{r-1}]\,(s^{2^r})$$

Final test: $f_r \stackrel{?}{\in} \mathsf{RS}_r$

$\downarrow$

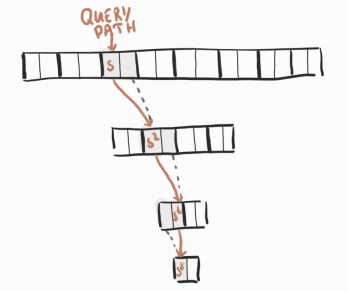
output: 1 or 0

FRI Protocol: Soundness

What can go wrong?

- **COMMIT**:
  Event $\textsc{Bad}_i$ = "at round i, $\textsc{Fold}\,[\cdot, \alpha_i]$ does not preserve distance to the code."
- **QUERY**: chosen queried path does not catch any errors.

$$\Pr\left[\mathcal{V} \text{ accepts }\right] \leq \underbrace{\Pr_{\alpha_i \in \mathbb{F}}\left[\cup_{i=0}^{r-1} \textsc{Bad}_i\right]}_{\mathsf{err}_{\mathsf{commit}}} + \underbrace{\Pr_{s \in D}\left[\mathcal{V} \text{ accepts } \mid \cap_{i=0}^{r-1} \overline{\textsc{Bad}_i}\right]}_{\mathsf{err}_{\mathsf{query}}}$$

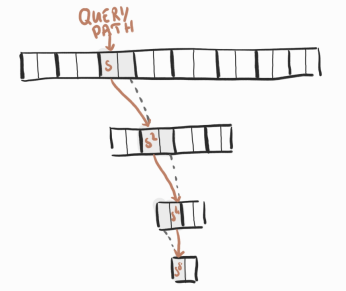


FRI Protocol: Soundness

What can go wrong?

- **COMMIT**:
  Event $\textsc{Bad}_i$ = "at round i, $\textsc{Fold}\,[\cdot, \alpha_i]$ does not preserve distance to the code."
- **QUERY**: chosen queried path does not catch any errors.

$$\Pr\left[\mathcal{V} \text{ accepts }\right] \leq \underbrace{\Pr_{\alpha_i \in \mathbb{F}}\left[\cup_{i=0}^{r-1} \textsc{Bad}_i\right]}_{\mathsf{err}_{\mathsf{commit}}} + \underbrace{\Pr_{s \in D}\left[\mathcal{V} \text{ accepts } \mid \cap_{i=0}^{r-1} \overline{\textsc{Bad}_i}\right]}_{\mathsf{err}_{\mathsf{query}}}$$



Theorem: Soundness [Ben–Sasson-Kopparty-Saraf'18, Ben–Sasson-Carmon-Ishai-Kopparty-Saraf'20]

For any $\delta < 1 - \sqrt{\rho}$, if $\Delta(f, \mathsf{RS}[\mathbb{F}, D, k]) > \delta$, then $\mathcal{V}$ accepts with proba at most

$$\begin{aligned} \mathsf{err}(\delta) &< \mathsf{err}_{\mathsf{commit}} + (\mathsf{err}_{\mathsf{query}})^l \\ &< r \cdot \varepsilon(|\mathbb{F}|, n, \rho) + (1-\delta)^l \end{aligned}$$

after l repetitions of the QUERY phase.

Soundness analysis I

Let $\delta_i = \min\left(\Delta(f_i, \mathsf{RS}_i), 1 - \sqrt{\rho})\right)$.

Bounding $\mathsf{err}_{\mathsf{commit}}$:

$\mathsf{BAD}_i =$ "$\Delta(\textsc{Fold}\left[f_i, \alpha_i\right], \mathsf{RS}_{i+1}) < \delta_i$".

By distance preservation: $\Pr[\mathsf{BAD}_i] \leq \varepsilon$,

thus $\underbrace{\Pr\left[\cup_{i=0}^{r-1} \mathsf{BAD}_i\right]}_{\mathsf{err}_{\mathsf{commit}}} \leq r \cdot \varepsilon$

Soundness analysis I

Let $\delta_i = \min\left(\Delta(f_i, \mathsf{RS}_i), 1 - \sqrt{\rho})\right)$.

Bounding $\mathsf{err}_{\mathsf{commit}}$:

$\mathsf{BAD}_i =$ "$\Delta(\textsc{Fold}\left[f_i, \alpha_i\right], \mathsf{RS}_{i+1}) < \delta_i$".

By distance preservation: $\Pr[\mathsf{BAD}_i] \leq \varepsilon$,

thus $\underbrace{\Pr\left[\cup_{i=0}^{r-1} \mathsf{BAD}_i\right]}_{\mathsf{err}_{\mathsf{commit}}} \leq r \cdot \varepsilon$

Bounding $\mathsf{err}_{\mathsf{query}}$:

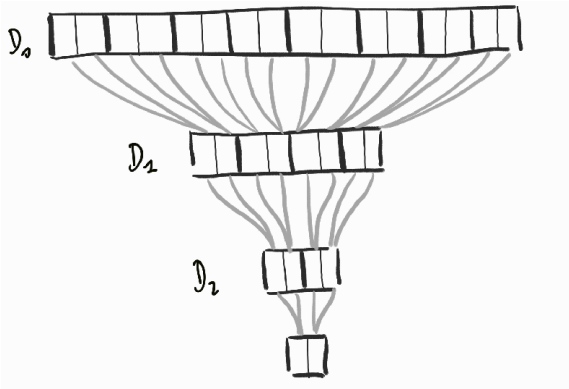
Suppose no BAD_i occurs.

Consider graph with vertices $D_0 \cup D_1 \cup \cdots \cup D_r$,

and we'll say that:

- $y \in D_{i+1}$ is **green** if $f_{i+1}(y) = \textsc{Fold}\left[f_i, \alpha_i\right](y)$
- $y \in D_{i+1}$ is **red** if $f_{i+1}(y) \neq \textsc{Fold}\left[f_i, \alpha_i\right](y)$

$\rightsquigarrow \mathcal{V}$ **accepts iff every vertices of the queried path are green**.



Soundness analysis I

Let $\delta_i = \min\left(\Delta(f_i, \mathsf{RS}_i), 1 - \sqrt{\rho})\right)$.

Bounding $\mathsf{err}_{\mathsf{commit}}$:

$\mathsf{BAD}_i =$ "$\Delta(\textsc{Fold}\,[f_i, \alpha_i]\,, \mathsf{RS}_{i+1}) < \delta_i$".

By distance preservation: $\Pr[\mathsf{BAD}_i] \leq \varepsilon$,

thus $\underbrace{\Pr\left[\cup_{i=0}^{r-1} \mathsf{BAD}_i\right]}_{\mathsf{err}_{\mathsf{commit}}} \leq r \cdot \varepsilon$

Bounding $\mathsf{err}_{\mathsf{query}}$:

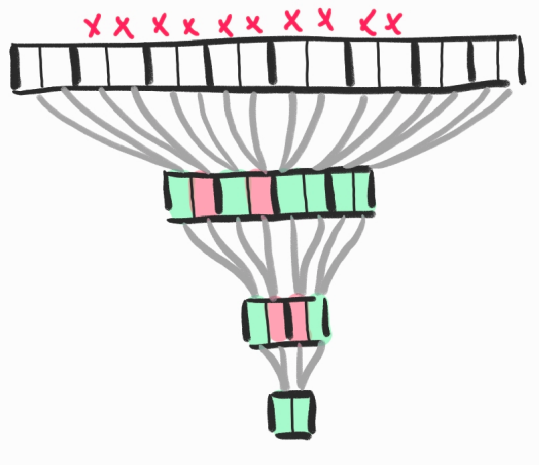
Suppose no BAD_i occurs.

Consider graph with vertices $D_0 \cup D_1 \cup \cdots \cup D_r$,

and we'll say that:

- $y \in D_{i+1}$ is **green** if $f_{i+1}(y) = \textsc{Fold}\,[f_i, \alpha_i]\,(y)$
- $y \in D_{i+1}$ is **red** if $f_{i+1}(y) \neq \textsc{Fold}\,[f_i, \alpha_i]\,(y)$

$\rightsquigarrow \mathcal{V}$ **accepts iff every vertices of the queried path are green.**



$\Pr[\mathcal{V} \text{ accepts}] = \frac{6}{16} = \frac{3}{8}$

Soundness analysis I

Let $\delta_i = \min\left(\Delta(f_i, \mathsf{RS}_i), 1 - \sqrt{\rho}\right)$.

Bounding $\mathsf{err}_{\mathsf{commit}}$:

$\mathsf{BAD}_i =$ "$\Delta(\textsc{Fold}\,[f_i, \alpha_i]\,, \mathsf{RS}_{i+1}) < \delta_i$".

By distance preservation: $\Pr[\mathsf{BAD}_i] \leq \varepsilon$,

thus $\underbrace{\Pr\left[\cup_{i=0}^{r-1} \mathsf{BAD}_i\right]}_{\mathsf{err}_{\mathsf{commit}}} \leq r \cdot \varepsilon$

Bounding $\mathsf{err}_{\mathsf{query}}$:

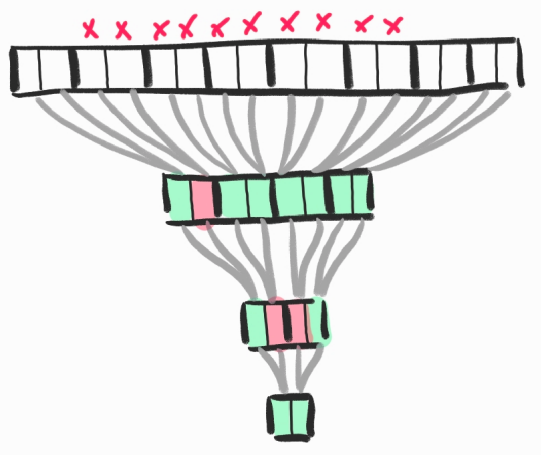
Suppose no BAD_i occurs.

Consider graph with vertices $D_0 \cup D_1 \cup \cdots \cup D_r$,

and we'll say that:

- $y \in D_{i+1}$ is **green** if $f_{i+1}(y) = \textsc{Fold}\,[f_i, \alpha_i]\,(y)$
- $y \in D_{i+1}$ is **red** if $f_{i+1}(y) \neq \textsc{Fold}\,[f_i, \alpha_i]\,(y)$

$\rightsquigarrow$ $\mathcal{V}$ **accepts iff every vertices of the queried path are green.**



$\Pr[\mathcal{V}\text{ accepts}] = \frac{6}{16} = \frac{3}{8}$

Soundness analysis I

Let $\delta_i = \min\left(\Delta(f_i, \mathsf{RS}_i), 1 - \sqrt{\rho}\right)$.

Bounding $\mathsf{err}_{\mathsf{commit}}$:

$\mathsf{BAD}_i =$ "$\Delta(\textsc{Fold}\,[f_i, \alpha_i]\,, \mathsf{RS}_{i+1}) < \delta_i$".

By distance preservation: $\Pr[\mathsf{BAD}_i] \leq \varepsilon$,

thus $\underbrace{\Pr\left[\cup_{i=0}^{r-1} \mathsf{BAD}_i\right]}_{\mathsf{err}_{\mathsf{commit}}} \leq r \cdot \varepsilon$

Bounding $\mathsf{err}_{\mathsf{query}}$:

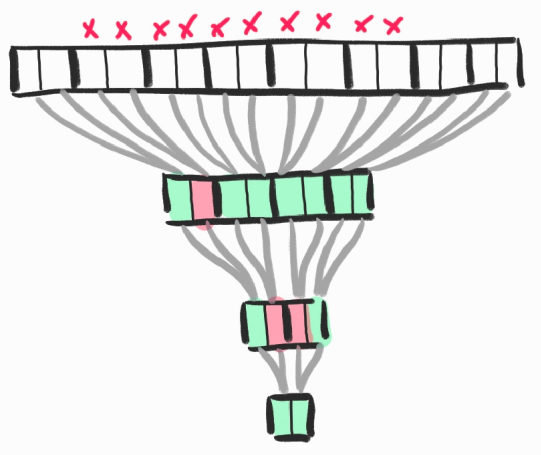
Suppose no BAD_i occurs.

Consider graph with vertices $D_0 \cup D_1 \cup \cdots \cup D_r$,

and we'll say that:

- $y \in D_{i+1}$ is **green** if $f_{i+1}(y) = \textsc{Fold}\,[f_i, \alpha_i]\,(y)$
- $y \in D_{i+1}$ is **red** if $f_{i+1}(y) \neq \textsc{Fold}\,[f_i, \alpha_i]\,(y)$

$\rightsquigarrow$ $\mathcal{V}$ **accepts iff every vertices of the queried path are green**.



$\Pr[\mathcal{V} \text{ accepts}] = \frac{6}{16} = \frac{3}{8}$

Modify the entries of $f_1, \ldots, f_{r-1}$ to keep only the **last red vertex** along the path.

Modification process does not affect **green** vertices $\rightsquigarrow$ rejection proba does not increase.

Soundness analysis II

Considering the modified oracles, define $E_{i+1} = \{y \in D_{i+1} \mid y \text{ is red, i.e. } f_{i+1}(y) \neq \textsc{Fold}\left[f_i, \alpha_i\right](y)\}$.
If no BAD_i **occurs,** $\mathcal{V}$ **rejects with proba at most**

$$\Pr_{s \in D_0}\left[\exists i \in \{1, \ldots, r\}, s^{2^i} \in E_i\right] = \sum_{i=1}^{r} \Pr_{s \in D_0}\left[s^{2^i} \in E_i\right] = \sum_{i=1}^{r} \frac{|E_i|}{|D_i|}.$$

Soundness analysis II

Considering the modified oracles, define $E_{i+1} = \{y \in D_{i+1} \mid y \text{ is red, i.e. } f_{i+1}(y) \neq \textsc{Fold}\left[f_i, \alpha_i\right](y)\}$.
If no BAD_i **occurs,** $\mathcal{V}$ **rejects with proba at most**

$$\Pr_{s \in D_0}\left[\exists i \in \{1, \ldots, r\}, s^{2^i} \in E_i\right] = \sum_{i=1}^{r} \Pr_{s \in D_0}\left[s^{2^i} \in E_i\right] = \sum_{i=1}^{r} \frac{|E_i|}{|D_i|}.$$

Triangular inequality:

$$\underbrace{\Delta(\textsc{Fold}\left[f_i, \alpha_i\right], f_{i+1})}_{=\frac{|E_{i+1}|}{|D_{i+1}|}} + \Delta(f_{i+1}, \mathsf{RS}_{i+1}) \geq \underbrace{\Delta(\textsc{Fold}\left[f_i, \alpha_i\right], \mathsf{RS}_{i+1})}_{\geq \delta_i}.$$

Soundness analysis II

Considering the modified oracles, define $E_{i+1} = \{y \in D_{i+1} \mid y \text{ is red, i.e. } f_{i+1}(y) \neq \textsc{Fold}\left[f_i, \alpha_i\right](y)\}$.
If no BAD_i **occurs,** $\mathcal{V}$ **rejects with proba at most**

$$\Pr_{s \in D_0}\left[\exists i \in \{1, \ldots, r\}, s^{2^i} \in E_i\right] = \sum_{i=1}^{r} \Pr_{s \in D_0}\left[s^{2^i} \in E_i\right] = \sum_{i=1}^{r} \frac{|E_i|}{|D_i|}.$$

Triangular inequality:

$$\underbrace{\Delta(\textsc{Fold}\left[f_i, \alpha_i\right], f_{i+1})}_{=\frac{|E_{i+1}|}{|D_{i+1}|}} + \Delta(f_{i+1}, \mathsf{RS}_{i+1}) \geq \underbrace{\Delta(\textsc{Fold}\left[f_i, \alpha_i\right], \mathsf{RS}_{i+1})}_{\geq \delta_i}.$$

We deduce that for all $i < r - 1$, $\frac{|E_{i+1}|}{|D_{i+1}|} \geq \delta_i - \delta_{i+1}$, since:

- If $\delta_{i+1} < \delta_i$, then $\delta_{i+1} < 1 - \sqrt{\rho}$, and thus $\delta_{i+1} = \Delta(f_{i+1}, \mathsf{RS}_{i+1})$.
- Otherwise $\delta_i - \delta_{i+1} \leq 0$, thus it's clear.

$$\delta_i = \min\left(\Delta(f_i, \mathsf{RS}_i), 1 - \sqrt{\rho}\right)$$

Soundness analysis II

Considering the modified oracles, define $E_{i+1} = \{y \in D_{i+1} \mid y \text{ is red, i.e. } f_{i+1}(y) \neq \textsc{Fold}\left[f_i, \alpha_i\right](y)\}$.
If no BAD_i **occurs,** $\mathcal{V}$ **rejects with proba at most**

$$\Pr_{s \in D_0}\left[\exists i \in \{1, \ldots, r\}, s^{2^i} \in E_i\right] = \sum_{i=1}^{r} \Pr_{s \in D_0}\left[s^{2^i} \in E_i\right] = \sum_{i=1}^{r} \frac{|E_i|}{|D_i|}.$$

Triangular inequality:

$$\underbrace{\Delta(\textsc{Fold}\left[f_i, \alpha_i\right], f_{i+1})}_{=\frac{|E_{i+1}|}{|D_{i+1}|}} + \Delta(f_{i+1}, \mathsf{RS}_{i+1}) \geq \underbrace{\Delta(\textsc{Fold}\left[f_i, \alpha_i\right], \mathsf{RS}_{i+1})}_{\geq \delta_i}.$$

We deduce that for all $i < r - 1$, $\frac{|E_{i+1}|}{|D_{i+1}|} \geq \delta_i - \delta_{i+1}$, since:

- If $\delta_{i+1} < \delta_i$, then $\delta_{i+1} < 1 - \sqrt{\rho}$, and thus $\delta_{i+1} = \Delta(f_{i+1}, \mathsf{RS}_{i+1})$.
- Otherwise $\delta_i - \delta_{i+1} \leq 0$, thus it's clear.

$\delta_i = \min\left(\Delta(f_i, \mathsf{RS}_i), 1 - \sqrt{\rho}\right)$

Thus, $\sum_{i=1}^{r} \frac{|E_i|}{|D_i|} \geq \sum_{i=0}^{r-1}(\delta_i - \delta_{i+1}) \geq \delta_0 - \delta_r = \delta_0$.

Soundness analysis II

Considering the modified oracles, define $E_{i+1} = \{y \in D_{i+1} \mid y \text{ is red, i.e. } f_{i+1}(y) \neq \textsc{Fold}\left[f_i, \alpha_i\right](y)\}$.
If no BAD_i **occurs,** $\mathcal{V}$ **rejects with proba at most**

$$\Pr_{s \in D_0}\left[\exists i \in \{1, \ldots, r\}, s^{2^i} \in E_i\right] = \sum_{i=1}^{r} \Pr_{s \in D_0}\left[s^{2^i} \in E_i\right] = \sum_{i=1}^{r} \frac{|E_i|}{|D_i|}.$$

Triangular inequality:

$$\underbrace{\Delta(\textsc{Fold}\left[f_i, \alpha_i\right], f_{i+1})}_{=\frac{|E_{i+1}|}{|D_{i+1}|}} + \Delta(f_{i+1}, \mathsf{RS}_{i+1}) \geq \underbrace{\Delta(\textsc{Fold}\left[f_i, \alpha_i\right], \mathsf{RS}_{i+1})}_{\geq \delta_i}.$$

We deduce that for all $i < r - 1$, $\frac{|E_{i+1}|}{|D_{i+1}|} \geq \delta_i - \delta_{i+1}$, since:

- If $\delta_{i+1} < \delta_i$, then $\delta_{i+1} < 1 - \sqrt{\rho}$, and thus $\delta_{i+1} = \Delta(f_{i+1}, \mathsf{RS}_{i+1})$. $\qquad \delta_i = \min\left(\Delta(f_i, \mathsf{RS}_i), 1 - \sqrt{\rho}\right)$
- Otherwise $\delta_i - \delta_{i+1} \leq 0$, thus it's clear.

Thus, $\sum_{i=1}^{r} \frac{|E_i|}{|D_i|} \geq \sum_{i=0}^{r-1} (\delta_i - \delta_{i+1}) \geq \delta_0 - \delta_r = \delta_0$.

Conclude: If $\delta = \Delta(f_0, \mathsf{RS}_0) > 0$, then $\Pr(\mathcal{V} \text{ accept}) \leq O_\rho\left(\frac{n^2}{|\mathbb{F}|}\right) + \left(1 - \min\left(\delta, 1 - \sqrt{\rho}\right)\right)$. $\square$

Conclusion

In a nutshell:

Arithmetization:

- valid execution trace $\rightsquigarrow$ RS codewords
- incorrect execution trace $\rightsquigarrow$ words far from any RS codewords

Low-degree test: key-ingredient is a "folding operator" such that:

- codeword of big RS code $\mapsto$ codeword of small RS code
- locally computable
- word far from big RS code $\mapsto$ words far from small RS code (w.h.p)

Conclusion

In a nutshell:
Arithmetization:

- valid execution trace $\rightsquigarrow$ RS codewords
- incorrect execution trace $\rightsquigarrow$ words far from any RS codewords

Low-degree test: key-ingredient is a "folding operator" such that:

- codeword of big RS code $\mapsto$ codeword of small RS code
- locally computable
- word far from big RS code $\mapsto$ words far from small RS code (w.h.p)

What about other codes?
Proximity testing to algebraic codes via IOPP, with linear prover and logarithmic verifier:

- Algebraic Geometry codes (joint work with Jade Nardi)
  `https://eccc.weizmann.ac.il/report/2020/165/`
- Reed-Muller and Tensor Products of RS codes (joint work with Daniel)

The challenging part is to construct folding operators satisfying "distance preservation".