Codes on any Cayley Graph have an Interactive Oracle Proof of Proximity

Hugo Delavenne, Louise Lallemand

CANS 2025, Osaka

Thursday 20th November 2025

- This is a follow up of [DMR25] with **Tanguy Medevielle** and **Élina Roussel**.
- There is an **error** in [DMR25, DL25]...
  we can **fix** it...
  but slightly **worsening complexity**.

[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs.
In *2025 IEEE International Symposium on Information Theory (ISIT)*, 2025

[DL25] Hugo Delavenne and Louise Lallemand. Codes on any Cayley Graph have an Interactive Oracle Proof of Proximity, 2025.
This work

1. Interactive Oracle Proofs of Proximity
2. Codes on graphs
3. Folding graphs
4. Cayley graphs
5. Flowering protocol

1 Interactive Oracle Proofs of Proximity

2 Codes on graphs

3 Folding graphs

4 Cayley graphs

5 Flowering protocol

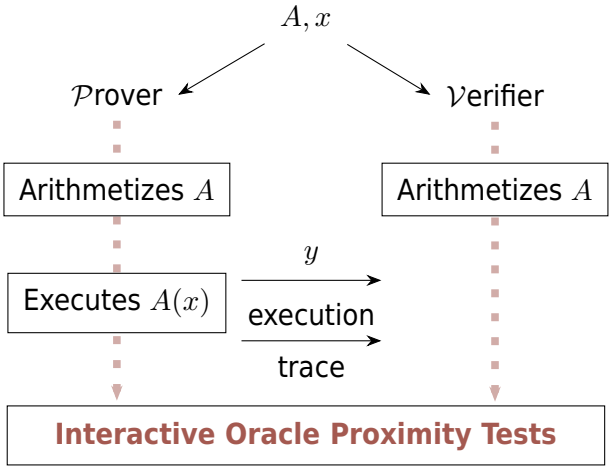
- Motivation: Code-based SNARKs
- Error correcting codes
- Folding Reed-Solomon codewords
- Successive foldings

SNARK means

- **S**uccinct
- **N**on-interactive
- **AR**gument of
- **K**nowledge

Goal: $\mathcal{P}$ convices $\mathcal{V}$ that

$$y = A(x)$$



An **error correcting code** C is a linear subspace of $\mathbb{F}^N$ with a **distance**.

Consider a support $\mathcal{L}$, $|\mathcal{L}| = N$. View $C \subseteq \{f : \mathcal{L} \to \mathbb{F}\}$.

Definition **Relative Hamming distance**

Let $f, g : \mathcal{L} \to \mathbb{F}$.

$$\Delta(f, g) := \frac{1}{N}|\{x \in \mathcal{L} \mid f(x) \neq g(x)\}|$$

Example **Reed-Solomon code**

Let $\mathcal{L} \subseteq \mathbb{F}$, $|\mathcal{L}| = N$, and $K < |\mathcal{L}|$.

$$\mathrm{RS}[N, K] := \{f : \mathcal{L} \to \mathbb{F} \mid f \in \mathbb{F}[X]_{\leqslant K-1}\}$$

Idea: Reduce testing $C := \mathbf{RS}[|\mathcal{L}_0|, K]$ to $C' := \mathbf{RS}[|\mathcal{L}_1|, K/2]$ using

$$f(X) =: f_{\mathsf{even}}(X^2) + X f_{\mathsf{odd}}(X^2).$$

Definition **Folding operator [BBHR18]**

Let $f : \mathcal{L}_0 \to \mathbb{F}$ and $\alpha \in \mathbb{F}$. Let $\mathcal{L}_1 := \{x^2 \mid x, -x \in \mathcal{L}_0\}$.

$$\begin{aligned} \mathsf{Fold}[f,\alpha](X^2) &:= f_{\mathsf{even}}(X^2) + \alpha f_{\mathsf{odd}}(X^2) \\ &= \tfrac{f(X)+f(-X)}{2} + \alpha \tfrac{f(X)-f(-X)}{2X} \end{aligned}$$

Locality: $\mathcal{V}$ computes $\mathsf{Fold}[f,\alpha](x^2)$ with **2 queries** to f

Completeness: If $f \in C$ then $\mathbf{Fold}[f,\alpha] \in C'$ for all α

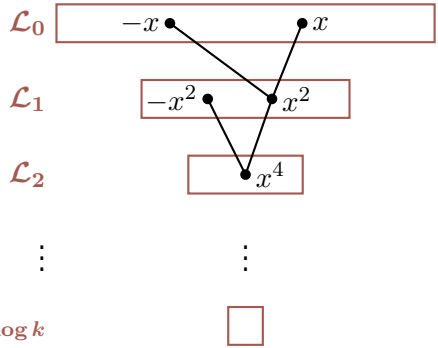
Soundness: If $\Delta(f, C) > \delta$ then $\Delta(\mathbf{Fold}[f,\alpha], C') > \delta$ w.h.p. over α

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast Reed-Solomon Interactive Oracle Proofs of Proximity.
In *45th international colloquium on automata, languages, and programming (ICALP 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018

$\mathcal{P}$ performs **successive foldings** on

$$\mathcal{L}_{i+1} := \{x^2 \mid x, -x \in \mathcal{L}_i\}$$

Field restriction:
$\mathcal{L}_0$ must have k^{th} roots of unity
$\rightarrow \mathbb{F}$ extension of $\mathbb{F}_p$ with $p = 2^{64} - 2^{32} + 1$



[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast Reed-Solomon Interactive Oracle Proofs of Proximity.
In *45th international colloquium on automata, languages, and programming (ICALP 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018

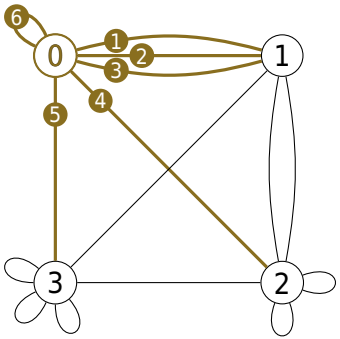
- Regular Indexed Multigraphs (RIM)
- Words and codes on graphs

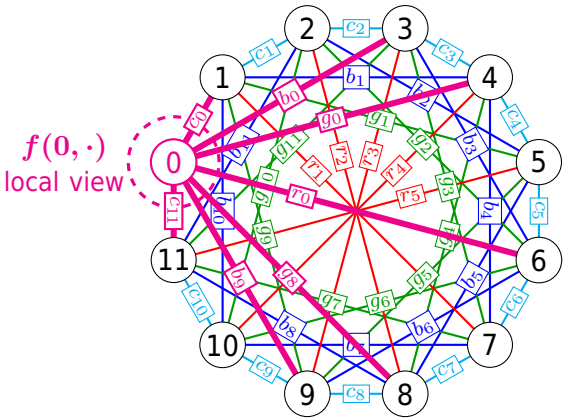
$\Gamma = (V, E)$ is a n-**RIM**:

- **Multigraph**: multiple edges and loops
- **Regular**: same number n of edges
- **Indexed**: edge is $(v, \ell) \in V \times [n]$
  Write $E(v, \ell) \in V$ the neighbor of v by ℓ

Denote $\mathcal{L} = V \times [n]/\sim_\Gamma$ **set of edges**

where $(v, \ell) \sim_\Gamma (v', \ell')$ if $E(v, \ell) = v'$ and $E(v', \ell') = v$





Word $f : \mathcal{L} \to \mathbb{F}$ on a graph Γ

Definition **Code $\mathcal{C}[\Gamma, k]$**

Given Γ a n-RIM and $k \leqslant n$,

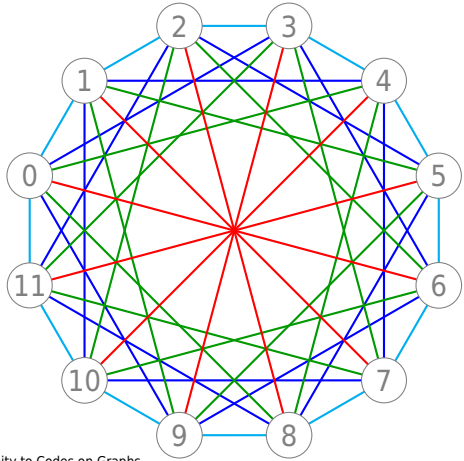
$f \in \mathcal{C}[\Gamma, k] \iff \forall v,\ \boldsymbol{f(v, \cdot)} \in \mathsf{RS}[n, k]$

Example

For $v = 0$, being a codeword requires $(\boldsymbol{c_0, b_0, g_0, r_0, g_8, b_9, c_{11}}) \in \mathsf{RS}[7, k]$.

- Folding graphs
- Disjoint flowering example
- Non disjoint flowering example

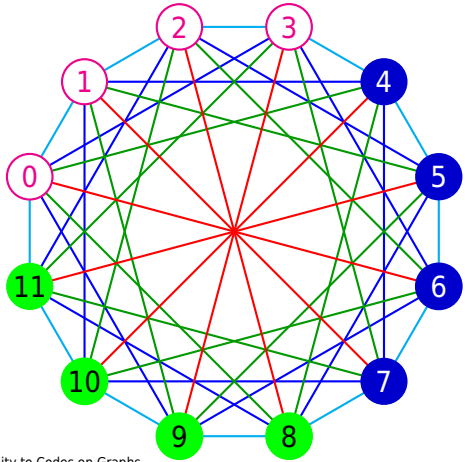
Folding graphs



[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs.
In *2025 IEEE International Symposium on Information Theory (ISIT)*, 2025

[DL25] Hugo Delavenne and Louise Lallemand. Codes on any Cayley Graph have an Interactive Oracle Proof of Proximity, 2025.
This work

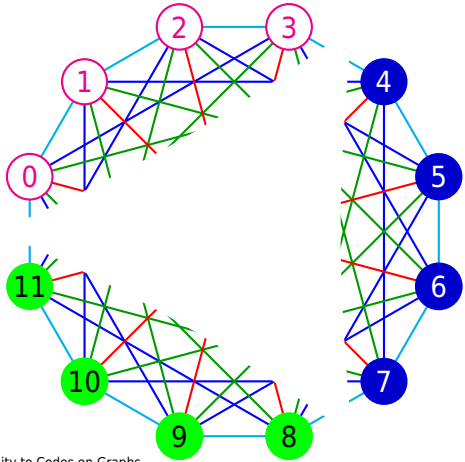
- Choose subsets of vertices $V_0, ..., V_m \subseteq V$



[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs.
In *2025 IEEE International Symposium on Information Theory (ISIT)*, 2025

[DL25] Hugo Delavenne and Louise Lallemand. Codes on any Cayley Graph have an Interactive Oracle Proof of Proximity, 2025.
This work

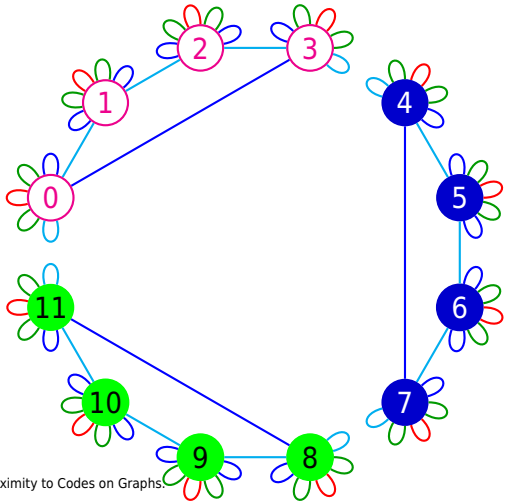
- Choose subsets of vertices $V_0, ..., V_m \subseteq V$
- Cut outgoing edges



[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs.
In *2025 IEEE International Symposium on Information Theory (ISIT)*, 2025

[DL25] Hugo Delavenne and Louise Lallemand. Codes on any Cayley Graph have an Interactive Oracle Proof of Proximity, 2025.
This work

- ▷ Choose subsets of vertices $V_0, ..., V_m \subseteq V$
- ▷ Cut outgoing edges
- ▶ Turn floating edges into **petals**

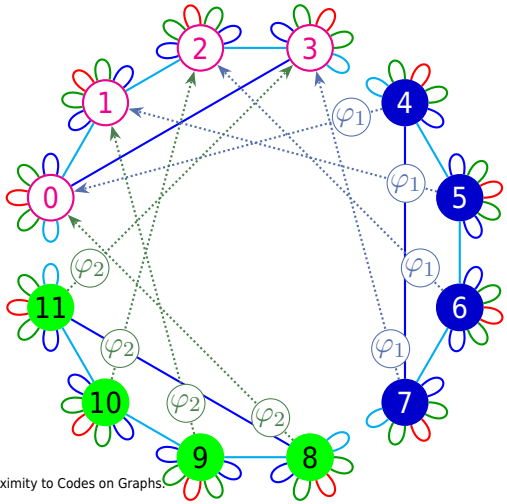


[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs.
In *2025 IEEE International Symposium on Information Theory (ISIT)*, 2025

[DL25] Hugo Delavenne and Louise Lallemand. Codes on any Cayley Graph have an Interactive Oracle Proof of Proximity, 2025.
This work

- ▷ Choose subsets of vertices $V_0, ..., V_m \subseteq V$
- ▷ Cut outgoing edges
- ▷ Turn floating edges into **petals**
- ▶ Make sure cuts are **isomorphic**
  Define the **Fold** for $f : \mathcal{L} \to \mathbb{F}, \alpha \in \mathbb{F}$

$$\text{Fold}[f, \alpha](v, \ell) := \sum_{i=0}^{m-1} \alpha^i f(\varphi_i^{-1}(v), \ell)$$



[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs.
In *2025 IEEE International Symposium on Information Theory (ISIT)*, 2025

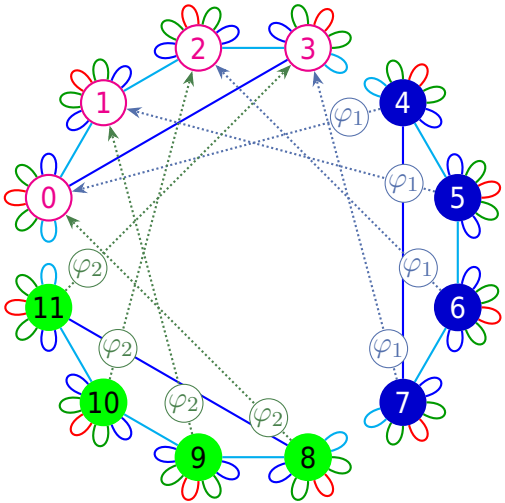
[DL25] Hugo Delavenne and Louise Lallemand. Codes on any Cayley Graph have an Interactive Oracle Proof of Proximity, 2025.
This work

- Choose subsets of vertices $V_0, ..., V_m \subseteq V$
- Cut outgoing edges
- Turn floating edges into **petals**
- Make sure cuts are **isomorphic**

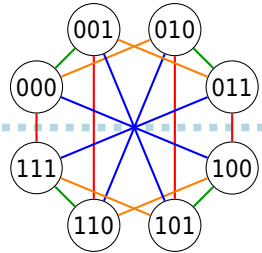
  Define the **Fold** for $f : \mathcal{L} \to \mathbb{F}, \alpha \in \mathbb{F}$

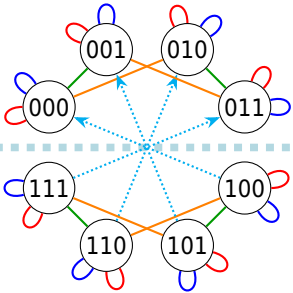
$$\mathsf{Fold}[f, \alpha] := \sum_{i=0}^{m-1} \alpha^i f_i, \quad f_i := \sum_{j=0}^{m-1} (-1)^{b_{ij}} f \circ \varphi_j$$

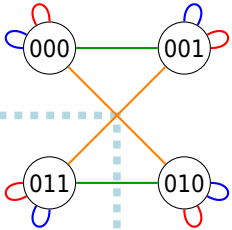
FIX

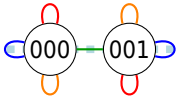


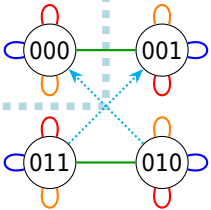
Disjoint flowering example





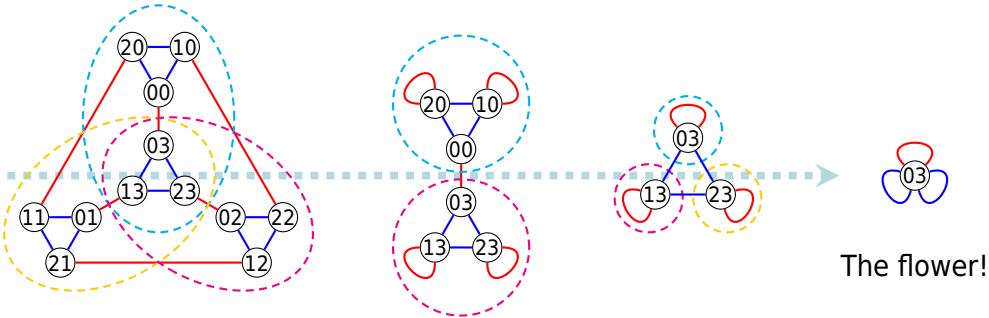






The flower!

Non disjoint flowering example



The flower!

- Cayley graphs
- Expander graphs
- Cutting Cayley graphs

Definition [Cay78]

Let $(G, \cdot)$ and $S \subseteq G$ such that

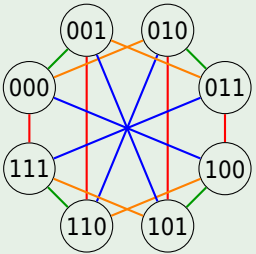
- $|S| = n$ (arity)
- $G = \langle S \rangle$ (connected)
- $S^{-1} \subseteq S$. (non-oriented)

Define $\mathbf{Cay}(G, S) = (V, E)$ by

- $V := G$
- $E(g, s) = g \cdot s$.

Example

With $G = (\mathbb{F}_2^3, +)$ and $S = \{100, 010, 001, 111\}$,



[Cay78] Arthur Cayley. Desiderata and Suggestions: No. 2. The Theory of Groups: Graphical Representation.
American Journal of Mathematics, 1(2):174–176, 1878

Graph expansion $\varepsilon \in [0,1]$ gives connectivity, random walk convergence, etc

Theorem **LPS Ramanujan graphs [LPS88]**

Let $n-1 \neq q$ be primes such that $n-1, q \equiv 1 \mod 4$. We can construct $G_{n,q}, S_{n,q}$

- $G_{n,q} = \mathrm{PGL}_2(\mathbb{F}_q)$ or $\mathrm{PSL}_2(\mathbb{F}_q)$ and $|S_{n,q}| = n$
- $\mathrm{Cay}(G_{n,q}, S_{n,q})$ is optimal expander, with $\varepsilon(q)$ constant

Lemma **[AC88]**

Let Γ be ε-expander and $\delta := 1 - \frac{k+1}{n}$

$\mathcal{C}[\Gamma, k]$ has **minimum distance** $\geqslant \delta(\delta - \varepsilon)$

Conjecture **[Sar19]**

The diameter of LPS graphs is

$$\leqslant (4/3) \cdot \log_n(q) \leqslant \frac{1}{2} \log N$$

[LPS88] Alexander Lubotzky, Ralph Phillips, and Peter Sarnak. Ramanujan graphs. ***Combinatorica*, 8(3):261–277, 1988**

[AC88] Noga Alon and Fan Chung. Explicit construction of linear sized tolerant networks. ***Discrete Mathematics*, 72(1-3):15–19, 1988**

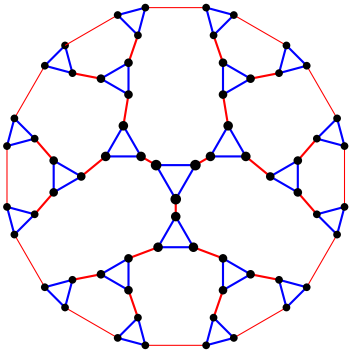
[Sar19] Naser T Sardari. Diameter of Ramanujan graphs and random Cayley graphs. ***Combinatorica*, 39(2):427–446, 2019**

Cutting Cayley graphs

If $G = \langle \{s_1, ..., s_n\} \rangle$ then $G = \left\{ \bullet_{i=1}^{\mathrm{diam}(\Gamma)} s_1^* \cdots s_n^* \right\}$ ($s^* = s^j$ for any $j \in \mathbb{Z}$)

Assume $S = \{\boldsymbol{b}, \boldsymbol{b}^{-1}, \boldsymbol{r}\}$ and $G = \langle S \rangle$ is

$$G = \{\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\}$$



[DL25] Hugo Delavenne and Louise Lallemand. Codes on any Cayley Graph have an Interactive Oracle Proof of Proximity, 2025.
This work

Cutting Cayley graphs

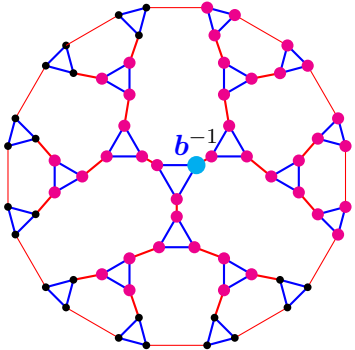
If $G = \langle \{s_1, ..., s_n\} \rangle$ then $G = \left\{ \bullet_{i=1}^{\mathrm{diam}(\Gamma)} s_1^* \cdots s_n^* \right\}$ ($s^* = s^j$ for any $j \in \mathbb{Z}$)

Assume $S = \{b, b^{-1}, r\}$ and $G = \langle S \rangle$ is

$$G = \{b^* r^* b^* r^* b^* r^* b^*\}$$

First cuts are:

- $V_{0,-1} = \{b^{-1} \cdot r^* b^* r^* b^* r^* b^*\}$



[DL25] Hugo Delavenne and Louise Lallemand. Codes on any Cayley Graph have an Interactive Oracle Proof of Proximity, 2025. **This work**

Cutting Cayley graphs

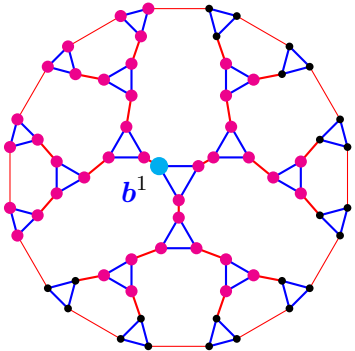
If $G = \langle \{s_1, ..., s_n\} \rangle$ then $G = \left\{ \bullet_{i=1}^{\mathrm{diam}(\Gamma)} \; s_1^* \cdots s_n^* \right\}$ ($s^* = s^j$ for any $j \in \mathbb{Z}$)

Assume $S = \{\boldsymbol{b}, \boldsymbol{b}^{-1}, \boldsymbol{r}\}$ and $G = \langle S \rangle$ is

$$G = \{\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\}$$

First cuts are:

- ▷ $V_{0,-1} = \{\boldsymbol{b}^{-1} \cdot \boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\}$
- ▶ $V_{0,1} = \{\boldsymbol{b}^1 \cdot \boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\}$



[DL25] Hugo Delavenne and Louise Lallemand. Codes on any Cayley Graph have an Interactive Oracle Proof of Proximity, 2025. **This work**

Cutting Cayley graphs

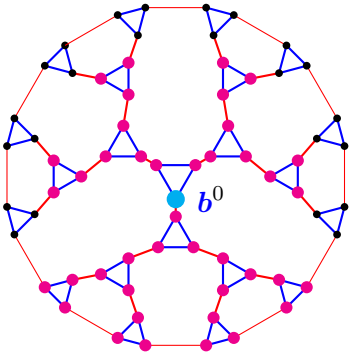
If $G = \langle \{s_1, ..., s_n\} \rangle$ then $G = \left\{ \bullet_{i=1}^{\mathrm{diam}(\Gamma)} \; s_1^* \cdots s_n^* \right\}$ ($s^* = s^j$ for any $j \in \mathbb{Z}$)

Assume $S = \{\boldsymbol{b}, \boldsymbol{b}^{-1}, \boldsymbol{r}\}$ and $G = \langle S \rangle$ is

$$G = \{\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\}$$

First cuts are:

- ▷ $\boldsymbol{V_{0,-1}} = \{\boldsymbol{b}^{-1} \cdot \boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\}$
- ▷ $\boldsymbol{V_{0,1}} = \{\boldsymbol{b}^{1} \cdot \boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\}$
- ▶ $\boldsymbol{V_{0,0}} = \{\boldsymbol{b}^{0} \cdot \boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\}$



[DL25] Hugo Delavenne and Louise Lallemand. Codes on any Cayley Graph have an Interactive Oracle Proof of Proximity, 2025.
This work

Cutting Cayley graphs

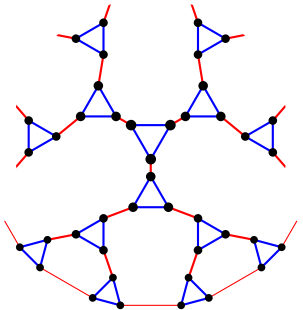
If $G = \langle \{s_1, ..., s_n\} \rangle$ then $G = \left\{ \bullet_{i=1}^{\mathrm{diam}(\Gamma)} s_1^* \cdots s_n^* \right\}$ ($s^* = s^j$ for any $j \in \mathbb{Z}$)

Assume $S = \{\boldsymbol{b}, \boldsymbol{b}^{-1}, \boldsymbol{r}\}$ and $G = \langle S \rangle$ is

$$G = \{\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\}$$

First cuts are:

- $\boldsymbol{V_{0,-1}} = \{\boldsymbol{b}^{-1} \cdot \boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\}$
- $\boldsymbol{V_{0,1}} = \{\boldsymbol{b}^{1} \cdot \boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\}$
- $\boldsymbol{V_{0,0}} = \{\boldsymbol{b}^{0} \cdot \boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\}$



[DL25] Hugo Delavenne and Louise Lallemand. Codes on any Cayley Graph have an Interactive Oracle Proof of Proximity, 2025. **This work**

Cutting Cayley graphs

If $G = \langle \{s_1, ..., s_n\} \rangle$ then $G = \left\{ \bullet_{i=1}^{\mathrm{diam}(\Gamma)} s_1^* \cdots s_n^* \right\}$ ($s^* = s^j$ for any $j \in \mathbb{Z}$)

Assume $S = \{\boldsymbol{b}, \boldsymbol{b}^{-1}, \boldsymbol{r}\}$ and $G = \langle S \rangle$ is

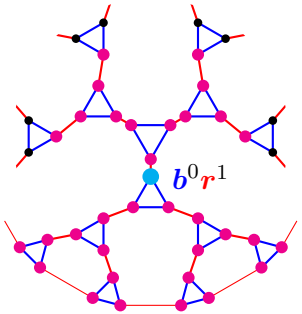
$$G = \{\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\}$$

First cuts are:

- $\boldsymbol{V_{0,-1}} = \{\boldsymbol{b}^{-1} \cdot \boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\}$
- $\boldsymbol{V_{0,1}} = \{\boldsymbol{b}^{1} \cdot \boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\}$
- $\boldsymbol{V_{0,0}} = \{\boldsymbol{b}^{0} \cdot \boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\}$

Second cuts are:

- $\boldsymbol{V_{1,1}} = \{\boldsymbol{b}^0\boldsymbol{r}^1 \cdot \boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\}$



Cutting Cayley graphs

If $G = \langle \{s_1, ..., s_n\} \rangle$ then $G = \left\{ \bullet_{i=1}^{\mathrm{diam}(\Gamma)} \; s_1^* \cdots s_n^* \right\}$ ($s^* = s^j$ for any $j \in \mathbb{Z}$)

Assume $S = \{\boldsymbol{b}, \boldsymbol{b}^{-1}, \boldsymbol{r}\}$ and $G = \langle S \rangle$ is

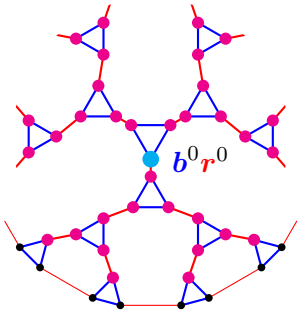
$$G = \{\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\}$$

First cuts are:

- $\boldsymbol{V_{0,-1}} = \{\boldsymbol{b}^{-1} \cdot \boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\}$
- $\boldsymbol{V_{0,1}} = \{\boldsymbol{b}^{1} \cdot \boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\}$
- $\boldsymbol{V_{0,0}} = \{\boldsymbol{b}^{0} \cdot \boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\}$

Second cuts are:

- $\boldsymbol{V_{1,1}} = \{\boldsymbol{b}^{0}\boldsymbol{r}^{1} \cdot \boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\}$
- $\boldsymbol{V_{1,0}} = \{\boldsymbol{b}^{0}\boldsymbol{r}^{0} \cdot \boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\boldsymbol{r}^*\boldsymbol{b}^*\}$



Cutting Cayley graphs

If $G = \langle \{s_1, ..., s_n\} \rangle$ then $G = \left\{ \bullet_{i=1}^{\mathrm{diam}(\Gamma)} \; s_1^* \cdots s_n^* \right\}$ ($s^* = s^j$ for any $j \in \mathbb{Z}$)

Assume $S = \{b, b^{-1}, r\}$ and $G = \langle S \rangle$ is

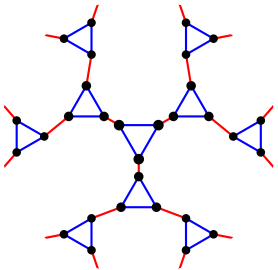
$$G = \{b^* r^* b^* r^* b^* r^* b^*\}$$

First cuts are:

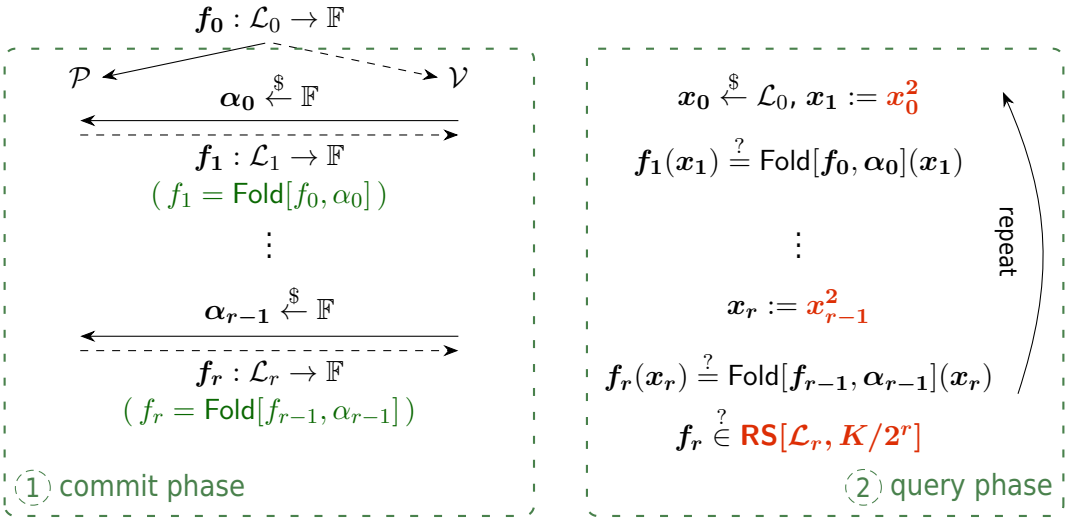
- $V_{0,-1} = \{b^{-1} \cdot r^* b^* r^* b^* r^* b^*\}$
- $V_{0,1} = \{b^1 \cdot r^* b^* r^* b^* r^* b^*\}$
- $V_{0,0} = \{b^0 \cdot r^* b^* r^* b^* r^* b^*\}$

Second cuts are:

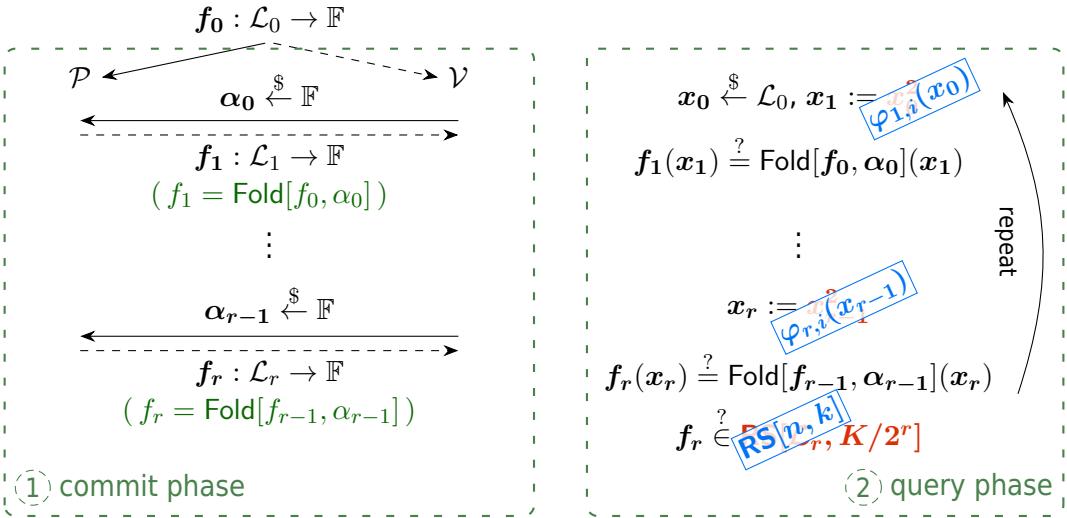
- $V_{1,1} = \{b^0 r^1 \cdot b^* r^* b^* r^* b^*\}$
- $V_{1,0} = \{b^0 r^0 \cdot b^* r^* b^* r^* b^*\}$



- **FRI** and **Flowering** protocols
- **FRI** and **Flowering** complexity comparison



[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast Reed-Solomon Interactive Oracle Proofs of Proximity.
In *45th international colloquium on automata, languages, and programming (ICALP 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018



[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast Reed-Solomon Interactive Oracle Proofs of Proximity.
In *45th international colloquium on automata, languages, and programming (ICALP 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018

FRI and Flowering complexity comparison

To test δ-proximity to a code $[N > 2^{20}, K > 2^{18}, D]$ with λ bits of security, with

$$\delta_1 := \min\left(\delta, 1 - \sqrt{1 - D/N}\right) \qquad \delta_2 := \min\left(\delta, 1 - \sqrt[4]{1 - D/N}\right)$$

Protocol	Flowering	BaseFold	FRI	STIR
Encoding	$O(N)$ (hopefully)	$O(N \log N)$		
Prover	$O(nN \log N)$	$O(N)$		
Verifier	$O(n\frac{\lambda}{\delta}\log^2 N)$	$O(\frac{\lambda}{\delta_2}\log^2 N)$	$O(\frac{\lambda}{\delta_1}\log^2 N)$	$O_{\delta_1}((\lambda^2 + \lambda \log\log K)\log N)$
Field size	$\Omega(2^\lambda N^3)$		$\Omega(2^\lambda N^{3.5}/K^{1.5})$	$\Omega(\lambda 2^\lambda K^2 N^{3.5})$
Structure	Any field		$\lvert\mathbb{F}\rvert - 1$ smooth	$2^{\lceil \log K \rceil} \mid \lvert\mathbb{F}\rvert - 1$

[DL25] Hugo Delavenne and Louise Lallemand. Codes on any Cayley Graph have an Interactive Oracle Proof of Proximity, 2025.
This work

[ZCF24] Hadas Zeilberger, Binyi Chen, and Ben Fisch. BaseFold: Efficient Field-Agnostic Polynomial Commitment Schemes from Foldable Codes.
In *Annual International Cryptology Conference*, pages 138–169. Springer, 2024

[BCI+23] Eli Ben-Sasson, Dan Carmon, Yuval Ishai, Swastik Kopparty, and Shubhangi Saraf. Proximity Gaps for Reed-Solomon Codes.
***J. ACM*, 70(5), October 2023**

[ACFY24] Gal Arnon, Alessandro Chiesa, Giacomo Fenzi, and Eylon Yogev. STIR: Reed-Solomon Proximity Testing with Fewer Queries.
In *Annual International Cryptology Conference*, pages 380–413. Springer, 2024

FRI and Flowering complexity comparison

To test δ-proximity to a code $[N > 2^{20}, K > 2^{18}, D]$ with λ bits of security, with

$$\delta_1 := \min\left(\delta, 1 - \sqrt{1 - D/N}\right) \qquad \delta_2 := \min\left(\delta, 1 - \sqrt[4]{1 - D/N}\right)$$

Protocol	Flowering	BaseFold	FRI	STIR
Encoding	$O(N)$ (hopefully)	$O(N \log N)$		
Prover	$O(nN \log N)$	$O(N)$		
Verifier	$O(n\frac{\lambda}{\delta_2} \log^2 N)$ FIX	$O(\frac{\lambda}{\delta_2} \log^2 N)$	$O(\frac{\lambda}{\delta_1} \log^2 N)$	$O_{\delta_1}((\lambda^2 + \lambda \log\log K) \log N)$
Field size	$\Omega(2^\lambda N^3)$		$\Omega(2^\lambda N^{3.5}/K^{1.5})$	$\Omega(\lambda 2^\lambda K^2 N^{3.5})$
Structure	Any field		$\lvert\mathbb{F}\rvert - 1$ smooth	$2^{\lceil \log K \rceil} \mid \lvert\mathbb{F}\rvert - 1$

[DL25] Hugo Delavenne and Louise Lallemand. Codes on any Cayley Graph have an Interactive Oracle Proof of Proximity, 2025.
This work

[ZCF24] Hadas Zeilberger, Binyi Chen, and Ben Fisch. BaseFold: Efficient Field-Agnostic Polynomial Commitment Schemes from Foldable Codes.
In *Annual International Cryptology Conference*, pages 138–169. Springer, 2024

[BCI+23] Eli Ben-Sasson, Dan Carmon, Yuval Ishai, Swastik Kopparty, and Shubhangi Saraf. Proximity Gaps for Reed-Solomon Codes.
***J. ACM*, 70(5), October 2023**

[ACFY24] Gal Arnon, Alessandro Chiesa, Giacomo Fenzi, and Eylon Yogev. STIR: Reed-Solomon Proximity Testing with Fewer Queries.
In *Annual International Cryptology Conference*, pages 380–413. Springer, 2024

▷ Conclusion

Not (yet?) competitive

- **STIR** has better complexity
- **BaseFold** has better complexity without field restriction
- Need to **improve cuts** and encoding!

Applications requiring graphs

- Verifiable Secret Sharing
- Other distributed protocols?

Thank you!