Non-Linear Repair of Reed–Solomon Codes.

Itzhak Tamo
Joint work with Roni Con

February 21, 2023

Outline

Outline

- The repair problem

Outline

- The repair problem
- Repairing RS codes - known results

Outline

- The repair problem
- Repairing RS codes - known results
- Repairing RS codes over prime fields

Outline

- The repair problem
- Repairing RS codes - known results
- Repairing RS codes over prime fields
- Summary and open questions

Information-era

Storage systems in information era

Storage systems in information era

- Goal: Reliable data storage over distributed nodes

Storage systems in information era

- Goal: Reliable data storage over distributed nodes
- Nodes can fail

Storage systems in information era

- Goal: Reliable data storage over distributed nodes
- Nodes can fail
- Solution: Add redundancy (erasure-correcting codes)

Storage systems in information era

- Goal: Reliable data storage over distributed nodes
- Nodes can fail
- Solution: Add redundancy (erasure-correcting codes)
- Redundancy = Money

MDS codes

MDS codes

- A **code**: $C : \mathbb{F}_q^k \to \mathbb{F}_q^n$

MDS codes

- A **code**: $C : \mathbb{F}_q^k \to \mathbb{F}_q^n$
- $[n, k]_q$ code $= k$-dimensional linear subspace of $\mathbb{F}_q^n$

MDS codes

- A **code**: $C : \mathbb{F}_q^k \to \mathbb{F}_q^n$
- $[n, k]_q$ code $=$ k-dimensional linear subspace of $\mathbb{F}_q^n$
- MDS $=$ optimal resiliency-redundancy tradeoff

MDS codes

- A **code**: $C : \mathbb{F}_q^k \to \mathbb{F}_q^n$
- $[n, k]_q$ code $=$ k-dimensional linear subspace of $\mathbb{F}_q^n$
- MDS $=$ optimal resiliency-redundancy tradeoff
- $[n, k]_q$ MDS code can correct any $n - k$ erasures

MDS codes

- A **code**: $C : \mathbb{F}_q^k \to \mathbb{F}_q^n$
- $[n, k]_q$ code $=$ k-dimensional linear subspace of $\mathbb{F}_q^n$
- MDS $=$ optimal resiliency-redundancy tradeoff
- $[n, k]_q$ MDS code can correct any $n - k$ erasures
- $(c_1, c_2, \ldots, c_n) \in C$

MDS codes

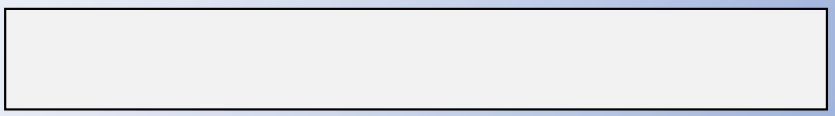
- A **code**: $C : \mathbb{F}_q^k \to \mathbb{F}_q^n$
- $[n, k]_q$ code $=$ k-dimensional linear subspace of $\mathbb{F}_q^n$
- MDS $=$ optimal resiliency-redundancy tradeoff
- $[n, k]_q$ MDS code can correct any $n - k$ erasures
- $(c_1, c_2, \ldots, c_n) \in C \to (?, ?, c_3, \ldots, ?, c_n)$

MDS codes

- A **code**: $C : \mathbb{F}_q^k \to \mathbb{F}_q^n$
- $[n, k]_q$ code $=$ k-dimensional linear subspace of $\mathbb{F}_q^n$
- MDS $=$ optimal resiliency-redundancy tradeoff
- $[n, k]_q$ MDS code can correct any $n - k$ erasures
- $(c_1, c_2, \ldots, c_n) \in C \to (?, ?, c_3, \ldots, ?, c_n) \to (c_1, c_2, \ldots, c_n)$

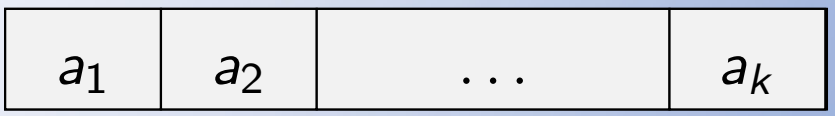
Codes for storage

file

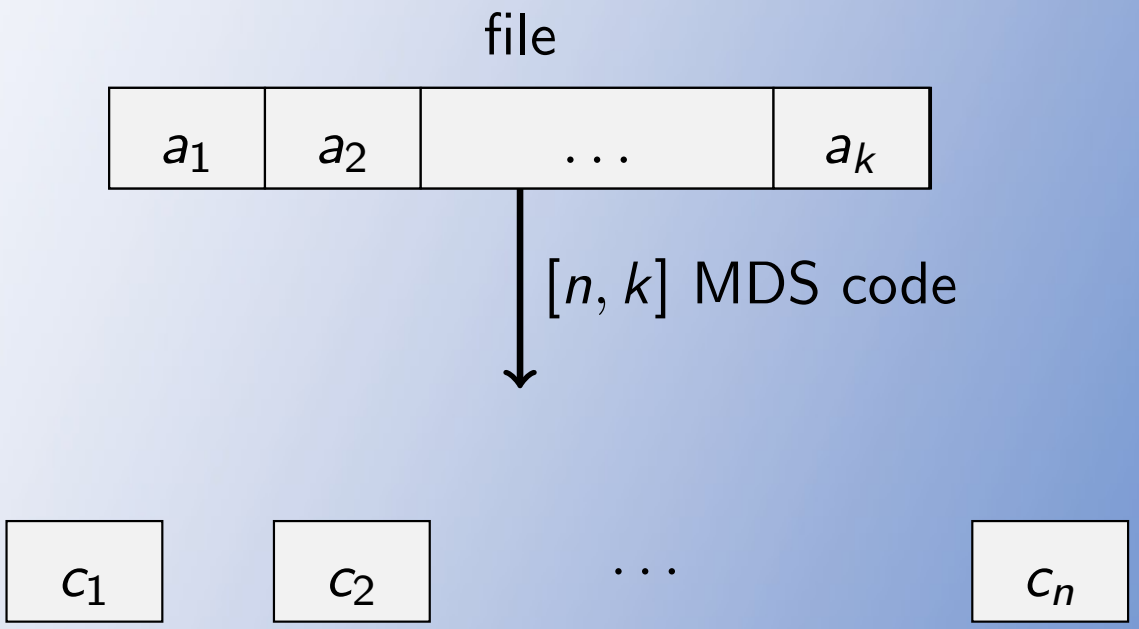


Codes for storage

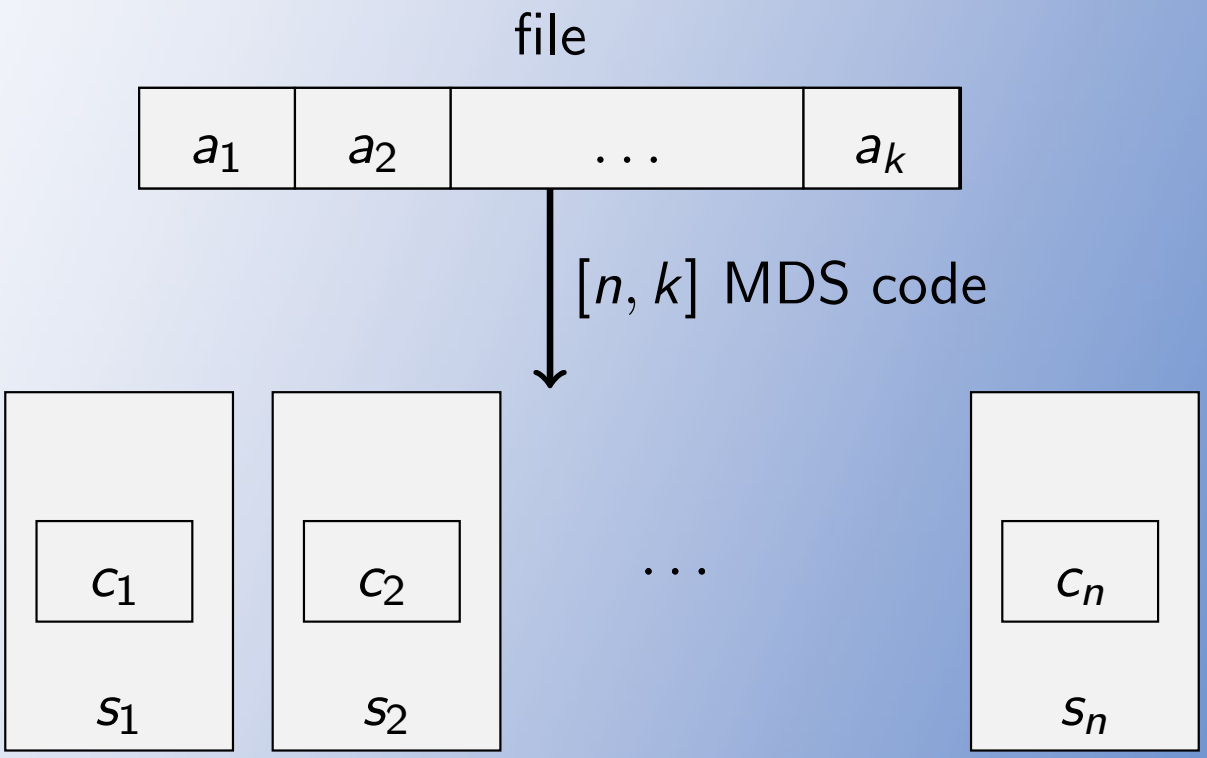
file



Codes for storage



Codes for storage



The repair problem

The repair problem

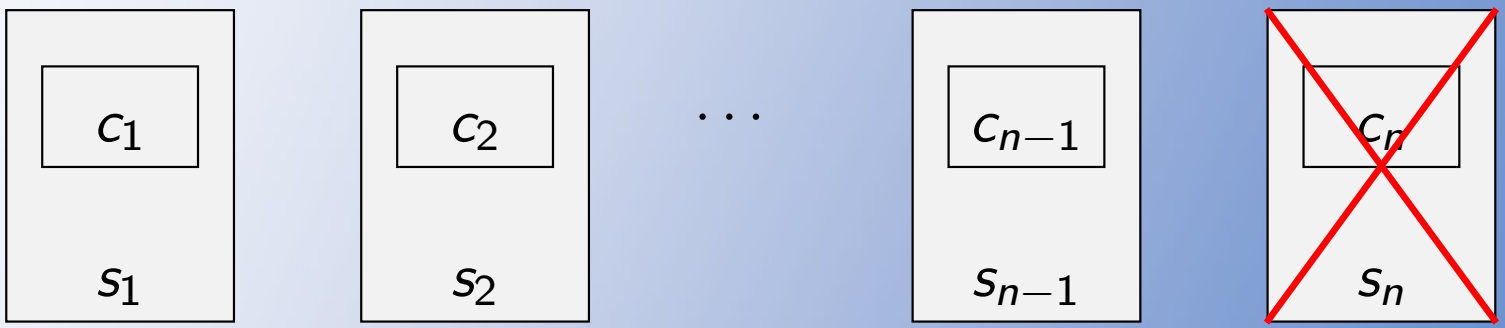
- Resiliency to worst-case scenario of $n - k$ erasures

The repair problem

- Resiliency to worst-case scenario of $n - k$ erasures
- Common scenario - a single erasure

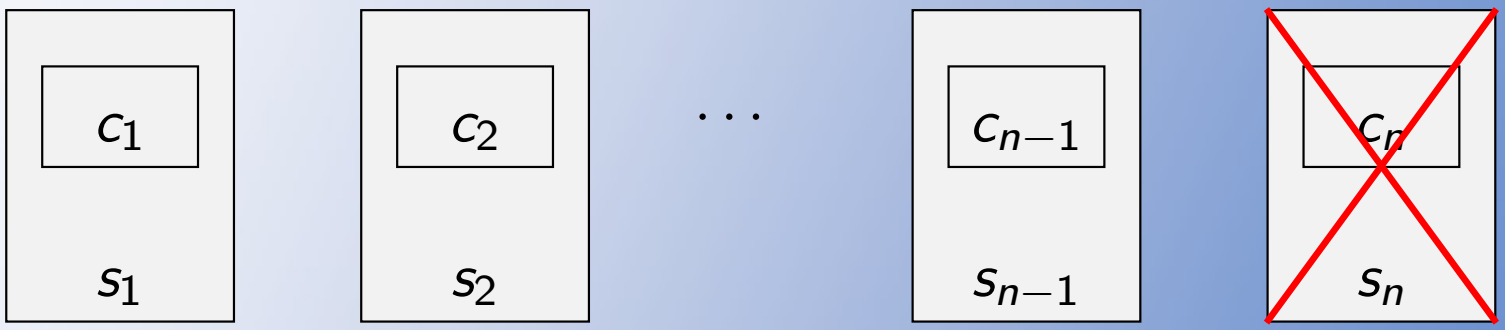
The repair problem

- Resiliency to worst-case scenario of $n - k$ erasures
- Common scenario - a single erasure



The repair problem

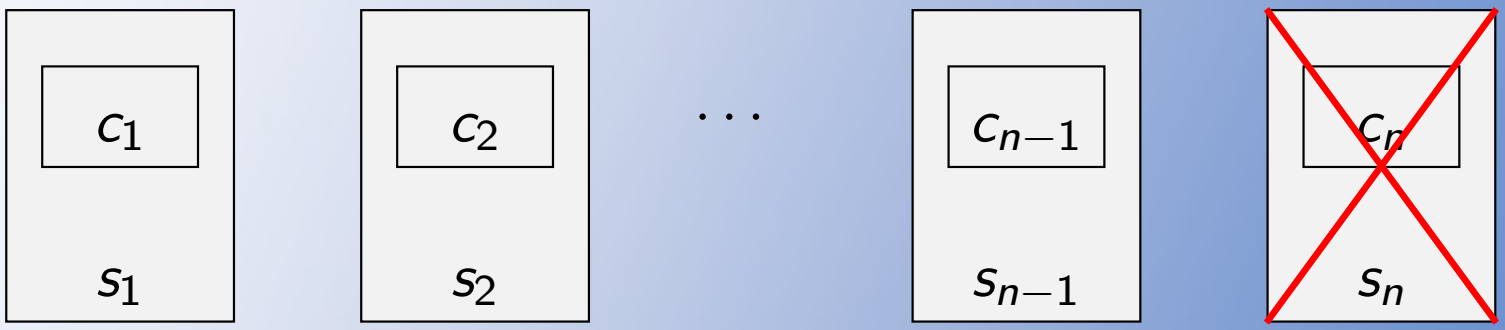
- Resiliency to worst-case scenario of $n - k$ erasures
- Common scenario - a single erasure



- Goal: Efficiently recover the erased node

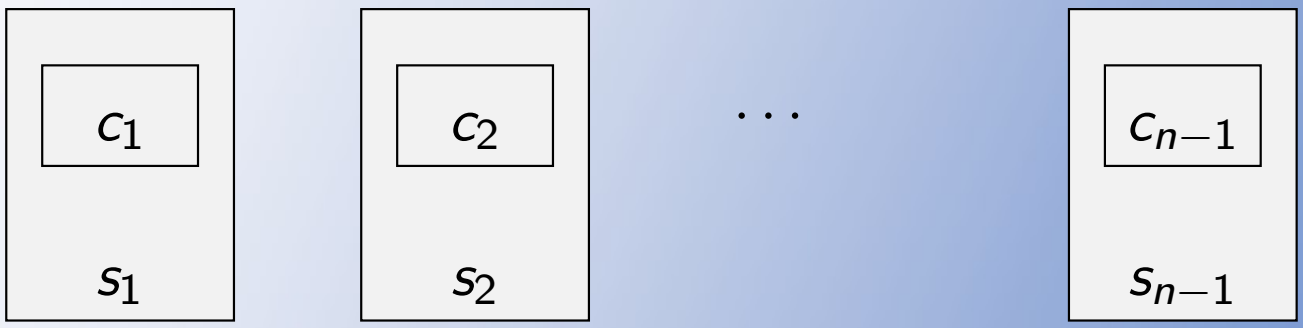
The repair problem

- Resiliency to worst-case scenario of $n - k$ erasures
- Common scenario - a single erasure

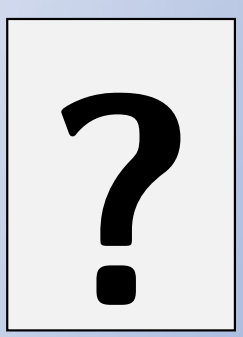


- Goal: Efficiently recover the erased node
- Repair scheme - Recovers lost data from remaining nodes

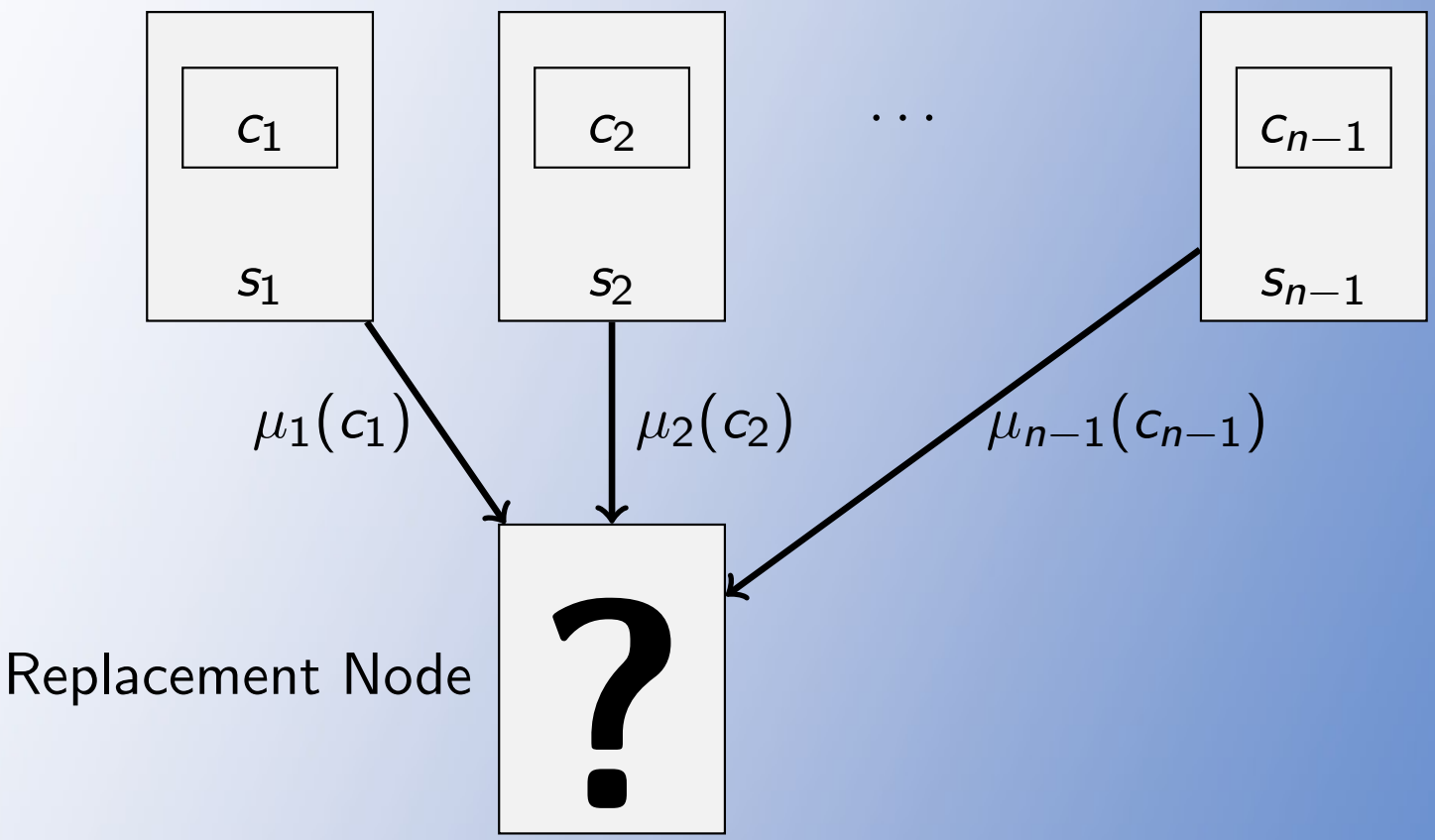
Repair scheme



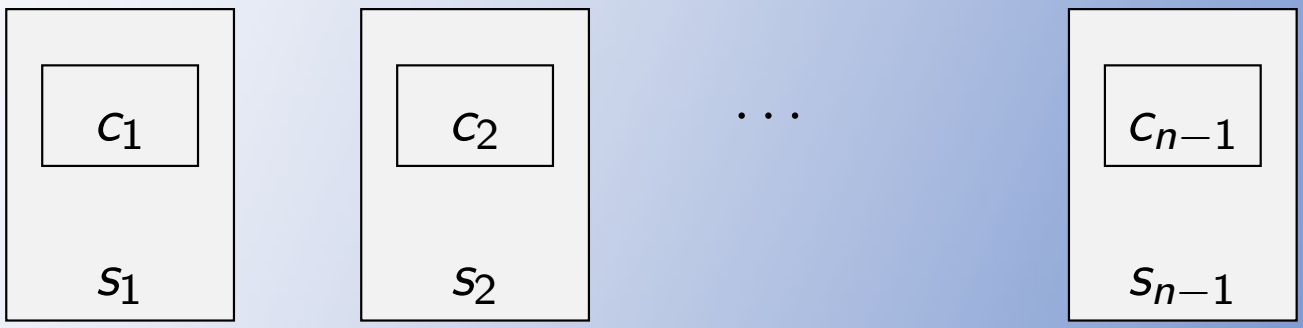
Replacement Node



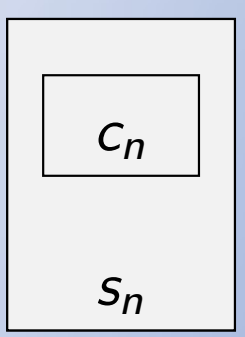
Repair scheme



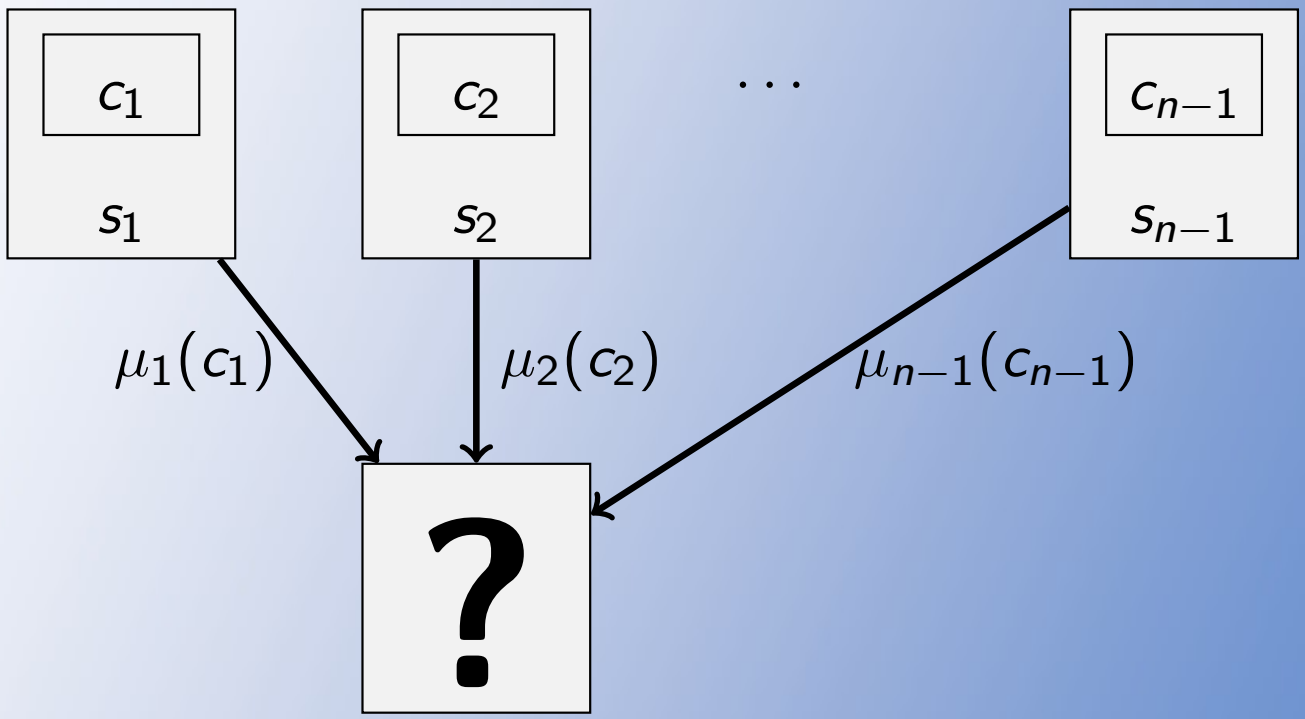
Repair scheme



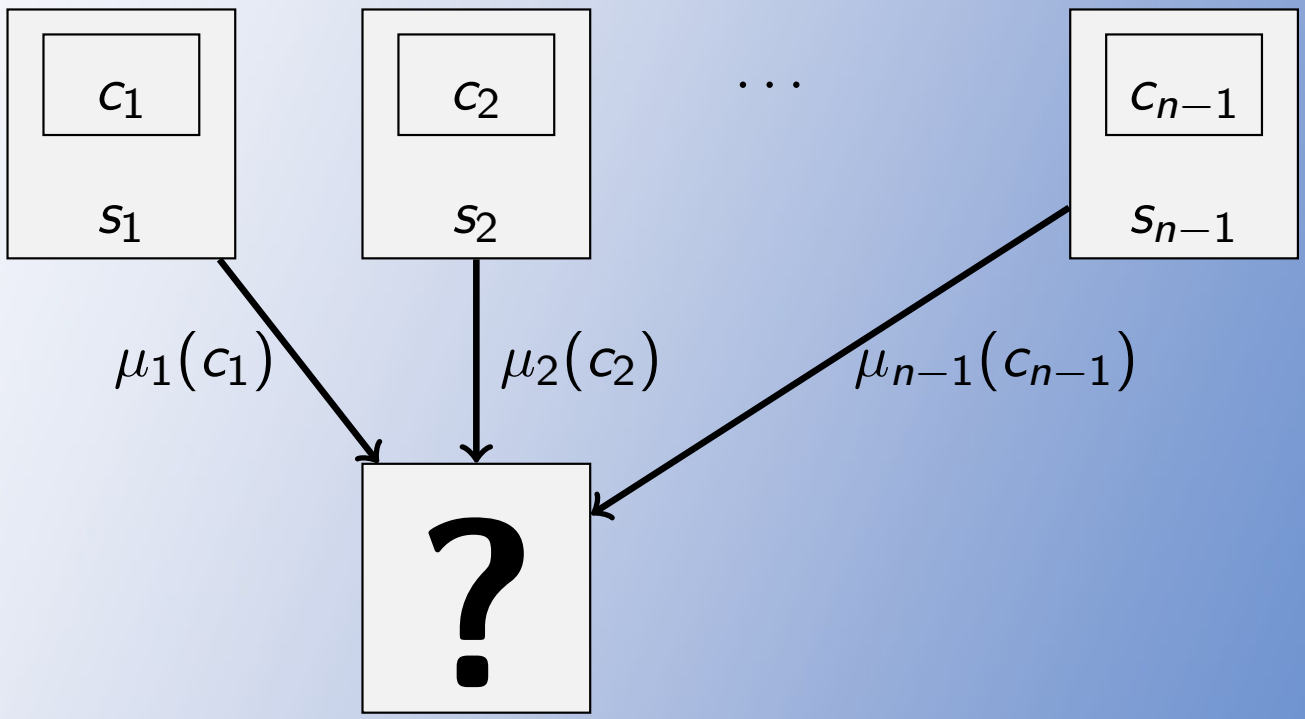
Replacement Node



Linear repair scheme

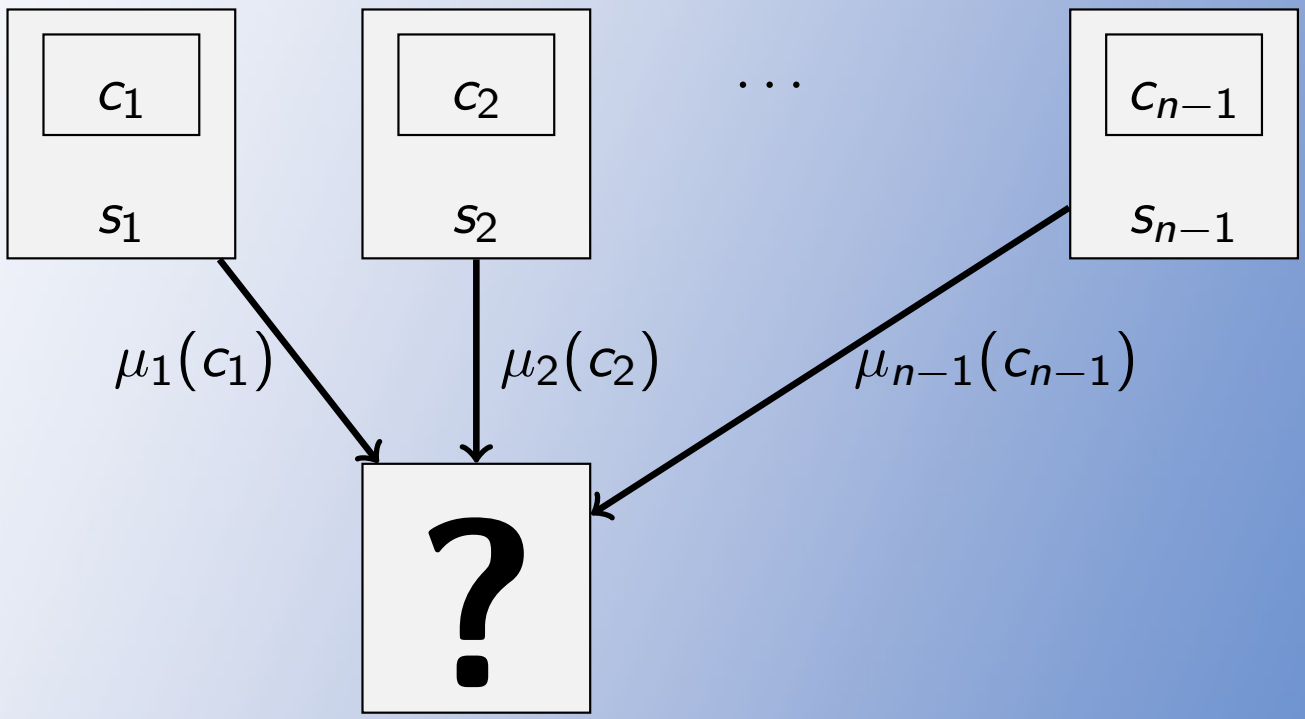


Linear repair scheme



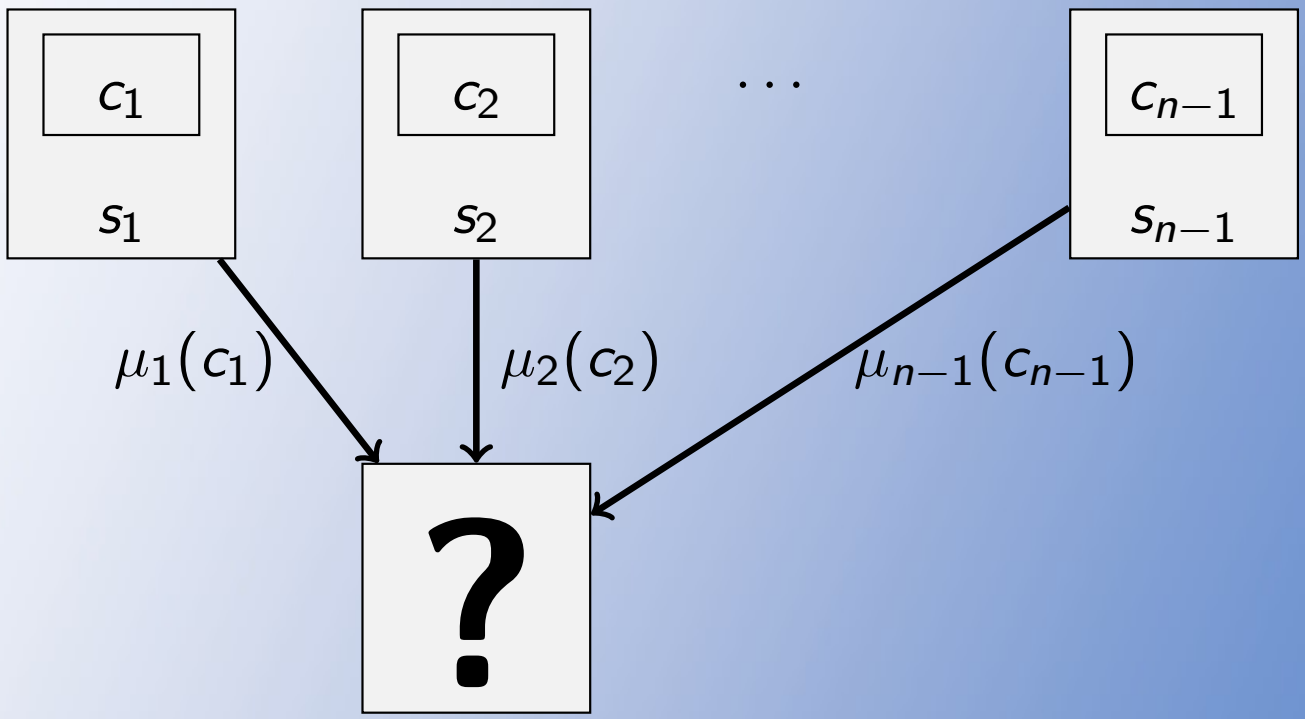
- Linear Repair - μ_i's are linear functions over base field

Linear repair scheme



- Linear Repair - μ_i's are linear functions over base field
  - $\mu_i : \mathbb{F}_{p^\ell} \to \mathbb{F}_p^m$

Linear repair scheme



- Linear Repair - μ_i's are linear functions over base field
  - $\mu_i : \mathbb{F}_{p^\ell} \to \mathbb{F}_p^m$
  - $\mu_i(\alpha x + \beta y) = \alpha\mu_i(\alpha) + \beta\mu_i(y), x, y \in \mathbb{F}_{p^\ell}, \alpha, \beta \in \mathbb{F}_p$

Cost of a repair scheme

Cost of a repair scheme

- Bandwidth - Total number of transmitted bits

Cost of a repair scheme

- Bandwidth - Total number of transmitted bits

$$\sum_{i=1}^{n-1} \log_2(|\,\mathrm{Image}(\mu_i)|)$$

The repair problem

The repair problem

- Design **MDS codes** with low **bandwidth repair schemes**.

The repair problem

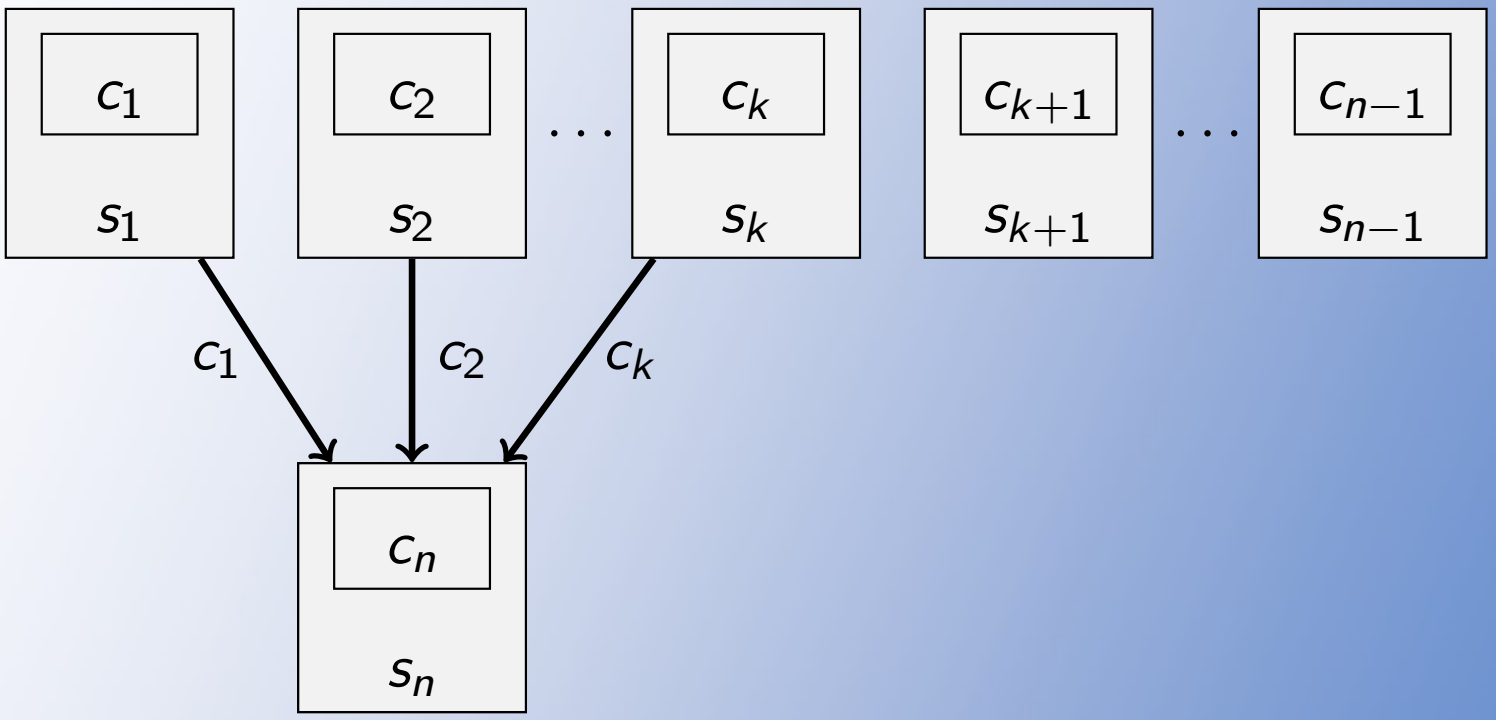
- Design **MDS codes** with low **bandwidth repair schemes**.
- Initiated by Dimakis et al. (2010)

The repair problem

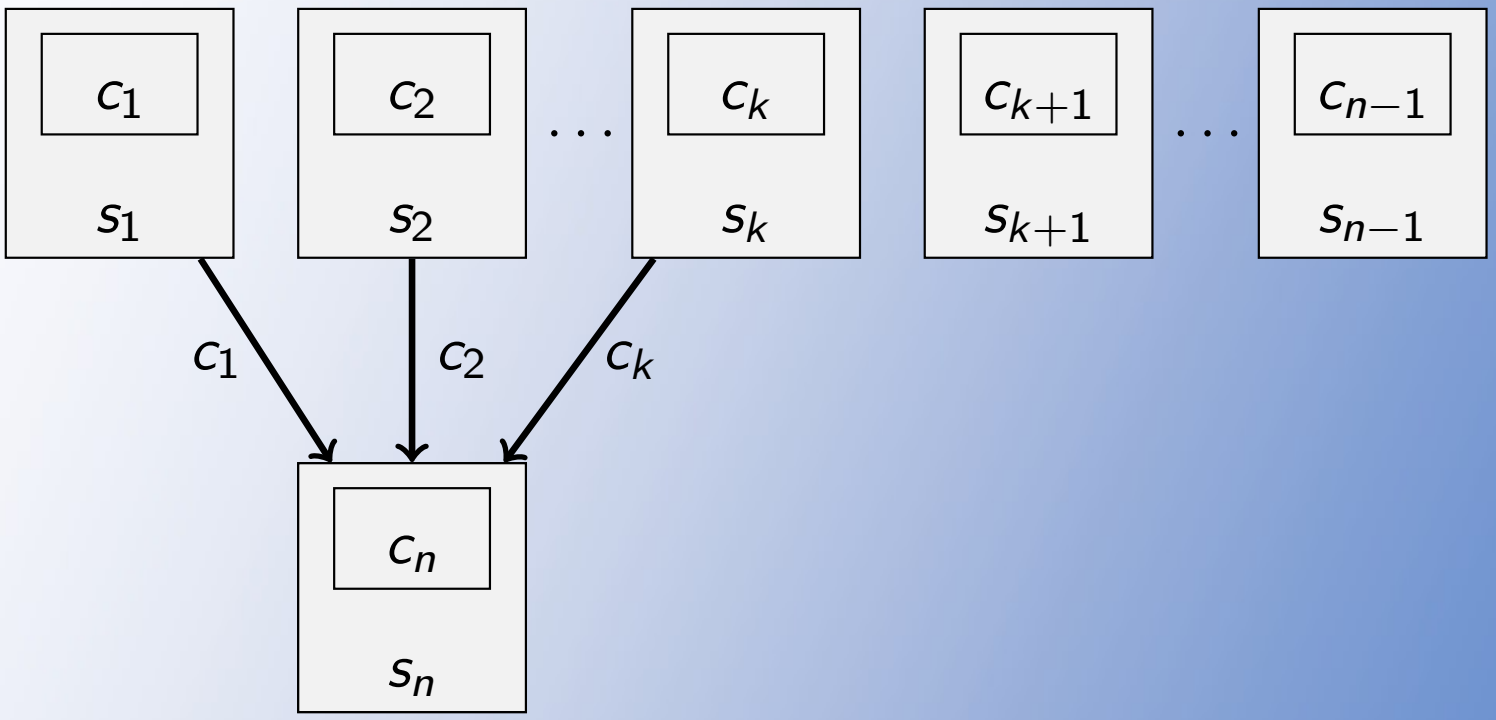
- Design **MDS codes** with low **bandwidth repair schemes**.
- Initiated by Dimakis et al. (2010)
- Important in large distributed storage systems: Google, Facebook, etc.

Trivial repair: MDS property

Trivial repair: MDS property

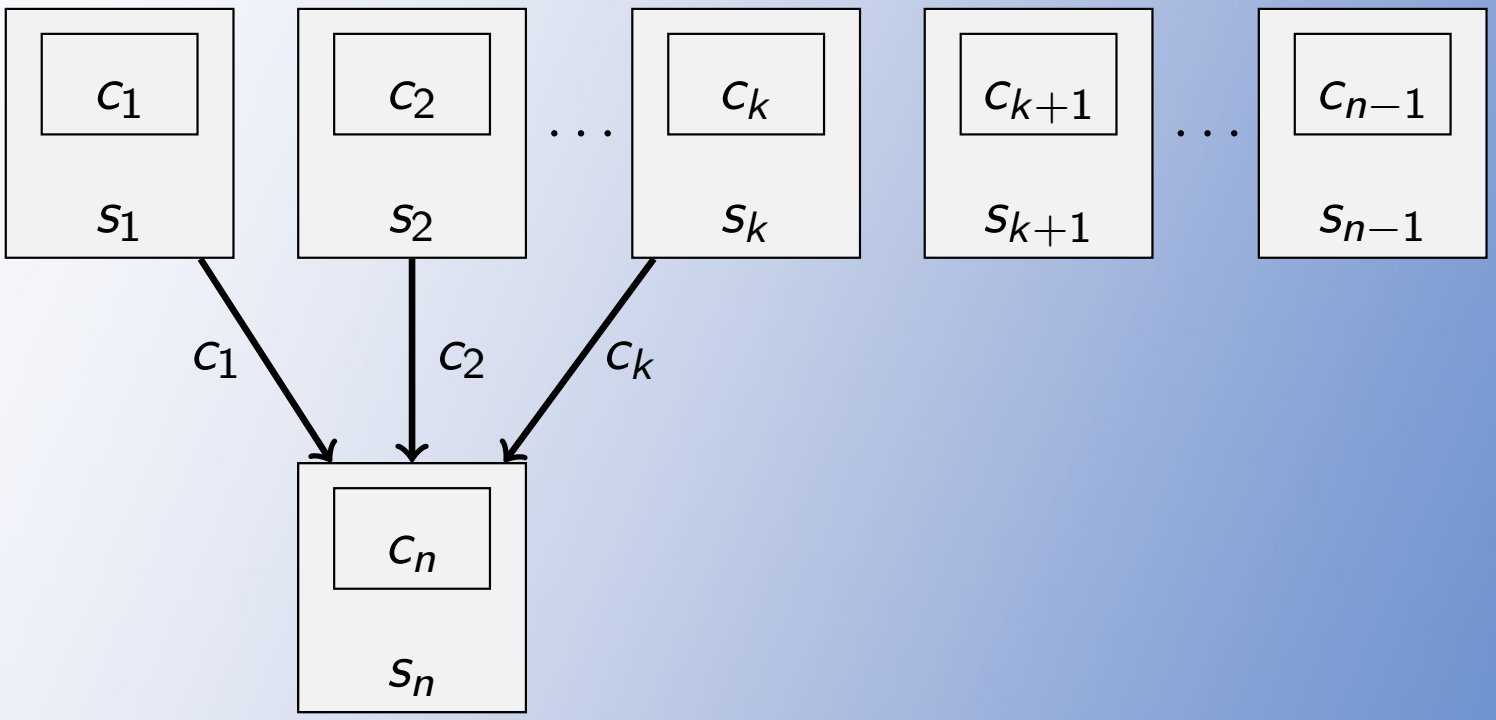


Trivial repair: MDS property



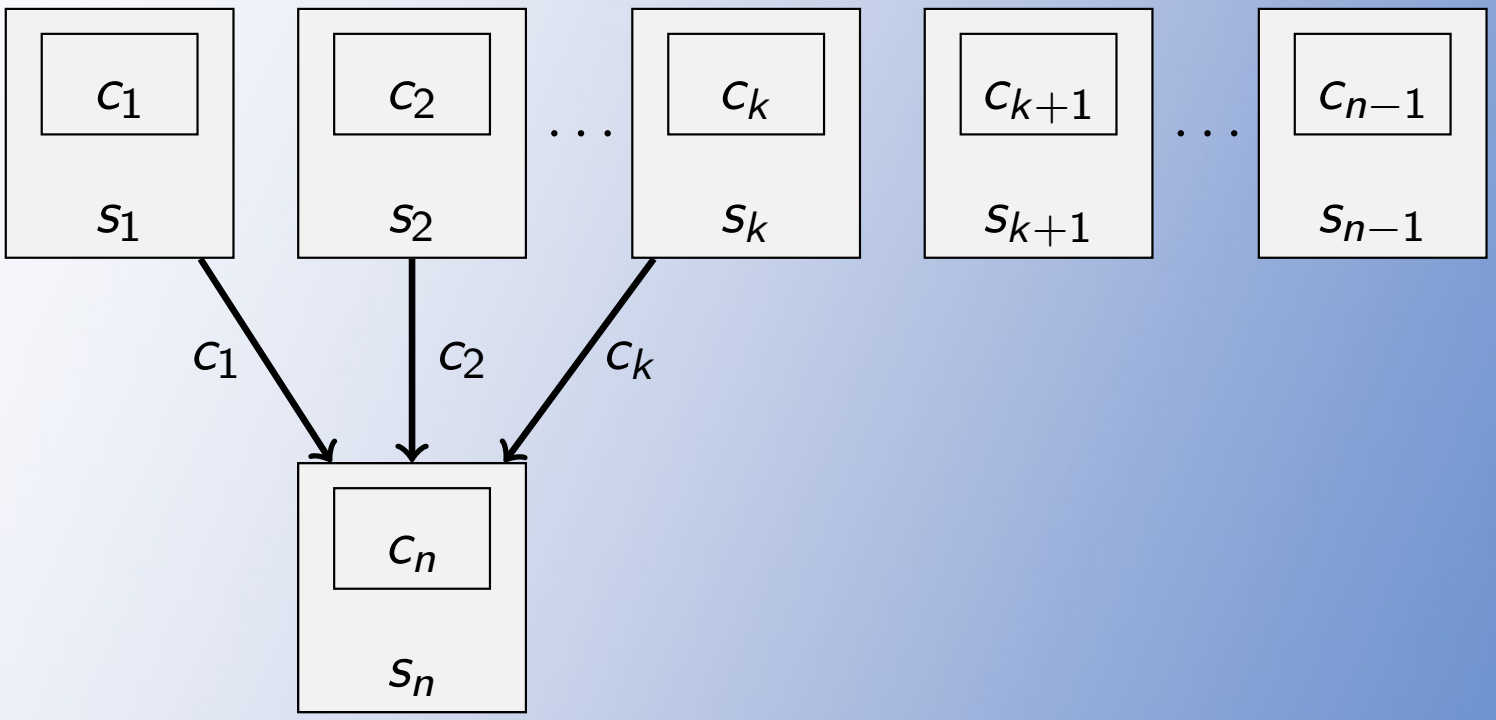
- Linear repair

Trivial repair: MDS property



- Linear repair
- Bandwidth: $k \cdot \log(q)$

Trivial repair: MDS property



- Linear repair
- Bandwidth: $k \cdot \log(q)$
- Optimal?

Cut-Set bound and MSR codes

Cut-Set bound and MSR codes

- d - number of helpers.

Cut-Set bound and MSR codes

- d - number of helpers. $k \leq d \leq n-1$

Cut-Set bound and MSR codes

- d - number of helpers. $k \leq d \leq n-1$
- Cut-Set bound:

$$\text{bandwidth} \geq \frac{d}{d-k+1} \log(q) \tag{1}$$

Cut-Set bound and MSR codes

- d - number of helpers. $k \leq d \leq n-1$
- Cut-Set bound:

$$\text{bandwidth} \geq \frac{d}{d-k+1} \log(q) \tag{1}$$

- Observations:

Cut-Set bound and MSR codes

- d - number of helpers. $k \leq d \leq n-1$
- Cut-Set bound:

$$\text{bandwidth} \geq \frac{d}{d-k+1} \log(q) \tag{1}$$

- Observations:Tight for $d = k$

Cut-Set bound and MSR codes

- d - number of helpers. $k \leq d \leq n-1$
- Cut-Set bound:

$$\text{bandwidth} \geq \frac{d}{d-k+1} \log(q) \tag{1}$$

- Observations:Tight for $d = k$; Increasing d is better

Cut-Set bound and MSR codes

- d - number of helpers. $k \leq d \leq n-1$
- Cut-Set bound:

$$\text{bandwidth} \geq \frac{d}{d-k+1} \log(q) \tag{1}$$

- Observations:Tight for $d = k$; Increasing d is better
- $[n, k, d]$-MSR code:

Cut-Set bound and MSR codes

- d - number of helpers. $k \leq d \leq n-1$
- Cut-Set bound:

$$\text{bandwidth} \geq \frac{d}{d-k+1}\log(q) \tag{1}$$

- Observations:Tight for $d = k$; Increasing d is better
- $[n, k, d]$-MSR code:
  1. $[n, k]$ MDS code

Cut-Set bound and MSR codes

- d - number of helpers. $k \leq d \leq n-1$
- Cut-Set bound:

$$\text{bandwidth} \geq \frac{d}{d-k+1} \log(q) \tag{1}$$

- Observations:Tight for $d = k$; Increasing d is better
- $[n, k, d]$-MSR code:
  1. $[n, k]$ MDS code
  2. Attains (1) for any failed node and d helpers

Cut-Set bound and MSR codes

- d - number of helpers. $k \leq d \leq n-1$
- Cut-Set bound:

$$\text{bandwidth} \geq \frac{d}{d-k+1}\log(q) \tag{1}$$

- Observations:Tight for $d = k$; Increasing d is better
- $[n, k, d]$-MSR code:
  1. $[n, k]$ MDS code
  2. Attains (1) for any failed node and d helpers
- Constructions of MSR codes are known: [YB17], [RSK11], [WTB16]

Reed-Solomon (RS) codes

Reed-Solomon (RS) codes

- $[n, k]$ RS code:

Reed-Solomon (RS) codes

- $[n, k]$ RS code: $\alpha_1, \ldots, \alpha_n \in \mathbb{F}$ distinct evaluation points

Reed-Solomon (RS) codes

- $[n, k]$ RS code: $\alpha_1, \ldots, \alpha_n \in \mathbb{F}$ distinct evaluation points

$$\{(f(\alpha_1), \ldots, f(\alpha_n)) \mid f \in \mathbb{F}[x], \deg(f) < k\}$$

Reed-Solomon (RS) codes

- $[n, k]$ RS code: $\alpha_1, \ldots, \alpha_n \in \mathbb{F}$ distinct evaluation points

$$\{(f(\alpha_1), \ldots, f(\alpha_n)) \mid f \in \mathbb{F}[x], \deg(f) < k\}$$

- RS codes are MDS codes

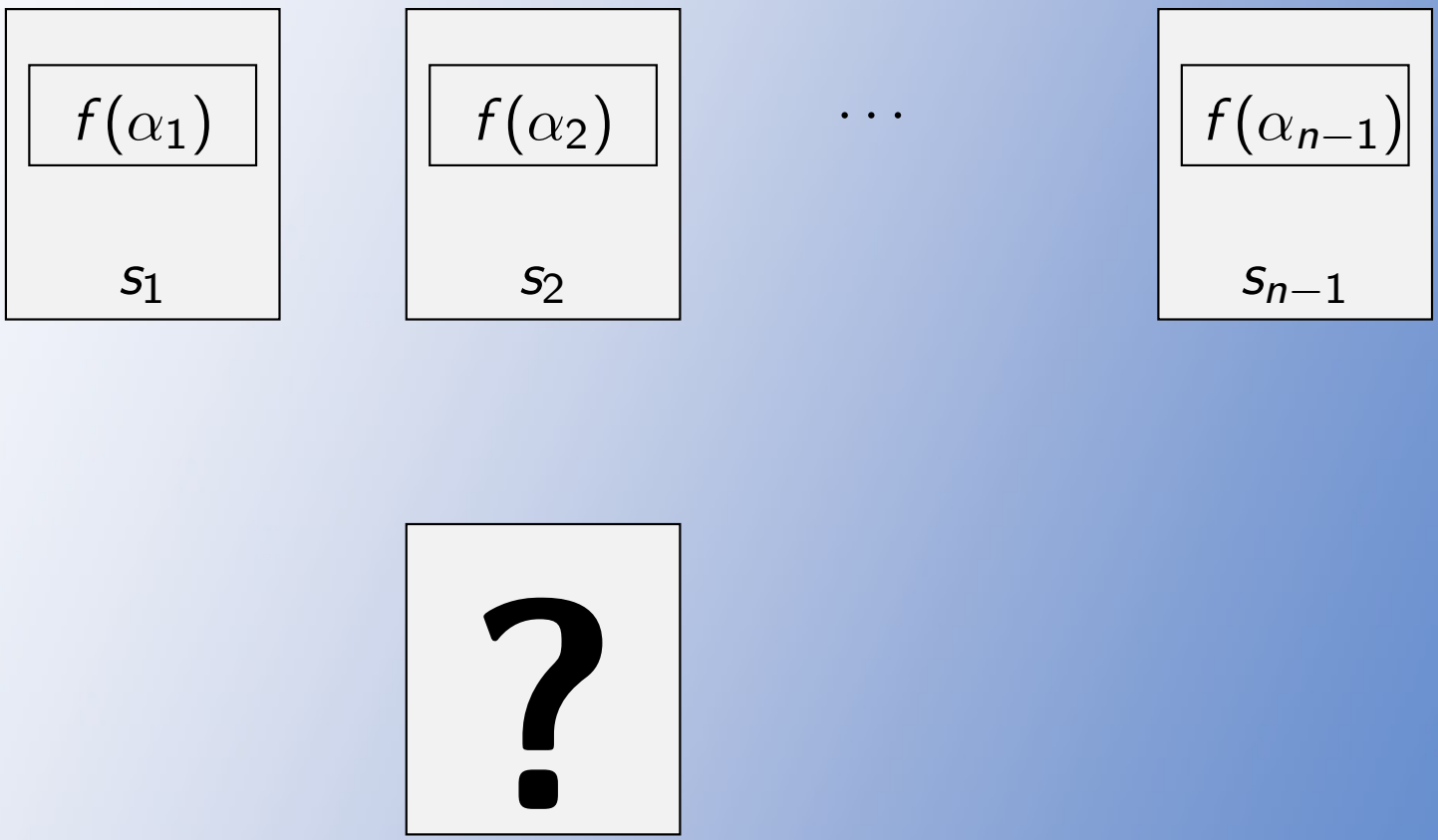
Reed-Solomon (RS) codes

- $[n, k]$ RS code: $\alpha_1, \ldots, \alpha_n \in \mathbb{F}$ distinct evaluation points

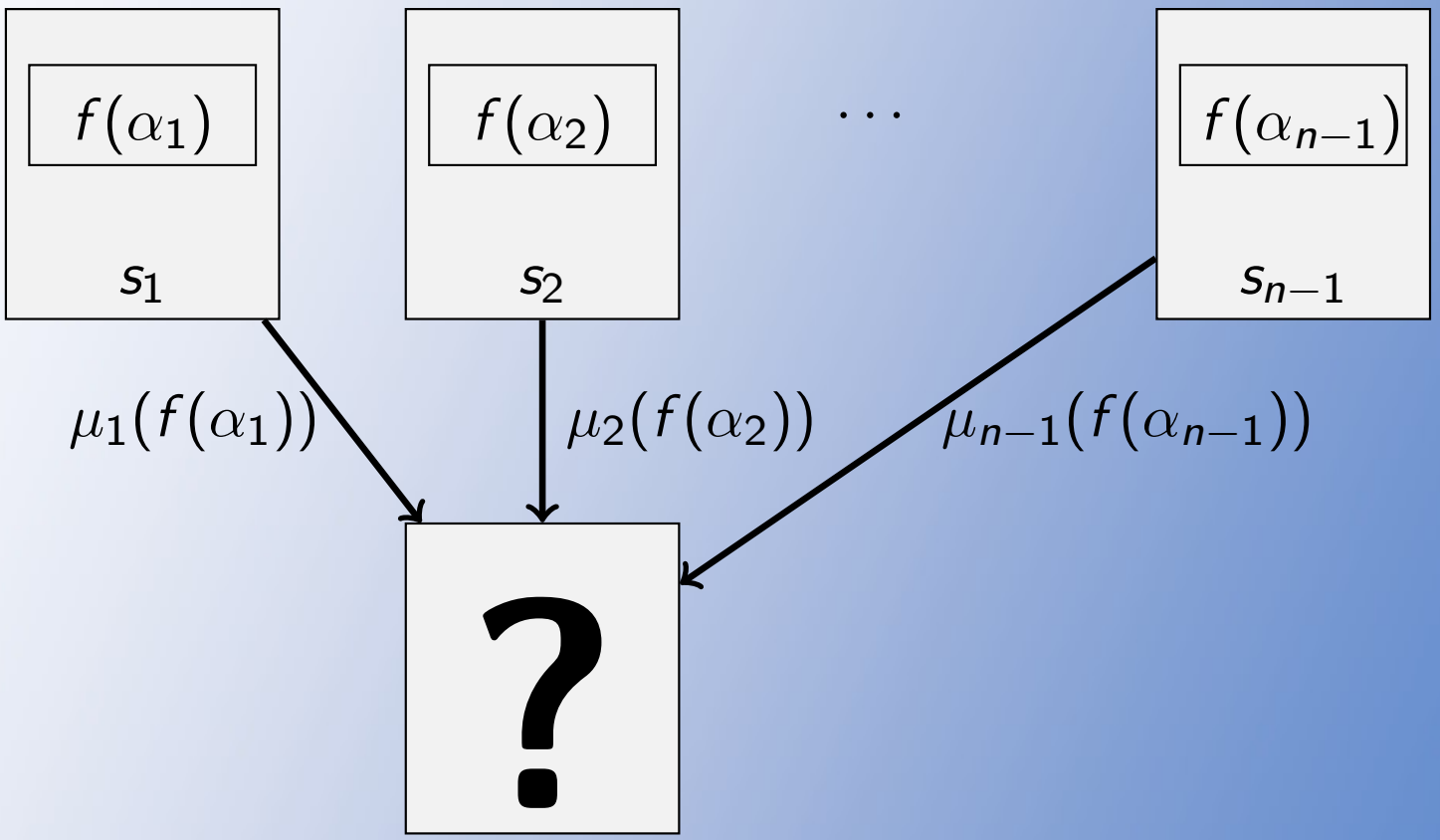
$$\{(f(\alpha_1), \ldots, f(\alpha_n)) \mid f \in \mathbb{F}[x], \deg(f) < k\}$$

- RS codes are MDS codes
- Important in theory and practice!

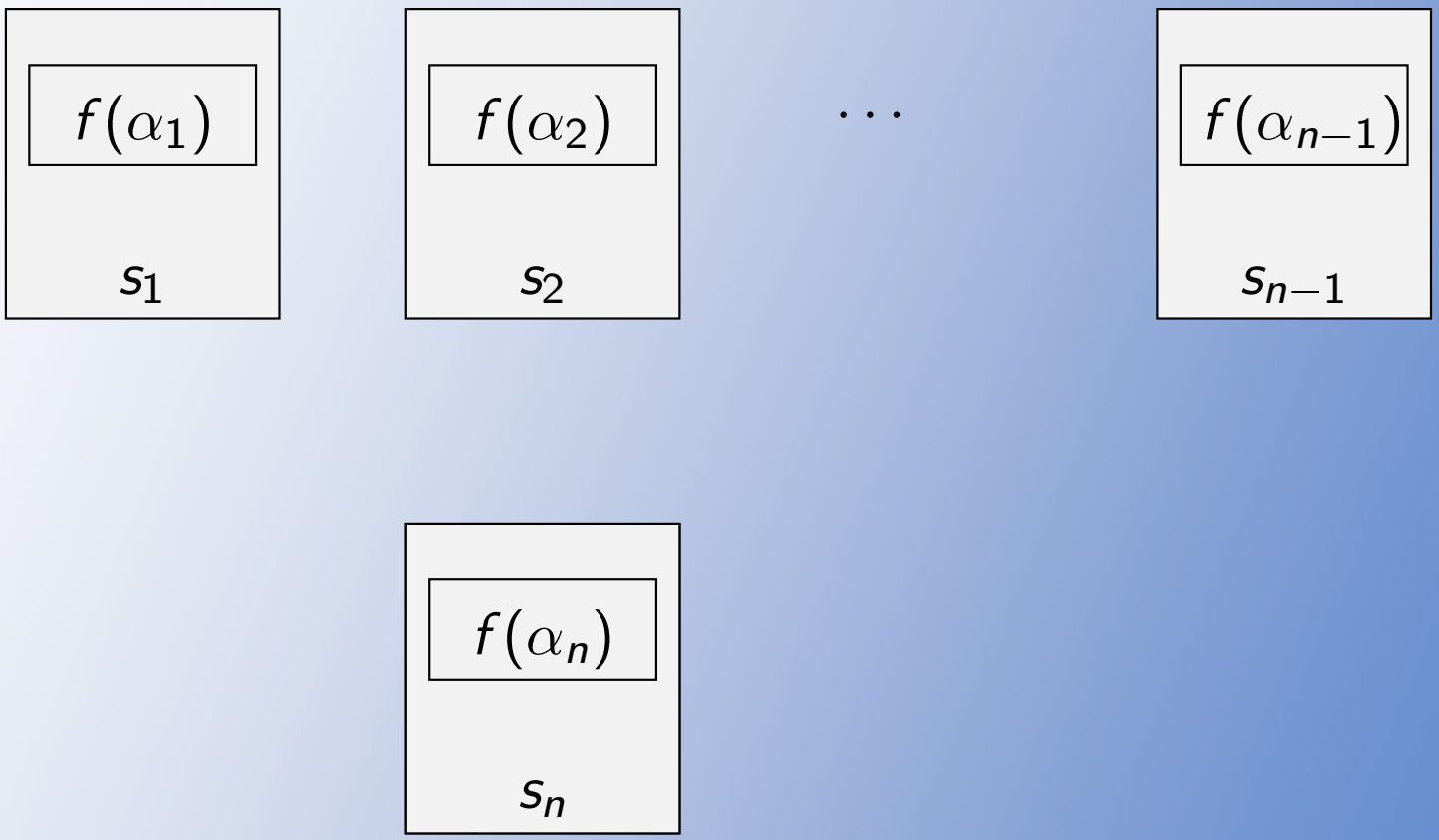
Repair of RS codes



Repair of RS codes



Repair of RS codes



RS codes - Known results

RS codes - Known results

- Non-trivial repair is possible [SPDC14]

RS codes - Known results

- Non-trivial repair is possible [SPDC14]
- Characterization of linear repair schemes of RS codes [GW17]

RS codes - Known results

- Non-trivial repair is possible [SPDC14]
- Characterization of linear repair schemes of RS codes [GW17]
  - Non-trivial linear repair

RS codes - Known results

- Non-trivial repair is possible [SPDC14]
- Characterization of linear repair schemes of RS codes [GW17]
  - Non-trivial linear repair $\rightarrow$ field extension.

RS codes - Known results

- Non-trivial repair is possible [SPDC14]
- Characterization of linear repair schemes of RS codes [GW17]
  - Non-trivial linear repair $\rightarrow$ field extension.
- Construction of $[n, k, d]$-MSR RS codes for $k < d < n$ [TYB17]

RS codes - Known results

- Non-trivial repair is possible [SPDC14]
- Characterization of linear repair schemes of RS codes [GW17]
  - Non-trivial linear repair $\rightarrow$ field extension.
- Construction of $[n, k, d]$-MSR RS codes for $k < d < n$ [TYB17]
  - Field size: $2^{n^n} \geq q$

RS codes - Known results

- Non-trivial repair is possible [SPDC14]
- Characterization of linear repair schemes of RS codes [GW17]
  - Non-trivial linear repair $\rightarrow$ field extension.
- Construction of $[n, k, d]$-MSR RS codes for $k < d < n$ [TYB17]
  - Field size: $2^{n^n} \geq q \geq 2^{k^k}$

RS codes - Known results

- Non-trivial repair is possible [SPDC14]
- Characterization of linear repair schemes of RS codes [GW17]
  - Non-trivial linear repair $\rightarrow$ field extension.
- Construction of $[n, k, d]$-MSR RS codes for $k < d < n$ [TYB17]
  - Field size: $2^{n^n} \geq q \geq 2^{k^k}$; Infeasible in practice

RS codes - Known results

- Non-trivial repair is possible [SPDC14]
- Characterization of linear repair schemes of RS codes [GW17]
  - Non-trivial linear repair $\rightarrow$ field extension.
- Construction of $[n, k, d]$-MSR RS codes for $k < d < n$ [TYB17]
  - Field size: $2^{n^n} \geq q \geq 2^{k^k}$; Infeasible in practice
- MSR codes + linear repair $\rightarrow$ very large alphabet size [AG19]

RS codes - Known results

- Non-trivial repair is possible [SPDC14]
- Characterization of linear repair schemes of RS codes [GW17]
  - Non-trivial linear repair $\rightarrow$ field extension.
- Construction of $[n, k, d]$-MSR RS codes for $k < d < n$ [TYB17]
  - Field size: $2^{n^n} \geq q \geq 2^{k^k}$; Infeasible in practice
- MSR codes + linear repair $\rightarrow$ very large alphabet size [AG19]
- Nonlinear repair scheme?

RS codes - Known results

- Non-trivial repair is possible [SPDC14]
- Characterization of linear repair schemes of RS codes [GW17]
  - Non-trivial linear repair $\rightarrow$ field extension.
- Construction of $[n, k, d]$-MSR RS codes for $k < d < n$ [TYB17]
  - Field size: $2^{n^n} \geq q \geq 2^{k^k}$; Infeasible in practice
- MSR codes + linear repair $\rightarrow$ very large alphabet size [AG19]
- Nonlinear repair scheme?
- Starting point: RS codes over prime fields

Connections to Shamir's secret sharing (scheme)

Connections to Shamir's secret sharing (scheme)

- Shamir's secret sharing (SSS)$\Longleftrightarrow$ RS code

Connections to Shamir's secret sharing (scheme)

- Shamir's secret sharing (SSS)$\Longleftrightarrow$ RS code
- Repairing RS code $\rightarrow$ vulnerability of SSS to local leakage

Connections to Shamir's secret sharing (scheme)

- Shamir's secret sharing (SSS)$\Longleftrightarrow$ RS code
- Repairing RS code $\rightarrow$ vulnerability of SSS to local leakage
- Is SSS resilient to local leakage?

Connections to Shamir's secret sharing (scheme)

- Shamir's secret sharing (SSS)$\Longleftrightarrow$ RS code
- Repairing RS code $\rightarrow$ vulnerability of SSS to local leakage
- Is SSS resilient to local leakage?
- Initiated by Benhamouda et al. 2019

Connections to Shamir's secret sharing (scheme)

- Shamir's secret sharing (SSS)$\Longleftrightarrow$ RS code
- Repairing RS code $\rightarrow$ vulnerability of SSS to local leakage
- Is SSS resilient to local leakage?
- Initiated by Benhamouda et al. 2019
- Main message: For some parameters adversary learns nothing

Connections to Shamir's secret sharing (scheme)

- Shamir's secret sharing (SSS)$\Longleftrightarrow$ RS code
- Repairing RS code $\rightarrow$ vulnerability of SSS to local leakage
- Is SSS resilient to local leakage?
- Initiated by Benhamouda et al. 2019
- Main message: For some parameters adversary learns nothing
- Conjecture: $(n, \alpha n)$-SSS is 1-bit local leakage resilient for any constant $\alpha > 0$ and $n \approx p$

Our results

Our results

- n constant, p grows

Our results

- n constant, p grows

Theorem

Almost all $[n, 2]_p$ *RS codes are* **asymptotically** $[n, 2, d]$ *MSR codes*

Our results

- n constant, p grows

Theorem

Almost all $[n, 2]_p$ *RS codes are* **asymptotically** $[n, 2, d]$ *MSR codes*

$$bandwidth = \frac{d}{d-1}\log(p) + O_n(1)$$

Our results

- n constant, p grows

Theorem

Almost all $[n, 2]_p$ *RS codes are* **asymptotically** $[n, 2, d]$ *MSR codes*

$$bandwidth = \frac{d}{d-1}\log(p) + O_n(1) \approx \log(p)$$

for large d

Our results

- n constant, p grows

Theorem

Almost all $[n, 2]_p$ *RS codes are* **asymptotically** $[n, 2, d]$ *MSR codes*

$$bandwidth = \frac{d}{d-1}\log(p) + O_n(1) \approx \log(p)$$

for large d

- Don't recover the polynomial

Our results - continued

Our results - continued

Theorem

Explicit construction of $[n, k]_p$ *RS code,* $k < d \leq n/2$ *with*

Our results - continued

Theorem

Explicit construction of $[n, k]_p$ RS code, $k < d \leq n/2$ with

1. *Asymptotically optimal bandwidth:*

Our results - continued

Theorem

Explicit construction of $[n, k]_p$ *RS code,* $k < d \leq n/2$ *with*

1. *Asymptotically optimal bandwidth:* $\frac{d}{d-k+1}\log(p) + O_n(1)$

Our results - continued

Theorem

Explicit construction of $[n, k]_p$ *RS code,* $k < d \leq n/2$ *with*

1. *Asymptotically optimal bandwidth:* $\frac{d}{d-k+1}\log(p) + O_n(1)$
2. *Many* d*-subsets*

Our results - continued

Theorem

Explicit construction of $[n,k]_p$ RS code, $k < d \leq n/2$ with

1. *Asymptotically optimal bandwidth:* $\frac{d}{d-k+1}\log(p) + O_n(1)$
2. *Many d-subsets*

Theorem

Full length $[p,k]_p$ RS code (for large p) and $d > k$ with

Our results - continued

Theorem

Explicit construction of $[n, k]_p$ *RS code,* $k < d \leq n/2$ *with*

1. *Asymptotically optimal bandwidth:* $\frac{d}{d-k+1}\log(p) + O_n(1)$
2. *Many* d*-subsets*

Theorem

Full length $[p, k]_p$ *RS code (for large* p*) and* $d > k$ *with*

- $\Omega(p)$ *repair sets of size* d *for each symbol*

Our results - continued

Theorem

Explicit construction of $[n,k]_p$ RS code, $k < d \leq n/2$ with

1. *Asymptotically optimal bandwidth:* $\frac{d}{d-k+1}\log(p) + O_n(1)$
2. *Many d-subsets*

Theorem

Full length $[p,k]_p$ RS code (for large p) and $d > k$ with

- $\Omega(p)$ *repair sets of size d for each symbol*
- *Asymptotically optimal bandwidth:*

Our results - continued

Theorem

Explicit construction of $[n,k]_p$ *RS code,* $k < d \leq n/2$ *with*

1. *Asymptotically optimal bandwidth:* $\frac{d}{d-k+1}\log(p) + O_n(1)$
2. *Many* d*-subsets*

Theorem

Full length $[p,k]_p$ *RS code (for large* p*) and* $d > k$ *with*

- $\Omega(p)$ *repair sets of size* d *for each symbol*
- *Asymptotically optimal bandwidth:* $\frac{d}{d-k+1}\log(p) + O_{d,k}(1)$

Our results - continued

Our results - continued

- Bounds on exponential sums $\rightarrow$ repair schemes

Our results - continued

- Bounds on exponential sums $\rightarrow$ repair schemes

Theorem (Repair scheme via Kloosterman sum)

Our results - continued

- Bounds on exponential sums $\rightarrow$ repair schemes

Theorem (Repair scheme via Kloosterman sum)

Explicit $[n, 3]_p$ *RS code, where*

Our results - continued

- Bounds on exponential sums $\rightarrow$ repair schemes

Theorem (Repair scheme via Kloosterman sum)

Explicit $[n, 3]_p$ *RS code, where*

- $\exp(\ln^{2/3}(p)) \leq n \leq \sqrt{p}$

Our results - continued

- Bounds on exponential sums $\rightarrow$ repair schemes

Theorem (Repair scheme via Kloosterman sum)

Explicit $[n, 3]_p$ *RS code, where*

- $\exp(\ln^{2/3}(p)) \leq n \leq \sqrt{p}$
- *Repair scheme by downloading* 3 *bits from each surviving node*

Our results - continued

Theorem (Improved cut-set bound)

Our results - continued

Theorem (Improved cut-set bound)

Repairing an $[n, k]_p$ *RS with* d *helpers requires at least*

$$\frac{d}{d-k+1}(\log_2(p) + \log_2(k) - 1) \text{ bits}$$

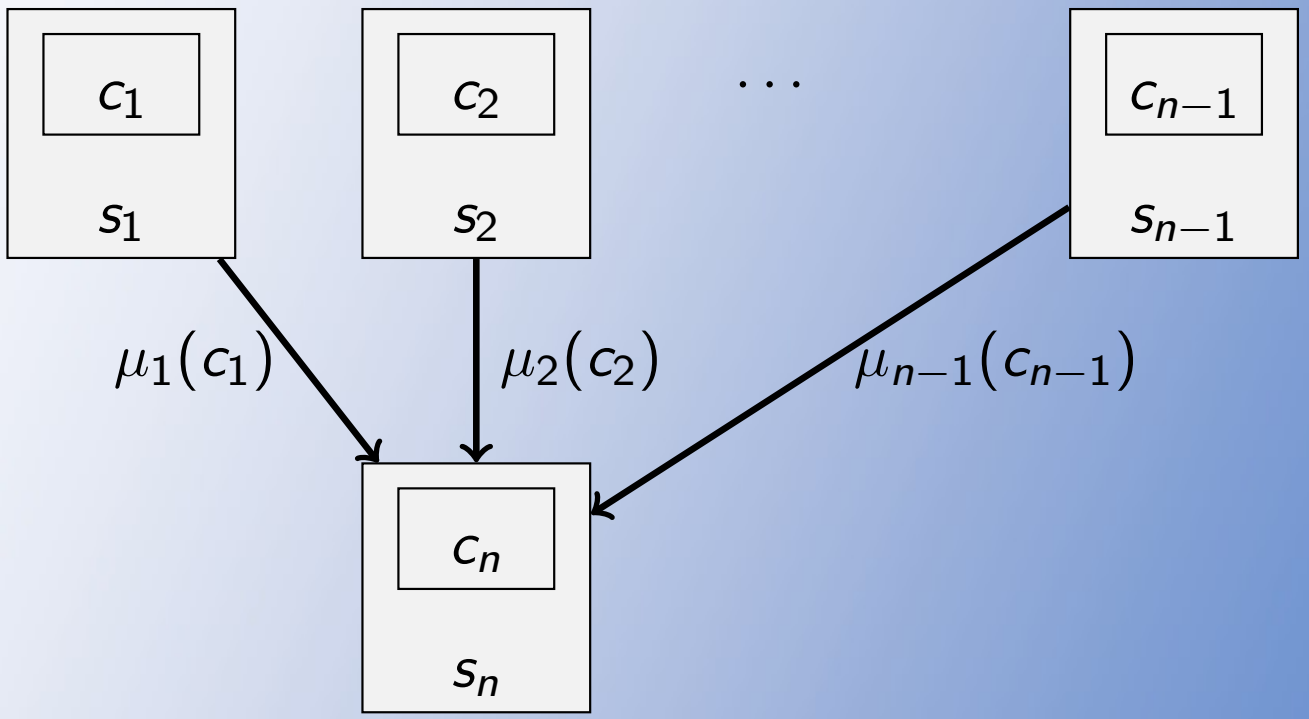
Non-linear repair - *n*th node

Non-linear repair - nth node

- Repair functions $\mu_i : \mathbb{F}_p \to A$

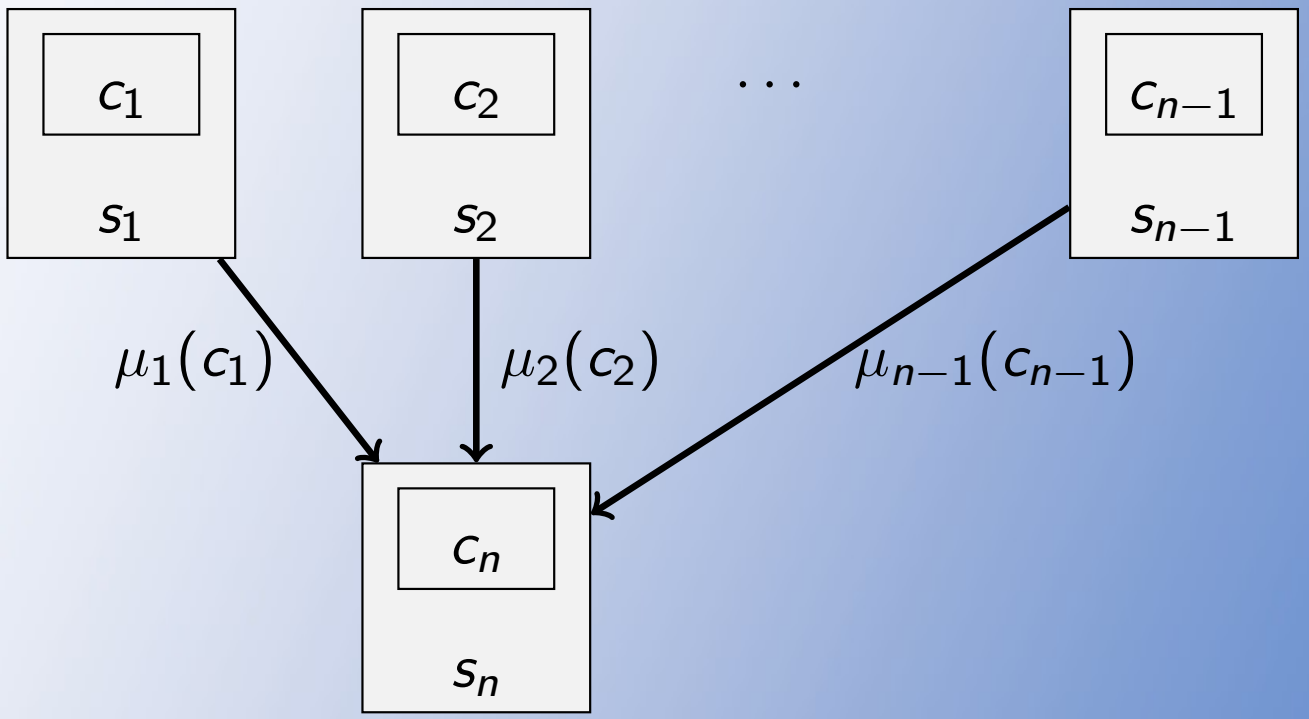
Non-linear repair - nth node

- Repair functions $\mu_i : \mathbb{F}_p \to A$



Non-linear repair - nth node

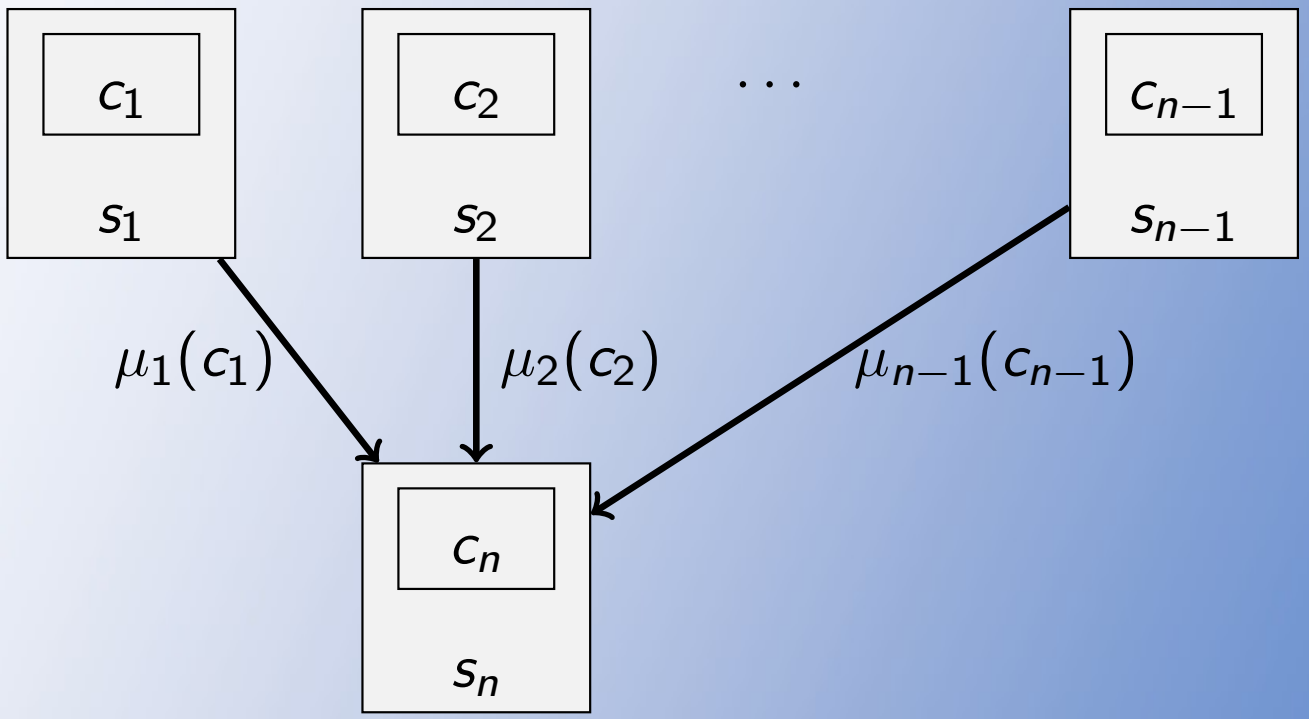
- Repair functions $\mu_i : \mathbb{F}_p \to A$



- Function $\Longleftrightarrow$ Partition of $\mathbb{F}_p$

Non-linear repair - nth node

- Repair functions $\mu_i : \mathbb{F}_p \to A$

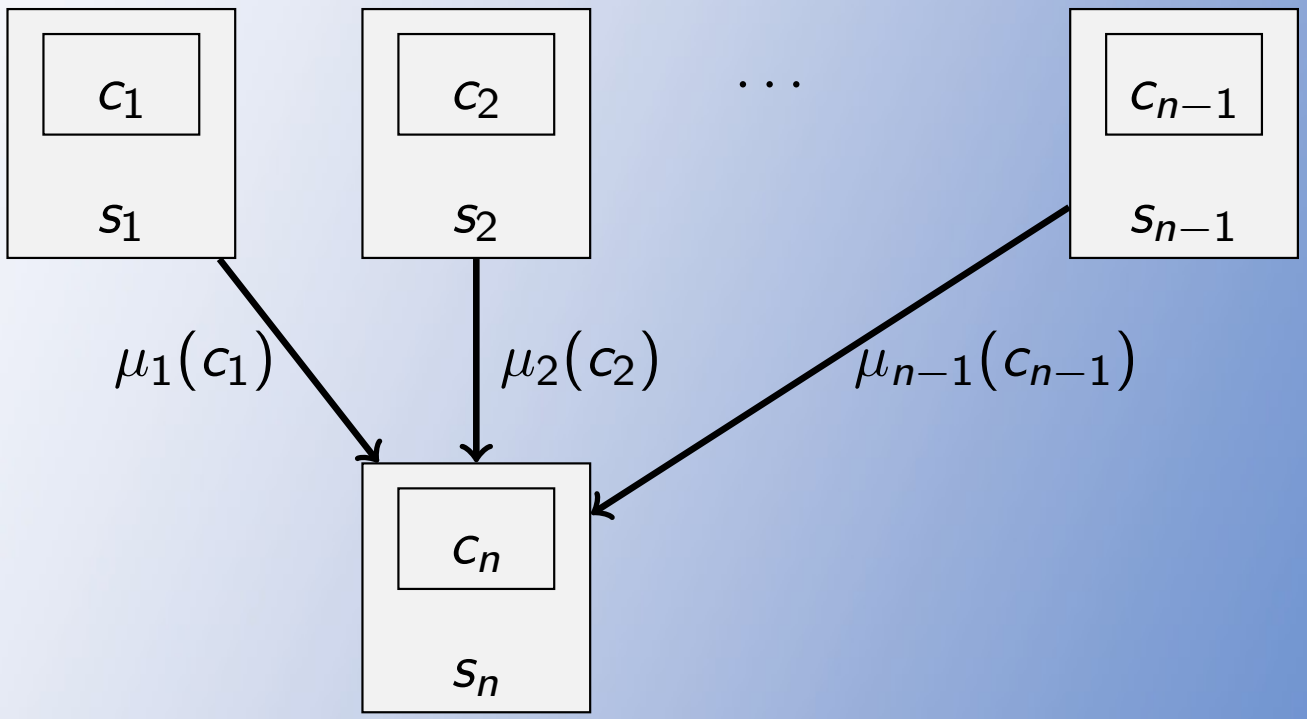


- Function $\iff$ Partition of $\mathbb{F}_p$

$$\{\mu_i^{-1}(a) : a \in A\}$$

Non-linear repair - nth node

- Repair functions $\mu_i : \mathbb{F}_p \to A$



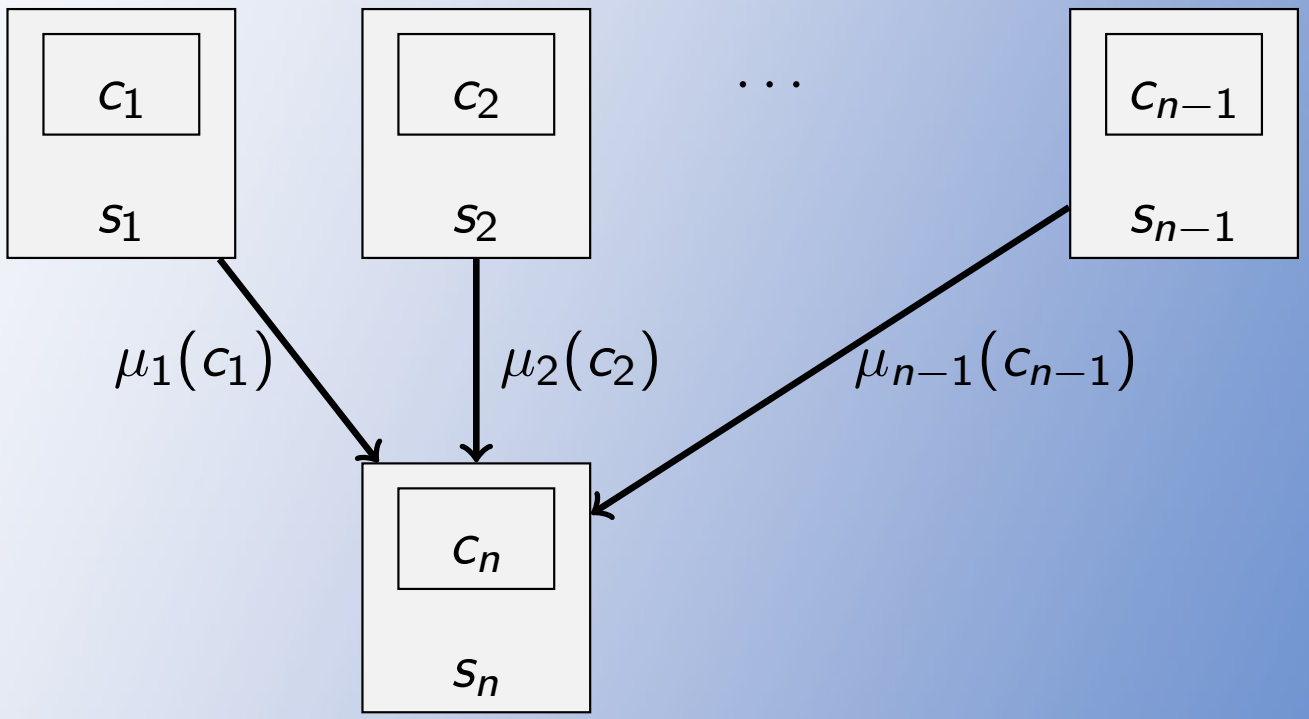
- Function $\Longleftrightarrow$ Partition of $\mathbb{F}_p$

$$\{\mu_i^{-1}(a) : a \in A\}$$

- Which partitions?

Non-linear repair - nth node

- Repair functions $\mu_i : \mathbb{F}_p \to A$



- Function $\Longleftrightarrow$ Partition of $\mathbb{F}_p$

$$\{\mu_i^{-1}(a) : a \in A\}$$

- Which partitions?
- A_j^i - jth set of ith partition

Repair scheme - condition

Repair scheme - condition

- Successful repair $\Longleftrightarrow$

Repair scheme - condition

- Successful repair $\Longleftrightarrow$ For $c, c' \in C$ s.t.

$$\mu_i(c_i) = \mu_i(c'_i), i = 1, \ldots, n-1,$$

Repair scheme - condition

- Successful repair $\Longleftrightarrow$ For $c, c' \in C$ s.t.

$$\mu_i(c_i) = \mu_i(c'_i), i = 1, \ldots, n-1,$$

then $c_n = c'_n$

Repair scheme - condition

- Successful repair $\Longleftrightarrow$ For $c, c' \in C$ s.t.

$$\mu_i(c_i) = \mu_i(c'_i), i = 1, \ldots, n-1,$$

then $c_n = c'_n$

- RS codes: $f(x), g(x)$ polynomials of degree $< k$

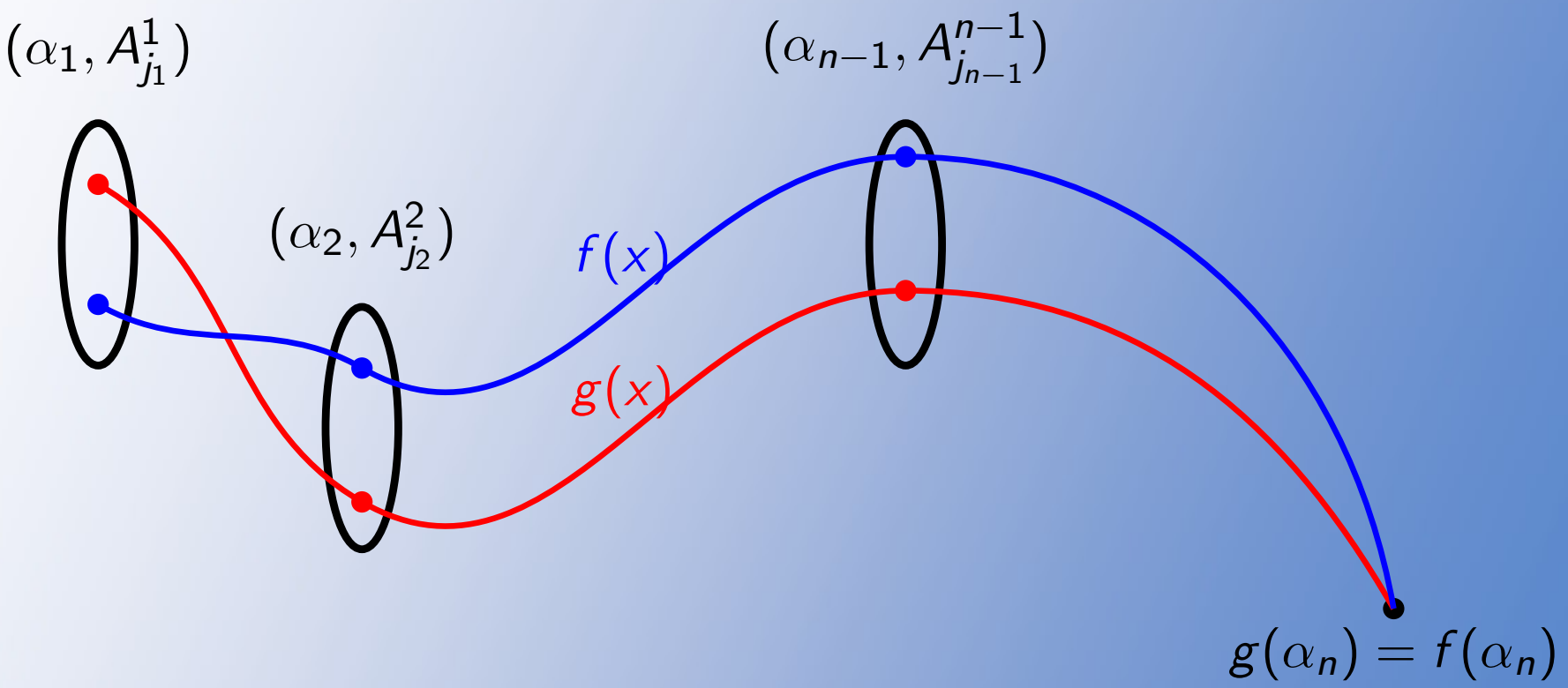
Repair scheme - condition

- Successful repair $\iff$ For $c, c' \in C$ s.t.

$$\mu_i(c_i) = \mu_i(c'_i), i = 1, \ldots, n-1,$$

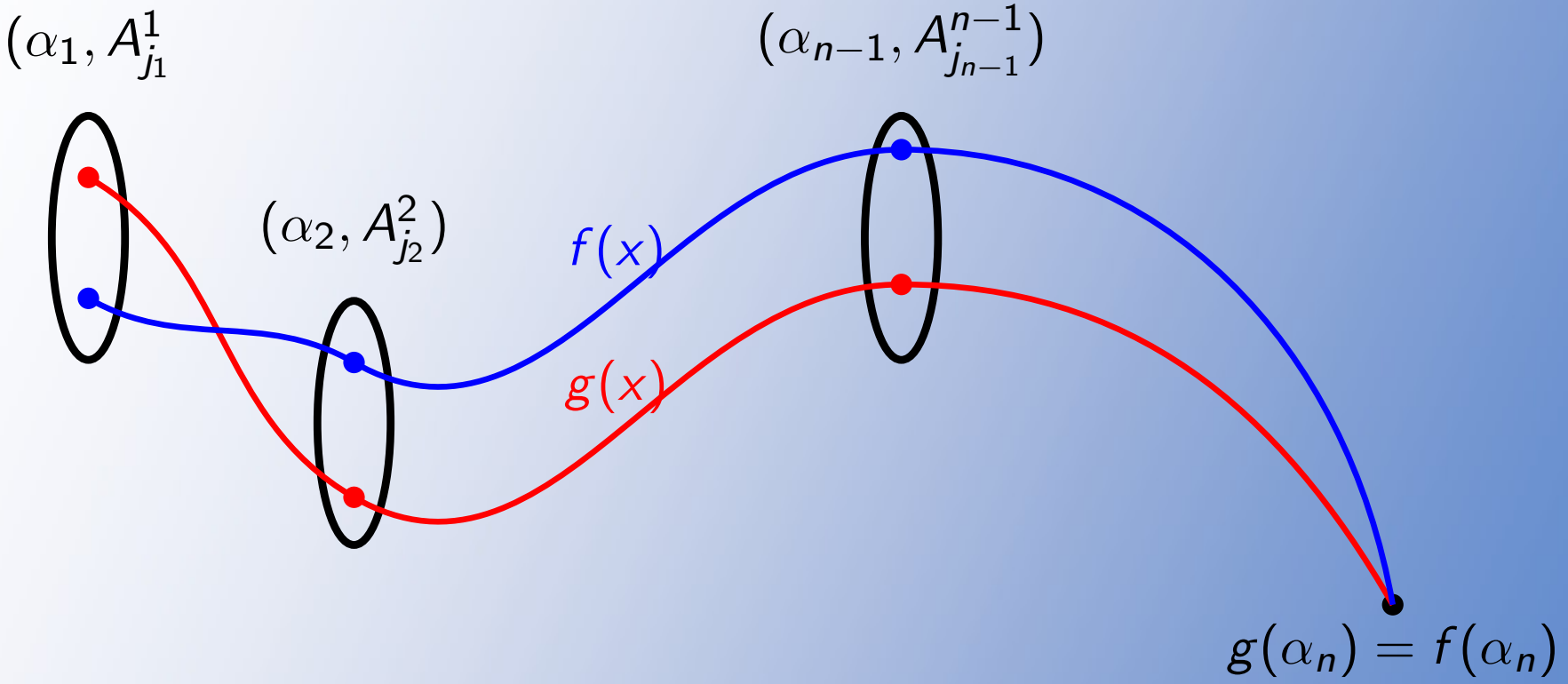
  then $c_n = c'_n$

- RS codes: $f(x), g(x)$ polynomials of degree $< k$

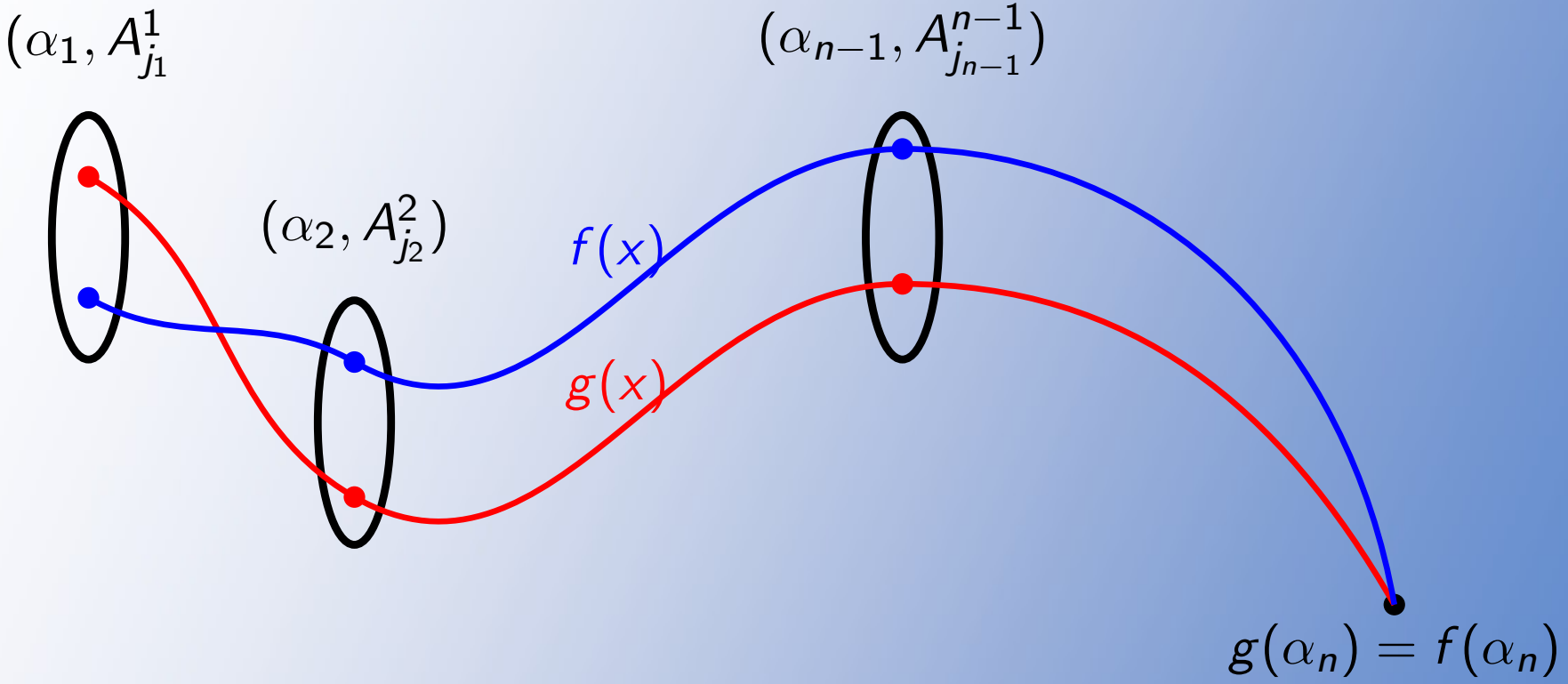


Repair scheme - condition

Repair scheme - condition

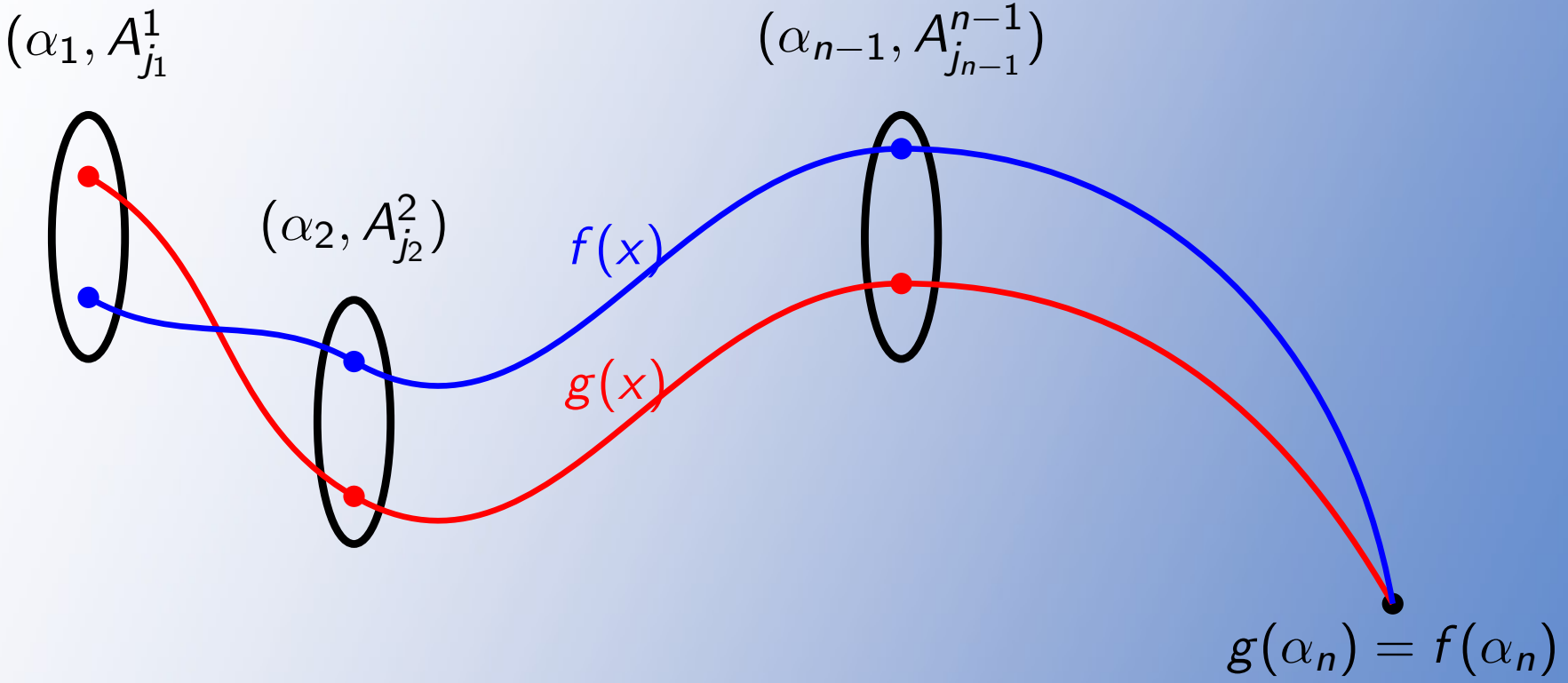


Repair scheme - condition



- Sufficient condition:

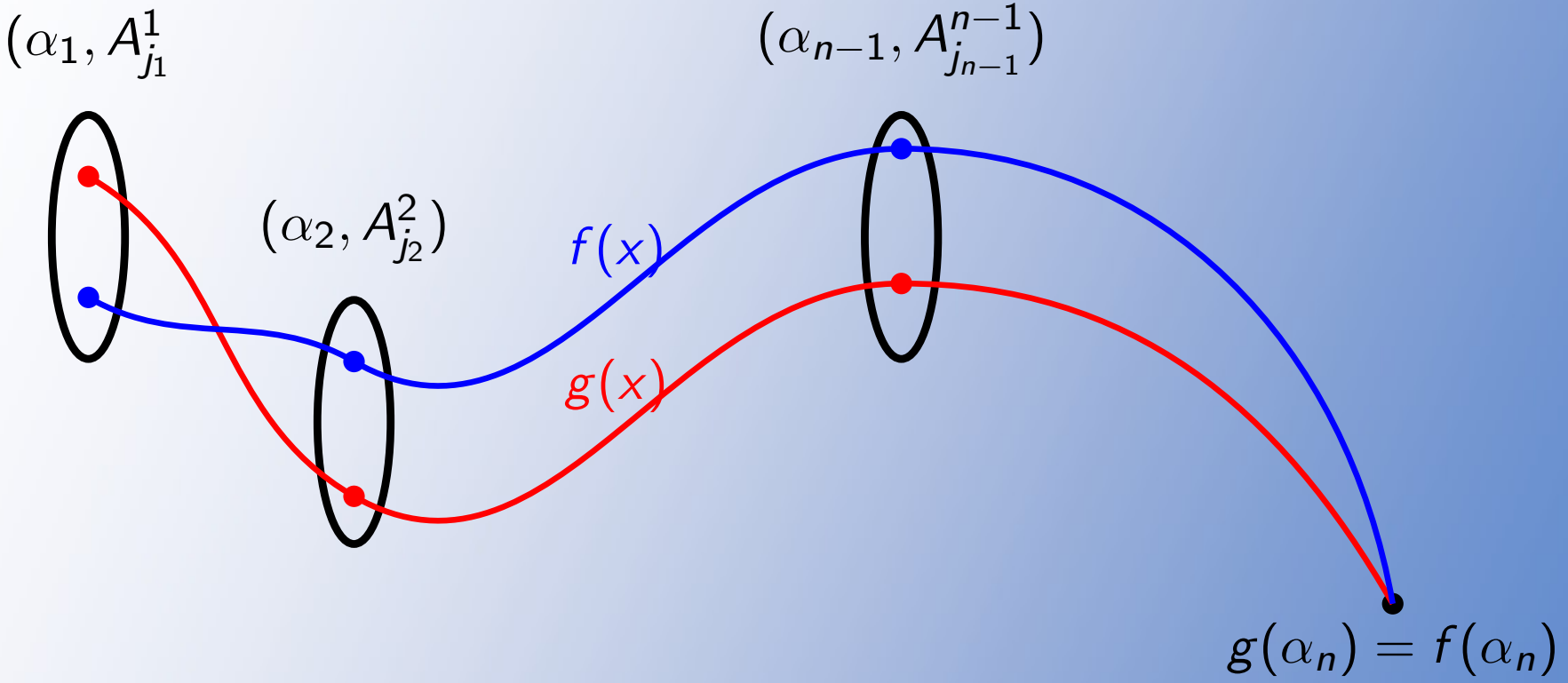
Repair scheme - condition



- Sufficient condition: For any polynomial $h(x), \deg(h) < k$, and sets A^i_j, if

$$h(\alpha_i) \in A^i_j - A^i_j, i = 1, \ldots, n-1$$

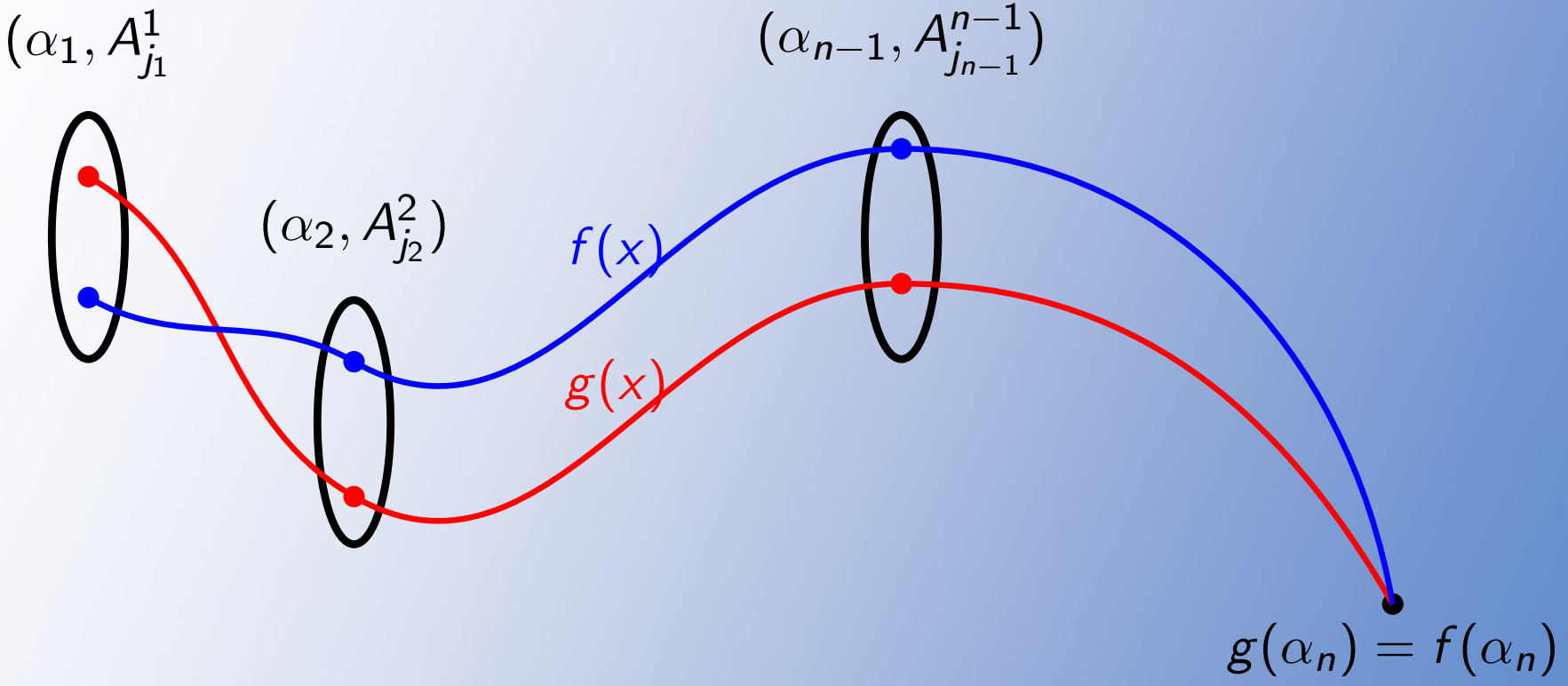
Repair scheme - condition



- Sufficient condition: For any polynomial $h(x), \deg(h) < k$, and sets A^i_j, if

$$h(\alpha_i) \in A^i_j - A^i_j, i = 1, \ldots, n-1 \rightarrow h(\alpha_n) = 0$$

Repair scheme - condition



- Sufficient condition: For any polynomial $h(x), \deg(h) < k$, and sets A^i_j, if

$$h(\alpha_i) \in A^i_j - A^i_j, i = 1, \ldots, n-1 \rightarrow h(\alpha_n) = 0$$

- $A - B = \{a - b : a \in A, b \in B\}$

Repair scheme - condition

- Sufficient condition: For any polynomial $h(x), \deg(h) < k$, and sets A_j^i, if

$$h(\alpha_i) \in A_j^i - A_j^i, i = 1, \ldots, n-1 \rightarrow h(\alpha_n) = 0$$

Repair scheme - condition

- Sufficient condition: For any polynomial $h(x), \deg(h) < k$, and sets A_j^i, if

$$h(\alpha_i) \in A_j^i - A_j^i, i = 1, \ldots, n-1 \rightarrow h(\alpha_n) = 0$$

Proposition

Sufficient condition is indeed sufficient

Repair scheme - condition

- Sufficient condition: For any polynomial $h(x), \deg(h) < k$, and sets A_j^i, if

$$h(\alpha_i) \in A_j^i - A_j^i, i = 1, \ldots, n-1 \rightarrow h(\alpha_n) = 0$$

Proposition

Sufficient condition is indeed sufficient

Proof.

Repair scheme - condition

- Sufficient condition: For any polynomial $h(x), \deg(h) < k$, and sets A_j^i, if

$$h(\alpha_i) \in A_j^i - A_j^i, i = 1, \ldots, n-1 \rightarrow h(\alpha_n) = 0$$

Proposition

Sufficient condition is indeed sufficient

Proof.

- Let $f(x), g(x)$ polynomials of deg $< k$

Repair scheme - condition

- Sufficient condition: For any polynomial $h(x), \deg(h) < k$, and sets A_j^i, if

$$h(\alpha_i) \in A_j^i - A_j^i, i = 1, \ldots, n-1 \rightarrow h(\alpha_n) = 0$$

Proposition

Sufficient condition is indeed sufficient

Proof.

- Let $f(x), g(x)$ polynomials of deg $< k$
- Assume $f(\alpha_i), g(\alpha_i) \in A_{j_i}^i$ for $i = 1, \ldots, n-1$

Repair scheme - condition

- Sufficient condition: For any polynomial $h(x), \deg(h) < k$, and sets A_j^i, if

$$h(\alpha_i) \in A_j^i - A_j^i, i = 1, \ldots, n-1 \rightarrow h(\alpha_n) = 0$$

Proposition

Sufficient condition is indeed sufficient

Proof.

- Let $f(x), g(x)$ polynomials of deg $< k$
- Assume $f(\alpha_i), g(\alpha_i) \in A_{j_i}^i$ for $i = 1, \ldots, n-1$
- $h := f - g$

Repair scheme - condition

- Sufficient condition: For any polynomial $h(x), \deg(h) < k$, and sets A_j^i, if

$$h(\alpha_i) \in A_j^i - A_j^i, i = 1, \ldots, n-1 \rightarrow h(\alpha_n) = 0$$

Proposition

Sufficient condition is indeed sufficient

Proof.

- Let $f(x), g(x)$ polynomials of deg $< k$
- Assume $f(\alpha_i), g(\alpha_i) \in A_{j_i}^i$ for $i = 1, \ldots, n-1$
- $h := f - g$ satisfies $h(\alpha_i) \in A_{j_i}^i - A_{j_i}^i$ for $i = 1, \ldots, n-1$

Repair scheme - condition

- Sufficient condition: For any polynomial $h(x), \deg(h) < k$, and sets A_j^i, if

$$h(\alpha_i) \in A_j^i - A_j^i, i = 1, \ldots, n-1 \rightarrow h(\alpha_n) = 0$$

Proposition

Sufficient condition is indeed sufficient

Proof.

- Let $f(x), g(x)$ polynomials of deg $< k$
- Assume $f(\alpha_i), g(\alpha_i) \in A_{j_i}^i$ for $i = 1, \ldots, n-1$
- $h := f - g$ satisfies $h(\alpha_i) \in A_{j_i}^i - A_{j_i}^i$ for $i = 1, \ldots, n-1$
- $\rightarrow 0 = h(\alpha_n)$

Repair scheme - condition

- Sufficient condition: For any polynomial $h(x), \deg(h) < k$, and sets A_j^i, if

$$h(\alpha_i) \in A_j^i - A_j^i, i = 1, \ldots, n-1 \rightarrow h(\alpha_n) = 0$$

Proposition

Sufficient condition is indeed sufficient

Proof.

- Let $f(x), g(x)$ polynomials of deg $< k$
- Assume $f(\alpha_i), g(\alpha_i) \in A_{j_i}^i$ for $i = 1, \ldots, n-1$
- $h := f - g$ satisfies $h(\alpha_i) \in A_{j_i}^i - A_{j_i}^i$ for $i = 1, \ldots, n-1$
- $\rightarrow 0 = h(\alpha_n) = f(\alpha_n) - g(\alpha_n)$

Which partitions?

- Sufficient condition: For any polynomial $h(x), \deg(h) < k$, and sets A_j^i, if

$$h(\alpha_i) \in A_j^i - A_j^i, i = 1, \ldots, n-1 \rightarrow h(\alpha_n) = 0$$

Which partitions?

- Sufficient condition: For any polynomial $h(x), \deg(h) < k$, and sets A_j^i, if

$$h(\alpha_i) \in A_j^i - A_j^i, i = 1, \ldots, n-1 \rightarrow h(\alpha_n) = 0$$

- Random partitions

Which partitions?

- Sufficient condition: For any polynomial $h(x), \deg(h) < k$, and sets A_j^i, if

$$h(\alpha_i) \in A_j^i - A_j^i, i = 1, \ldots, n-1 \rightarrow h(\alpha_n) = 0$$

- Random partitions $\rightarrow |A_j^i - A_j^i| \approx |A_j^i|^2$ w.h.p.

Which partitions?

- Sufficient condition: For any polynomial $h(x), \deg(h) < k$, and sets A_j^i, if

$$h(\alpha_i) \in A_j^i - A_j^i, i = 1, \ldots, n-1 \rightarrow h(\alpha_n) = 0$$

- Random partitions $\rightarrow |A_j^i - A_j^i| \approx |A_j^i|^2$ w.h.p. $\rightarrow$ Bad idea

Which partitions?

- Sufficient condition: For any polynomial $h(x), \deg(h) < k$, and sets A_j^i, if

$$h(\alpha_i) \in A_j^i - A_j^i, i = 1, \ldots, n-1 \rightarrow h(\alpha_n) = 0$$

- Random partitions $\rightarrow |A_j^i - A_j^i| \approx |A_j^i|^2$ w.h.p. $\rightarrow$ Bad idea
- Need:

Which partitions?

- Sufficient condition: For any polynomial $h(x), \deg(h) < k$, and sets A_j^i, if

$$h(\alpha_i) \in A_j^i - A_j^i, i = 1, \ldots, n-1 \rightarrow h(\alpha_n) = 0$$

- Random partitions $\rightarrow |A_j^i - A_j^i| \approx |A_j^i|^2$ w.h.p. $\rightarrow$ Bad idea
- Need: $|A_j^i - A_j^i|$ small for any A_j^i (small expansion)

Which partitions?

- Sufficient condition: For any polynomial $h(x), \deg(h) < k$, and sets A_j^i, if

$$h(\alpha_i) \in A_j^i - A_j^i, i = 1, \ldots, n-1 \rightarrow h(\alpha_n) = 0$$

- Random partitions $\rightarrow |A_j^i - A_j^i| \approx |A_j^i|^2$ w.h.p. $\rightarrow$ Bad idea
- Need: $|A_j^i - A_j^i|$ small for any A_j^i (small expansion)
- Solution:

Which partitions?

- Sufficient condition: For any polynomial $h(x), \deg(h) < k$, and sets A_j^i, if

$$h(\alpha_i) \in A_j^i - A_j^i, i = 1, \ldots, n-1 \rightarrow h(\alpha_n) = 0$$

- Random partitions $\rightarrow |A_j^i - A_j^i| \approx |A_j^i|^2$ w.h.p. $\rightarrow$ Bad idea
- Need: $|A_j^i - A_j^i|$ small for any A_j^i (small expansion)
- Solution: A_j^i's partition $\mathbb{F}_p$ into arithmetic progressions

Which partitions? - continued

Which partitions? - continued

- $\mathcal{A} = \{A_0, \ldots, A_{\lfloor p/t \rfloor}\}$ a partition of $\mathbb{F}_p$ to arithmetic progressions of length t

Which partitions? - continued

- $\mathcal{A} = \{A_0, \ldots, A_{\lfloor p/t \rfloor}\}$ a partition of $\mathbb{F}_p$ to arithmetic progressions of length t
  1. $A_0 = \{0, 1, \ldots, t-1\}$

Which partitions? - continued

- $\mathcal{A} = \{A_0, \ldots, A_{\lfloor p/t \rfloor}\}$ a partition of $\mathbb{F}_p$ to arithmetic progressions of length t
  1. $A_0 = \{0, 1, \ldots, t-1\}$
  2. $A_{i+1} = A_i + t$

Which partitions? - continued

- $\mathcal{A} = \{A_0, \ldots, A_{\lfloor p/t \rfloor}\}$ a partition of $\mathbb{F}_p$ to arithmetic progressions of length t
  1. $A_0 = \{0, 1, \ldots, t-1\}$
  2. $A_{i+1} = A_i + t$ (except $A_{\lfloor p/t \rfloor}$)

Which partitions? - continued

- $\mathcal{A} = \{A_0, \ldots, A_{\lfloor p/t \rfloor}\}$ a partition of $\mathbb{F}_p$ to arithmetic progressions of length t
  1. $A_0 = \{0, 1, \ldots, t-1\}$
  2. $A_{i+1} = A_i + t$ (except $A_{\lfloor p/t \rfloor}$)
- Same partition for each node?

Which partitions? - continued

- $\mathcal{A} = \{A_0, \ldots, A_{\lfloor p/t \rfloor}\}$ a partition of $\mathbb{F}_p$ to arithmetic progressions of length t
  1. $A_0 = \{0, 1, \ldots, t-1\}$
  2. $A_{i+1} = A_i + t$ (except $A_{\lfloor p/t \rfloor}$)
- Same partition for each node? Let $A_j^i = A_j$

Which partitions? - continued

- $\mathcal{A} = \{A_0, \ldots, A_{\lfloor p/t \rfloor}\}$ a partition of $\mathbb{F}_p$ to arithmetic progressions of length t
  1. $A_0 = \{0, 1, \ldots, t-1\}$
  2. $A_{i+1} = A_i + t$ (except $A_{\lfloor p/t \rfloor}$)
- Same partition for each node? Let $A_j^i = A_j$
- Sufficient condition: For any polynomial $h(x), \deg(h) < k$, and sets A_j^i, if

$$h(\alpha_i) \in A_j^i - A_j^i = [-(t-1), t-1], i = 1, \ldots, n-1 \rightarrow h(\alpha_n) = 0$$

Which partitions? - continued

- $\mathcal{A} = \{A_0, \ldots, A_{\lfloor p/t \rfloor}\}$ a partition of $\mathbb{F}_p$ to arithmetic progressions of length t
  1. $A_0 = \{0, 1, \ldots, t-1\}$
  2. $A_{i+1} = A_i + t$ (except $A_{\lfloor p/t \rfloor}$)
- Same partition for each node? Let $A^i_j = A_j$
- Sufficient condition: For any polynomial $h(x), \deg(h) < k$, and sets A^i_j, if

$$h(\alpha_i) \in A^i_j - A^i_j = [-(t-1), t-1], i = 1, \ldots, n-1 \rightarrow h(\alpha_n) = 0$$

- Different partitions (functions)

nth node repair scheme

nth node repair scheme

- $\mathcal{A} = \{A_0, \ldots, A_{\lfloor p/t \rfloor}\}$ partitions $\mathbb{F}_p$ to arithm. progr.

nth node repair scheme

- $\mathcal{A} = \{A_0, \ldots, A_{\lfloor p/t \rfloor}\}$ partitions $\mathbb{F}_p$ to arithm. progr.
- $\gamma\mathcal{A} = \{\gamma \cdot A_0, \ldots, \gamma \cdot A_{\lfloor p/t \rfloor}\}$ partitions $\mathbb{F}_p$ to arithm. progr. $(\gamma \neq 0)$

nth node repair scheme

- $\mathcal{A} = \{A_0, \ldots, A_{\lfloor p/t \rfloor}\}$ partitions $\mathbb{F}_p$ to arithm. progr.
- $\gamma\mathcal{A} = \{\gamma \cdot A_0, \ldots, \gamma \cdot A_{\lfloor p/t \rfloor}\}$ partitions $\mathbb{F}_p$ to arithm. progr. $(\gamma \neq 0)$
- $\gamma_1, \ldots, \gamma_{n-1} \in \mathbb{F}_p^*$.

nth node repair scheme

- $\mathcal{A} = \{A_0, \ldots, A_{\lfloor p/t \rfloor}\}$ partitions $\mathbb{F}_p$ to arithm. progr.
- $\gamma\mathcal{A} = \{\gamma \cdot A_0, \ldots, \gamma \cdot A_{\lfloor p/t \rfloor}\}$ partitions $\mathbb{F}_p$ to arithm. progr. $(\gamma \neq 0)$
- $\gamma_1, \ldots, \gamma_{n-1} \in \mathbb{F}_p^*$.

c_1	c_2	$\ldots$	c_{n-1}
$\gamma_1 A_0$	$\gamma_2 A_0$	$\ldots$	$\gamma_{n-1} A_0$
$\gamma_1 A_1$	$\gamma_2 A_1$		$\gamma_{n-1} A_1$
$\gamma_1 A_2$	$\gamma_2 A_2$		$\gamma_{n-1} A_2$
$\vdots$	$\vdots$		$\vdots$
$\gamma_1 A_{\lfloor p/t \rfloor}$	$\gamma_2 A_{\lfloor p/t \rfloor}$	$\ldots$	$\gamma_{n-1} A_{\lfloor p/t \rfloor}$

nth node repair scheme - continued

nth node repair scheme - continued

- Node i transmits set index that contains c_i

nth node repair scheme - continued

- Node i transmits set index that contains c_i

c_1	c_2	$\cdots$	c_{n-1}
$c_1 \in \gamma_1 A_0$	$\gamma_2 A_0$	$\cdots$	$\gamma_{n-1} A_0$
$\gamma_1 A_1$	$\gamma_2 A_1$		$\gamma_{n-1} A_1$
$\gamma_1 A_2$	$c_2 \in \gamma_2 A_2$		$\gamma_{n-1} A_2$
$\vdots$	$\vdots$		$\vdots$
$\gamma_1 A_{\lfloor p/t \rfloor}$	$\gamma_2 A_{\lfloor p/t \rfloor}$	$\cdots$	$c_{n-1} \in \gamma_{n-1} A_{\lfloor p/t \rfloor}$

nth node repair scheme - continued

- Node i transmits set index that contains c_i

c_1	c_2	$\cdots$	c_{n-1}
$c_1 \in \gamma_1 A_0$	$\gamma_2 A_0$	$\cdots$	$\gamma_{n-1} A_0$
$\gamma_1 A_1$	$\gamma_2 A_1$		$\gamma_{n-1} A_1$
$\gamma_1 A_2$	$c_2 \in \gamma_2 A_2$		$\gamma_{n-1} A_2$
$\vdots$	$\vdots$		$\vdots$
$\gamma_1 A_{\lfloor p/t \rfloor}$	$\gamma_2 A_{\lfloor p/t \rfloor}$	$\cdots$	$c_{n-1} \in \gamma_{n-1} A_{\lfloor p/t \rfloor}$
$\downarrow$	$\downarrow$	$\cdots$	$\downarrow$
0	2	$\cdots$	$\lfloor p/t \rfloor$

nth node repair scheme - continued

- Node i transmits set index that contains c_i

c_1	c_2	$\cdots$	c_{n-1}
$c_1 \in \gamma_1 A_0$	$\gamma_2 A_0$	$\cdots$	$\gamma_{n-1} A_0$
$\gamma_1 A_1$	$\gamma_2 A_1$		$\gamma_{n-1} A_1$
$\gamma_1 A_2$	$c_2 \in \gamma_2 A_2$		$\gamma_{n-1} A_2$
$\vdots$	$\vdots$		$\vdots$
$\gamma_1 A_{\lfloor p/t \rfloor}$	$\gamma_2 A_{\lfloor p/t \rfloor}$	$\cdots$	$c_{n-1} \in \gamma_{n-1} A_{\lfloor p/t \rfloor}$
$\downarrow$	$\downarrow$	$\cdots$	$\downarrow$
0	2	$\cdots$	$\lfloor p/t \rfloor$

- Bandwidth $= (n-1)\log(\lceil p/t \rceil)$ bits

nth node repair scheme - continued

- Node i transmits set index that contains c_i

c_1	c_2	$\cdots$	c_{n-1}
$c_1 \in \gamma_1 A_0$	$\gamma_2 A_0$	$\cdots$	$\gamma_{n-1} A_0$
$\gamma_1 A_1$	$\gamma_2 A_1$		$\gamma_{n-1} A_1$
$\gamma_1 A_2$	$c_2 \in \gamma_2 A_2$		$\gamma_{n-1} A_2$
$\vdots$	$\vdots$		$\vdots$
$\gamma_1 A_{\lfloor p/t \rfloor}$	$\gamma_2 A_{\lfloor p/t \rfloor}$	$\cdots$	$c_{n-1} \in \gamma_{n-1} A_{\lfloor p/t \rfloor}$
$\downarrow$	$\downarrow$	$\cdots$	$\downarrow$
0	2	$\cdots$	$\lfloor p/t \rfloor$

- Bandwidth $= (n-1)\log(\lceil p/t \rceil)$ bits

Proposition

Sufficient condition holds $\rightarrow$ A repair scheme of nth node with bandwidth

$$(n-1) \cdot \log_2(\lceil p/t \rceil)$$

Repair scheme - condition

Repair scheme - condition

- Sufficient condition: For any polynomial $h(x), \deg(h) < k$, if

$$h(\alpha_i) \in \gamma_i \cdot [-(t-1), t-1], \forall i \rightarrow h(\alpha_n) = 0$$

Repair scheme - condition

- Sufficient condition: For any polynomial $h(x), \deg(h) < k$, if

$$h(\alpha_i) \in \gamma_i \cdot [-(t-1), t-1], \forall i \rightarrow h(\alpha_n) = 0$$

- Bandwidth - $(n-1)\log(\lceil p/t \rceil)$

Repair scheme - condition

- Sufficient condition: For any polynomial $h(x), \deg(h) < k$, if

$$h(\alpha_i) \in \gamma_i \cdot [-(t-1), t-1], \forall i \rightarrow h(\alpha_n) = 0$$

- Bandwidth - $(n-1)\log(\lceil p/t \rceil)$
- Need:

Repair scheme - condition

- Sufficient condition: For any polynomial $h(x), \deg(h) < k$, if

$$h(\alpha_i) \in \gamma_i \cdot [-(t-1), t-1], \forall i \rightarrow h(\alpha_n) = 0$$

- Bandwidth - $(n-1)\log(\lceil p/t \rceil)$
- Need:
  1. Find α_i's and γ_i's

Repair scheme - condition

- Sufficient condition: For any polynomial $h(x), \deg(h) < k$, if

$$h(\alpha_i) \in \gamma_i \cdot [-(t-1), t-1], \forall i \rightarrow h(\alpha_n) = 0$$

- Bandwidth - $(n-1)\log(\lceil p/t \rceil)$
- Need:
  1. Find α_i's and γ_i's
  2. Maximize t

A glimpse of the proof

Theorem

Almost all $[n, 2]_p$ *RS codes are* **asymptotically** $[n, 2, d]$ *MSR codes*

$$bandwidth = \frac{d}{d-1}\log(p) + O_n(1)$$

A glimpse of the proof

Theorem

Almost all $[n, 2]_p$ *RS codes are* **asymptotically** $[n, 2, d]$ *MSR codes*

$$bandwidth = \frac{d}{d-1}\log(p) + O_n(1)$$

- Proof:

A glimpse of the proof

Theorem

Almost all $[n, 2]_p$ *RS codes are* **asymptotically** $[n, 2, d]$ *MSR codes*

$$bandwidth = \frac{d}{d-1}\log(p) + O_n(1)$$

- Proof:
  - Assume $\alpha_n = 0, d = n - 1$

A glimpse of the proof

Theorem

Almost all $[n, 2]_p$ *RS codes are* **asymptotically** $[n, 2, d]$ *MSR codes*

$$bandwidth = \frac{d}{d-1}\log(p) + O_n(1)$$

- Proof:
  - Assume $\alpha_n = 0, d = n-1 \rightarrow$ bandwidth $= \frac{n-1}{n-2}\log(p) + O_n(1)$

A glimpse of the proof

Theorem

Almost all $[n, 2]_p$ *RS codes are* **asymptotically** $[n, 2, d]$ *MSR codes*

$$bandwidth = \frac{d}{d-1}\log(p) + O_n(1)$$

- Proof:
  - Assume $\alpha_n = 0, d = n-1 \rightarrow$ bandwidth $= \frac{n-1}{n-2}\log(p) + O_n(1)$
  - Proof by counting argument

A glimpse of the proof

- Set $\gamma_i := \alpha_i$

A glimpse of the proof

- Set $\gamma_i := \alpha_i$
- Assume sufficient condition is violated:

A glimpse of the proof

- Set $\gamma_i := \alpha_i$
- Assume sufficient condition is violated: polynomial $f(x), \deg(f) \leq 1$ s.t.

$$f(\alpha_i) \in \alpha_i[-t, t] \text{ but } f(0) \neq 0$$

A glimpse of the proof

- Set $\gamma_i := \alpha_i$
- Assume sufficient condition is violated: polynomial $f(x), \deg(f) \leq 1$ s.t.

$$f(\alpha_i) \in \alpha_i[-t, t] \text{ but } f(0) \neq 0$$

- Define

$$g(x) := f(x) - \frac{f(\alpha_1)}{\alpha_1} x$$

A glimpse of the proof

- Set $\gamma_i := \alpha_i$
- Assume sufficient condition is violated: polynomial $f(x), \deg(f) \leq 1$ s.t.

$$f(\alpha_i) \in \alpha_i[-t, t] \text{ but } f(0) \neq 0$$

- Define

$$g(x) := f(x) - \frac{f(\alpha_1)}{\alpha_1} x$$

- g satisfies:

A glimpse of the proof

- Set $\gamma_i := \alpha_i$
- Assume sufficient condition is violated: polynomial $f(x), \deg(f) \leq 1$ s.t.

$$f(\alpha_i) \in \alpha_i[-t, t] \text{ but } f(0) \neq 0$$

- Define

$$g(x) := f(x) - \frac{f(\alpha_1)}{\alpha_1} x$$

- g satisfies:
  - $g(\alpha_1) = 0$.

A glimpse of the proof

- Set $\gamma_i := \alpha_i$
- Assume sufficient condition is violated: polynomial $f(x), \deg(f) \leq 1$ s.t.

$$f(\alpha_i) \in \alpha_i[-t, t] \text{ but } f(0) \neq 0$$

- Define

$$g(x) := f(x) - \frac{f(\alpha_1)}{\alpha_1} x$$

- g satisfies:
  - $g(\alpha_1) = 0$.
  - $g(0) = f(0) \neq 0$.

A glimpse of the proof

- Set $\gamma_i := \alpha_i$
- Assume sufficient condition is violated: polynomial $f(x), \deg(f) \leq 1$ s.t.

$$f(\alpha_i) \in \alpha_i[-t, t] \text{ but } f(0) \neq 0$$

- Define

$$g(x) := f(x) - \frac{f(\alpha_1)}{\alpha_1} x$$

- g satisfies:
  - $g(\alpha_1) = 0$.
  - $g(0) = f(0) \neq 0$.
  - $\deg(g) = 1$.

A glimpse of the proof

- Set $\gamma_i := \alpha_i$
- Assume sufficient condition is violated: polynomial $f(x), \deg(f) \leq 1$ s.t.

$$f(\alpha_i) \in \alpha_i[-t, t] \text{ but } f(0) \neq 0$$

- Define

$$g(x) := f(x) - \frac{f(\alpha_1)}{\alpha_1} x$$

- g satisfies:
  - $g(\alpha_1) = 0$.
  - $g(0) = f(0) \neq 0$.
  - $\deg(g) = 1$.
  - $\rightarrow g(x) = m \cdot (x - \alpha_1)$, $m \neq 0$

A glimpse of the proof

- Set $\gamma_i := \alpha_i$
- Assume sufficient condition is violated: polynomial $f(x), \deg(f) \leq 1$ s.t.

$$f(\alpha_i) \in \alpha_i[-t, t] \text{ but } f(0) \neq 0$$

- Define

$$g(x) := f(x) - \frac{f(\alpha_1)}{\alpha_1} x$$

- g satisfies:
  - $g(\alpha_1) = 0$.
  - $g(0) = f(0) \neq 0$.
  - $\deg(g) = 1$.
  - $\rightarrow g(x) = m \cdot (x - \alpha_1)$, $m \neq 0$
  - $g(\alpha_i) = f(\alpha_i) - \frac{f(\alpha_1)}{\alpha_1} \alpha_i$

A glimpse of the proof

- Set $\gamma_i := \alpha_i$
- Assume sufficient condition is violated: polynomial $f(x), \deg(f) \leq 1$ s.t.

$$f(\alpha_i) \in \alpha_i[-t, t] \text{ but } f(0) \neq 0$$

- Define

$$g(x) := f(x) - \frac{f(\alpha_1)}{\alpha_1} x$$

- g satisfies:
  - $g(\alpha_1) = 0$.
  - $g(0) = f(0) \neq 0$.
  - $\deg(g) = 1$.
  - $\rightarrow g(x) = m \cdot (x - \alpha_1)$, $m \neq 0$
  - $g(\alpha_i) = f(\alpha_i) - \frac{f(\alpha_1)}{\alpha_1}\alpha_i \in \alpha_i[-2t, 2t]$.

A glimpse of the proof

- Set $\gamma_i := \alpha_i$
- Assume sufficient condition is violated: polynomial $f(x), \deg(f) \leq 1$ s.t.

$$f(\alpha_i) \in \alpha_i[-t, t] \text{ but } f(0) \neq 0$$

- Define

$$g(x) := f(x) - \frac{f(\alpha_1)}{\alpha_1} x$$

- g satisfies:
  - $g(\alpha_1) = 0$.
  - $g(0) = f(0) \neq 0$.
  - $\deg(g) = 1$.
  - $\rightarrow g(x) = m \cdot (x - \alpha_1)$, $m \neq 0$
  - $g(\alpha_i) = f(\alpha_i) - \frac{f(\alpha_1)}{\alpha_1}\alpha_i \in \alpha_i[-2t, 2t]$.
    $f(\alpha_i), \frac{f(\alpha_1)}{\alpha_1}\alpha_i \in \alpha_i[-t, t]$

A glimpse of the proof

- $g(x) = m(x - \alpha_1)$ where $m \neq 0$.

A glimpse of the proof

- $g(x) = m(x - \alpha_1)$ where $m \neq 0$. $g(\alpha_i) \in \alpha_i[-2t, 2t]$

A glimpse of the proof

- $g(x) = m(x - \alpha_1)$ where $m \neq 0$. $g(\alpha_i) \in \alpha_i[-2t, 2t]$
- p^3 possible values for $\alpha_1, \alpha_2, \alpha_n$

A glimpse of the proof

- $g(x) = m(x - \alpha_1)$ where $m \neq 0$. $g(\alpha_i) \in \alpha_i[-2t, 2t]$
- p^3 possible values for $\alpha_1, \alpha_2, \alpha_n$
- $4t$ possible values for m

A glimpse of the proof

- $g(x) = m(x - \alpha_1)$ where $m \neq 0$. $g(\alpha_i) \in \alpha_i[-2t, 2t]$
- p^3 possible values for $\alpha_1, \alpha_2, \alpha_n$
- $4t$ possible values for m

$$g(\alpha_2) = m(\alpha_2 - \alpha_1)$$

A glimpse of the proof

- $g(x) = m(x - \alpha_1)$ where $m \neq 0$. $g(\alpha_i) \in \alpha_i[-2t, 2t]$
- p^3 possible values for $\alpha_1, \alpha_2, \alpha_n$
- $4t$ possible values for m

$$g(\alpha_2) = m(\alpha_2 - \alpha_1) \in \alpha_2[-2t, 2t] \setminus \{0\}$$

A glimpse of the proof

- $g(x) = m(x - \alpha_1)$ where $m \neq 0$. $g(\alpha_i) \in \alpha_i[-2t, 2t]$
- p^3 possible values for $\alpha_1, \alpha_2, \alpha_n$
- $4t$ possible values for m

$$g(\alpha_2) = m(\alpha_2 - \alpha_1) \in \alpha_2[-2t, 2t] \setminus \{0\}$$

- $4t$ possible values for $\alpha_i, i = 3, \ldots, n-1$.

A glimpse of the proof

- $g(x) = m(x - \alpha_1)$ where $m \neq 0$. $g(\alpha_i) \in \alpha_i[-2t, 2t]$
- p^3 possible values for $\alpha_1, \alpha_2, \alpha_n$
- $4t$ possible values for m

$$g(\alpha_2) = m(\alpha_2 - \alpha_1) \in \alpha_2[-2t, 2t] \setminus \{0\}$$

- $4t$ possible values for $\alpha_i, i = 3, \ldots, n-1$.

$$g(\alpha_i) = m(\alpha_i - \alpha_1) = \alpha_i \cdot t' \quad \text{for } t' \in [-2t, 2t] \setminus \{0\}$$

A glimpse of the proof

- $g(x) = m(x - \alpha_1)$ where $m \neq 0$. $g(\alpha_i) \in \alpha_i[-2t, 2t]$
- p^3 possible values for $\alpha_1, \alpha_2, \alpha_n$
- $4t$ possible values for m

$$g(\alpha_2) = m(\alpha_2 - \alpha_1) \in \alpha_2[-2t, 2t] \setminus \{0\}$$

- $4t$ possible values for $\alpha_i, i = 3, \ldots, n-1$.

$$g(\alpha_i) = m(\alpha_i - \alpha_1) = \alpha_i \cdot t' \quad \text{for } t' \in [-2t, 2t] \setminus \{0\}$$

- $p^3(4t)^{n-2}$ bad selections of $\alpha_1, \ldots, \alpha_n$

A glimpse of the proof

- $g(x) = m(x - \alpha_1)$ where $m \neq 0$. $g(\alpha_i) \in \alpha_i[-2t, 2t]$
- p^3 possible values for $\alpha_1, \alpha_2, \alpha_n$
- $4t$ possible values for m

$$g(\alpha_2) = m(\alpha_2 - \alpha_1) \in \alpha_2[-2t, 2t] \setminus \{0\}$$

- $4t$ possible values for $\alpha_i, i = 3, \ldots, n-1$.

$$g(\alpha_i) = m(\alpha_i - \alpha_1) = \alpha_i \cdot t' \quad \text{for } t' \in [-2t, 2t] \setminus \{0\}$$

- $p^3(4t)^{n-2}$ bad selections of $\alpha_1, \ldots, \alpha_n$
- Set $t = (\epsilon p/4)^{\frac{n-3}{n-2}}$

A glimpse of the proof

- $g(x) = m(x - \alpha_1)$ where $m \neq 0$. $g(\alpha_i) \in \alpha_i[-2t, 2t]$
- p^3 possible values for $\alpha_1, \alpha_2, \alpha_n$
- $4t$ possible values for m

$$g(\alpha_2) = m(\alpha_2 - \alpha_1) \in \alpha_2[-2t, 2t] \setminus \{0\}$$

- $4t$ possible values for $\alpha_i, i = 3, \ldots, n-1$.

$$g(\alpha_i) = m(\alpha_i - \alpha_1) = \alpha_i \cdot t' \quad \text{for } t' \in [-2t, 2t] \setminus \{0\}$$

- $p^3(4t)^{n-2}$ bad selections of $\alpha_1, \ldots, \alpha_n$
- Set $t = (\epsilon p/4)^{\frac{n-3}{n-2}} \rightarrow p^3(4t)^{n-2} \leq \epsilon p^n$

A glimpse of the proof

- $g(x) = m(x - \alpha_1)$ where $m \neq 0$. $g(\alpha_i) \in \alpha_i[-2t, 2t]$
- p^3 possible values for $\alpha_1, \alpha_2, \alpha_n$
- $4t$ possible values for m

$$g(\alpha_2) = m(\alpha_2 - \alpha_1) \in \alpha_2[-2t, 2t] \setminus \{0\}$$

- $4t$ possible values for $\alpha_i, i = 3, \ldots, n-1$.

$$g(\alpha_i) = m(\alpha_i - \alpha_1) = \alpha_i \cdot t' \quad \text{for } t' \in [-2t, 2t] \setminus \{0\}$$

- $p^3(4t)^{n-2}$ bad selections of $\alpha_1, \ldots, \alpha_n$
- Set $t = (\epsilon p/4)^{\frac{n-3}{n-2}} \rightarrow p^3(4t)^{n-2} \leq \epsilon p^n$
- Bandwidth $= (n-1)\log(p/t) = \frac{n-1}{n-2}\log(p) + O_n(1)$

Summary and open questions

Summary and open questions

- The repair problem

Summary and open questions

- The repair problem
- RS codes are important

Summary and open questions

- The repair problem
- RS codes are important
- Repairing RS codes over prime fields is possible

Summary and open questions

- The repair problem
- RS codes are important
- Repairing RS codes over prime fields is possible
- First example of nonlinear repair of RS codes

Summary and open questions

- The repair problem
- RS codes are important
- Repairing RS codes over prime fields is possible
- First example of nonlinear repair of RS codes

Questions:

Summary and open questions

- The repair problem
- RS codes are important
- Repairing RS codes over prime fields is possible
- First example of nonlinear repair of RS codes

Questions:

- Construct $[n, k, d]$ MSR RS codes over $\mathbb{F}_p$ (explicit or existence)

Summary and open questions

- The repair problem
- RS codes are important
- Repairing RS codes over prime fields is possible
- First example of nonlinear repair of RS codes

Questions:

- Construct $[n, k, d]$ MSR RS codes over $\mathbb{F}_p$ (explicit or existence)
- Efficient repair algorithm

Summary and open questions

- The repair problem
- RS codes are important
- Repairing RS codes over prime fields is possible
- First example of nonlinear repair of RS codes

Questions:

- Construct $[n, k, d]$ MSR RS codes over $\mathbb{F}_p$ (explicit or existence)
- Efficient repair algorithm
- Prove the SSS leakage resilience conjecture

Summary and open questions

- The repair problem
- RS codes are important
- Repairing RS codes over prime fields is possible
- First example of nonlinear repair of RS codes

Questions:

- Construct $[n, k, d]$ MSR RS codes over $\mathbb{F}_p$ (explicit or existence)
- Efficient repair algorithm
- Prove the SSS leakage resilience conjecture
- Understand the limits of the repair - relation between n, k, d, p

Summary and open questions

- The repair problem
- RS codes are important
- Repairing RS codes over prime fields is possible
- First example of nonlinear repair of RS codes

Questions:

- Construct $[n, k, d]$ MSR RS codes over $\mathbb{F}_p$ (explicit or existence)
- Efficient repair algorithm
- Prove the SSS leakage resilience conjecture
- Understand the limits of the repair - relation between n, k, d, p
- Twist on the interpolation problem -

Summary and open questions

- The repair problem
- RS codes are important
- Repairing RS codes over prime fields is possible
- First example of nonlinear repair of RS codes

Questions:

- Construct $[n, k, d]$ MSR RS codes over $\mathbb{F}_p$ (explicit or existence)
- Efficient repair algorithm
- Prove the SSS leakage resilience conjecture
- Understand the limits of the repair - relation between n, k, d, p
- Twist on the interpolation problem - other applications?

Thank you!


Omar Alrabiah and Venkatesan Guruswami.
An exponential lower bound on the sub-packetization of MSR codes.
In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 979–985, 2019.


Venkatesan Guruswami and Mary Wootters.
Repairing Reed-Solomon codes.
IEEE transactions on Information Theory, 63(9):5684–5698, 2017.


Korlakai Vinayak Rashmi, Nihar B Shah, and P Vijay Kumar.
Optimal exact-regenerating codes for distributed storage at the MSR and MBR points via a product-matrix construction.
IEEE Transactions on Information Theory, 57(8):5227–5239, 2011.


Karthikeyan Shanmugam, Dimitris S Papailiopoulos, Alexandros G Dimakis, and Giuseppe Caire.
A repair framework for scalar MDS codes.

IEEE Journal on Selected Areas in Communications, 32(5):998–1007, 2014.


Itzhak Tamo, Min Ye, and Alexander Barg.
Optimal repair of Reed-Solomon codes: Achieving the cut-set bound.
In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 216–227. IEEE, 2017.


Zhiying Wang, Itzhak Tamo, and Jehoshua Bruck.
Explicit minimum storage regenerating codes.
IEEE Transactions on Information Theory, 62(8):4466–4480, 2016.


Min Ye and Alexander Barg.
Explicit constructions of optimal-access MDS codes with nearly optimal sub-packetization.
IEEE Transactions on Information Theory, 63(10):6307–6317, 2017.