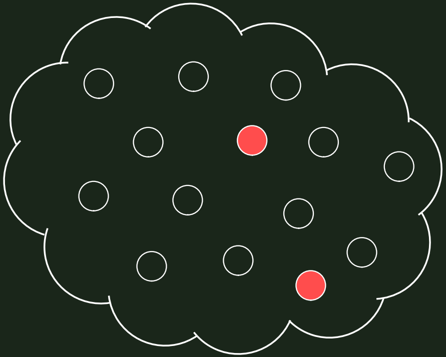
Polytope Codes in Networks, Storage, and Multiple Descriptions

Oliver Kosut

Joint work with Lang Tong, David Tse, Aaron Wagner, and Xiaoqing Fan

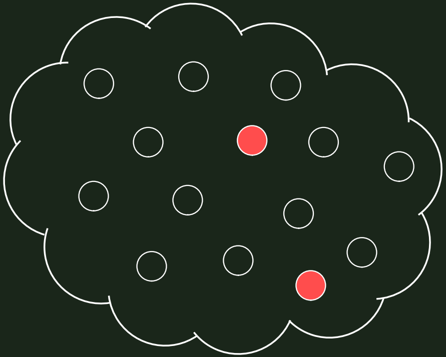
April 1, 2015

Networks with Active Adversaries



Distributed system in the presence of active omniscient adversaries

Networks with Active Adversaries



Distributed system in the presence of active omniscient adversaries

Applications:

- Man-in-the-middle attacks
- Wireless jamming attacks
- Distributed storage systems

Polytope Codes

A new-ish coding paradigm using:

- linear constructions on the integers
- covariance matrices as checksums

Polytope Codes

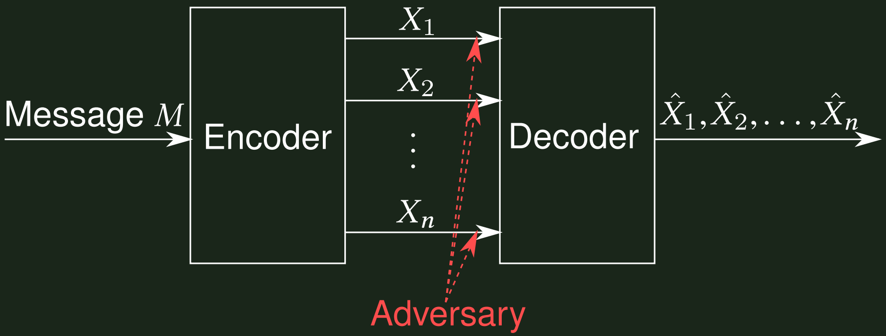
A new-ish coding paradigm using:

- linear constructions on the integers
- covariance matrices as checksums

Advantages:

- Partial decoding
- Distributed detection and correction of adversarial errors

Classical Coding Formulation



- X_i in finite field $\mathbb{F}$
- Adversary may replace any z packets (min. distance $d \geq 2z + 1$)
- Decoder must output all packets without error
- Fundamental limit: Singleton bound $k \leq n - 2z$ where k is dimension of message — achievable by MDS codes

Classical setting

Must decode all information

Network setting

Partial information may do

— any partial information

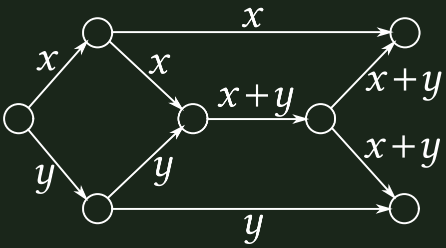
Classical setting

Must decode all information

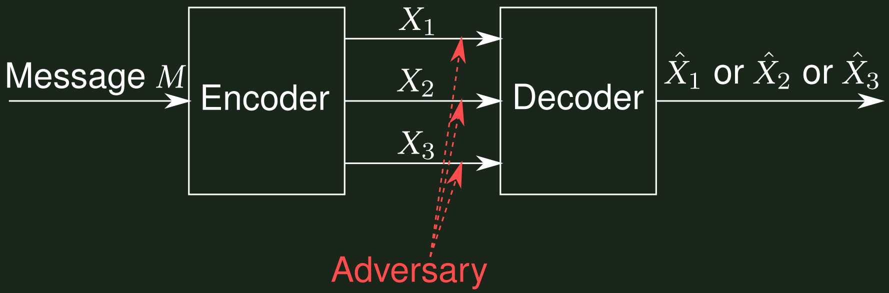
Network setting

Partial information may do

— any partial information



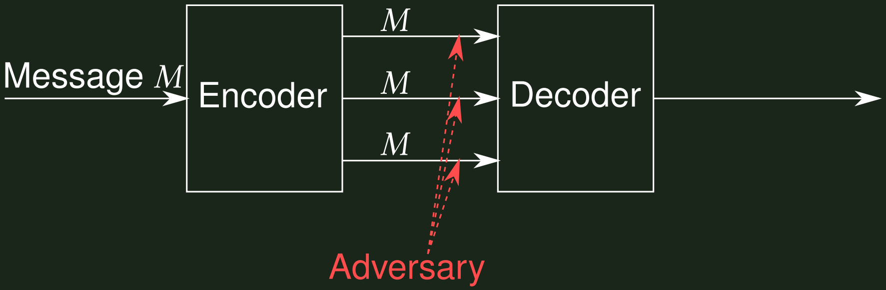
Motivating Toy Problem



- $M \in \{1, 2, \ldots, 2^{qR}\}$
- $X_i \in \{1, 2, \ldots, 2^q\}$
- M must be recoverable from any two of X_1, X_2, X_3
- Adversary may replace one of the three packets
- Decoder must output one packet without error

Finite Field Constructions

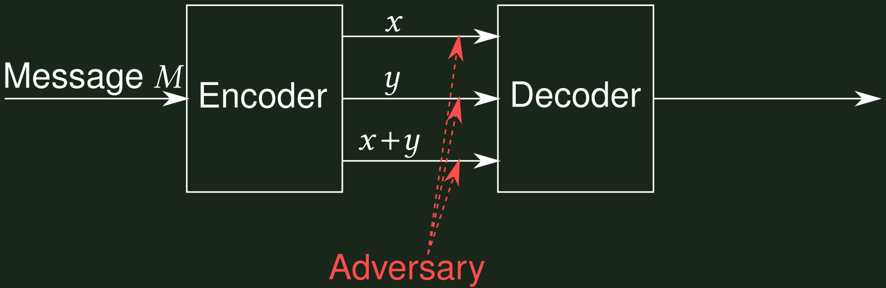
(3,1) MDS code: Let $M \in \mathbb{F}$



Achieves $R = 1$

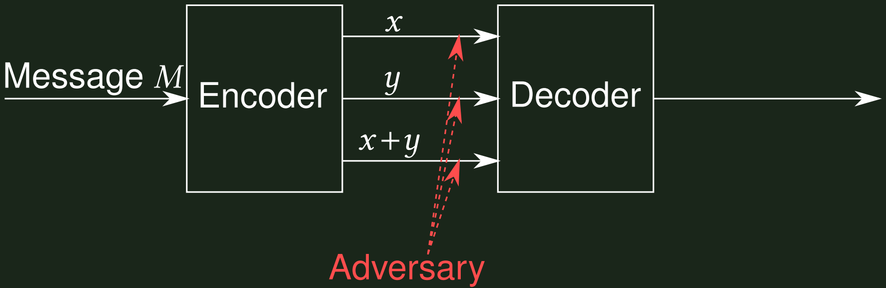
Finite Field Constructions

(3,2) MDS code: Let $M = (x, y)$, $x, y \in \mathbb{F}$



Finite Field Constructions

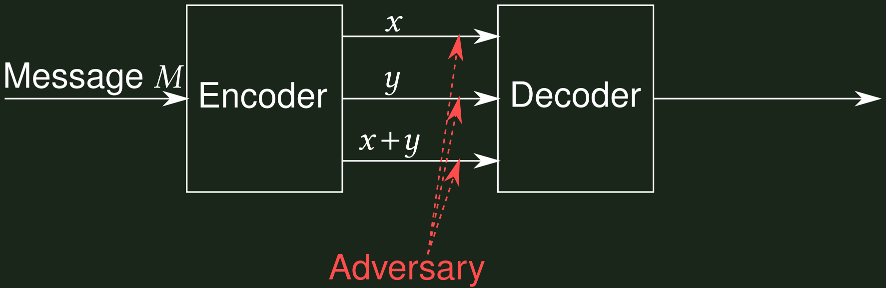
(3,2) MDS code: Let $M = (x, y)$, $x, y \in \mathbb{F}$



- If adversary alters one of the packets, decoder cannot tell which

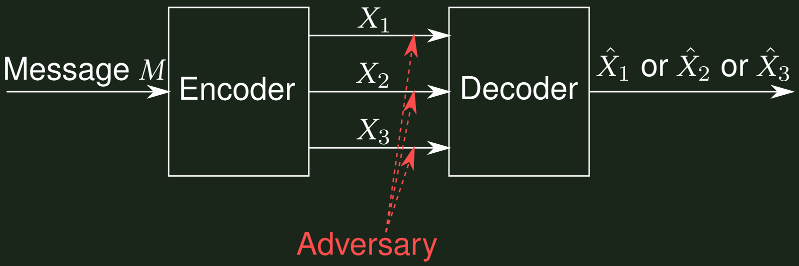
Finite Field Constructions

(3,2) MDS code: Let $M = (x, y)$, $x, y \in \mathbb{F}$

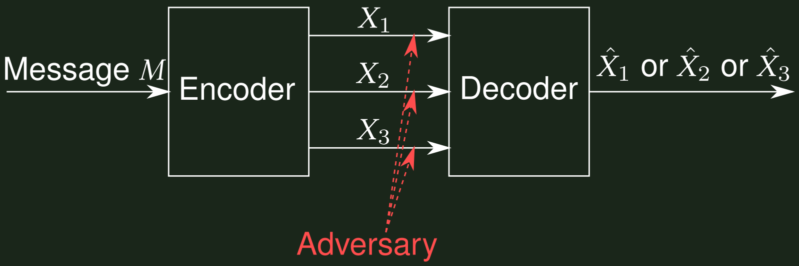


- If adversary alters one of the packets, decoder cannot tell which
- Finite field code cannot do better than $R = 1$

What would it take to achieve $R = 2$?

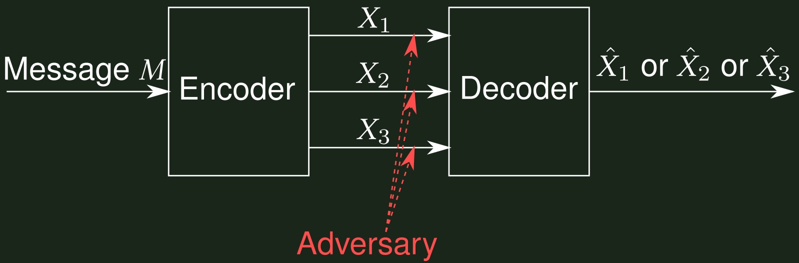


What would it take to achieve $R = 2$?



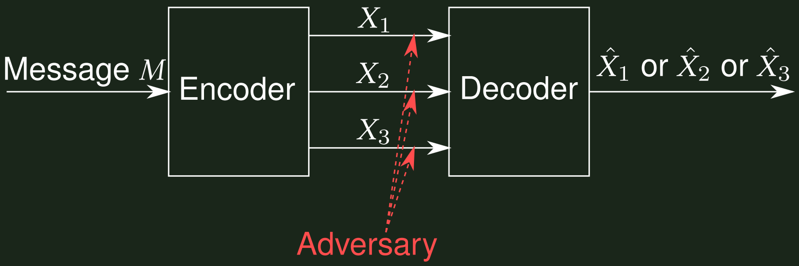
- $H(X_i, X_j) = H(M) = 2q$

What would it take to achieve $R = 2$?



- $H(X_i, X_j) = H(M) = 2q$
- Thus $I(X_i; X_j) = 0$

What would it take to achieve $R = 2$?

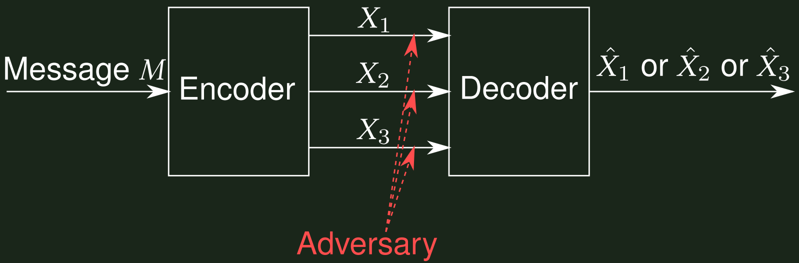


- $H(X_i, X_j) = H(M) = 2q$
- Thus $I(X_i; X_j) = 0$
- But if the packets are pairwise independent, then adversary may replace X_3 with an independent copy, yielding distribution

$$p(x_1)\,p(x_2)\,p(x_3)$$

Decoder cannot tell which is correct

What would it take to achieve $R = 2$?



- $H(X_i, X_j) = H(M) =$ ~~$2q$~~ $(2 - \epsilon)q$
- Thus $I(X_i; X_j) =$ ~~0~~ ϵq
- But if the packets are pairwise independent, then adversary may replace X_3 with an independent copy, yielding distribution

$$p(x_1)\, p(x_2)\, p(x_3)$$

Decoder cannot tell which is correct

A Polytope Code Construction

- Let $M = (x^N, y^N)$ where $x^N, y^N \in \{1, 2, 3, \ldots, 2^k\}^N$

A Polytope Code Construction

- Let $M = (x^N, y^N)$ where $x^N, y^N \in \{1, 2, 3, \ldots, 2^k\}^N$
- Let $z^N = x^N + y^N$ [x^N, y^N, z^N sit in a polytope]

A Polytope Code Construction

- Let $M = (x^N, y^N)$ where $x^N, y^N \in \{1, 2, 3, \ldots, 2^k\}^N$
- Let $z^N = x^N + y^N$ $\quad [x^N, y^N, z^N$ sit in a polytope]
- Construct the covariance

$$\Sigma^\star = \begin{bmatrix} x^N \\ y^N \\ z^N \end{bmatrix} \begin{bmatrix} x^N \\ y^N \\ z^N \end{bmatrix}^T = \begin{bmatrix} \langle x^N, x^N \rangle & \langle x^N, y^N \rangle & \langle x^N, z^N \rangle \\ \langle x^N, y^N \rangle & \langle y^N, y^N \rangle & \langle y^N, z^N \rangle \\ \langle x^N, z^N \rangle & \langle y^N, z^N \rangle & \langle z^N, z^N \rangle \end{bmatrix}$$

A Polytope Code Construction

- Let $M = (x^N, y^N)$ where $x^N, y^N \in \{1, 2, 3, \dots, 2^k\}^N$
- Let $z^N = x^N + y^N$ [x^N, y^N, z^N sit in a polytope]
- Construct the covariance

$$\Sigma^\star = \begin{bmatrix} x^N \\ y^N \\ z^N \end{bmatrix} \begin{bmatrix} x^N \\ y^N \\ z^N \end{bmatrix}^T = \begin{bmatrix} \langle x^N, x^N \rangle & \langle x^N, y^N \rangle & \langle x^N, z^N \rangle \\ \langle x^N, y^N \rangle & \langle y^N, y^N \rangle & \langle y^N, z^N \rangle \\ \langle x^N, z^N \rangle & \langle y^N, z^N \rangle & \langle z^N, z^N \rangle \end{bmatrix}$$

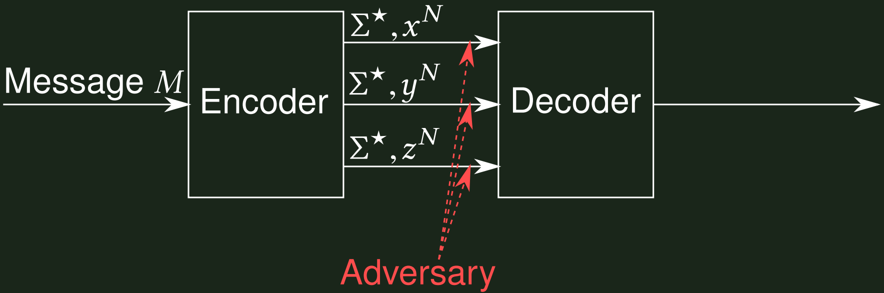
- $\Sigma^\star$ takes infinitesimal rate compared to x^N for large N

A Polytope Code Construction

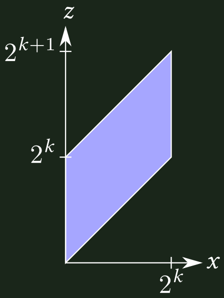
- Let $M = (x^N, y^N)$ where $x^N, y^N \in \{1, 2, 3, \ldots, 2^k\}^N$
- Let $z^N = x^N + y^N$ [x^N, y^N, z^N sit in a polytope]
- Construct the covariance

$$\Sigma^\star = \begin{bmatrix} x^N \\ y^N \\ z^N \end{bmatrix} \begin{bmatrix} x^N \\ y^N \\ z^N \end{bmatrix}^T = \begin{bmatrix} \langle x^N, x^N \rangle & \langle x^N, y^N \rangle & \langle x^N, z^N \rangle \\ \langle x^N, y^N \rangle & \langle y^N, y^N \rangle & \langle y^N, z^N \rangle \\ \langle x^N, z^N \rangle & \langle y^N, z^N \rangle & \langle z^N, z^N \rangle \end{bmatrix}$$

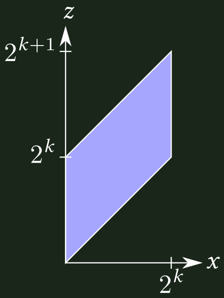
- $\Sigma^\star$ takes infinitesimal rate compared to x^N for large N



MDS structure

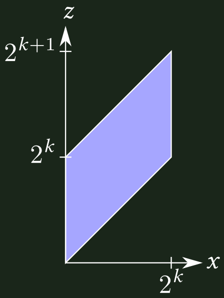


MDS structure



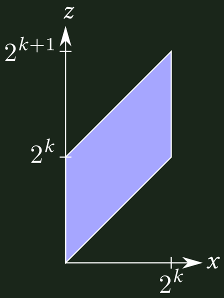
- $x^N, y^N \in \{1, 2, \ldots, 2^k\}^N$: Number of bits = kN

MDS structure



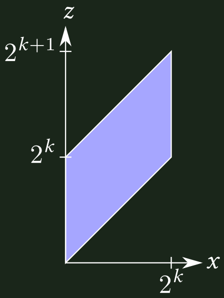
- $x^N, y^N \in \{1, 2, \ldots, 2^k\}^N$: Number of bits = kN
- $z^N \in \{1, 2, \ldots, 2^{k+1}\}^N$: Number of bits = $(k+1)N \approx kN$ for large k

MDS structure



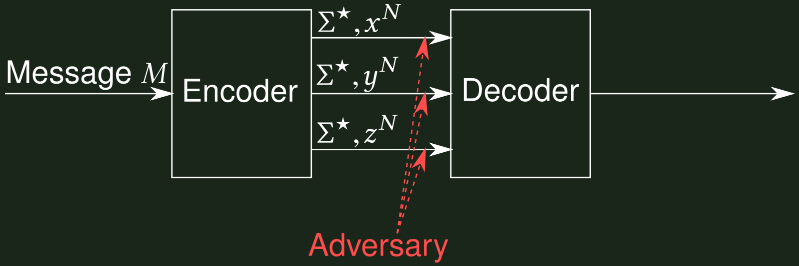
- $x^N, y^N \in \{1, 2, \ldots, 2^k\}^N$: Number of bits = kN
- $z^N \in \{1, 2, \ldots, 2^{k+1}\}^N$: Number of bits = $(k+1)N \approx kN$ for large k
- Thus x^N, y^N, z^N are nearly pairwise independent

MDS structure

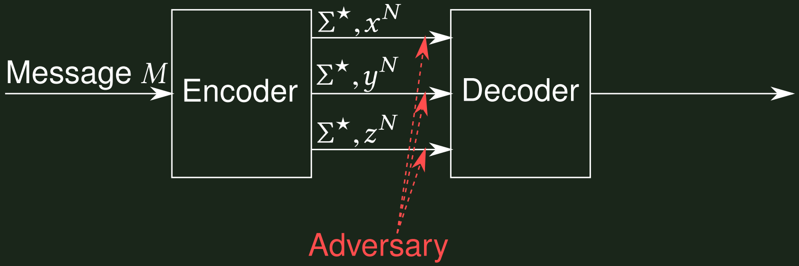


- $x^N, y^N \in \{1, 2, \ldots, 2^k\}^N$: Number of bits = kN
- $z^N \in \{1, 2, \ldots, 2^{k+1}\}^N$: Number of bits = $(k+1)N \approx kN$ for large k
- Thus x^N, y^N, z^N are nearly pairwise independent
- (x^N, y^N, z^N) form a $(3, 2)$ MDS polytope code

Decoding

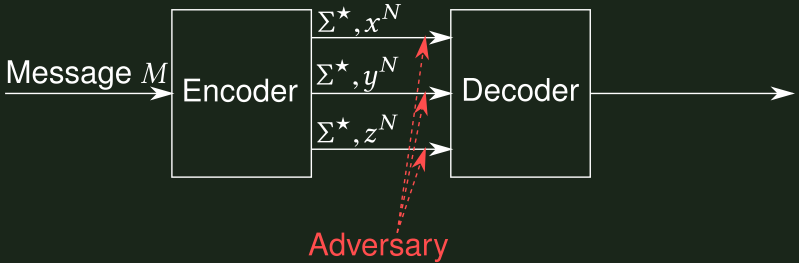

Message M
Encoder
$\Sigma^\star, x^N$
$\Sigma^\star, y^N$
$\Sigma^\star, z^N$
Decoder
Adversary

Decoding



- Recover the should-be covariance $\Sigma^\star$ using majority rule

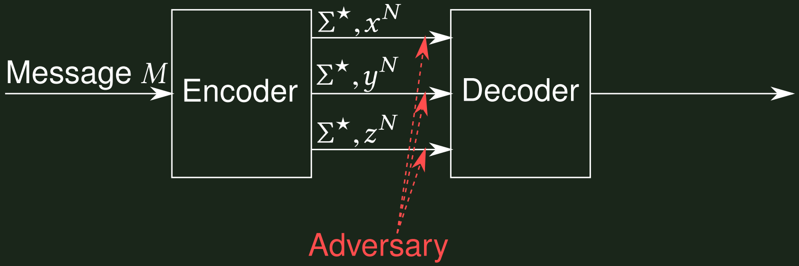
Decoding



- Recover the should-be covariance $\Sigma^\star$ using majority rule
- Given x^N, y^N, z^N form the actually-is covariance

$$\Sigma = \begin{bmatrix} \langle x^N, x^N \rangle & \langle x^N, y^N \rangle & \langle x^N, z^N \rangle \\ \langle x^N, y^N \rangle & \langle y^N, y^N \rangle & \langle y^N, z^N \rangle \\ \langle x^N, z^N \rangle & \langle y^N, z^N \rangle & \langle z^N, z^N \rangle \end{bmatrix}$$

Decoding

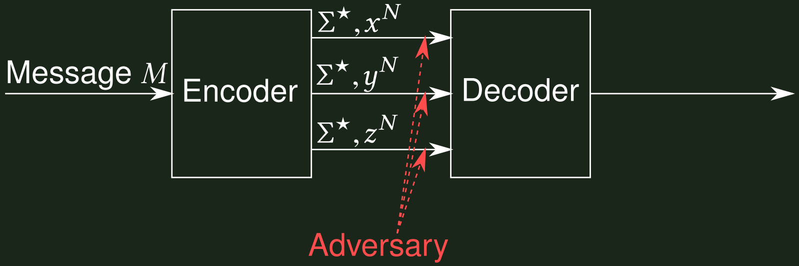


- Recover the should-be covariance $\Sigma^\star$ using majority rule
- Given x^N, y^N, z^N form the actually-is covariance

$$\Sigma = \begin{bmatrix} \langle x^N, x^N \rangle & \langle x^N, y^N \rangle & \langle x^N, z^N \rangle \\ \langle x^N, y^N \rangle & \langle y^N, y^N \rangle & \langle y^N, z^N \rangle \\ \langle x^N, z^N \rangle & \langle y^N, z^N \rangle & \langle z^N, z^N \rangle \end{bmatrix}$$

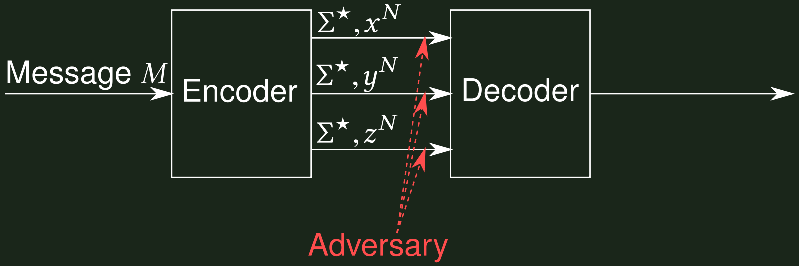
- By comparing $\Sigma^\star$ with Σ, the decoder can always find a trustworthy packet

Decoding



Suppose $\Sigma \neq \Sigma^\star$:

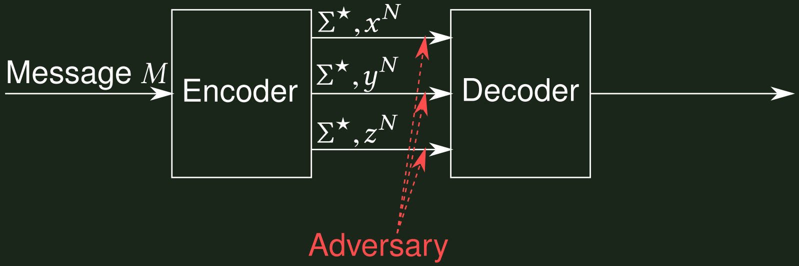
Decoding



Suppose $\Sigma \neq \Sigma^\star$:

- If $\Sigma_{xx} \neq \Sigma^\star_{xx}$, then x^N is corrupted — y^N and z^N are safe

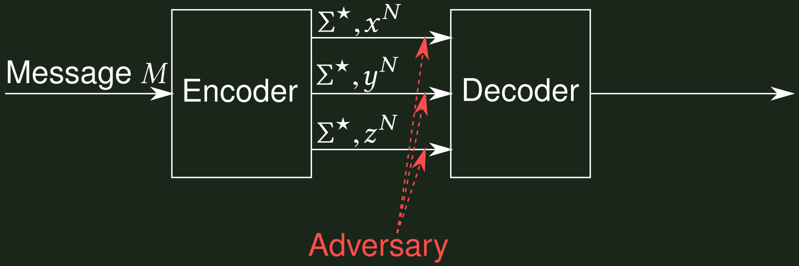
Decoding



Suppose $\Sigma \neq \Sigma^\star$:

- If $\Sigma_{xx} \neq \Sigma^\star_{xx}$, then x^N is corrupted — y^N and z^N are safe
- If $\Sigma_{xy} \neq \Sigma^\star_{xy}$, then either x^N or y^N is corrupted — z^N is safe

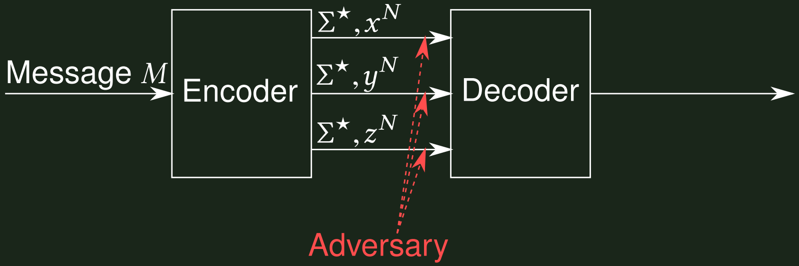
Decoding



Suppose $\Sigma \neq \Sigma^\star$:

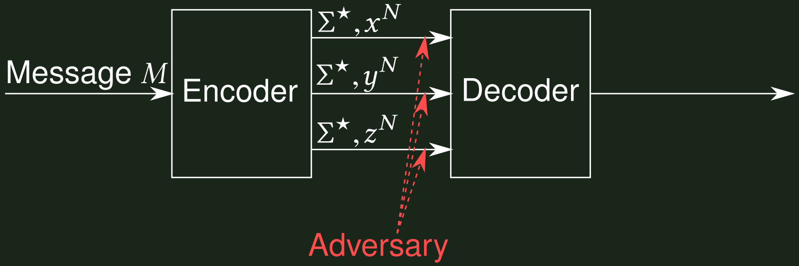
- If $\Sigma_{xx} \neq \Sigma^\star_{xx}$, then x^N is corrupted — y^N and z^N are safe
- If $\Sigma_{xy} \neq \Sigma^\star_{xy}$, then either x^N or y^N is corrupted — z^N is safe
- Can always identify one safe packet

Decoding



Suppose $\Sigma = \Sigma^\star$:

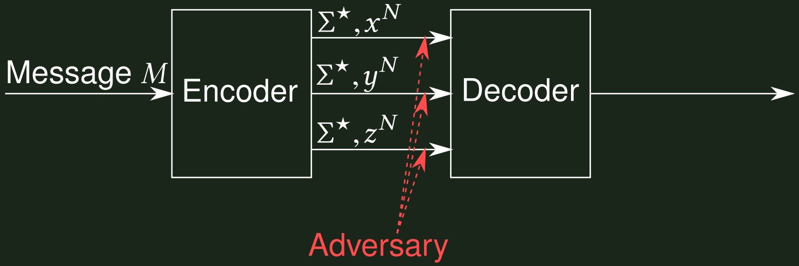
Decoding



Suppose $\Sigma = \Sigma^\star$:

- All quadratic functions of x^N, y^N, z^N must be uncorrupted

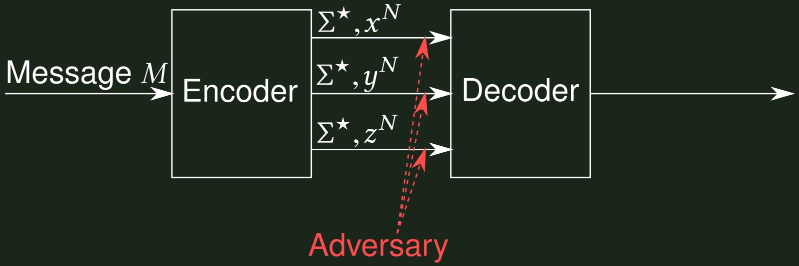
Decoding



Suppose $\Sigma = \Sigma^\star$:

- All quadratic functions of x^N, y^N, z^N must be uncorrupted
- $\left\|x^N + y^N - z^N\right\|^2 = 0 \quad \Longrightarrow \quad x^N + y^N - z^N = 0$

Decoding



Suppose $\Sigma = \Sigma^\star$:

- All quadratic functions of x^N, y^N, z^N must be uncorrupted
- $\left\|x^N + y^N - z^N\right\|^2 = 0 \quad \Longrightarrow \quad x^N + y^N - z^N = 0$
- Therefore all packets are trustworthy

Outline

- Polytope codes in general
- Polytope codes in network coding
- Polytope codes in distributed storage systems
- Polytope codes in multiple descriptions

Outline

- Polytope codes in general
- Polytope codes in network coding
- Polytope codes in distributed storage systems
- Polytope codes in multiple descriptions

Generic polytope code constructions

- Message $m \in \{1, 2, \ldots, 2^k\}^{R \times N}$

Generic polytope code constructions

- Message $m \in \{1, 2, \ldots, 2^k\}^{R \times N}$
- Calculate covariance $\Sigma^\star = m\, m^T$ — included in all packets

Generic polytope code constructions

- Message $m \in \{1, 2, \ldots, 2^k\}^{R \times N}$
- Calculate covariance $\Sigma^\star = m\, m^T$ — included in all packets
- Packet data is in the form $x^N = a^T m$ for integer vector $a \in \mathbb{Z}^R$

Generic polytope code constructions

- Message $m \in \{1, 2, \ldots, 2^k\}^{R \times N}$
- Calculate covariance $\Sigma^\star = m\,m^T$ — included in all packets
- Packet data is in the form $x^N = a^T m$ for integer vector $a \in \mathbb{Z}^R$
- $x_i = \sum_j a_j m_{ji} \leq \sum_j a_j 2^k \leq 2^{k+\Delta}$ for sufficiently large k
  — requires $(k + \Delta)N$ bits to store

Generic polytope code constructions

- Message $m \in \{1, 2, \ldots, 2^k\}^{R \times N}$
- Calculate covariance $\Sigma^\star = m\, m^T$ — included in all packets
- Packet data is in the form $x^N = a^T m$ for integer vector $a \in \mathbb{Z}^R$
- $x_i = \sum_j a_j m_{ji} \leq \sum_j a_j 2^k \leq 2^{k+\Delta}$ for sufficiently large k
  — requires $(k + \Delta)N$ bits to store

These constructions can mimic most finite field linear codes

Main property

Given some subset of packets $y^N = \begin{bmatrix} x_1^N \\ x_2^N \\ \vdots \\ x_p^N \end{bmatrix} = Am$

Main property

Given some subset of packets $y^N = \begin{bmatrix} x_1^N \\ x_2^N \\ \vdots \\ x_p^N \end{bmatrix} = Am$

- Form $\Sigma_y = (y^N)\,(y^N)^T$

Main property

Given some subset of packets $y^N = \begin{bmatrix} x_1^N \\ x_2^N \\ \vdots \\ x_p^N \end{bmatrix} = Am$

- Form $\Sigma_y = (y^N)\,(y^N)^T$
- Without corruption, $\Sigma_y = A\Sigma^\star A^T$

Main property

Given some subset of packets $y^N = \begin{bmatrix} x_1^N \\ x_2^N \\ \vdots \\ x_p^N \end{bmatrix} = Am$

- Form $\Sigma_y = (y^N)\,(y^N)^T$
- Without corruption, $\Sigma_y = A\Sigma^\star A^T$
- If $\Sigma \neq A^T \Sigma^\star A$, then corrupted packets may be localized

Main property

Given some subset of packets $y^N = \begin{bmatrix} x_1^N \\ x_2^N \\ \vdots \\ x_p^N \end{bmatrix} = Am$

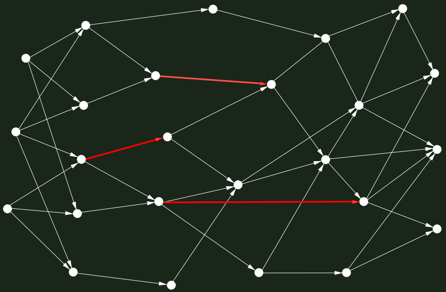
- Form $\Sigma_y = (y^N)\,(y^N)^T$
- Without corruption, $\Sigma_y = A\Sigma^\star A^T$
- If $\Sigma \neq A^T\Sigma^\star A$, then corrupted packets may be localized
- If $\Sigma = A^T\Sigma^\star A$, then all quadratic functions are uncorrupted:

  For C satisfying $CA = 0$, $\|Cy^N\|^2 = 0$, so $Cy^N = 0$, i.e. all linear constraints match

Outline

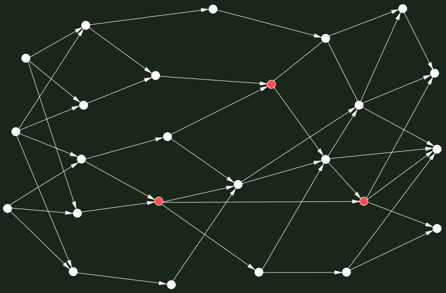
- Polytope codes in general
- Polytope codes in network coding
- Polytope codes in distributed storage systems
- Polytope codes in multiple descriptions

Network Error Correction



- Directed graph of rate-limited noise-free channels
- Omniscient adversary can control some subset of the network
- Possible adversary control models:
  - Any z edges
  - Any z nodes
  - Any z edges/nodes from a specific area

Network Error Correction



- Directed graph of rate-limited noise-free channels
- Omniscient adversary can control some subset of the network
- Possible adversary control models:
  - Any z edges
  - Any z nodes
  - Any z edges/nodes from a specific area

Theorem (Cai-Yeung (2006))

For a single multicast, and an adversary that controls any z unit-capacity edges:

$$C = \textit{min-cut} - 2z$$

Theorem (Cai-Yeung (2006))

For a single multicast, and an adversary that controls any z unit-capacity edges:

$$C = \text{min-cut} - 2z$$

- Converse via network version of the Singleton bound
- Achievability via network version of (linear) MDS codes

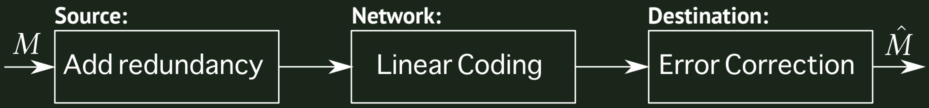
Theorem (Cai-Yeung (2006))

For a single multicast, and an adversary that controls any z unit-capacity edges:

$$C = \text{min-cut} - 2z$$

- Converse via network version of the Singleton bound
- Achievability via network version of (linear) MDS codes

Can be viewed as a separation theorem:



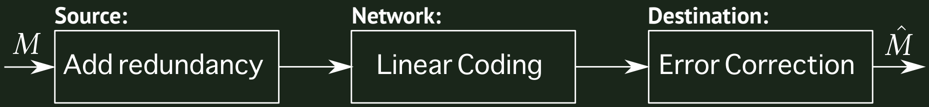
Theorem (Cai-Yeung (2006))

For a single multicast, and an adversary that controls any z unit-capacity edges:

$$C = \textit{min-cut} - 2z$$

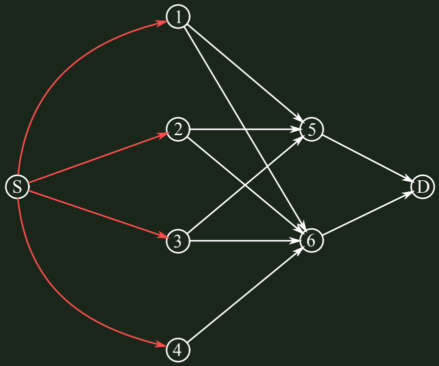
- Converse via network version of the Singleton bound
- Achievability via network version of (linear) MDS codes

Can be viewed as a separation theorem:



Polytope codes allow error detection/correction inside the network

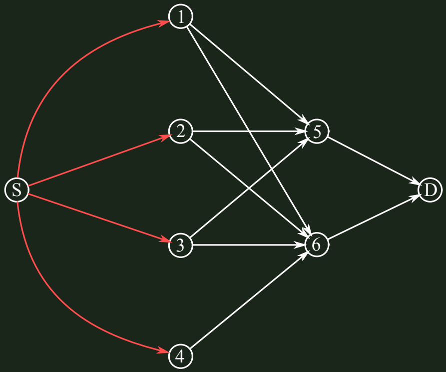
The Caterpillar Network



- Single unicast from S to D
- All links have unit capacity
- Adversary may control any one of the red edges
- Simple upper bound: $C \leq 2$

Polytope Code Achievability

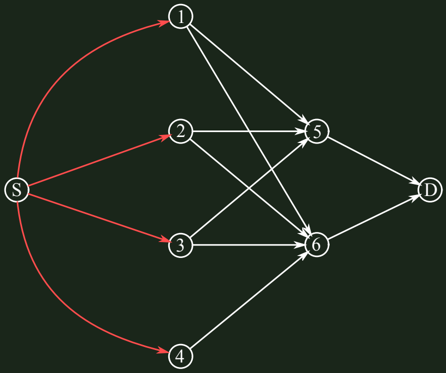
Let message $m = (x^N, y^N)$, where $x^N, y^N \in \{1, \ldots, 2^k\}^N$



Polytope Code Achievability

Let message $m = (x^N, y^N)$, where $x^N, y^N \in \{1, \ldots, 2^k\}^N$

$$z^N = x^N + y^N$$
$$w^N = x^N + 2y^N$$
$$\Sigma^\star = m\,m^T$$

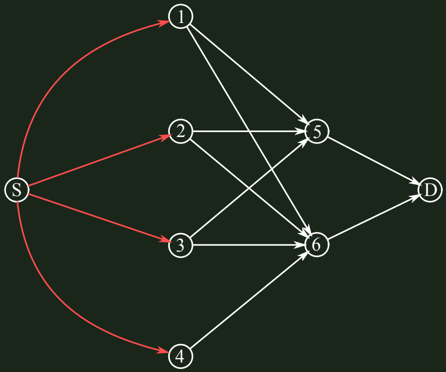


Polytope Code Achievability

Let message $m = (x^N, y^N)$, where $x^N, y^N \in \{1, \ldots, 2^k\}^N$

$$z^N = x^N + y^N$$
$$w^N = x^N + 2y^N$$
$$\Sigma^\star = m\, m^T$$

(x^N, y^N, z^N, w^N) form a $(4,2)$ MDS polytope code

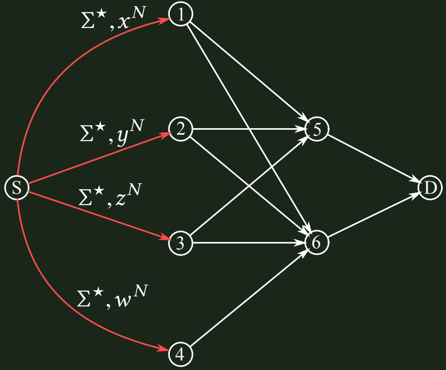


Polytope Code Achievability

Let message $m = (x^N, y^N)$, where $x^N, y^N \in \{1, \ldots, 2^k\}^N$

$$z^N = x^N + y^N$$
$$w^N = x^N + 2y^N$$
$$\Sigma^\star = m\, m^T$$

(x^N, y^N, z^N, w^N) form a $(4,2)$ MDS polytope code

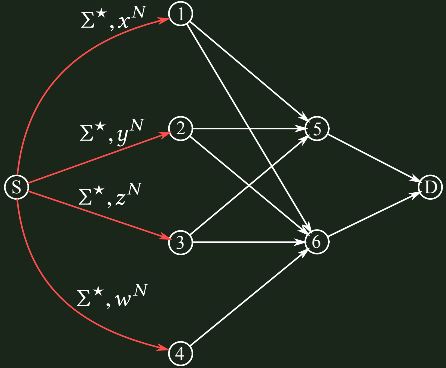


Polytope Code Achievability

Let message $m = (x^N, y^N)$, where $x^N, y^N \in \{1, \ldots, 2^k\}^N$

$$z^N = x^N + y^N$$
$$w^N = x^N + 2y^N$$
$$\Sigma^\star = m\,m^T$$

(x^N, y^N, z^N, w^N) form a $(4, 2)$ MDS polytope code



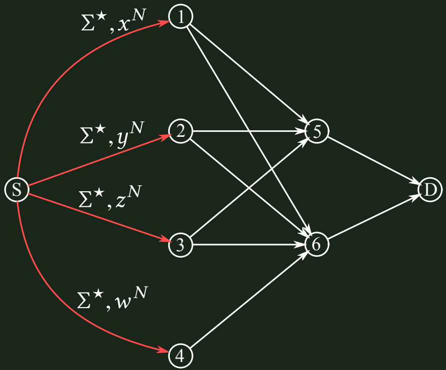
- At node 5, determine one uncorrupted packet

Polytope Code Achievability

Let message $m = (x^N, y^N)$, where $x^N, y^N \in \{1, \ldots, 2^k\}^N$

$$z^N = x^N + y^N$$
$$w^N = x^N + 2y^N$$
$$\Sigma^\star = m\, m^T$$

(x^N, y^N, z^N, w^N) form a $(4, 2)$ MDS polytope code



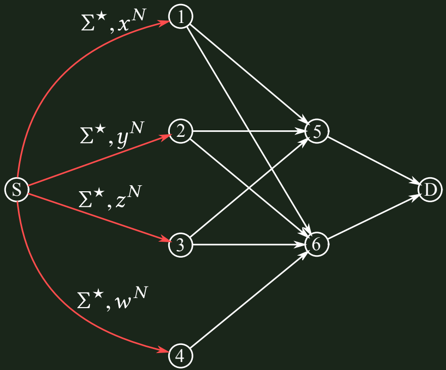
- At node 5, determine one uncorrupted packet
- At node 6, decode the message and transmit a different uncorrupted packet

Polytope Code Achievability

Let message $m = (x^N, y^N)$, where $x^N, y^N \in \{1, \ldots, 2^k\}^N$

$$z^N = x^N + y^N$$
$$w^N = x^N + 2y^N$$
$$\Sigma^\star = m\, m^T$$

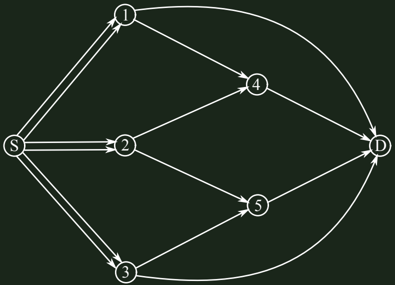
(x^N, y^N, z^N, w^N) form a $(4,2)$ MDS polytope code



- At node 5, determine one uncorrupted packet
- At node 6, decode the message and transmit a different uncorrupted packet

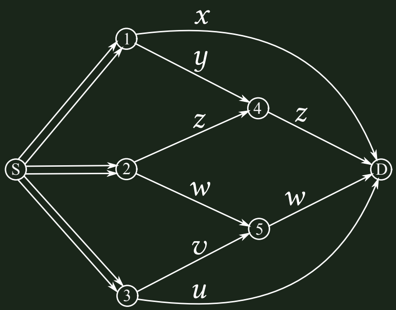
No finite field linear code achieves this rate

Cockroach Network



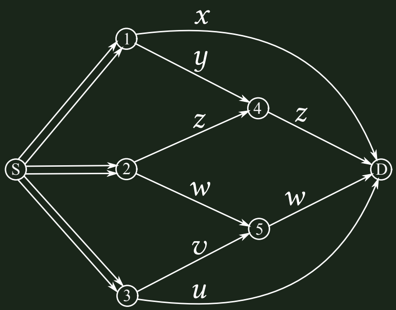
- One node is controlled by the adversary — controls all outgoing messages

Cockroach Network



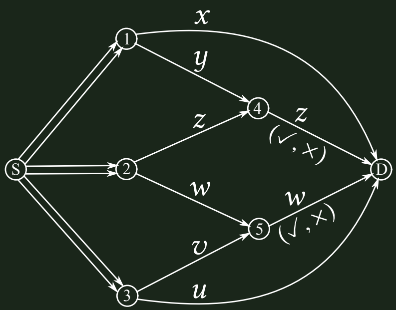
- One node is controlled by the adversary — controls all outgoing messages
- Let $(x^N, y^N, z^N, w^N, v^N, u^N)$ be a $(6, 2)$ MDS polytope code

Cockroach Network



- One node is controlled by the adversary — controls all outgoing messages
- Let $(x^N, y^N, z^N, w^N, v^N, u^N)$ be a $(6, 2)$ MDS polytope code
- $\Sigma^\star$ included in all packets

Cockroach Network



- One node is controlled by the adversary — controls all outgoing messages
- Let $(x^N, y^N, z^N, w^N, v^N, u^N)$ be a $(6, 2)$ MDS polytope code
- $\Sigma^\star$ included in all packets
- Nodes 4 and 5 compare covariance of incoming pair of packets — transmit outcome of comparison

A Class of Networks Solved by Polytope Codes

Theorem (Kosut-Tong-Tse (2014))

Polytope codes achieve the cut-set bound if

- *Network is planar*
- *1 adversary node*
- *No node has more than 2 unit-capacity output edges*
- *No node has more outputs than inputs*

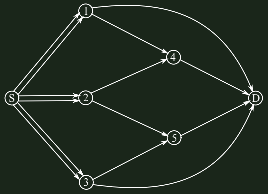
A Class of Networks Solved by Polytope Codes

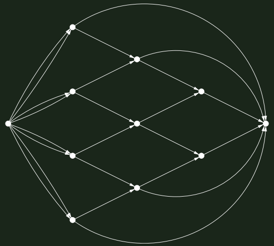
Theorem (Kosut-Tong-Tse (2014))

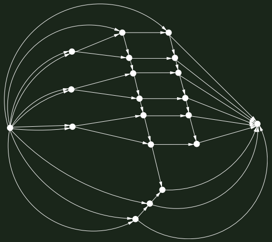
Polytope codes achieve the cut-set bound if

- *Network is planar*
- *1 adversary node*
- *No node has more than 2 unit-capacity output edges*
- *No node has more outputs than inputs*

Examples:



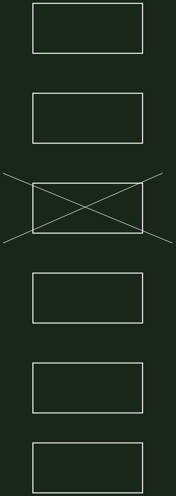




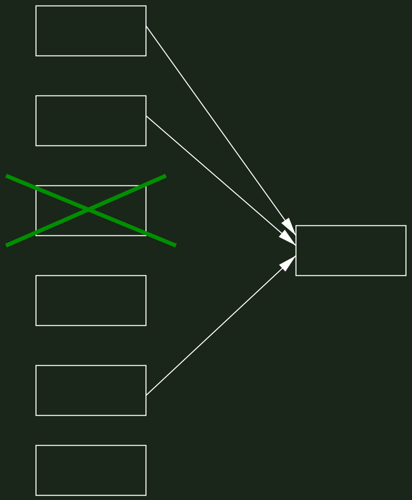
Outline

- Polytope codes in general
- Polytope codes in network coding
- Polytope codes in distributed storage systems
- Polytope codes in multiple descriptions

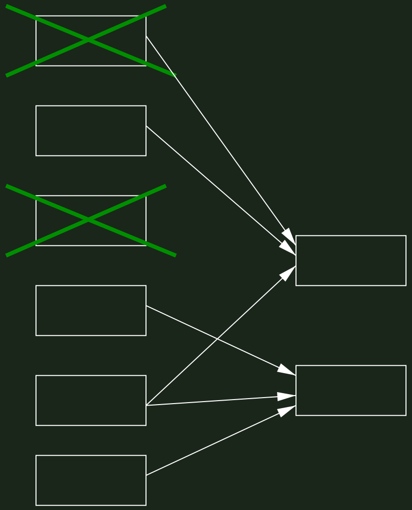
Distributed Storage Systems



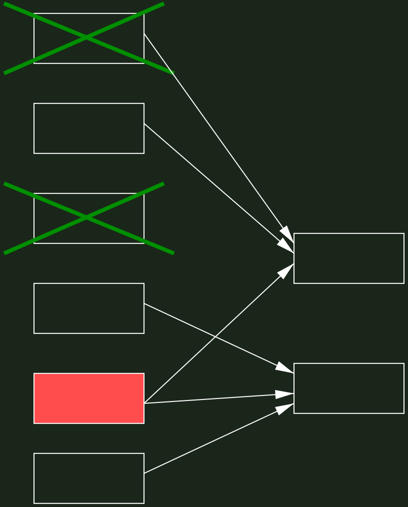
Distributed Storage Systems



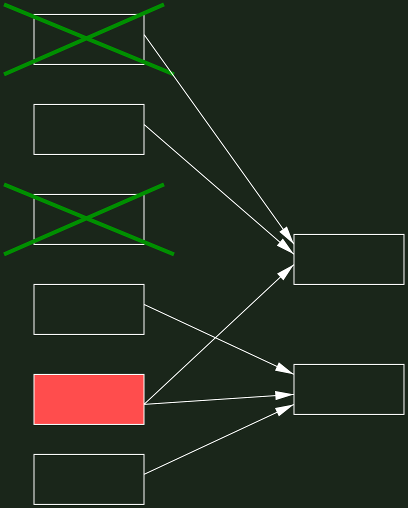
Distributed Storage Systems



Distributed Storage Systems

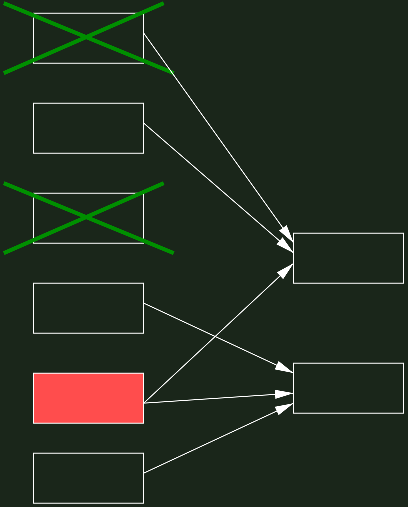


Distributed Storage Systems



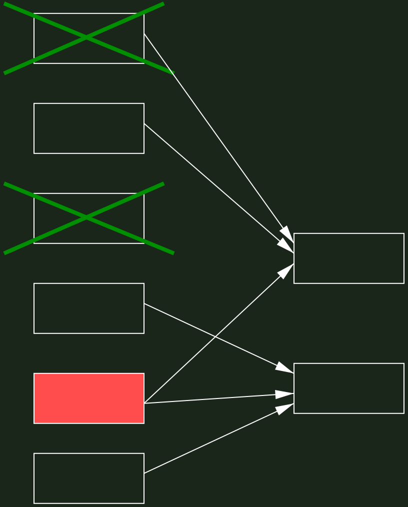
- Single adversarial node may transmit many times

Distributed Storage Systems



- Single adversarial node may transmit many times
- Naturally suited to the node-based adversary model

Distributed Storage Systems



- Single adversarial node may transmit many times
- Naturally suited to the node-based adversary model
- Functional repair rather than exact repair

Parameters

- α: Storage capacity of single node
- β: Download bandwidth when forming new node
- n: Number of active storage nodes
- k: Number of nodes contacted by data collector (DC) to recover file
- d: Number of nodes contacted to construct new node
- z: Number of (simultaneous) adversarial nodes

Existing Bounds

- Pawar-El Rouayheb-Ramchandran (2011): Storage capacity is upper bounded by

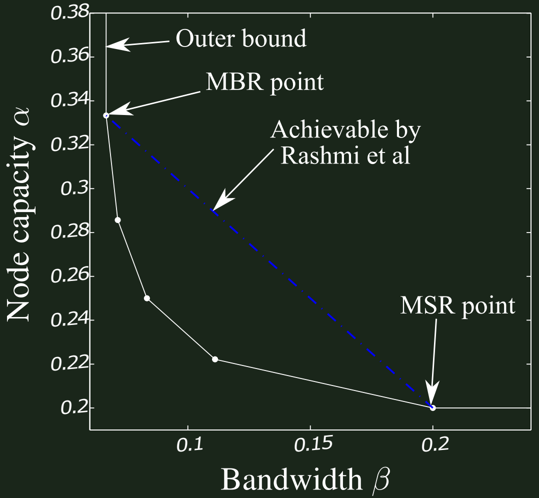
$$C \leq \sum_{i=0}^{k-2z-1} \min\{(d-2z-i)\beta, \alpha\}$$

  Identical to bound without adversaries where $k \to k-2z$ and $d \to d-2z$

- Rashmi et al (2012): The Minimum Storage Regeneration (MSR) and Minimum Bandwidth Regeneration (MBR) points are achievable with exact repair

Existing Bounds, Ctd.

Parameters: $n = 8$, $k = d = 7$, $z = 1$



Structure of Polytope Code for DSS

- Initial file to store $m \in \{1, 2, \ldots, 2^k\}^{R \times N}$

Structure of Polytope Code for DSS

- Initial file to store $m \in \{1, 2, \ldots, 2^k\}^{R \times N}$
- Covariance $\Sigma^\star = m\, m^T$

Structure of Polytope Code for DSS

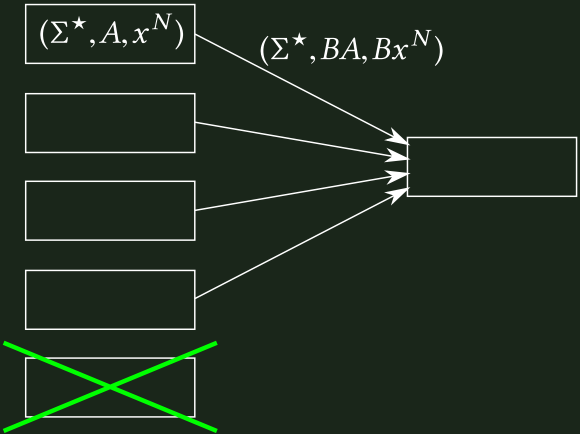
- Initial file to store $m \in \{1, 2, \ldots, 2^k\}^{R \times N}$
- Covariance $\Sigma^\star = m\, m^T$
- All packets are of the form $(\Sigma^\star, A, x^N)$ where initially $x^N = Am$

Structure of Polytope Code for DSS

- Initial file to store $m \in \{1, 2, \ldots, 2^k\}^{R \times N}$
- Covariance $\Sigma^\star = m\, m^T$
- All packets are of the form $(\Sigma^\star, A, x^N)$ where initially $x^N = Am$
- For storage packet $x^N \in \{1, 2, \ldots, 2^k\}^{\alpha \times N}$
  For transmission packet $x^N \in \{1, 2, \ldots, 2^k\}^{\beta \times N}$

Messages for new node

Choose linear transformation $B \in \mathbb{Z}^{\beta \times \alpha}$



New Node Construction

Given $(\Sigma^\star, A_i, y_i^N)$ for $i = 1, 2, \ldots, d$

New Node Construction

Given $(\Sigma^\star, A_i, y_i^N)$ for $i = 1, 2, \ldots, d$

- Recover $\Sigma^\star$ using majority rule

New Node Construction

Given $(\Sigma^\star, A_i, y_i^N)$ for $i = 1, 2, \ldots, d$

- Recover $\Sigma^\star$ using majority rule
- Form $A = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_d \end{bmatrix}$ and $y^N = \begin{bmatrix} y_1^N \\ y_2^N \\ \vdots \\ y_d^N \end{bmatrix}$

New Node Construction

Given $(\Sigma^\star, A_i, y_i^N)$ for $i = 1, 2, \ldots, d$

- Recover $\Sigma^\star$ using majority rule
- Form $A = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_d \end{bmatrix}$ and $y^N = \begin{bmatrix} y_1^N \\ y_2^N \\ \vdots \\ y_d^N \end{bmatrix}$
- Compare $A\Sigma^\star A^T$ to $\Sigma_y = (y^N)(y^N)^T$

New Node Construction

Given $(\Sigma^\star, A_i, y_i^N)$ for $i = 1, 2, \ldots, d$

- Recover $\Sigma^\star$ using majority rule
- Form $A = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_d \end{bmatrix}$ and $y^N = \begin{bmatrix} y_1^N \\ y_2^N \\ \vdots \\ y_d^N \end{bmatrix}$
- Compare $A\Sigma^\star A^T$ to $\Sigma_y = (y^N)(y^N)^T$
- Form syndrome graph on the vertex set $\{1, 2, \ldots, d\}$ with edge (i, j) if

$$\begin{bmatrix} A_i \\ A_j \end{bmatrix} \Sigma^\star \begin{bmatrix} A_i \\ A_j \end{bmatrix}^T = \begin{bmatrix} y_i^N \\ y_j^N \end{bmatrix} \begin{bmatrix} y_i^N \\ y_j^N \end{bmatrix}^T$$

New Node Construction

Given $(\Sigma^\star, A_i, y_i^N)$ for $i = 1, 2, \ldots, d$

- Recover $\Sigma^\star$ using majority rule
- Form $A = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_d \end{bmatrix}$ and $y^N = \begin{bmatrix} y_1^N \\ y_2^N \\ \vdots \\ y_d^N \end{bmatrix}$
- Compare $A\Sigma^\star A^T$ to $\Sigma_y = (y^N)(y^N)^T$
- Form syndrome graph on the vertex set $\{1, 2, \ldots, d\}$ with edge (i, j) if

$$\begin{bmatrix} A_i \\ A_j \end{bmatrix} \Sigma^\star \begin{bmatrix} A_i \\ A_j \end{bmatrix}^T = \begin{bmatrix} y_i^N \\ y_j^N \end{bmatrix} \begin{bmatrix} y_i^N \\ y_j^N \end{bmatrix}^T$$

- Goal: Find trustworthy packets from which to form stored data

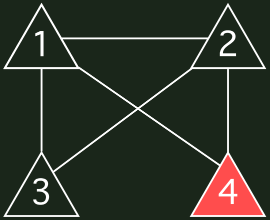
Syndrome Graphs

The honest nodes form a clique of size $d - z$

Syndrome Graphs

The honest nodes form a clique of size $d - z$

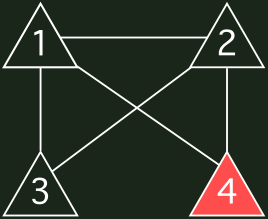
Example: $d = 4$ and $z = 1$:



Syndrome Graphs

The honest nodes form a clique of size $d - z$

Example: $d = 4$ and $z = 1$:

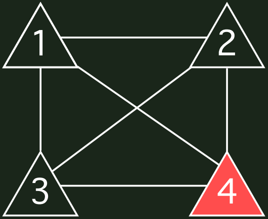


- Use packets 1 and 2 to form stored data
- This is the typical case where $d - 2z$ trustworthy packets can be identified

Syndrome Graphs

The honest nodes form a clique of size $d - z$

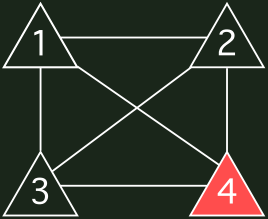
Example: $d = 4$ and $z = 1$:



Syndrome Graphs

The honest nodes form a clique of size $d - z$

Example: $d = 4$ and $z = 1$:



- Use all packets to form stored data
- Linear constraints (because covariances match) mean the adversary data is uncorrupted

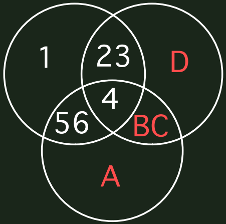
Syndrome Graphs

The honest nodes form a clique of size $d - z$

Example: $d = 10$ and $z = 4$

- Call honest nodes 1,2,3,4,5,6 and adversary nodes A,B,C,D
- Three cliques of size 6:

123456
456ABC
234BCD



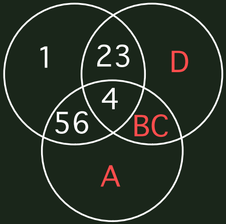
Syndrome Graphs

The honest nodes form a clique of size $d - z$

Example: $d = 10$ and $z = 4$

- Call honest nodes 1,2,3,4,5,6 and adversary nodes A,B,C,D
- Three cliques of size 6:

123456
456ABC
234BCD



- Use packet 4 to form stored data
- Less than $d - 2z$ trustworthy packets!

Algorithm to find trustworthy packets

1. Discard all packets not in a clique of size $d - z$
2. Pick packets i where edge (i, j) is in the syndrome graph for all remaining packets j

Algorithm to find trustworthy packets

1. Discard all packets not in a clique of size $d - z$
2. Pick packets i where edge (i, j) is in the syndrome graph for all remaining packets j

- Any chosen adversarial packet must match covariances with all $d - z$ honest nodes

Algorithm to find trustworthy packets

1. Discard all packets not in a clique of size $d - z$
2. Pick packets i where edge (i, j) is in the syndrome graph for all remaining packets j

- Any chosen adversarial packet must match covariances with all $d - z$ honest nodes
- If $R \leq (d - z)\beta$, then linear constraints ensure all stored data is uncorrupted

Algorithm to find trustworthy packets

1. Discard all packets not in a clique of size $d - z$
2. Pick packets i where edge (i, j) is in the syndrome graph for all remaining packets j

- Any chosen adversarial packet must match covariances with all $d - z$ honest nodes
- If $R \leq (d - z)\beta$, then linear constraints ensure all stored data is uncorrupted
- This procedure always finds at least $d - v_z$ packets where

$$v_z = (\lfloor \tfrac{z}{2} \rfloor + 1)(\lceil \tfrac{z}{2} \rceil + 1)$$

z	1	2	3	4	5	6
v_z	2	4	6	9	12	16

Note $v_z = 2z$ only for $z \leq 3$

Resulting Achievability Bound

Theorem (Kosut (2013))

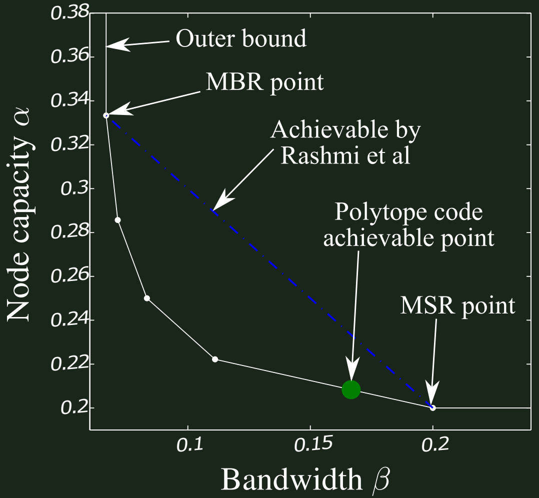
There exists a distributed storage code achieving rate

$$\min\left\{\sum_{i=0}^{k-v_z-1}\min\{(d-v_z-i)\beta,\alpha\},\ (d-z)\beta,\ (k-z)\alpha\right\}.$$

where $v_z = (\lfloor \frac{z}{2} \rfloor + 1)(\lceil \frac{z}{2} \rceil + 1)$.

Achievability Plot

Parameters: $n = 8$, $k = d = 7$, $z = 1$

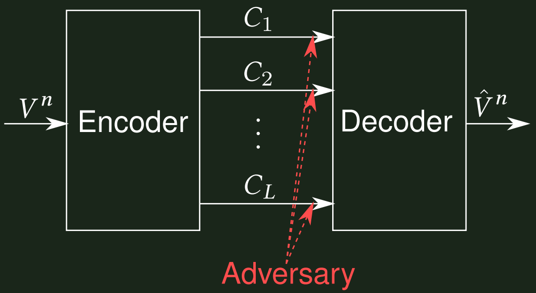


Outline

- Polytope codes in general
- Polytope codes in network coding
- Polytope codes in distributed storage systems
- Polytope codes in multiple descriptions

Adversarial Multiple Descriptions

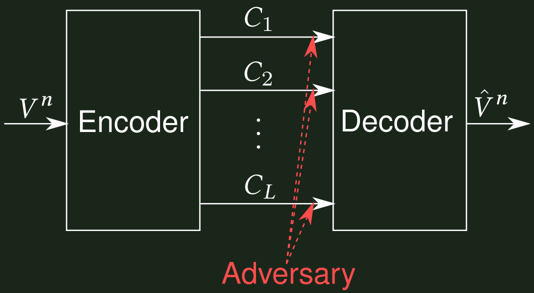
Problem formulated in Fan-Wagner-Ahmed (2013)



Construct a single code that fails gracefully — fewer adversarial packets gives smaller distortion

Adversarial Multiple Descriptions

Problem formulated in Fan-Wagner-Ahmed (2013)

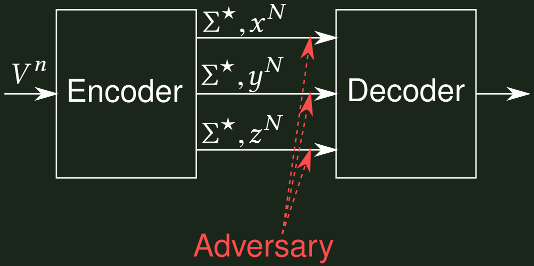


Construct a single code that fails gracefully — fewer adversarial packets gives smaller distortion

- $V^n \in \{0,1\}^n$
- $C_i \in \{1, 2, \ldots, 2^{nR}\}$
- Adversary controls z packets
- Distortion: $D = \sum_{i=1}^{n} d(X_i, \hat{X}_i)$ where d is the erasure distortion

3-Description Example

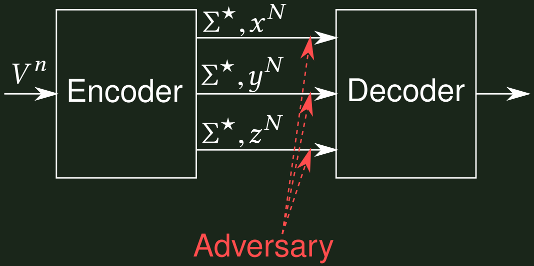
- $R = 1/2$
- Write $V^n = (x^N, y^N)$ where $x^N, y^N \in \{1, 2, \ldots, 2^k\}^N$
- $z^N = x^N + y^N$



- If $z = 0$, then entire source sequence can be decoded, so $D = 0$
- If $z = 1$, then one trustworthy packet (half the message) can be identified, so $D = 1/2$

3-Description Example

- $R = 1/2$
- Write $V^n = (x^N, y^N)$ where $x^N, y^N \in \{1, 2, \ldots, 2^k\}^N$
- $z^N = x^N + y^N$



- If $z = 0$, then entire source sequence can be decoded, so $D = 0$
- If $z = 1$, then one trustworthy packet (half the message) can be identified, so ~~$D = 1/2$~~

  Problem: z^N is not a systematic part of source V^n

3-Description Example

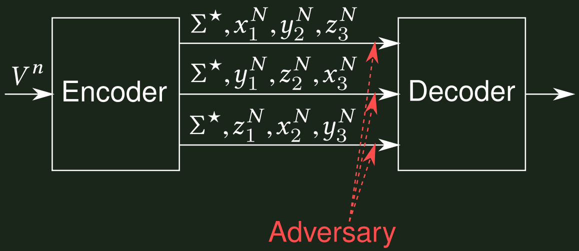
- $V^n = (V_1^{n/3}, V_2^{n/3}, V_3^{n/3})$, and write $V_i^{n/3} = (x_i^N, y_i^N)$

3-Description Example

- $V^n = (V_1^{n/3}, V_2^{n/3}, V_3^{n/3})$, and write $V_i^{n/3} = (x_i^N, y_i^N)$
- $z_i^N = x_i^N + y_i^N$ for $i = 1, 2, 3$

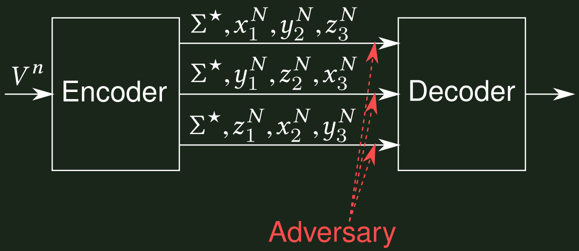
3-Description Example

- $V^n = (V_1^{n/3}, V_2^{n/3}, V_3^{n/3})$, and write $V_i^{n/3} = (x_i^N, y_i^N)$
- $z_i^N = x_i^N + y_i^N$ for $i = 1, 2, 3$



3-Description Example

- $V^n = (V_1^{n/3}, V_2^{n/3}, V_3^{n/3})$, and write $V_i^{n/3} = (x_i^N, y_i^N)$
- $z_i^N = x_i^N + y_i^N$ for $i = 1, 2, 3$



- Decoder can always identify one trustworthy packet, containing two systematic parts of V^n
- Thus $D = 2/3$

Conclusions

- Polytope codes operate on the integers and can mimic most finite field codes
- Covariances are used as checksums, allowing for:
  - Partial decoding
  - Distributed error detection/correction
- Polytope codes outperform finite field codes, but many achievable results have no matching converse
  — seems to be very hard to find the best polytope code
- All results for omniscient adversary — weaker adversary models require different techniques