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
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 $M \in \mathbb{F}$

Message $M$ \hspace{1cm} Encoder \hspace{1cm} Decoder

$M$ $M$ $M$

Adversary

Achieves $R = 1$
(3,2) MDS code: Let \( M = (x, y) \), \( x, y \in \mathbb{F} \)
(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 $R = 2$?

$H(X_i, X_j) = H(M) = 2$.

Thus $I(X_i; X_j) = 0$.

But if the packets are pairwise independent, then the 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 $R = 2$?

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

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

$H(X_i, X_j) = H(M) = 2q \left(2 - \epsilon\right)q$

Thus $I(X_i; X_j) = \emptyset \epsilon q$

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
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 \text{ 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 $z^N = x^N + y^N$  \[[x^N, y^N, z^N \text{ sit in a polytope}]\]
- Construct the covariance

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

$\Sigma^*$ 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\text{ sit in a polytope}]
- Construct the covariance

\[
\Sigma^* = \begin{bmatrix}
    x^N \\
y^N \\
z^N
\end{bmatrix}
\begin{bmatrix}
x^N \\
y^N \\
z^N
\end{bmatrix}^T
= \begin{bmatrix}
    \langle x^N, x^N \rangle & \langle x^N, y^N \rangle & \langle x^N, z^N \rangle \\
    \langle x^N, y^N \rangle & \langle y^N, y^N \rangle & \langle y^N, z^N \rangle \\
    \langle x^N, z^N \rangle & \langle y^N, z^N \rangle & \langle z^N, z^N \rangle
\end{bmatrix}
\]

- $\Sigma^*$ 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

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

- $\Sigma^*$ takes infinitesimal rate compared to $x^N$ for large $N$
Thus $x^N, y^N, z^N$ are nearly pairwise independent. ($x^N, y^N, z^N$) form a $(3, 2)$ MDS polytope code.
$x^N, y^N \in \{1, 2, \ldots, 2^k\}^N$: Number of bits = $kN$
- $x^N, y^N \in \{1, 2, \ldots, 2^k\}^N$: 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
\[
\begin{align*}
\text{MDS structure} \\
\begin{array}{l}
\text{\(x_N, y_N \in \{1, 2, \ldots, 2^k\}^N\): Number of bits = } kN \\
\text{\(z_N \in \{1, 2, \ldots, 2^{k+1}\}^N\): Number of bits = } (k + 1)N \approx kN \text{ for large } k \\
\text{Thus } x_N, y_N, z_N \text{ are nearly pairwise independent} \\
\text{(}x_N, y_N, z_N) \text{ form a } (3, 2) \text{ MDS polytope code}
\end{array}
\end{align*}
\]
Decoding

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

Given $x^N$, $y^N$, $z^N$ form the actually-is covariance $\Sigma = \begin{bmatrix} \langle x^N, x^N \rangle & \langle x^N, y^N \rangle & \langle x^N, z^N \rangle \\ \langle y^N, y^N \rangle & \langle y^N, z^N \rangle \\ \langle z^N, z^N \rangle \end{bmatrix}$

By comparing $\Sigma^*$ with $\Sigma$, the decoder can always find a trustworthy packet.
Decoding

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

Message $M$ → Encoder $\Sigma^*, x^N$ $\Sigma^*, y^N$ $\Sigma^*, z^N$ → Decoder

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

Given $x^N, y^N, z^N$ form the actually-is covariance

$$\Sigma = \begin{bmatrix}
    \langle x^N, x^N \rangle & \langle x^N, y^N \rangle & \langle x^N, z^N \rangle \\
    \langle x^N, y^N \rangle & \langle y^N, y^N \rangle & \langle y^N, z^N \rangle \\
    \langle x^N, z^N \rangle & \langle y^N, z^N \rangle & \langle z^N, z^N \rangle
\end{bmatrix}$$
Recover the should-be covariance $\Sigma^*$ 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 $\Sigma^*$ with $\Sigma$, the decoder can always find a trustworthy packet.
Suppose $\Sigma \neq \Sigma^*$:
Suppose $\Sigma \neq \Sigma^*$:

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

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

- If $\Sigma_{xy} \neq \Sigma^*_{xy}$, then either $x^N$ or $y^N$ is corrupted — $z^N$ is safe.
Decoding

Suppose $\Sigma \neq \Sigma^*$:

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

All quadratic functions of $x_N$, $\frac{1}{y.N}$, $z_N$ must be uncorrupted. Therefore all packets are trustworthy.
Decoding

Suppose $\Sigma = \Sigma^*$:

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

Suppose $\Sigma = \Sigma^*$:

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

$$\|x^N + y^N - z^N\|^2 = 0 \implies x^N + y^N - z^N = 0$$
Suppose $\Sigma = \Sigma^*$:

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

- $\left\| x^N + y^N - z^N \right\|^2 = 0 \implies x^N + y^N - z^N = 0$

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

Calculate covariance $\Sigma^{\star} = mm^T$ — included in all packets

Packet data is in the form $x_N = a^T_m$ for integer vector $a \in \mathbb{Z}^R$

$x_i = \sum_j a_j m_{ji}$

$1 \leq \sum_j a_j 2^{k} \leq 2^k + \Delta$ for sufficiently large $k$ — requires $(k + \Delta)N$ bits to store

These constructions can mimic most finite field linear codes
Generic polytope code constructions

- Message $m \in \{1, 2, \ldots, 2^k\}^{R \times N}$
- Calculate covariance $\Sigma^* = mm^T$ — included in all packets
Generic polytope code constructions

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

Calculate covariance $\Sigma^* = mm^T$ — 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 \( \Sigma^* = mm^T \) — included in all packets

- Packet data is in the form \( x^N = a^Tm \) for integer vector \( a \in \mathbb{Z}^R \)

- \( x_i = \sum_j a_jm_{ji} \leq \sum_j a_j2^k \leq 2^{k+\Delta} \) 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 $y^N = \begin{bmatrix} x_1^N \\ x_2^N \\ \vdots \\ x_p^N \end{bmatrix} = Am$
Given some subset of packets $y^N = \begin{bmatrix} x_1^N \\ x_2^N \\ \vdots \\ x_p^N \end{bmatrix} = Am$

- Form $\Sigma_y = (y^N) (y^N)^T$
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, $\Sigma_y = A\Sigma^*A^T$
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, $\Sigma_y = A\Sigma^*A^T$
- If $\Sigma \neq A^T\Sigma^*A$, 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, \( \Sigma_y = A\Sigma^*A^T \)
- If \( \Sigma \neq A^T\Sigma^*A \), then corrupted packets may be localized
- If \( \Sigma = A^T\Sigma^*A \), 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
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:

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

Source: Add redundancy

Network: Linear Coding

Destination: Error Correction

Polytope codes allow error detection/correction inside the network
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:

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

$$z^N = x^N + y^N$$
$$w^N = x^N + 2y^N$$
$$\Sigma^* = mm^T$$
Let message $m = (x^N, y^N)$, where $x^N, y^N \in \{1, \ldots, 2^k\}^N$

\[
\begin{align*}
z^N &= x^N + y^N \\
w^N &= x^N + 2y^N \\
\Sigma^* &= mm^T
\end{align*}
\]

$(x^N, y^N, z^N, w^N)$ form a $(4,2)$ MDS polytope code
Let message $m = (x^N, y^N)$, where $x^N, y^N \in \{1, \ldots, 2^k\}^N$.

$z^N = x^N + y^N$

$w^N = x^N + 2y^N$

$\Sigma^* = mm^T$

$(x^N, y^N, z^N, w^N)$ form a $(4,2)$ MDS polytope code.
Let message $m = (x^N, y^N)$, where $x^N, y^N \in \{1, \ldots, 2^k\}^N$

$$
\begin{align*}
  z^N &= x^N + y^N \\
  w^N &= x^N + 2y^N \\
  \Sigma^* &= mm^T
\end{align*}
$$

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

\[
z^N = x^N + y^N
\]
\[
w^N = x^N + 2y^N
\]
\[
\Sigma^* = mm^T
\]

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

$$z^N = x^N + y^N$$
$$w^N = x^N + 2y^N$$
$$\Sigma^* = mm^T$$

$(x^N, y^N, z^N, w^N)$ 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
One node is controlled by the adversary — controls all outgoing messages
One node is controlled by the adversary — controls all outgoing messages

Let \((x^N, y^N, z^N, w^N, \nu^N, u^N)\) be a \((6, 2)\) MDS polytope code
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

\(\Sigma\) included in all packets
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

\(\Sigma^*\) included in all packets

Nodes 4 and 5 compare covariance of incoming pair of packets — transmit outcome of comparison
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
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

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

Single adversarial node may transmit many times

Naturally suited to the node-based adversary model

Functional repair rather than exact repair
Distributed Storage Systems

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

Single adversarial node may transmit many times

Naturally suited to the node-based adversary model

Functional repair rather than exact repair
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

- $\alpha$: Storage capacity of single node
- $\beta$: 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: $n = 8$, $k = d = 7$, $z = 1$

Outer bound

MBR point

Achievable by Rashmi et al

MSR point
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 $\Sigma^* = mm^T$
Structure of Polytope Code for DSS

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

Covariance $\Sigma^* = mm^T$

All packets are of the form $(\Sigma^*, A, x^N)$ where initially $x^N = Am$

For storage packet $x^N \in \{1, 2, \ldots, 2^k\}^{\alpha \times N}$
For transmission packet $x^N \in \{1, 2, \ldots, 2^k\}^{\beta \times N}$
Choose linear transformation $B \in \mathbb{Z}^{\beta \times \alpha}$
New Node Construction

Given $(\Sigma^*, A_i, y_i^N)$ for $i = 1, 2, \ldots, d$
New Node Construction

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

- Recover \(\Sigma^*\) using majority rule
New Node Construction

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

- Recover \(\Sigma^*\) 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 \(A \Sigma^* A^T\) to \(\Sigma^* = (y_i^N)(y_i^N)^T\)

Form syndrome graph on the vertex set \(\{1, 2, \ldots, d\}\) with edge \((i, j)\) if \(\begin{bmatrix} A_i \\ A_j \end{bmatrix} \Sigma^* \begin{bmatrix} A_i \\ A_j \end{bmatrix}^T = \begin{bmatrix} y_i^N \\ y_j^N \end{bmatrix} \begin{bmatrix} y_i^N \\ y_j^N \end{bmatrix}^T\)

Goal: Find trustworthy packets from which to form stored data
New Node Construction

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

- Recover \(\Sigma^*\) 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 \(A\Sigma^*A^T\) to \(\Sigma_y = (y^N)(y^N)^T\)
New Node Construction

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

- Recover \(\Sigma^*\) 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 \(A \Sigma^* A^T\) to \(\Sigma_y = (y^N)(y^N)^T\)

- Form syndrome graph on the vertex set \(\{1, 2, \ldots, d\}\) with edge \((i, j)\) if

\[
\begin{bmatrix} A_i \\ A_j \end{bmatrix} \Sigma^* \begin{bmatrix} A_i \\ A_j \end{bmatrix}^T = \begin{bmatrix} y_i^N \\ y_j^N \end{bmatrix} \begin{bmatrix} y_i^N \\ y_j^N \end{bmatrix}^T
\]
New Node Construction

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

- Recover \(\Sigma^*\) 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 \(A\Sigma^*A^T\) to \(\Sigma_y = (y^N)(y^N)^T\)

- Form syndrome graph on the vertex set \(\{1, 2, \ldots, d\}\) with edge \((i, j)\) if

\[
\begin{bmatrix} A_i \\ A_j \end{bmatrix}\Sigma^*\begin{bmatrix} A_i \\ A_j \end{bmatrix}^T = \begin{bmatrix} y_i^N \\ y_j^N \end{bmatrix}\begin{bmatrix} y_i^N \\ y_j^N \end{bmatrix}^T
\]

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

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

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

![Diagram](image-url)
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:

  1. 123456
  2. 456ABC
  3. 234BCD

\[1\]
\[23\]
\[4\]
\[BC\]
\[56\]
\[A\]
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 $d - \nu_z$ packets where

\[
\nu_z = (\lfloor \frac{z}{2} \rfloor + 1)(\lceil \frac{z}{2} \rceil + 1)
\]

<table>
<thead>
<tr>
<th>$z$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\nu_z$</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>9</td>
<td>12</td>
<td>16</td>
</tr>
</tbody>
</table>

Note $\nu_z = 2z$ 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)

\[ V^n \rightarrow \text{Encoder} \rightarrow C_1 \rightarrow \cdots \rightarrow C_L \rightarrow \text{Decoder} \rightarrow \hat{V}^n \]

Adversary controls \( z \) packets

Distortion:
\[ D = \sum_{i=1}^{n} d(X_i, \hat{X}_i) \]
where \( d \) is the erasure distortion

Construct a single code that \textit{fails gracefully} — fewer adversarial packets gives smaller distortion
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 $V^n$
$V^n = (V_1^{n/3}, V_2^{n/3}, V_3^{n/3})$, 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 \( V_{i/n}^{n/3} = (x_i^N, y_i^N) \)
- \( z_i^N = x_i^N + y_i^N \) for \( i = 1, 2, 3 \)
\[ V^n = \left( V_1^{n/3}, V_2^{n/3}, V_3^{n/3} \right), \text{ and write } V_i^{n/3} = (x_i^N, y_i^N) \]

\[ z_i^N = x_i^N + y_i^N \text{ for } i = 1, 2, 3 \]

Decoder can always identify one trustworthy packet, containing two systematic parts of \( V^n \). Thus \( D = \frac{2}{3} \).
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.