On data-hiding and network coding
Security aspects of network coding

• Network coding operates by allowing mixing of data
• What are the security consequences of such mixtures?
• Three aspects:
  – Data hiding
  – Byzantine or pollution attacks – detection and correction
  – Verification
• We concentrate here on the first aspect
To what extent does network coding protect data?

- Mixture of data provides, at least locally, the equivalent of the one-time pad [CaiYeung02]
- Random network coding provides some data hiding
  - Trade-off between vulnerability and cost of transmissions
  - For low vulnerability, network coding decreases vulnerability and improves the cost-vulnerability trade-off
  - When there is high vulnerability because several nodes may collude, random network coding may increase vulnerability [TanMedard06]
- To consider such types of data hiding, we may envisage more careful consideration of the transfer matrix
  - No recovery – no Gaussian elimination
  - Partial or full recovery – partial or full diagonalization [LimaMedardBarros07]
- How to use network coding for good cryptographic approaches:
  - Protect the code [LimaBarrosVilelaMedard08] (we address this here)
  - Use network coding for verification without decoding or returning to trusted authority [ZhangKalkerMedardHan07]
Trade-off between cost and security

Four possible schemes for data transmission \((p = 0.01)\)

**Single path routing**

![Diagram of single path routing]

Cost: 2 (lowest), Vul: 0.020 (highest)

**Multipath routing**

![Diagram of multipath routing]

Cost: 2.75, Vul: \(5.9 \times 10^{-4}\)

**Single path coding**

![Diagram of single path coding]

Cost: 3.5 (highest), Vul: 0.010

**Multipath coding**

![Diagram of multipath coding]

Cost: 2.75, Vul: \(2.1 \times 10^{-5}\) (lowest)

\(p\) is the probability of being tapped
Problem formulation

- Wiretapper is interested in knowing $k \rho$ of the $r \rho$ transmitted processes
  - $\mathcal{W}_{\text{interest}}$ — the set of processes of interest
  - $p$ — probability of tapping any single link
  - $\mathcal{Y}_{\text{tapped}}$ — the random set of messages retrieved by the wiretapper

- Two performance parameters
  - Overall network cost, $\mu = \sum_{\{l \in E\}} c_l z_l$
  - Network vulnerability, $\nu = P[H(\mathcal{W}_{\text{interest}} | \mathcal{Y}_{\text{tapped}}) = 0]$

- Aim: minimize $\mathcal{F}(\mu, \nu)$, an increasing function of both $\mu$ and $\nu$
Problem approach

- Employ random linear network codes [HMSEK03]
- Minimize $\mathcal{F}(\mu, \nu)$, subject to:
  - Capacity constraints
  - Flow constraints on the virtual flows
- A general solution is difficult
  - Characteristics of $\mathcal{F}(\mu, \nu)$ are unknown
  - $\nu$ depends on the actual network code
Minimizing cost and finding disjoint paths

- Consider the graph representation $G = (V, E)$
- Each edge $l$ in $E$ has
  - Edge capacity of $d_l$
  - Cost per unit flow of $c_l$
  - Actual flow rate of $z_l$
  - Probability of being tapped of $p_l$
- $r$ independent, discrete random processes $W_1, W_2, \ldots, W_r$
  - Generated at source nodes $s_1, s_2, \ldots, s_r$
  - To be transmitted to $T = \{t_1, t_2, \ldots, t_{|T|}\}$
  - Each process has integer entropy rate of $\rho$
  - Can decompose $W_i$ into $\rho$ independent, discrete random processes with unit entropy rates
    \[ W_i = X_{(i-1)\rho+1}, X_{(i-1)\rho+2}, \ldots, X_{i\rho} \]
- To find subgraph of minimum cost:
Linear Cost Minimization

- Minimize

\[ \sum_{l \in E'} c_l z_l \]

subject to

\[ z_l = \rho \quad \forall l \text{ s.t. } \text{tail}(l) = \alpha \]

\[ z_l \geq x^{(t)}_l \quad \forall l \in E', t \in T \]

\[ \sum_{\{l : \text{head}(l) = v\}} x^{(t)}_l - \sum_{\{l : \text{tail}(l) = v\}} x^{(t)}_l = \sigma^{(t)}_v \quad \forall v \in V', t \in T \]

\[ d_l \geq x^{(t)}_l \geq 0 \quad \forall l \in E', t \in T \]

\[ \sigma^{(t)}_v = \begin{cases} r\rho & \text{if } v = \alpha \\ -r\rho & \text{if } v = t \\ 0 & \text{otherwise} \end{cases} \]
Convex Cost Minimization

- Minimize

$$\sum_{\{l \in E\}} \left( c_l z_l + \omega p_l z_l^2 \right)$$

subject to

$$z_l = \rho \quad \forall l \text{ s.t. } \text{tail}(l) = \alpha$$

$$z_l \geq x_{l}^{(t)} \quad \forall l \in E', t \in T$$

$$\sum_{\{l: \text{head}(l) = v\}} x_{l}^{(t)} - \sum_{\{l: \text{tail}(l) = v\}} x_{l}^{(t)} = \sigma_v^{(t)} \quad \forall v \in V', t \in T$$

where

$$d_l \geq x_l^{(t)} \geq 0 \quad \forall l \in E', t \in T$$

$$\sigma_v^{(t)} = \begin{cases} r\rho & \text{if } v = \alpha \\ -r\rho & \text{if } v = t \\ 0 & \text{otherwise} \end{cases}$$
Network coding model

- Directed graph $G' = (V', E')$ with unit capacity edges
- Random linear network coding
  - Information can be represented as symbols in $F_2$
  - Signal transmitted in edge $l$ is denoted $Y(l)$
    \[
    Y(l) = \sum_{\{i: X_i \text{ generated at tail}(l)\}} a_{i,l} X_i + \sum_{\{j: \text{head}(j) = \text{tail}(l)\}} f_{j,l} Y(j)
    \]
  - The $i$-th signal received at sink $t$ is denoted $Z(t,i)$
    \[
    Z(t,i) = \sum_{\{l: \text{head}(l) = i\}} b_{i,l} Y(l)
    \]
  - where $a_{i,l}$, $f_{j,l}$, and $b_{i}$ are random symbols in $F_2$
Network security

• Security against a wiretapping adversary
  – Interested in the first $k$ of $r$ messages ($W_1$, $W_2$, …, $W_k$)
  – Able to tap some unknown set of edges

\[ E_{tap} = \{ l_1, l_2, \ldots, l_{|E_{tap}|} \} \subseteq E \]

• Perfect information security criterion:

\[ H\left( W_1, W_2, \ldots, W_k \left| Y(l_1), Y(l_2), \ldots, Y(l_{|E_{tap}|}) \right. \right) = H\left( W_1, W_2, \ldots, W_k \right) \]

• Probability of full retrieval of messages of interest:

\[ P\left[ H\left( W_1, W_2, \ldots, W_k \left| Y(l_1), Y(l_2), \ldots, Y(l_{|E_{wp}|}) \right. \right) = 0 \right] \]
Results ($p = 0.02, r = 4, k = 1$)

Plot of vulnerability against cost for $p = 0.02, r = 4, k = 1$.
Results ($p = 0.1, r = 4, k = 1$)

Plot of vulnerability against cost for $p = 0.1, r = 4, k = 1$
Results – Comparing $p = 0.1$ and $p = 0.02$

- Cost of network coding is about half that of routing
- For $p = 0.02$:
  - Network coding yields lower network vulnerability
  - For network coding, the vulnerability decreases with increasing cost (and number of used paths)
- For $p = 0.1$:
  - Network coding yields higher network vulnerability
  - For network coding, the vulnerability increases with increasing cost (and number of used paths)
- Feasibility of this secure network coding scheme depends heavily on $p$
Results – Comparing SPC with MPC

Plot of vulnerability differences between SPC and MPC against $p$

- $r=2$
- $r=4$
- $r=6$
Results – Comparing SPR with SPC

Graph of vulnerability against $p$ for SPR and SPC

- SPR
- SPC, $r=2$
- SPC, $r=4$
- SPC, $r=6$
Results – Comparing MPR with MPC

Graph of vulnerability against \( p \) for MPR and MPC

- SPR
- MPR
- MPC, \( r=2 \)
- MPC, \( r=4 \)
- MPC, \( r=6 \)
Results – Comparing routing with coding

• SPC usually yields higher security than SPR for $0 \leq p \leq 0.1$

• MPC starts off with better performance than MPR
  – But as $p$ increases, it starts to fare worse than MPR
  – And eventually fares worse than SPR

• Two conflicting factors:
  – Individually, the messages sent in the network do not reveal much about the secure messages of interest
  – However, you just need to obtain the right number of degrees of freedom to figure out the secure messages
Heuristic approaches

- First heuristic
  - If every tapped process is unique and is a mixture of all $r \rho$ input processes
  - Then, $\nu$ is equal to $\nu'$, the probability that at least $r \rho$ processes are tapped
  - Minimize $\mu + \omega \nu'$

- Second heuristic
  - Attempt to decrease vulnerability by spreading information flow across links
  - Introduce to each link, a strictly convex and increasing cost function
  - Minimize $\mu + \omega' \mu_{convex}$
Simulation results for sample ISPs

Network vulnerabilities for three different networks

<table>
<thead>
<tr>
<th></th>
<th>Exodus</th>
<th></th>
<th>Telstra</th>
<th></th>
<th>Tiscali</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$p = 0.01$</td>
<td>$p = 0.1$</td>
<td>$p = 0.01$</td>
<td>$p = 0.1$</td>
<td>$p = 0.01$</td>
<td>$p = 0.1$</td>
</tr>
<tr>
<td>Single path routing</td>
<td>0.11</td>
<td>0.67</td>
<td>0.084</td>
<td>0.59</td>
<td>0.090</td>
<td>0.62</td>
</tr>
<tr>
<td>Multipath routing</td>
<td>0.025</td>
<td>0.39</td>
<td>0.044</td>
<td>0.45</td>
<td>0.047</td>
<td>0.47</td>
</tr>
<tr>
<td>Single path coding</td>
<td>0.054</td>
<td>0.62</td>
<td>0.060</td>
<td>0.59</td>
<td>0.083</td>
<td>0.69</td>
</tr>
<tr>
<td>Heuristic 1</td>
<td>0.012</td>
<td>0.57</td>
<td>0.028</td>
<td>0.55</td>
<td>0.023</td>
<td>0.60</td>
</tr>
<tr>
<td>Heuristic 2</td>
<td>0.013</td>
<td>0.73</td>
<td>0.024</td>
<td>0.60</td>
<td>0.068</td>
<td>0.86</td>
</tr>
</tbody>
</table>

- Simulation values: $r = 4$, $\rho = 10$, $k = 1$, $|T| = 4$, $\omega = 3 \times 10^5$, $\omega' = 300$
- Added quadratic cost function for second heuristic
- When $p$ is small, coding tend to perform better than routing
- When $p$ is small, heuristics tend to perform better than the other methods

Random network coding does not necessarily outperform other methods when there is high vulnerability
Simulation results - Exodus

Plot of $\mu + \omega V$ against $p$ (linear scale)

- Single path routing
- Multi path routing
- Single path coding
- Heuristic 1
- Heuristic 2
Simulation results - Exodus

Plot of $\mu v$ against $p$ (linear scale)

- Single path routing
- Multi path routing
- Single path coding
- Heuristic 1
- Heuristic 2
Refining the analysis

Network: DAG $G = (V, E)$

Network links
- no delays, losses
- unit capacity

$Y(e) = \sum_{l: X_l \text{ generated at } v} \alpha_{l,e} X(v, l) + \sum_{e': \text{head}(e')=\text{tail}(e)} \beta_{e',e} Y(e')$

- Transfer matrix
  \[ Z = xM \quad M = A(I - F)^{-1} B^T \]

- Coefficients uniformly at random $F_q \quad q = 2^m$
Algebraic Security Criterion

Level of security provided by RLNC

- Number of symbols that an intermediate node \( v \) has to guess in order to decode one of the transmitted symbols
- Partial transfer matrix

\[
\Delta_s(v) = \frac{K - \text{rank}(M'_{\Gamma_v(v)}) + l_d}{K}
\]
Algebraic Security: Security Characterization

- **Two cases w/ relevant information:**
  1. Partial transfer matrix has full rank
  2. Partial transfer matrix has diagonizable parts

\[ P(\Delta s > 0) \leq P(\exists v : \delta_i(v) > K) \]

- **Algebraic security: topology dependent!**
- Only necessary to consider the case

\[ K \leq \delta_i(v) \]
Algebraic Security (Diagonizable)

- Linear combination of independent and uniformly distributed values in $F_q$

- Product - Obtain a zero:
  1. $(a \in F_q) \times 0$
  2. $a \in F_q, b \in F_q, a \neq 0, b \neq 0, ab = 0$

- $q^2$ entries of the multiplicative table
- Number of zeros obtained by (2) grows

$$O(h(q)) < O(q^2)$$

$$P(X_{\text{lin}} = 0) \leq \frac{2q + h(q)}{q^2}$$

$$P(X_{\text{lin}} = 0)_{q \to \infty} = 0$$
Algebraic Security (Diagonizable)

- \[ M' = M^T \delta_1(v) \]  \[ \text{Gaussian elimination} \]

- Analyse probability \( p \) of having \( K-1 \) zeros in one or more lines of \( M' \)

\[
p = \binom{K}{K-1} \left( \frac{2q + h(q)}{q^2} \right) \left( 1 - \frac{2q + h(q)}{q^2} \right)^{K-1}
\]
Algebraic Security (Diagonizable)

- $X$: recoverable number of symbols
- $\delta_t(v)$: degrees of freedom

- If $\delta_t(v) = 1$
  \[ X = 1, \quad P(X = 1) = p \]

- If $1 < \delta_t(v) < K$
  \[ L = \delta_t(v) - l \]
  Lines to perform Gaussian elimination

- $L = l$ lines with $K - 1$ zeros
Algebraic Security (Diagonizable)

\[ P(X = x \mid L = l) \leq \left( \frac{\delta_l(v) - l}{x - (\delta_l(v) - l)} \right) p^{x - (\delta_l(v) - l)} (1 - p)^{2\delta_l(v) - 2l - x} \]

\[ P(X = x) \leq \sum_{l=0}^{x} \binom{\delta_l(v)}{l} p^l (1 - p)^{\delta_l(v) - l} P(X = x \mid L = l) \]
Algebraic Security (Diagonizable)

- \( P(l_d > 0) \) probability of recovering \( l_d \) symbols
- Intermediate nodes with \( \delta_l(v) \leq K - 1 \)
- Gaussian elimination

\[ P(l_d > 0) \rightarrow 0, \quad q \rightarrow \infty, \quad K \rightarrow \infty \]
An example: The complete DAG

- **Algebraic security:** topology dependent
- Node with $\delta_i(v) \geq K$
- Restricted nodes
- Network: complete DAG
- 1 source, 1 sink

$$|E| = \frac{n(n - 1)}{2} \quad n - 1 = \delta_i(v) + \delta_o(v)$$

$$\delta_o(v) = n - \text{order}(v) \quad \delta_i(v) = \text{order}(v) - 1$$
An example: The complete DAG

- **Algebraic security:** topology dependent
- Node with $\delta_i(v) \geq K$
- Restricted nodes
- Network: complete DAG
- 1 source, 1 sink

$$|E| = \frac{n(n-1)}{2}$$
$$n - 1 = \delta_i(v) + \delta_o(v)$$
$$\delta_o(v) = n - \text{order}(v)$$
$$\delta_i(v) = \text{order}(v) - 1$$
An example: The complete DAG

- **Algebraic security:** topology dependent
- Node with $\delta_i(v) \geq K$
- Restricted nodes
- Network: complete DAG
- 1 source, 1 sink

$$|E| = n(n - 1)/2 \quad n - 1 = \delta_i(v) + \delta_o(v)$$

$$\delta_o(v) = n - \text{order}(v) \quad \delta_i(v) = \text{order}(v) - 1$$
An example: The complete DAG

- **Algebraic security:** topology dependent
- Node with $\delta_i(v) \geq K$
- Restricted nodes
- Network: complete DAG
- 1 source, 1 sink

\[ |E| = \frac{n(n-1)}{2} \quad n - 1 = \delta_i(v) + \delta_o(v) \]

\[ \delta_o(v) = n - \text{order}(v) \quad \delta_i(v) = \text{order}(v) - 1 \]
An example: The complete DAG

- **Algebraic security:** topology dependent
- Node with \( \delta_l(v) \geq K \)
- Restricted nodes
- Network: complete DAG
- 1 source, 1 sink

\[
|E| = \frac{n(n-1)}{2} \quad n - 1 = \delta_l(v) + \delta_o(v) \\
\delta_o(v) = n - \text{order}(v) \quad \delta_l(v) = \text{order}(v) - 1
\]
An example: The complete DAG

- **Algebraic security:** *topology dependent*
- Node with $\delta_l(v) \geq K$
- *Restricted* nodes
- Network: complete DAG
- 1 source, 1 sink

\[
|E| = \frac{n(n - 1)}{2} \quad n - 1 = \delta_l(v) + \delta_o(v) \\
\delta_o(v) = n - \text{order}(v) \quad \delta_l(v) = \text{order}(v) - 1
\]
An example: The complete DAG

- **Algebraic security:** **topology dependent**
- Node with \( \delta_l(v) \geq K \)
- Restricted nodes
- Network: complete DAG
- 1 source, 1 sink

\[
|E| = \frac{n(n-1)}{2} \quad n - 1 = \delta_l(v) + \delta_o(v) \\
\delta_o(v) = n - \text{order}(v) \quad \delta_l(v) = \text{order}(v) - 1
\]
An example: The complete DAG

- **Algebraic security:** topology dependent
- Node with $\delta_i(v) \geq K$
- Restricted nodes
- Network: complete DAG
- 1 source, 1 sink

\[ |E| = \frac{n(n-1)}{2} \quad n - 1 = \delta_i(v) + \delta_o(v) \]
\[ \delta_o(v) = n - \text{order}(v) \quad \delta_i(v) = \text{order}(v) - 1 \]
An example: The complete DAG

- Partial transfer matrix received at a vertex
- Symbols: \( R = K + \theta, K, \theta \geq 0 \)

- If columns are linearly dependent:
  \[
  \left\{ x_{h_1} c_{h_1} + x_{h_2} c_{h_2} + \ldots + x_{h_n} c_{h_n} = (0\ldots0)^T \right\}
  \]

- Probability \( \rightarrow 0, q \rightarrow \infty, K \rightarrow \infty \)
- So a node that receives \( R \) symbols receives w.h.p rank

\[
\min(R, K)
\]

\[
\Delta_s(v) = \frac{K - \min(K, \text{order}(v))}{K}
\]
An example: The complete DAG

- **Secure max-flow**: maximum number of symbols that may be secured in a transmission by using RLNC

  \[ \phi_s = n - 1 \]

- Minimum number of symbols required for secure transmission

- “Dummy” symbols
Where do we go from here?

- The use of network coding as a cypher has some benefits, but they are highly dependent on topology and on the model of attack in terms of location of the attackers and their ability to collude.
- Can we still make use of the properties of network coding without having the vagaries of topology intervene?
- Two approaches:
  - Protect the code only
  - Use network coding for per packet verification without decoding
Protecting the code

• The goal is to lighten the overhead of cryptanalysis by making use of network coding
• A traditional approach will encode packets and operate on those encoded packets
• We propose to protect the code itself:
  – It is still possible to perform combinations on top of encrypted codes, by keeping the collective effect of the successive encodings upon the encrypted codes
  – We shall model the encryption of the codes as saying that the encoding matrices, based upon variants of random linear network coding, are given only to the source and sinks
• Main results:
  – We provide a characterization of the mutual information between the encoded data and the two elements that can lead to information disclosure:
    • the matrices of random coefficients
    • the original data itself.
  – Our results, some of which hold even with finite block lengths, show that, predicated on uniform distribution of the data to be encoded, information-theoretic security is achievable for any field size without loss in terms of decoding probability
  – The assumption of uniform distribution can be obtained through compression of encryption of even one of the encoded packets
Lemma 1 (From [1]): In Random Linear Network Coding, the conditional entropy of the payload $\gamma$ given the encoding matrix $A$ is:

$$H(\gamma|A) \geq n \log(q) \left(1 - f(q)\right),$$

where $O(f(q)) \leq O\left(\frac{1}{q}\right)$.

Theorem 1: The mutual information between the payload $\gamma$ and the encoding matrix $A$ is upper bounded by:

$$I(\gamma; A) \leq f(n, q)$$

where $f(n, q)$ is a function such that $O(f(n, q)) < O\left(\frac{n \log(q)}{q}\right)$.

[1] Ho et al.
Statement of results

- We can obtain tighter bounds if we restrict ourselves to invertible matrices

**Corollary 1:** The mutual information between the payload $\gamma$ and the encoding matrix $\mathbf{A}$, given that $\mathbf{A}$ is invertible and that $b_i \in \mathbb{F}_q \setminus \{0\}$, $0 \leq i \leq n$, is upper bounded by:

$$I(\gamma; \mathbf{A} | \{ \det(\mathbf{A}) \neq 0 \}) \leq n(\log(q) - \log(q - 1))$$

**Corollary 2:** If $\mathbf{A}$ is invertible and $b_i \in \mathbb{F}_q$, $0 \leq i \leq n$, the mutual information $I(\gamma; \mathbf{A} | \{ \det(\mathbf{A}) \neq 0 \})$ is equal to 0:

$$I(\gamma; \mathbf{A} | \{ \det(\mathbf{A}) \neq 0 \}) = 0$$
Statement of results

- To obtain further results, consider source encoding of the plaintext where the source symbols exclude the codeword zero
- We can use traditional source encoding, with random matrices that do not include the coefficient 0

\[ \text{Lemma 2: The probability of the sum } S_k = \sum_{i=0}^{k} \alpha_i \beta_i \text{ yielding the result } \phi \in F_q, \text{ conditioned on } (\alpha_1, \ldots, \alpha_k), \text{ where } \alpha_i \in F_q \setminus \{0\} \text{ and } \beta_i \in F_q, \text{ is:} \]

\[
P(S_k = 0 | (\alpha_1, \ldots, \alpha_k)) = p_0
\]

\[
P(S_k = \phi | (\alpha_1, \ldots, \alpha_k))_{\forall \phi \neq 0} = p_\phi
\]

\[
(p_0, p_\phi)_k = \left( \frac{(q-1)^{k-1} - z_{k-1}}{(q-1)^k}, \frac{(q-1)^k - z_k}{(q-1)^{k+1}} \right)
\]

\[
z_0 = 0, \ z_1 = (q - 1)
\]
Statement of results

Theorem 3: The mutual information between the payload $\gamma$ and the encoding matrix $A$, for the case in which $b_i$ is uniformly i.i.d and $b_i \in \mathbb{F}_q \setminus \{0\}$ is upper bounded by:

$$I(\gamma; A) = n \left( \log(q) - \sum_{i=0}^{n} \binom{n}{i} \frac{(q - 1)^{(n-i)}}{q^n} H(\gamma_i | Z(a_i) = i) \right)$$

where $H(\gamma_i | Z(a_i) = i) = -p_{0i} \log p_{0i} - (q - 1)p_{\phi i} \log p_{\phi i}$, and the values for $(p_0, p_{\phi})_i$ are given by the expression in Lemma 2. It follows that $\lim_{q \to \infty} I(\gamma; A) = 0$. 
Statement of results

Theorem 4: The mutual information between the payload $\gamma$ and the plaintext $b$, for the case in which $b_i$ is uniformly i.i.d and $a_{ij} \in \mathbb{F}_q \setminus \{0\}$ is bounded by the following expression:

$$I(\gamma; b) \leq \left( n \log(q) - \sum_{i=0}^{n} \binom{n}{i} \frac{i(q - 1)^{(n-i)}}{q^n} H(\gamma_i | Z(b) = i) \right)$$

where $H(\gamma_i | Z(b) = i)) = -p_{0i} \log p_{0i} - (q - 1)p_{\phi i} \log p_{\phi i},$ and the values for $(p_0, p_{\phi})_i$ are given by the expression in Lemma 2. It follows that $\lim_{q \to \infty} I(\gamma; b) = 0.$
Statement of results

- What is the effect of reusing the coding matrix?

**Theorem 6:** The information obtained about \( A \) given \( (\gamma_1, \gamma_2, \ldots, \gamma_m) = (A\beta_1, A\beta_2, \ldots, A\beta_n) \), for the case in which \( b_i \) is uniformly i.i.d and \( a_{ij} \in \mathbb{F}_q \setminus \{0\} \) is:

\[
I(\gamma_1, \gamma_2, \ldots, \gamma_m; A) = 0
\]

**Theorem 7:** The information obtained about \( (b_1, \ldots, b_m) \) given \( (\gamma_1, \gamma_2, \ldots, \gamma_m) = (A\beta_1, A\beta_2, \ldots, A\beta_n) \), for the case in which \( b_i \) is uniformly i.i.d and \( b_i \in \mathbb{F}_q \setminus \{0\} \), is:

\[
I(\gamma_1, \ldots, \gamma_m; b_1, \ldots, b_m) = 0,
\]
Results

Mutual information in function of field size, for several coding strategies for RLNC and $n = 4$. 
Conclusions for protecting the code

- Our results show that it is possible to achieve information theoretic security by performing optimal source coding on the payload information and protecting the code.
- It is also possible to choose between null mutual information on the coefficients or on the payload, which has an impact on practical choices for RLNC protocols, such as the size of the generation to choose for security.
- We are considering the evaluation of the impact of non-uniformities caused by imperfect source coding.
- This may be a promising approach to leverage the one-time pad aspects of network coding without having the drawbacks of topology dependence.