Some problems I (still) am interested in

Carlos Munuera

Aguilar de Campoo — 2023

The problems to which I refer in this talk belong to three topics:

- Locally recoverable codes
- Steganography and decoding
- Secret sharing

Locally recoverable codes

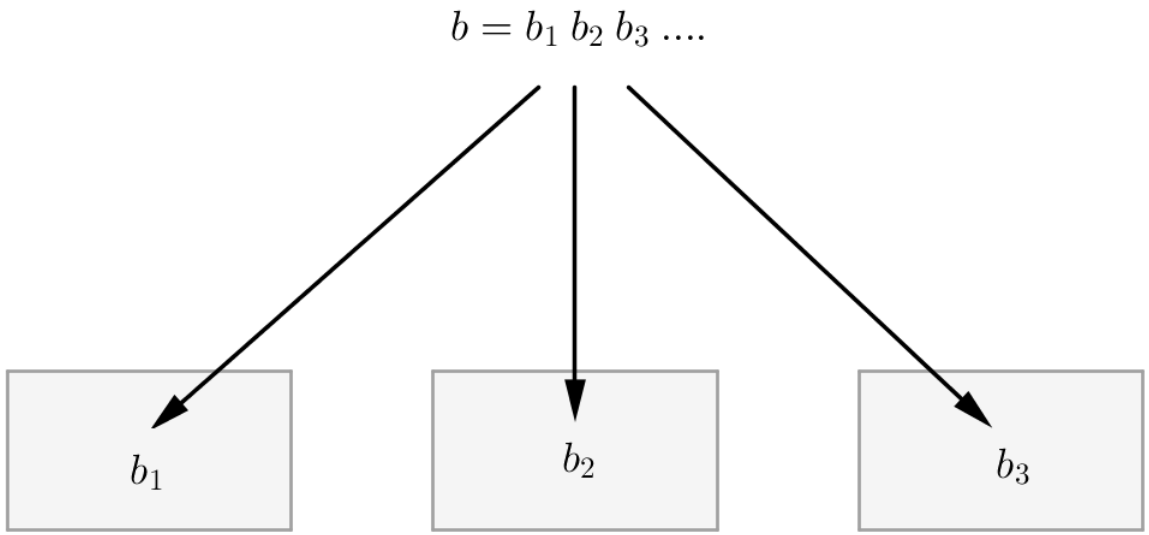
Distributed storage of information

Distributed storage splits data across multiple physical servers (clouds,..)

– The information to be stored is a long sequence b of symbols of a finite field $\mathbb{F}_\ell$.

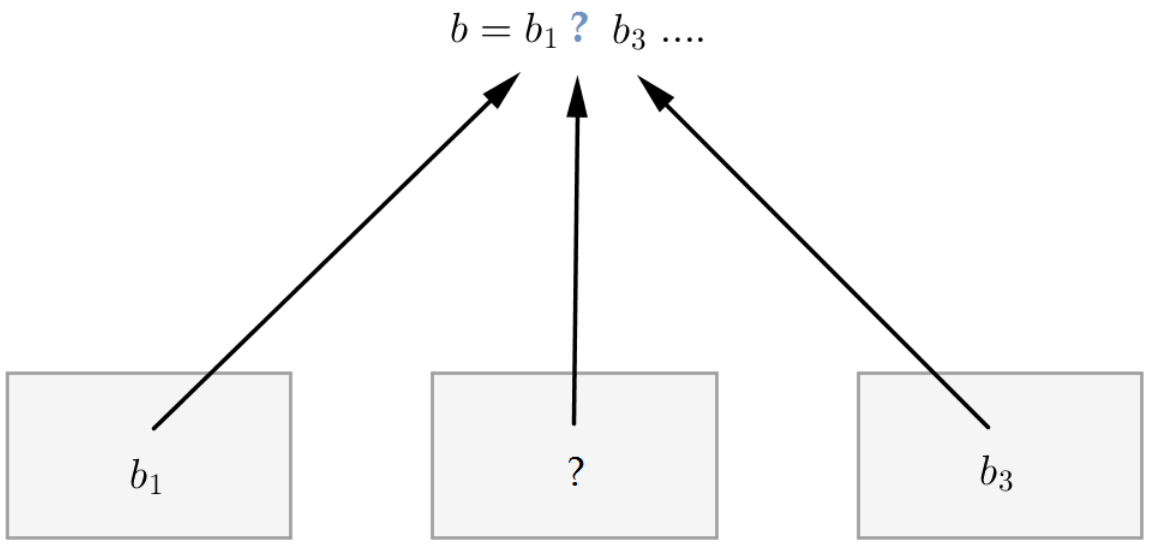
– The sequence is cut into blocks $b = b_1 b_2 b_3, ...$ of equal length, m.

– Each block is stored in a (different) available node.



Distributed storage of information

When a node fails, the information it contains is lost
resulting in the **repair problem**



Repair problem

- A trivial (but used) solution is to replicate the same data on different nodes. Clearly wasteful as it multiplies the resources required.

- Another more convenient solution is to use error correcting codes:
  - Since $\mathbb{F}_\ell^m \cong \mathbb{F}_{\ell^m}$, each block b can be seen as an element of $\mathbb{F}_q$, $q = \ell^m$.
  - Each node stores an element $b \in \mathbb{F}_q$.
  - We use a correcting code to recover the missing symbols.

with RS codes

We want to store $b_1, b_2 \cdots \in \mathbb{F}_q^*$

– We set integers $k < n \leq q$ and elements $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_q$.

– The vector $\mathbf{b} = (b_1, \ldots, b_k) \in \mathbb{F}_q^k$ will be stored in n nodes as

consider $f(x) = b_1 + b_2 x + \cdots + b_k x^{k-1}$

and we store $f(\alpha_i) = b_1 + b_2 \alpha_i + \cdots + b_k \alpha_i^{k-1}$ in the ith node.

The stored information $\mathbf{b}$ is retrieved from any k available nodes.

Distributed Storage using Codes

- To store the k-uple of information $(b_1, \ldots, b_k) \in \mathbb{F}_q^k$.
- We will use a code $\mathcal{C}$ of type $[n, k, d]$ over $\mathbb{F}_q$.
- $(b_1, \ldots, b_k)$ is encoded by $(c_1, \ldots, c_n)$ (stored in n nodes).

Any erasure or error in the c_i is corrected by means of $\mathcal{C}$.

Local recovery

The codes used in practice are long and their decoding is computationally expensive.

To make the repair problem efficient, it should be possible to solve it in a **local** way, that is, using only a small number of nodes

$$(c_1, \ldots\ldots, \underbrace{c_j, \ldots, c_i, \ldots, c_t}, \ldots\ldots, c_n)$$

LRC codes

Let $\mathcal{C}$ be a linear code $[n, k, d]$ over $\mathbb{F}_q$.

A coordinate i admits **local recovery of locality** $\leq r$ if there exists a **recovery set** $R_i \subseteq \{1, \ldots, n\}$, $i \notin R_i$ and $\#R_i = r$, such that for any word $\mathbf{u} \in \mathcal{C}$, the coordinate u_i is uniquely determined by the $\{u_j : j \in R_i\}$.

$\mathcal{C}$ is a **locally recoverable code (LRC) of locality** $\leq r$ if all coordinates support it.

The minimum r is the **locality** of $\mathcal{C}$.

Optimal LRC codes

Every code with distance $d > 1$ is LRC with locality $r \leq k$.
But we are interested in codes with small locality:

The locality of a $[n, k, d]$ code verifies the **Singleton-like bound**

$$k + d + \left\lceil \frac{k}{r} \right\rceil \leq n + 2$$

Codes achieving equality are called **optimal**.
(For example, MDS codes are optimal of locality k.)

The Tamo-Barg construction

- Let $\phi(x) \in \mathbb{F}_q[x]$ be a polynomial of degree $r+1$, constant over certain $\mathcal{P}_1, \ldots, \mathcal{P}_t \subset \mathbb{F}_q$, of cardinality $r+1$.

- Let $\mathcal{P} = \mathcal{P}_1 \cup \cdots \cup \mathcal{P}_t = \{\alpha_1, \ldots, \alpha_n\}$.

- Let $V = \left\{ \sum_{i=0}^{r-1} \sum_{j=0}^{s} a_{ij} \phi(x)^j x^i \right\}$ with $(r+1)s + (r-1) < n$ (so $\deg(f) < n$)

The code $ev_{\mathcal{P}}(V) = \{(f(\alpha_1), \ldots, f(\alpha_n)) : f \in V\}$ has

dimension $k = \dim(V) = r(s+1)$;
distance $d = n - (r+1)s - (r-1)$.

The Tamo-Barg construction

Let $\mathbf{c} \in \mathcal{C}$, $\mathbf{c} = ev_{\mathcal{P}}(f)$ for certain $f \in V$,

$$f = \sum_{i=0}^{r-1} \sum_{j=0}^{s} a_{ij} \phi(x)^j x^i$$

Suppose an erasure in $c_t = f(\alpha_t),\ \alpha_t \in \mathcal{P}_i$. Being ϕ constant over every $\mathcal{P}_i$:

- f acts as a polynomial f_i of degree $\leq r-1$ over $\mathcal{P}_i$;
- that can be interpolated from its values at r points any of $\mathcal{P}_i$.

$\mathcal{C}$ is LRC of locality r; the recovery sets are the $\mathcal{P}_i$.
It is **optimal** and $r \ll k$ (and also $n \leq q$).

Example

The polynomial $\phi(x) = x^3 \in \mathbb{F}_{13}[x]$
is constant over $\mathcal{P}_1 = \{1, 3, 9\}$, $\mathcal{P}_2 = \{2, 6, 5\}$, $\mathcal{P}_3 = \{4, 10, 12\} \subset \mathbb{F}_{13}$.

Let $r = 2$, $\mathcal{P} = \{1, 3, 9; 2, 6, 5; 4, 10, 12\}$,

$$V = \left\{ \sum_{i=0}^{1} \sum_{j=0}^{s} a_{ij} x^{3j} x^i \right\}$$

We get LRC codes $[9; 2; 8]; [9; 4; 5]; [9; 6; 2]$ of locality 2.

The Tamo-Barg construction

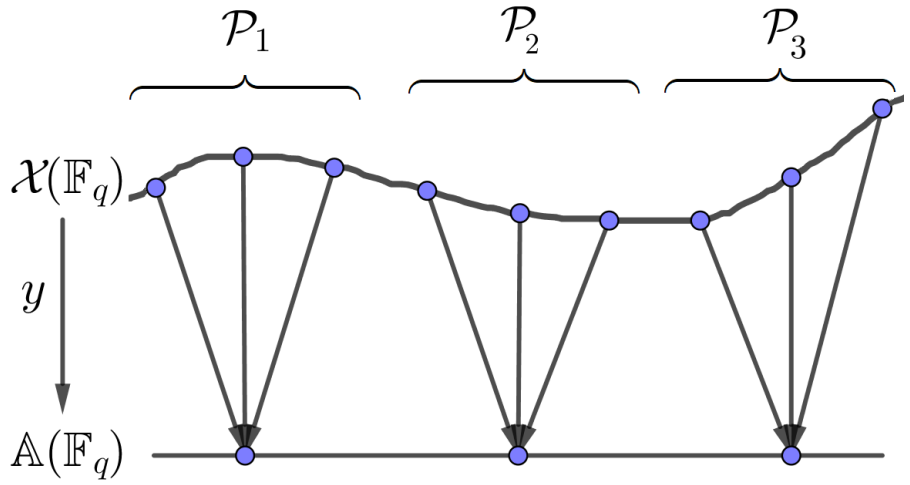
The Tamo-Barg construction leads us to two kinds of problems.

– The TB construction allows to obtain optimal codes for all lengths $n \leq q$
  What about $n > q$?

– Many research looking for 'good' polynomials
  Are they necessary?

A Geometric Interpretation

Consider the curve $\mathcal{X} : y = x^3$, and the functions $x, y \in \mathbb{F}_{13}(\mathcal{X})$



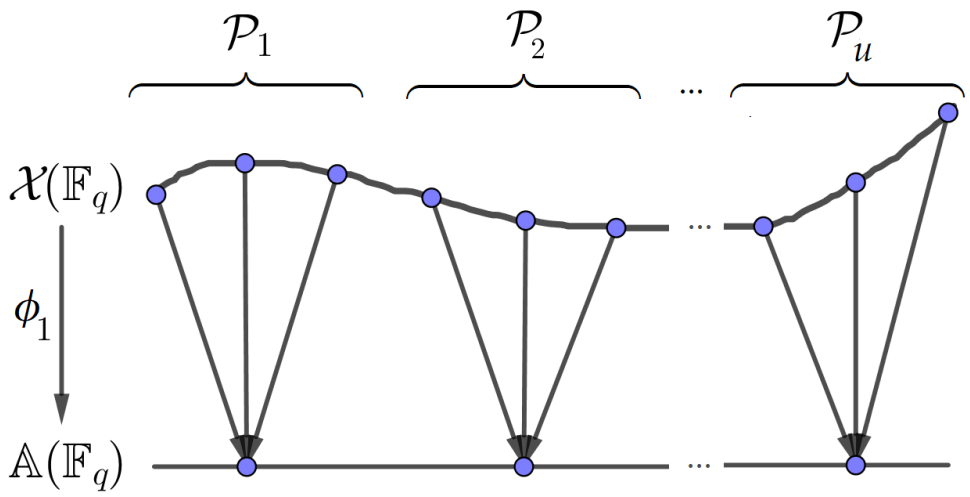
$\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2 \cup \mathcal{P}_3$, and the vector space

$$V = \bigoplus_{i=0}^{1} \langle 1, y, \ldots, y^{l_i} \rangle \; x^i \subset \mathbb{F}_{13}[x, y] \subset \mathbb{F}_{13}(\mathcal{X})$$

$\mathrm{ev}_{\mathcal{P}}(V)$ is a LRC code of locality $r = 2$.

A Geometric Interpretation

Consider a curve $\mathcal{X}$ and rational functions $\phi_1, \phi_2 \in \mathbb{F}_q(\mathcal{X})$



$\mathcal{P} = \mathcal{P}_1 \cup \cdots \cup \mathcal{P}_u$ fibres with $\geq r+1$ elements, and the space

$$V = \bigoplus_{i=0}^{r-1} \langle 1, \phi_1, \ldots, \phi_1^{l_i} \rangle \ \phi_2^i \subset \mathbb{F}_q[\phi_1, \phi_2] \subset \mathbb{F}_q(\mathcal{X})$$

$\mathrm{ev}_{\mathcal{P}}(V)$ is an LRC code of locality r.

Problems.

– Study the same idea with other objects:

$$\mathcal{A} \subset \mathbb{A}^m(\mathbb{F}_q),\ \phi_1, \ldots, \phi_t, \phi_{t+1} \in \mathbb{F}_q(x_1, \ldots, x_m),\ \phi = (\phi_1, \ldots, \phi_t),$$

$$\mathcal{P} = \mathcal{P}_1 \cup \cdots \cup \mathcal{P}_u \text{ fibres with } \geq r+1 \text{ elements,}$$

$$V = \sum_{i=0}^{r-1} V_i\ \phi_{t+1}^i, \qquad V_i \subset \mathbb{F}_q[\phi_1, \ldots, \phi_t]$$

$\mathcal{C} = \mathrm{ev}_{\mathcal{P}}(V)$ is a LRC code of locality r.

– Study the case of more than one erasure

For example

Let $\phi_1 = \dfrac{x}{y},\ \phi_2 = \dfrac{y-1}{z} \in \mathbb{F}_3(x,y,z)$
$\mathcal{A} = \{(x,y,z) \in \mathbb{A}^3(\mathbb{F}_3) : yz \neq 0\};\ \ \phi = \phi_1 : \mathcal{A} \subset \mathbb{A}^3(\mathbb{F}_3) \to \mathbb{A}^1(\mathbb{F}_3).$

$\phi^{-1}(0) = \{(0,0,1),(0,1,2),(0,2,1),(0,2,2)\},$
$\phi^{-1}(1) = \{(1,1,1),(1,1,2),(2,2,1),(2,2,2)\},$
$\phi^{-1}(2) = \{(1,2,1),(1,2,2),(2,1,1),(2,1,2)\}.$

$\mathcal{P} = \{(0,1,2),(0,2,1),(0,2,2);(1,1,2),(2,2,1),(2,2,2);$
$(1,2,1),(1,2,2),(2,1,2)\}$

$$V_1 = \langle 1, \frac{x}{y}\rangle \oplus \langle 1\rangle \frac{y-1}{z} \quad \text{and} \quad V_2 = \langle 1, \frac{x}{y}, \frac{x^2}{y^2}\rangle \oplus \langle 1, \frac{x}{y}\rangle \frac{y-1}{z}$$

we obtain $[9,3,6]$ and $[9,5,3]$ **optimal** LRC over $\mathbb{F}_3$.

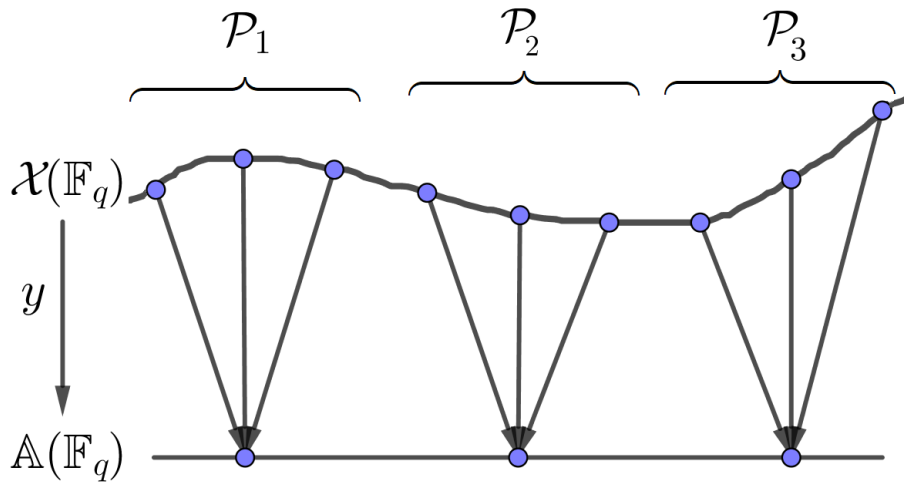
More on the Tamo-Barg construction

– The TB construction allows to obtain optimal codes for all lengths $n \leq q$
  What about $n > q$?

– Many research looking for 'good' polynomials
  Are they necessary?

More on the Tamo-Barg construction

Look at the previous example over $\mathbb{F}_{13}$



$$\mathcal{P} = \{1, 3, 9; 2, 6, 5; 4, 10, 12\} \sim$$
$$\sim \{(1, 1), (3, 1), (9, 1); (2, 8), (6, 8), (5, 8); (4, -1), (10, -1), (12, -1)\}$$
$$V = \bigoplus_{i=0}^{1} \langle 1, y, y^2 \rangle \, x^i = \langle 1, x, y, xy, y^2, xy^2 \rangle$$

More on the Tamo-Barg construction

Its generator matrix is

	1	3	9	2	6	5	4	10	12
1	1	1	1	1	1	1	1	1	1
x	1	3	9	2	6	5	4	10	12
y	1	1	1	8	8	8	-1	-1	-1
xy	1	3	9	3	9	1	-4	-10	-12
y^2	1	1	1	-1	-1	-1	1	1	1
xy^2	1	3	9	-2	-6	-5	4	10	12

More on the Tamo-Barg construction

We can follow the same method with other values
(not obtained from polynomials; these 'at random')

Take $q = 13, n = 9$, $\mathcal{P} = 1, 2, 3;\ 4, 5, 6;\ 7, 8, 9$, $y = 1, -1, 0$,

$$G = \left(\begin{array}{ccc|ccc|ccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \hline 1 & 1 & 1 & -1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 2 & -3 & -4 & -5 & 0 & 0 & 0 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{array}\right).$$

This is an optimal $[9, 5, 3]$

No polynomial provides these values of x and y!
Still optimal over $q = 5$ or 7.

Problems.

– Can the Tamo-Barg construction be carried **without** using polynomials (in terms of linear algebra only)?

– In such a case, can it be extended to codes of lengths $n > q$?

– When we obtain optimal codes?

Steganography and decoding

What is steganography?

Steganography is the science of transmitting secret messages in such a way that no one, apart from the sender and receiver, can detect the existence of communication.

steganography = steganos + graphein = **covered writing**

That is how steganography works: the secret message we want to protect is hidden into an apparently innocuous object, the **cover**.

Today's typical cover is a computer file. Media files are ideal for steganographic purposes because of their large size and redundancy.

Typical example: Hiding information in images

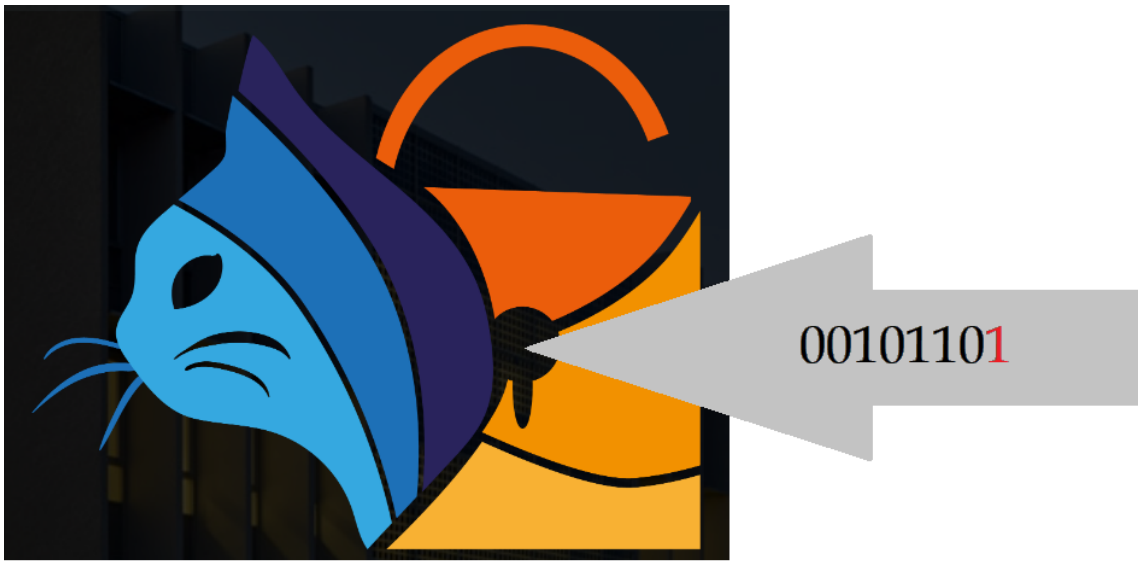
A digital image is stored in the computer as a sequence of bytes corresponding to its pixels.

To hide a secret message, the sender adjusts the color of some pixels in such a way to correspond to a message symbol. Due to the limited capacity of the human eye to distinguish colors the changes made are visually imperceptible.

The original image is deleted after changing it, so there is no possibility of comparison.

LSB steganography

Usually the changes of color are made by replacing the **least significant bits** of the selected pixels by message bits. This technique is **LSB steganography**.



From now on we will assume this form of stegangraphy

Steganography: instructions for use

Given a digital image, we first select a set of n pixels in which we shall embed the secret information: we get a **cover vector** $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$.
Let $\mathbf{m} = (m_1, \ldots, m_r) \in \mathbb{F}_2^r$ be the secret information to be embedded.

A **steganographic scheme of type** $[n, r]$ is a pair of functions

$$\mathrm{emb} : \mathbb{F}_2^n \times \mathbb{F}_2^r \longrightarrow \mathbb{F}_2^n \ , \quad \mathrm{rec} : \mathbb{F}_2^n \longrightarrow \mathbb{F}_2^r$$

such that $\mathrm{rec}(\mathrm{emb}(\mathbf{x}, \mathbf{m})) = \mathbf{m}$ for all $\mathbf{x}, \mathbf{m}$.

Steganography: instructions for use

- The condition $\mathrm{rec}(\mathrm{emb}(\mathbf{x}, \mathbf{m})) = \mathbf{m}$ implies

$$\mathrm{emb}(\mathbf{x}, \mathbf{m}) \in \mathrm{rec}^{-1}(\mathbf{m})$$

  so $\mathrm{emb}(\mathbf{x}, \mathbf{m})$ belongs to the (nonlinear) code $\mathcal{C}_{\mathbf{m}} = \mathrm{rec}^{-1}(\mathbf{m})$.

- Since we want the number of embedding changes as small as possible we take

$$\mathrm{emb}(\mathbf{x}, \mathbf{m}) = \mathrm{dec}(\mathbf{x})$$

  where dec is a decoding map for $\mathcal{C}_{\mathbf{m}}$.

- If rec is linear (with matrix $\mathbf{H}$), then $\mathrm{rec}^{-1}(\mathbf{0})$ is linear and $\mathcal{C}_{\mathbf{m}} = \mathrm{rec}^{-1}(\mathbf{m})$ is the coset $\{\mathbf{v} \in \mathbb{F}_2^n : \mathbf{H}\mathbf{v}^{\mathrm{t}} = \mathbf{m}^{\mathrm{t}}\}$.
  So embedding $\mathrm{emb}(\mathbf{x}, \mathbf{m})$ is solving $\mathbf{H}\mathbf{v}^{\mathrm{t}} = \mathbf{m}^{\mathrm{t}}$ with $d(\mathbf{v}, \mathbf{x})$ as small as possible.

Decoding maps

A decoding map $\mathrm{dec} : \mathcal{X} \subseteq \mathbb{F}_2^n \longrightarrow \mathcal{C}$ is

- **complete** if $\mathcal{X} = \mathbb{F}_2^n$;
- **minimum distance** if $d(\mathbf{x}, \mathrm{dec}(\mathbf{x})) = d(\mathbf{x}, \mathcal{C})$;
- **efficient** if it is polynomial in time and memory requirements.

but minimum distance complete decoding is a NP-complete problem.

Ussually we remove the requirement about completness.
But here we cannot do without it,
so we use codes providing complete decoding (Hamming).

Embedding with locked positions

The cover pixels storing the secret information are determined through a **selection rule**, agreed by the sender and the receiver.

When applying an agreed rule to a particular cover object, some pixels (coordinates of the cover vector) are not suitable for making changes. They are **locked**, preventing its modification during the embedding.

Embedding with locked positions

Using the previous ideas, we fix a code $\mathcal{C}$ and a parity check matrix $\mathbf{H}$.

If $L \subset \{1, \ldots, n\}$ is the set of locked positions
we set $\mathrm{emb}(\mathbf{x}, \mathbf{m}) = \mathbf{v} \in \mathbb{F}_2^{\mathrm{n}}$, with

$$[S] : \begin{cases} \mathbf{H}\mathbf{v}^{\mathrm{t}} & = \mathbf{m}^{\mathrm{t}}, \\ v_i & = x_i \ \text{ if } i \in L, \end{cases}$$

where $d(\mathbf{v}, \mathbf{x})$ is as small as possible.

Embedding with locked positions

To solve the system

$$[S]: \begin{cases} \mathbf{H}\mathbf{v}^{\mathtt{t}} & = \mathbf{m}^{\mathtt{t}}, \\ v_i & = x_i \ \text{ if } i \in L, \end{cases}$$

we delete coordinates corresponding to the positions of L, and the corresponding columns of $\mathbf{H}$. We arrive to a new system

$$\mathbf{D}\mathbf{u}^{\mathtt{t}} = \mathbf{z}^{\mathtt{t}}$$

where $\mathbf{D}$ is the matrix formed by the columns of $\mathbf{H}$ with indices not in L .

When $\mathbf{D}$ has full rank then it is the parity-check matrix of a code $\mathcal{D}$. Then the appropriate (nearest to $\mathbf{x}$) solution of $[S]$ can be found again by using a decoding map of $\mathcal{D}$.

But now D is a completely arbitrary code!

Embedding with locked positions

A decoding map $\mathrm{dec} : \mathcal{X} \subseteq \mathbb{F}_2^n \longrightarrow \mathcal{C}$ is

- **complete** if $\mathcal{X} = \mathbb{F}_2^n$;
- **minimum distance** if $d(\mathbf{x}, \mathrm{dec}(\mathbf{x})) = d(\mathbf{x}, \mathcal{C})$;
- **efficient** if it is polynomial in time and memory requirements.

but minimum distance complete decoding is a NP-complete problem.

Ussually we remove the requirement about completness.
But here we cannot do it.
We should remove the requirement about minimum distance!
(whenever our decoding map is not 'too far' to be minimum distance)

Problems.

– Design an **efficient** algorithm for decoding **any** linear code (not by minimum distance, but **not very far**).

– Estimate how far from minimum distance it is.

Up and down

In the previous scheme, each pixel can be changed in only one direction
according to its LSB $(1 \to 0,\ 0 \to 1)$,
which causes us to lose part of the ability to insert information.

What is important is the effect of the distortion caused by the insertion **in the image**

Problem (v.1).

– Design an embedding algorithm over $\mathbb{Z}/3\mathbb{Z}$

More up and more down

Instead of restricting the change in luminance of each pixel to one unit, we can allow changes of 2,3,... units, which leads to write the above problem as

Problem (v.2).

– Design an embedding algorithm over $\mathbb{Z}/m\mathbb{Z}$ (m odd)

More up and more down

In this case, the Hamming metric over $\mathbb{Z}/m\mathbb{Z}$ is not adequate.

The same happens in other practical applications of error-correcting codes.
E.g. bar codes

Problem (v.3).

– Design a decoding algorithm for codes over $\mathbb{Z}/m\mathbb{Z}$
with respect to a inner metric δ in $\mathbb{Z}/m\mathbb{Z}$.

Power rates in secret sharing

Secret sharing schemes

A **secret sharing scheme** is a method to share a secret among a set of participants, so that

(a) only some **authorized coalitions** can access the secret;
(b) no other coalition does.

The secret s is split into **shares** s_i given to the participants.

The usual notation

$\mathcal{P} = \{P_1, \cdots, P_n\}$	the participants
$\Gamma = \{A \subseteq \mathcal{P} \mid A \text{ authorized}\}$	the access structure
$\Gamma_0 = \{$ minimal authorized coalitions $\}$	the basis of Γ

Perfect and ramp

The scheme is called:

– **perfect** if nonauthorized coalitions can obtain nothing about the secret,

– **ramp** if some nonauthorized coalitions can obtain some partial information about the secret.

In the following we restrict to perfect secret sharing schemes.

Uses and generalizations

Many uses:

- key management
- cooperative criptography
- E-voting
- multiparty computation, ...

and many extensions:

- Proactive secret sharing
- Verifiable secret sharing
- Computationally secure secret sharing, ...

Threshold access structures and others

A (t, n) **threshold access structure** is

$$\Gamma = \{A \subseteq \mathcal{P} \ : \ |A| \geq t\}.$$

– Shamir (Lagrange interpolation)
– Blakley (projective geometry)
– Mignotte (Chinese remainder theorem)
– MDS codes
– ...

There exist also **not threshold** structures
and any monotone acces structure can be realized

but INCIBE only recognizes Shamir's scheme.

Access to the secret

In a threshold scheme all participants have the same opportunity to access the secret.

This is not true of general structures!
Eg $\Gamma = \{\{P_1, P_2\}\{P_3, P_4, P_5, P_6, P_7\}\}$

(a) How to **measure** the power of each participant?
(b) How to **obtain** structures with prefixed power rates?

This will depend only on the access structure and not on the scheme used!

Weighted structures

An answer could be weighted structures:
we give weights $w_1, \ldots, w_n$ to the paticipants, and

$$[t; w_1, \cdots, w_n] = \{A \subseteq \mathcal{P} \mid w_A = \sum_{P_i \in A} w_i \geq t\}$$

Two drawbacks:

– the size of shares increases;
– the weights **do not** measure the power of participants:

In $[6; 4, 3, 3, 1]$, the secret is recovered by **any two** participants from the first three, so P_1, P_2, P_3 have equal power, and the power of P_4 is zero.

Measuring the power

Let π_i be the power of P_i in a structure Γ.

The power of a participant depends on the need that others have of him to form a authorized coalition.

1. $0 \leq \pi_i \leq 1$ and $\pi_1 + \cdots + \pi_n = 1$.
2. A participant is **null** if it does not belong to any minimal coalition.
   If P_i is null, then $\pi_i = 0$.
3. The participant P_i is **over** P_j if for each coalition A, such that $P_i, P_j \notin A$, $A_i \cup P_j \in \Gamma \Rightarrow A \cup P_i \in \Gamma$. If P_i **is over** P_j**, then** $\pi_i \geq \pi_j$**.**

Measuring the power

Consider all possible orderings of the participants

$$P_{i_1} P_{i_2} \cdots P_{i_k} \cdots P_{i_n}.$$

We say that the participant P_{i_k} is **key** for the previous ordering if
$\{P_{i_1} P_{i_2} \cdots P_{i_k}\}$ recovers the secret,
but $\{P_{i_1}, P_{i_2}, \cdots, P_{i_{k-1}}\}$ does not.

Define the power of P_i as the number of rearrangements in which it is key,
divided by the total number $n!$

Measuring the power

Thus

$$\pi_i(\Gamma) = \frac{1}{n!} \sum_{S \in \Gamma_i} (|S| - 1)!(n - |S|)!$$

where Γ_i is the set of all coalitions for which P_i is key.

For example:

$\Gamma = [6; 4, 3, 3, 1]$, gives powers 1-1-1-0

$\Gamma = [5; 4, 1, 1, 1]$, gives powers 9-1-1-1

$\Gamma = \{\{1, 2, \}\{3, 4\}\}$, gives powers 1-1-1-1

Problems.

– Is this the right form to measure the powers? (others?)

– Include this new ingredient in the theory.

– How to construct structures with (or closest to) prefixed power rates.

– What about ramp secret sharing schemes?

References

I. Tamo, A. Barg: A family of optimal locally recoverable codes,
IEEE Transactions on Information Theory 60, 4661–4676.
doi: 10.1109/TIT.2014.2321280.

C. Munuera, W. Tenorio: Locally recoverable codes from rational maps,
Finite Fields and Their Applications 54 (2018), 80–100.
doi: 10.1016/j.ffa.2018.07.005.

J. Fridrich, M. Goljan, P. Lisonek, D. Soukal: Writing on wet paper,
IEEE Transactions on Signal Processing 53 (2005) 3923–3935.
doi: 10.1109/TSP.2005.855393

F. Carreras, A. Magaña, C. Munuera: The accessibility of an access structure,
RAIRO-Theoretical Informatics and Applications 40 (2006), 559–567. doi:
10.1051/ita:2006040