UNIVERSITÀ
DI TRENTO

# Coding Theory Cryptography and LEDAcrypt implementation

Giacomo Borin

Università di Trento

December 16, 2021

# Table of contents


# Quantum computers and Post-Quantum

**Problem**

Shor's algorithm on QC breaks Asymmetric Cryptography

We have the necessity to find new suitable encryption algorithms, some possibilities are:

- Lattice reduction problems (based on the difficulty of finding a minimal norm base for a lattice given a base with big norm)
- **Coding Theory based algorithms**
- Isogeny-Based Cryptography (a new version of ECC)
- other...

# The new requirements

NIST has open a call for proposal of new primitives for Post-Quantum Cryptography in 2017:

https://csrc.nist.gov/projects/post-quantum-cryptography

LEDAcrypt is a standardization based on QC-LDPC codes used on McEliene and Niederreiter Cryptosystem that arrived to round 2 (but failed to reach round 3) proposed by researchers from *Università politecnica delle Marche* and *Politecnico di Milano*.

# NP-Completeness

**Definition**

A decision problem $\mathcal{C}$ is NP-complete if:

- $\mathcal{C}$ is in NP (set of problems that can be solved in polynomial time by a nondeterministic Turing machine)
- Every problem in NP is reducible to $\mathcal{C}$ in polynomial time

NP-complete problem are belived to be resistant to QC attacks.

For example we have that:

The Nearest Codeword Problem (NCP) is NP-complete.


# Coding Theory

# Coding Theory in Plain English

When we send messages on a disturbed channel it is possible that one or more errors occours, thus we would like to be able to correct them.

For example if I sent you the message:
**ATTAXK THE ENEMUES AT DAWB**
you will be able to recover the original message.

This happens because the english words bring a quantity of redundant information (in fact not every characters combination is an english word).

# Idea

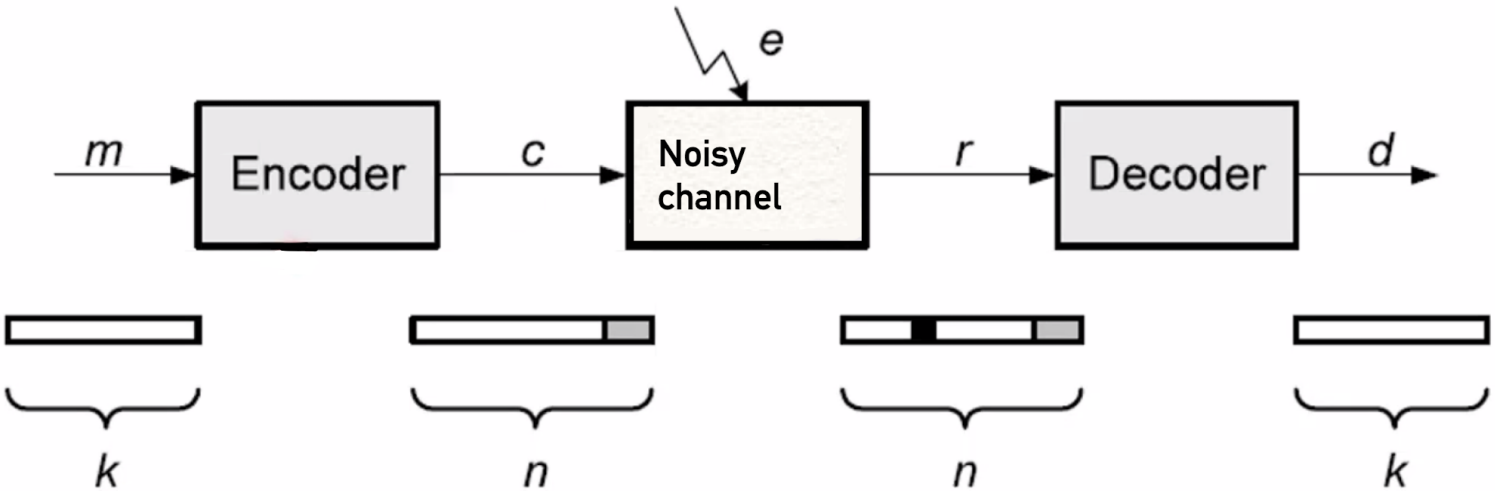



Figure: Example idea of Error correcting codes

## Definiton (Linear code)

A linear code is an injective linear map:

$$\mathcal{C}(n,k) : \mathbb{F}_2^k \to \mathbb{F}_2^n$$

This map is uniquely identified by the linear subspace of the image in $\mathbb{F}_2^n$, thus we call **codewords** the vectors of the image.

Using this map we can add $n-k$ bits of redundant information to the input string.

The matrix $G$ that represents the linear code is called **Generator matrix**.

We can also associate an $n-k \times n$ matrix $H$ called **Parity-Check matrix**, that maps a $n$ bit vector to $\mathbf{0}$ if and only if it is a codeword.

# Example I

For example if we want to send a 2 bit message and correct at least one error we can use this linear code:

$$G = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix} \text{ and } H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

thus we `encode` the 2 bit strings as:

$$(0,0) \mapsto (0,0,0,0,0)$$
$$(0,1) \mapsto (1,0,1,1,1)$$
$$(1,0) \mapsto (0,1,1,0,1)$$
$$(1,1) \mapsto (1,1,0,1,0)$$

# Distance

## Definiton (Hamming Distance)

The distance of two points is the number of different coordinates:

$$d(\boldsymbol{x}, \boldsymbol{y}) = \#\{i \mid \boldsymbol{x}_i \neq \boldsymbol{y}_i\}$$

For example

$$d((0,0,1,0,1),(0,1,1,0,0)) = 2$$

We define the minimum distance of a linear code the minimum Hamming distance between any two codewords.

# Example II

To have an idea of what's happening we use graphs.

Here vertices will represents strings and the vertices will be connected if the strings have Hamming distance 1 (we can pass from one to another with one flip).

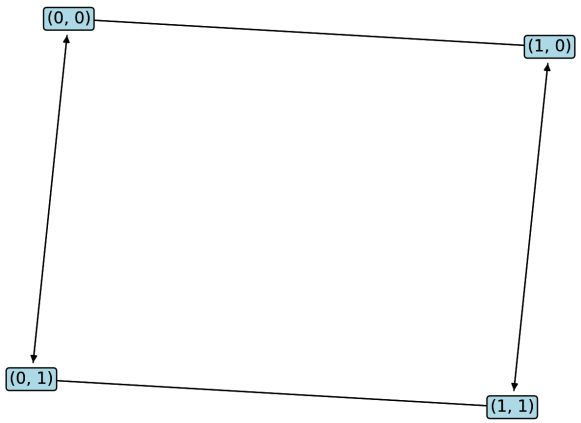



Figure: Representation of $\mathbb{F}_2^2$

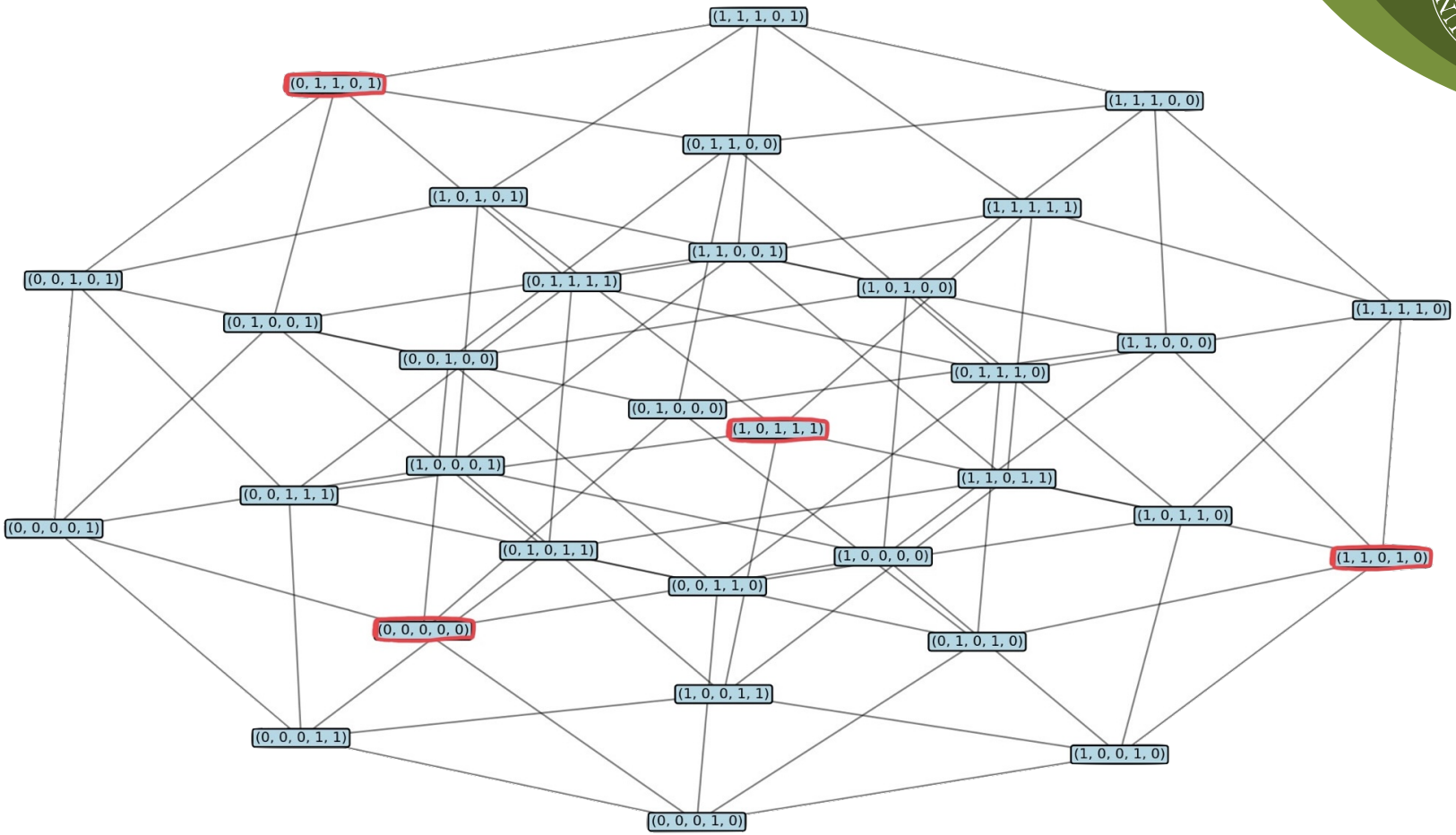


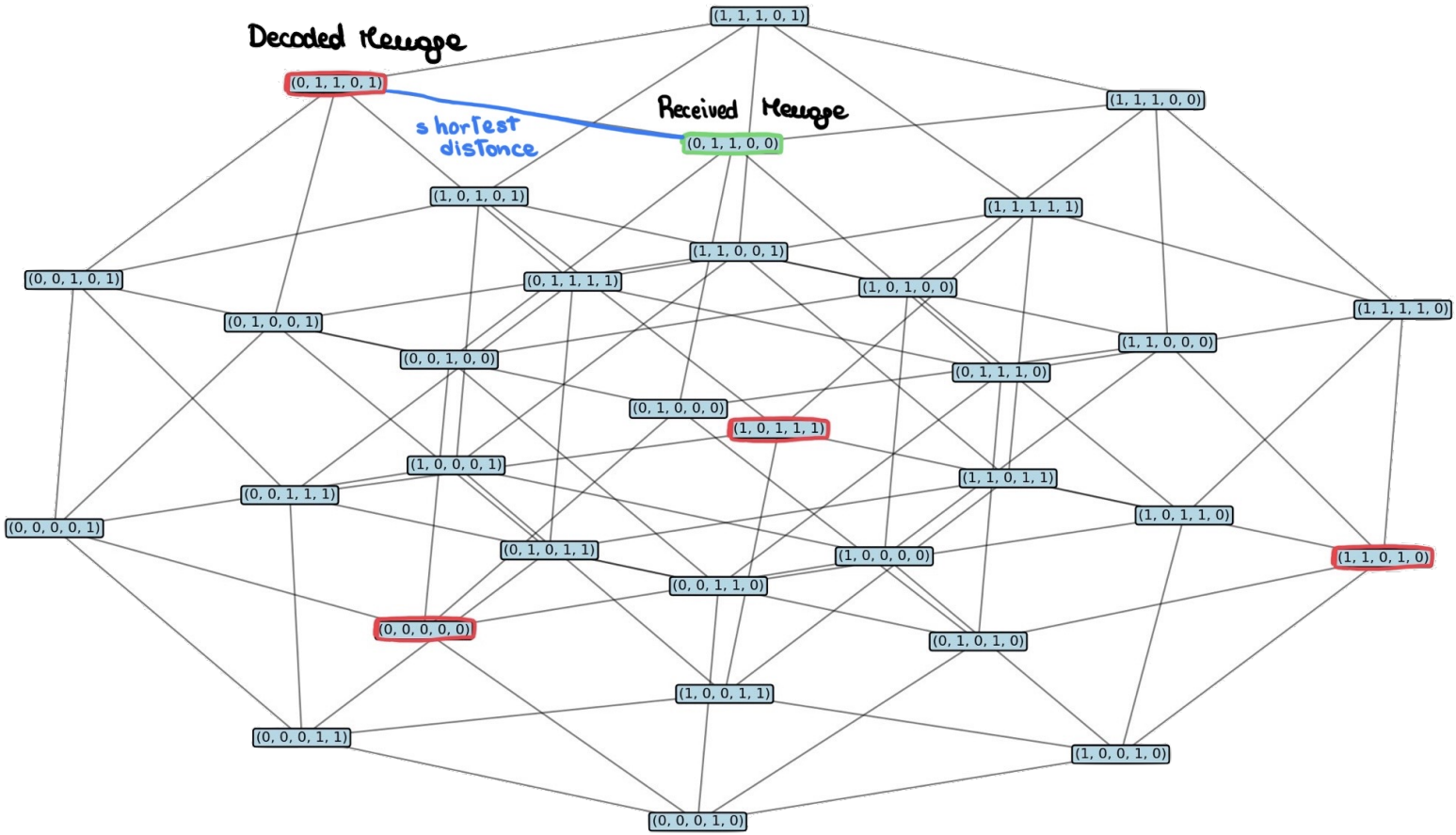

Figure: Immersion of $\mathbb{F}_2^2$ in $\mathbb{F}_2^5$

# Algorithm description

1. The first phase consist in the encoding: we add information to a $k$ bit string through a matrix.
2. Then the message is sent over a noisy channel.
3. The decoding algorithm is then able to invert a fixed number of errors looking for the nearest codeword.

We can see that if $d$ is the minimum distance, then we can correct $t$ errors if $t \leq 2d - 1$.

Suppose that we want to send $(0, 1)$. We encode it as $(0, 1, 1, 0, 1)$, but then $(0, 1, 1, 0, 0)$ is received.



Now we can properly state the NP-complete problem associated to linear codes:

**Definiton**

The decision problem **Nearest Codeword Problem** (NCP) is the set of instances given by a binary $k \times n$ matrix $G$, a vector $\boldsymbol{y} \in \mathbb{F}_2^n$ and a positive integer $d$ such that there exists a vector $\boldsymbol{x} \in \mathbb{F}_2^k$ such that $d(\boldsymbol{x}G, \boldsymbol{y}) \leq d$.


# McEliece Cryptosystem

The **McEliece Cryptosystem** is a triple of polynomial time algorithms

$$\Pi^{\mathrm{McE}} = \left(\mathsf{Keygen}^{\mathrm{McE}}, \mathsf{Enc}^{\mathrm{McE}}, \mathsf{Dec}^{\mathrm{McE}}\right)$$

That uses (linear) coding theory as key idea.

# Keygen

Consider a (secret) linear code $\mathcal{C}(n, k)$ with $G \in M_{k \times n}$ as generator matrix, an invertible scrambling matrix $S \in M_{k \times k}$ and a permutation matrix $P \in M_{n \times n}$.

Then we can define:

1. the secret key $\text{sk}^{\text{McE}} \leftarrow \{S, G, P\}$
2. the public key $\text{pk}^{\text{McE}} \leftarrow \{G'\}$ where $G' = S \cdot G \cdot P$

(the $\cdot$ represents matrix multiplication)

# Encryption

The *encryption* algorithm encode a $k$ bit string $\boldsymbol{m}$ with $G'$ and output a $n$ bit vector $\boldsymbol{x}$ on which are performed $t$ errors:

$$\boldsymbol{x} = \boldsymbol{m}G' + \boldsymbol{e} \leftarrow \mathrm{Enc}^{\mathrm{McE}}(\mathrm{pk}^{\mathrm{McE}}, \boldsymbol{m})$$

The vector $\boldsymbol{e}$ represents the $t$ (or less) random errors.

# Decryption

The *decryption* algorithm uses the decoding algorithm for linear codes (Decode) with the private key $\mathrm{sk}^{\mathrm{McE}}$ to perform an error correcting decoding on $\boldsymbol{x}P^{-1}$, then it uses the inverted scrambling matrix $S^{-1}$ to recover the plain-text.

$$\boldsymbol{m} = (\boldsymbol{m}S)S^{-1} = \mathrm{Decode}\left((\boldsymbol{m}S)G + \left(\boldsymbol{e}P^{-1}\right)\right)S^{-1} =$$
$$= \mathrm{Decode}\left(\boldsymbol{x}P^{-1}\right)S^{-1} \leftarrow \mathrm{Dec}^{\mathrm{McE}}(\mathrm{sk}^{\mathrm{McE}}, \boldsymbol{x})$$


# LEDAcrypt

# Key ideas

The McEliece cryptosystem is a well-known and trusted construction for longer than 40 years, to improve it the LEDAcrypt proposed a linear version with:

1. Matrices that are easy to store (ideally sparse)
2. Faster generation of $G$ and $S$ with the required properties
3. Faster decoding

(Fun fact: Classical McEliece cryptosystem for Goppa codes reached the 3rd round of standardization)

# LDPC codes

LDPC codes use (sparse) circular matrices, namely matrices (in $\mathbb{F}_2$) such that the shifted rows repeats them self, thus are of the form:

$$\begin{bmatrix} a_0 & a_1 & a_2 & \dots & a_n \\ a_1 & a_2 & a_3 & \dots & a_0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_n & a_0 & a_1 & \dots & a_{n-1} \end{bmatrix}$$

In particular the weight (number of non zero coordinates) of a column of $G$ (or $H$), denoted as $d_v$, is much smaller than its length.

# QC codes

A QC code is defined as a linear code that can be represented as matrix of blocks $p \times p$ such that encodes string of $k_0 p$ bits in codeword of $n_0 p$ bits and that if we shift a codeword of $n_0$ bits we get another codeword.

In our case $k_0 = n_0 - 1$ and the parity-check matrix will be formed by $n_0$ circular $p \times p$ blocks:

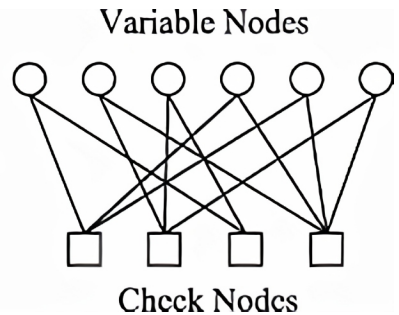
$$H = [H_0|H_1|...|H_{n_0-1}]$$

# Iterative decoding

The peculiar form of LDPC codes allows to devise an efficient iterative decoding procedure (if the parity check matrix is known).

This decoding algorithm derived from their possible representation as Tanner graphs ([Tan81])

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$



# Key Generation

An important part of the algorithm is to be able to generate genuine (pseudo)random parity-check and scrambling matrix from a secret and secure seed.

Also it is recommended to use the seed directly as private key, since the Keygen$^{\mathrm{McE}}$ algorithm is very fast.

Both the two matrices are composed of $p \times p$ circulant blocks

# Smart idea

We can represent a circular matrix as polynomial:

$$\{\text{circular matrix}\} \leftrightarrow \frac{\mathbb{F}_2[x]}{x^p + 1}$$

$$A = [a_{ij}]_{i,j=1}^{p} \leftrightarrow a(x) = \sum_{i=0}^{p-1} a_{0i} x^i$$

Using this and some correlated mathematical results is possible to put in relation the parity of the weight of the first row (the number of ones) and the singularity of the matrix.


# Cryptanalysis of LEDAcrypt

# Vulnerability I

Sadly LEDAcrypt didn't pass the round 2 of NIST call because in the article [Apo+20] they found a subset of weak keys such that there are bias in the structure of the public key that leak information on the structure of the codewords.

They also proved that it is not possible to identify the weak keys during Keygen and the vulnerability is intrinsic to the system.

# Vulnerability II

**Definiton**

A key is weak if the associated matrix $H \cdot S$ (equivalent to $G'$) presents a strong bias in the distribution of the non zero coefficients in the circular sub-matrices.

In particular the sub-matrices associated to **ultra weak keys** (for $n_0 = 2$) have only less than $\frac{p-1}{2}$ non zero coefficients that are all consecutive, such that the first rows has a distribution like this:

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

This matrices allows for an improved version of the ISD attack.


# Bibliography

# Bibliography I


*Classical McEliece post-quantum asymmetric cryptosystems*. https://classic.mceliece.org/index.html. Accessed: 2021-12-16.

*LEDAcrypt post-quantum asymmetric cryptosystems*. https://www.ledacrypt.org. Accessed: 2021-12-16.

D. Apon, R. Perlner, A. Robinson, and P. Santini. *Cryptanalysis of LEDAcrypt*. Cryptology ePrint Archive, Report 2020/455. https://ia.cr/2020/455. 2020.

# Bibliography II


R. G. Gallager. “Low-density parity-check codes”. English. In: *IRE Trans. Inf. Theory* 8 (1962), pp. 21–28.

R. J. McEliece. “A Public-Key Cryptosystem Based On Algebraic Coding Theory”. In: *Deep Space Network Progress Report* 44 (Jan. 1978), pp. 114–116.

R. Tanner. “A recursive approach to low complexity codes”. In: *IEEE Transactions on Information Theory* 27.5 (1981), pp. 533–547. DOI: 10.1109/TIT.1981.1056404.