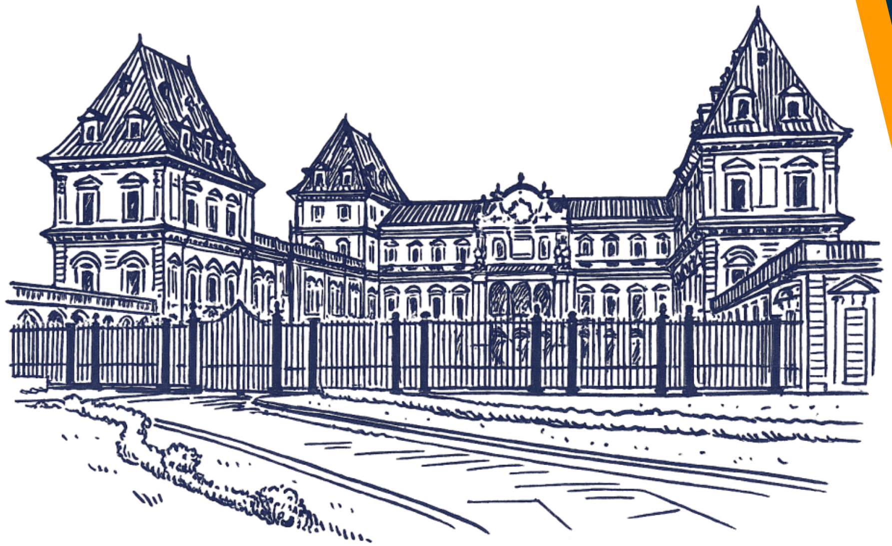
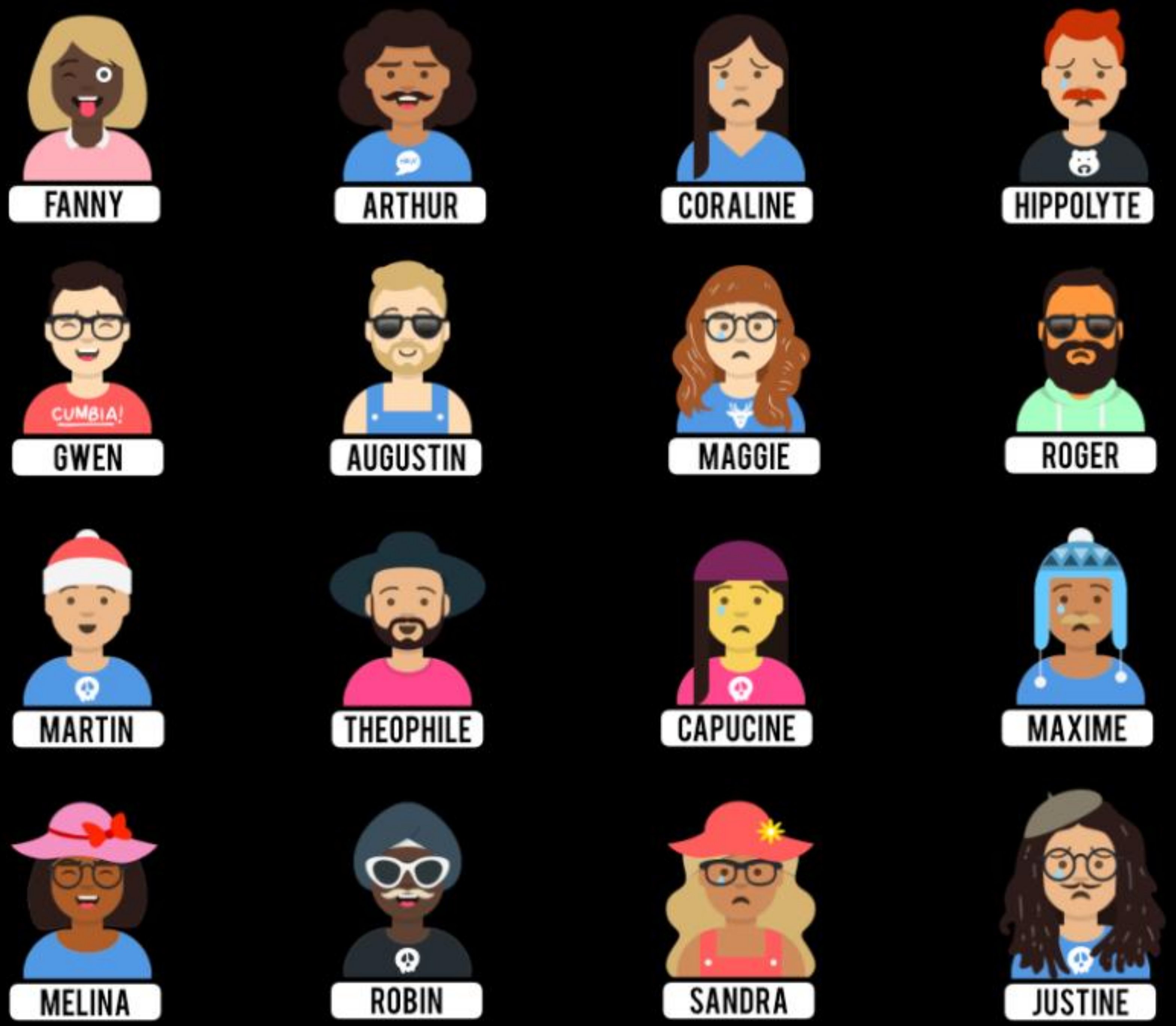


A Tale of Errors and Secrets

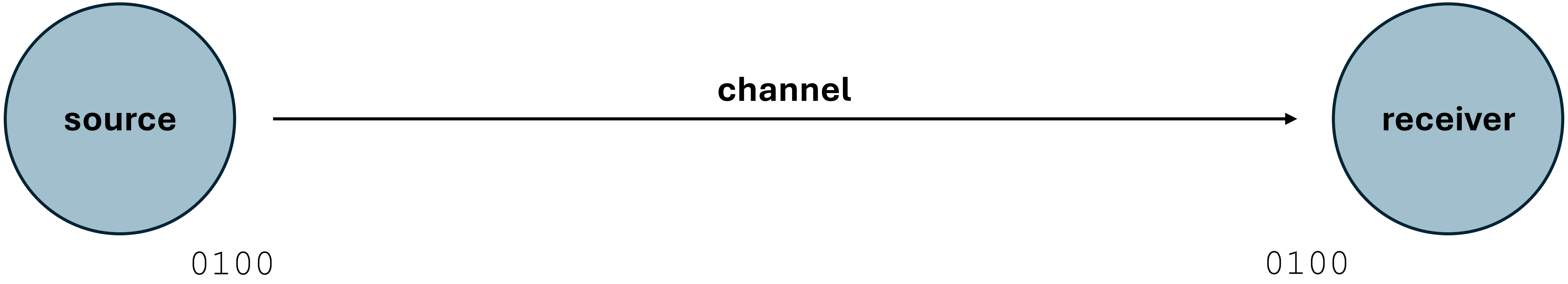
Leonardo Errati

Politecnico di Torino 1859 | CrypTO


FANNY
ARTHUR
CORALINE
HIPPOLYTE
CUMBIA!
GWEN
AUGUSTIN
MAGGIE
ROGER
MARTIN
THEOPHILE
CAPUCINE
MAXIME
MELINA
ROBIN
SANDRA
JUSTINE

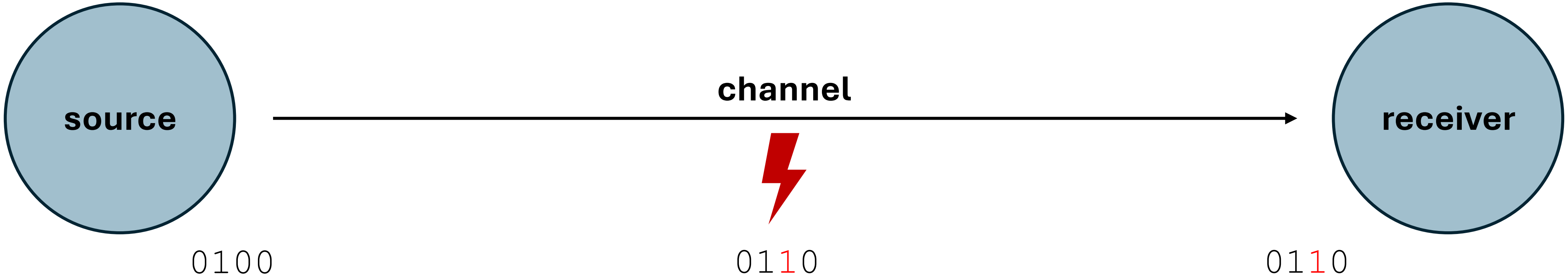
1: CODING THEORY

Coding theory



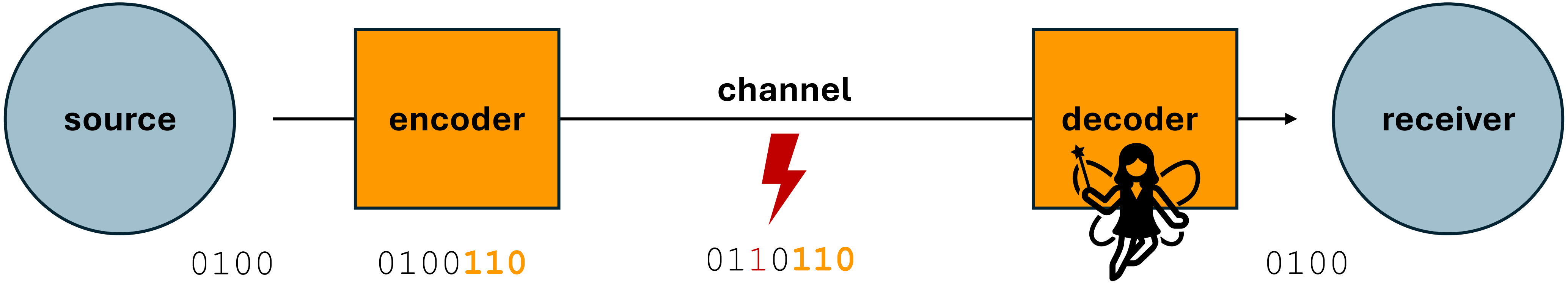
A	alphabet
k	input length

Coding theory



A alphabet
k input length

Coding theory



A	alphabet
k	input length
n	code length

Coding theory

Hackaday

Cosmic Ray Flips Bit, Assists Mario 64 Speedrunner

The leading theory is that this bit flip was caused by a cosmic ray event, though the likelihood of such an event is exceedingly rare. It's...

17 feb 2021

BBC

Bit flips: How cosmic rays grounded a fleet of aircraft

Radiation from space that led to more than 6000 Airbus aircraft needing emergency computer updates could become a growing problem.

1 dic 2025

“around one error per month per 256 MiB of RAM on a desktop computer”
(IBM, 1996)

Codes

A **code** is a map $\varphi: S^k \to S^n$.

We want it to

- be easy to invert
- be easy to memorise
- be efficient

$$\varphi:\ (x_1, x_2, x_3, x_4) \to ?$$

Codes

A **code** is a map $\varphi: S^k \to S^n$.

We want it to

- be easy to invert
- be easy to memorise
- be efficient

$$\varphi:\ (x_1, x_2, x_3, x_4) \to (x_1, x_2, x_3, x_4, x_5, x_6, x_7)$$

$$x_5 = f(x_1, x_2, x_3, x_4)$$
$$x_6 = f(x_1, x_2, x_3, x_4)$$
$$x_7 = f(x_1, x_2, x_3, x_4)$$

Linear codes

$$\varphi\colon (x_1, x_2, x_3, x_4) \to (x_1, x_2, x_3, x_4, x_5, x_6, x_7)$$

$$x_5 = x_1 + x_2 + x_4$$
$$x_6 = x_1 + x_3 + x_4$$
$$x_7 = x_2 + x_3 + x_4$$

Linear codes

A **linear code** is $C = \{xG : x \in S^k\}$.
The matrix is called **generator matrix**.

Usually $S = \mathrm{F}_q$, so C is a linear subspace of F_q^n.

$$\varphi\colon (x_1, x_2, x_3, x_4) \to (x_1, x_2, x_3, x_4, x_5, x_6, x_7)$$

$$x_5 = x_1 + x_2 + x_4$$
$$x_6 = x_1 + x_3 + x_4$$
$$x_7 = x_2 + x_3 + x_4$$

$$(x_1, x_2, x_3, x_4)\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} = (x_1, x_2, x_3, x_4, x_5, x_6, x_7)$$

Linear codes

Our constraints form a system of equations.

$$\begin{cases} x_1 + x_2 + x_4 + x_5 = 0 \\ x_1 + x_3 + x_4 + x_6 = 0 \\ x_2 + x_3 + x_4 + x_7 = 0 \end{cases}$$

$$x_5 = x_1 + x_2 + x_4$$
$$x_6 = x_1 + x_3 + x_4$$
$$x_7 = x_2 + x_3 + x_4$$

All elements of C satisfy it. This defines the parity check matrix, H.

$$\begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Linear codes

From generator G:

- The code is $C = \langle G \rangle$
- k linearly independent rows
- n columns

$$m \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} = c$$

From parity check H:

- The code is $C = ker(H)$
- $n - k$ linearly independent rows
- n columns

$$\begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} c^T = 0$$

Linear codes

From generator G:

- The code is $C = \langle G \rangle$
- k linearly independent rows
- n columns

From parity check H:

- The code is $C = ker(H)$
- $n - k$ linearly independent rows
- n columns

$$m \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} = c$$

$$\begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} c^T = 0$$

Linear codes

From generator G:

- The code is $C = \langle G \rangle$
- k linearly independent rows
- n columns

$$m \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} = c$$

From parity check H:

- The code is $C = ker(H)$
- $n - k$ linearly independent rows
- n columns

$$\begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} c^T = 0$$

Theorem.
If $G = (I_k \mid A)$, then $H = (-A^T \mid I_k)$.

Linear codes

Some codes are more equal than others.

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \qquad G' = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Theorem.
If $G = (I_k \mid A)$, then $H = (-A^T \mid I_k)$.

Linear codes

Some codes are more equal than others.

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

systematic form

$$G' = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

permuted systematic

Theorem.
If $G = (I_k \mid A)$, then $H = (-A^T \mid I_k)$.

Linear codes

Some codes are more equal than others.

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \qquad \longleftarrow \qquad G' = GP$$

systematic form

permuted systematic

Theorem.
If $G = (I_k \mid A)$, then $H = (-A^T \mid I_k)$.

Linear codes

Some codes are more equal than others.

permutation equivalent	$G' = GP$
permutation equivalent (2)	$G' = SGP$

Linear codes

Some codes are more equal than others.

permutation equivalent	$G' = GP$
permutation equivalent (2)	$G' = SGP$
linearly equivalent	$G' = SGPD$

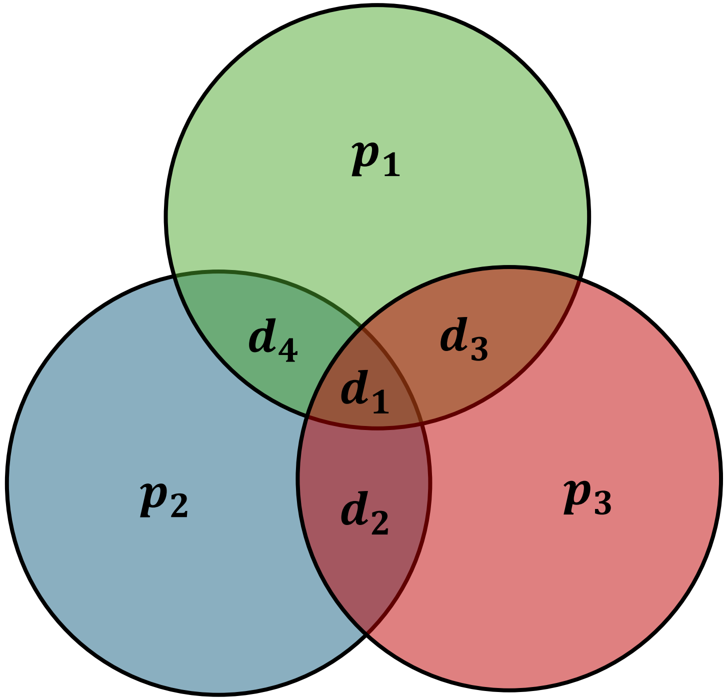
Linear codes

0	0000**000**
1	0001**011**
2	0010**101**
3	0011**110**
4	0100**110**
5	0101**101**
6	0110**011**
7	0111**000**
8	1000**111**
9	1001**100**
10	1010**010**
11	1011**001**
12	1100**001**
13	1101**010**
14	1110**100**
15	1111**111**

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Linear codes

0	0000**000**
1	0001**011**
2	0010**101**
3	0011**110**
4	0100**110**
5	0101**101**
6	0110**011**
7	0111**000**
8	1000**111**
9	1001**100**
10	1010**010**
11	1011**001**
12	1100**001**
13	1101**010**
14	1110**100**
15	1111**111**

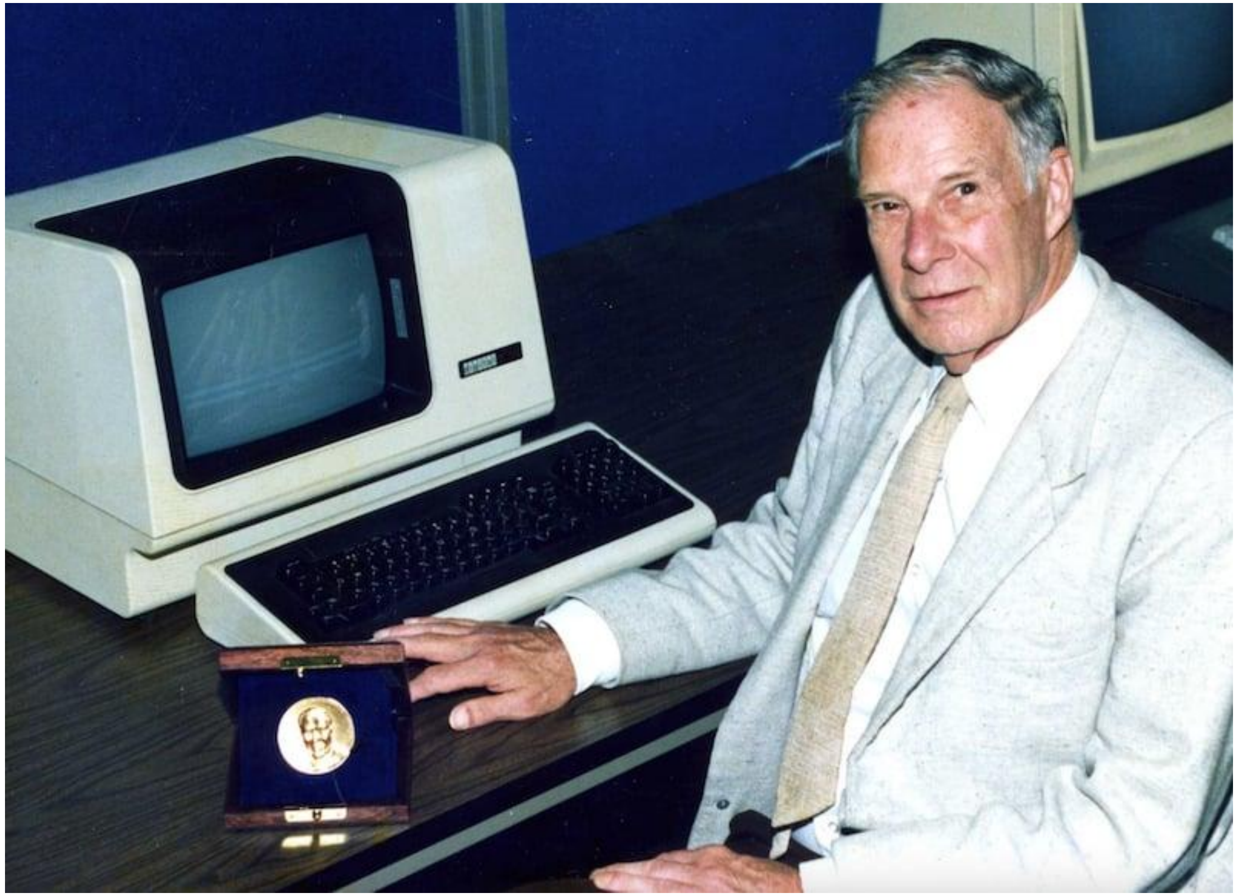


$$p_1 = d_1 + d_2 + d_4$$
$$p_2 = d_1 + d_3 + d_4$$
$$p_3 = d_2 + d_3 + d_4$$

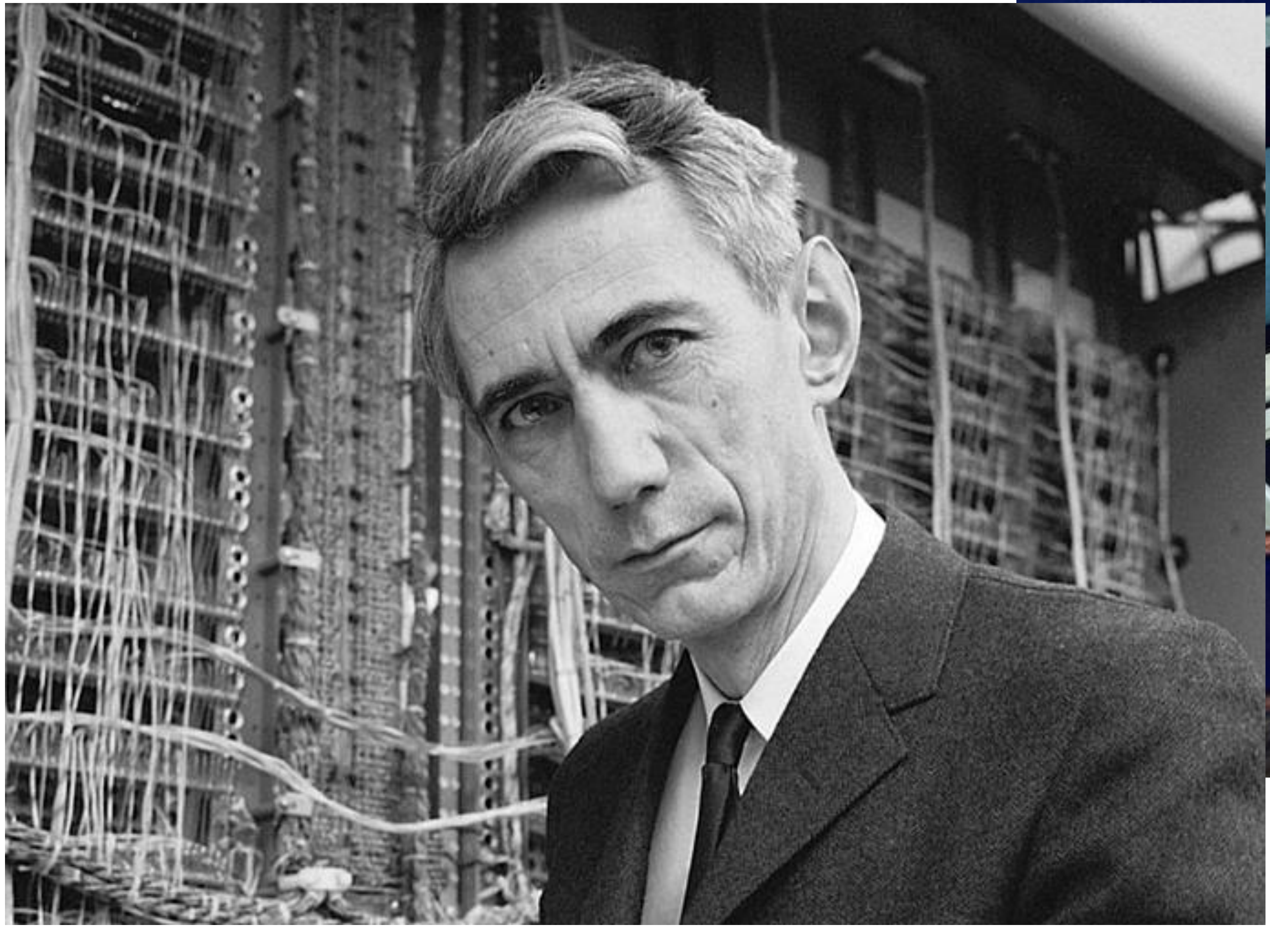
Linear codes

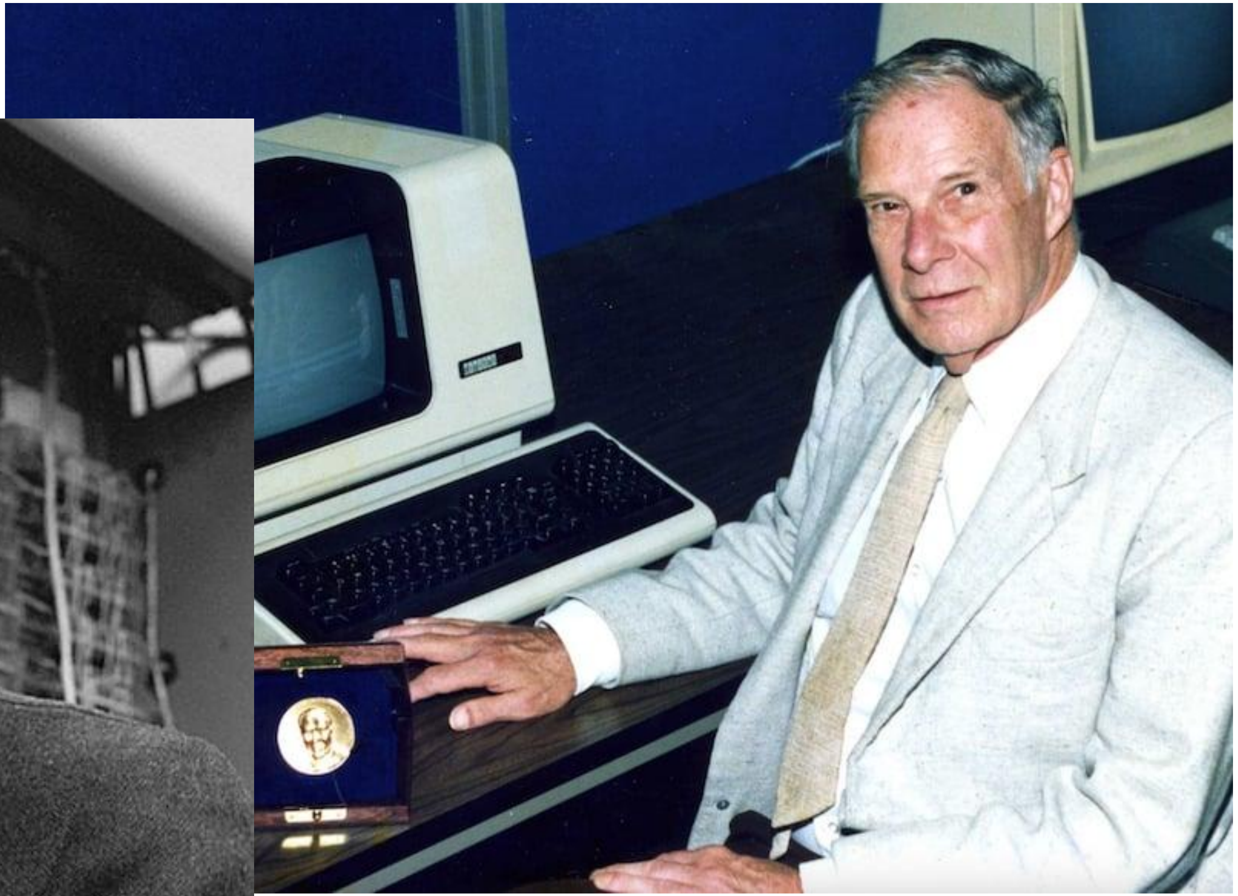
0	0000000
1	0001011
2	0010101
3	0011110
4	0100110
5	01011 0[illegible]
6	011[illegible]
7	[illegible]000
8	[illegible]000111
[illegible]	1001100
10	1010010
11	1011001
12	1100001
13	1101010
14	1110100
15	1111111

[7,4] Hamming code



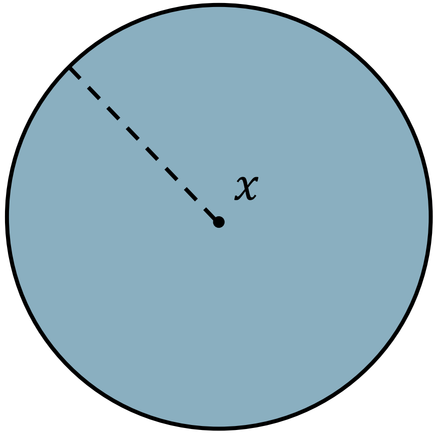
Linear codes





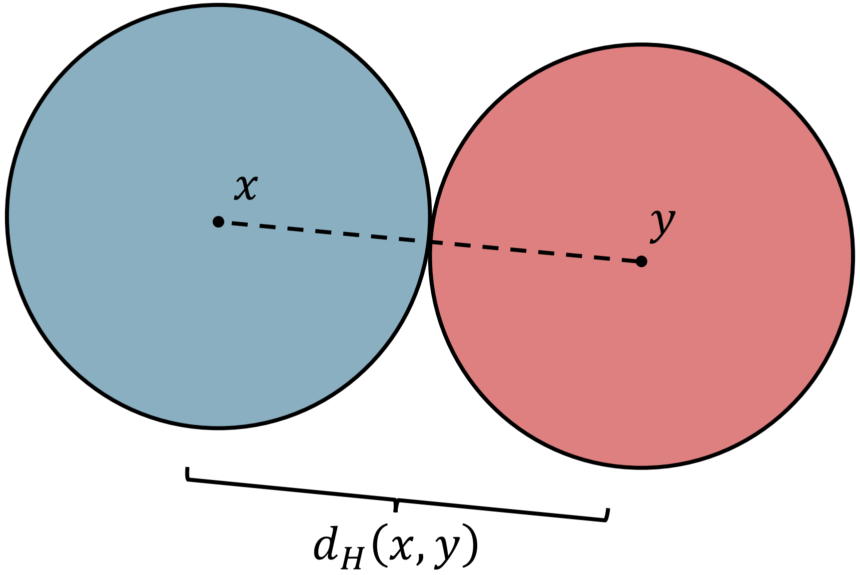
How to decode?

The **Hamming weight** of a codeword is $w_H(x) = \#\{i : x_i \neq 0\}$.
The **Hamming distance** of x, y is $d_H(x, y) = w_H(x - y)$.



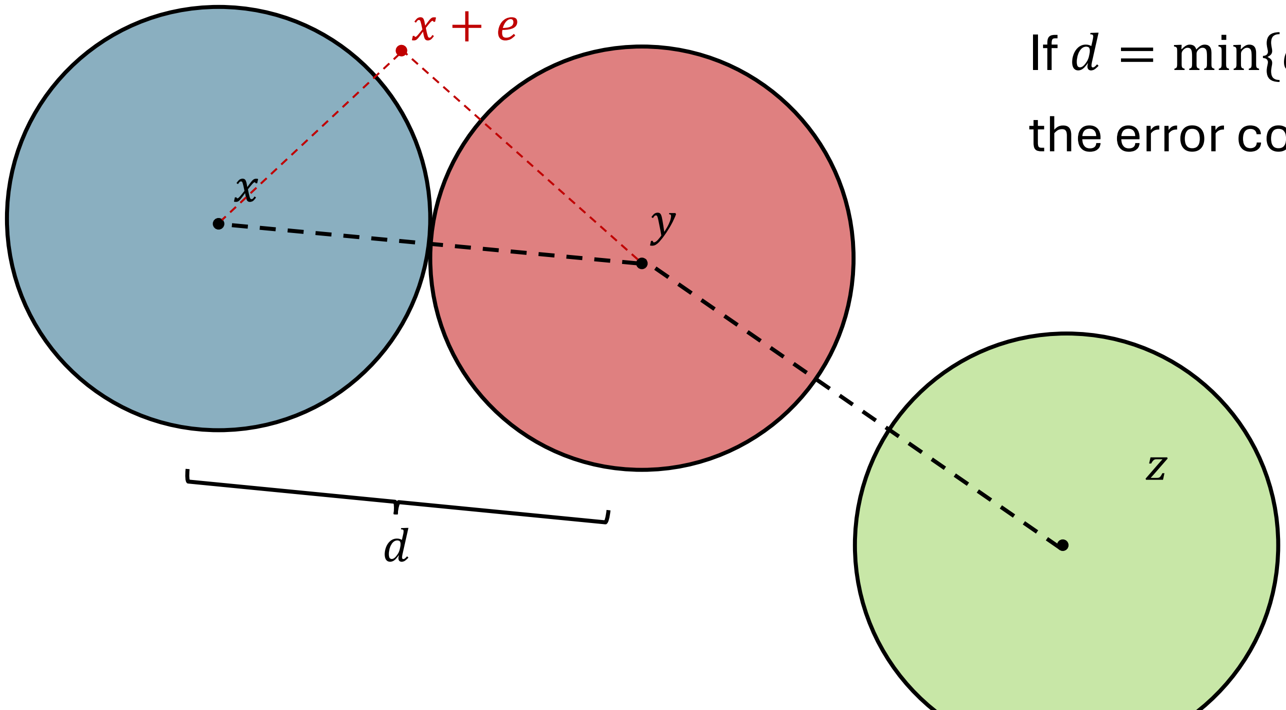
How to decode?

The **Hamming weight** of a codeword is $w_H(x) = \#\{i : x_i \neq 0\}$.
The **Hamming distance** of x, y is $d_H(x, y) = w_H(x - y)$.



How to decode?

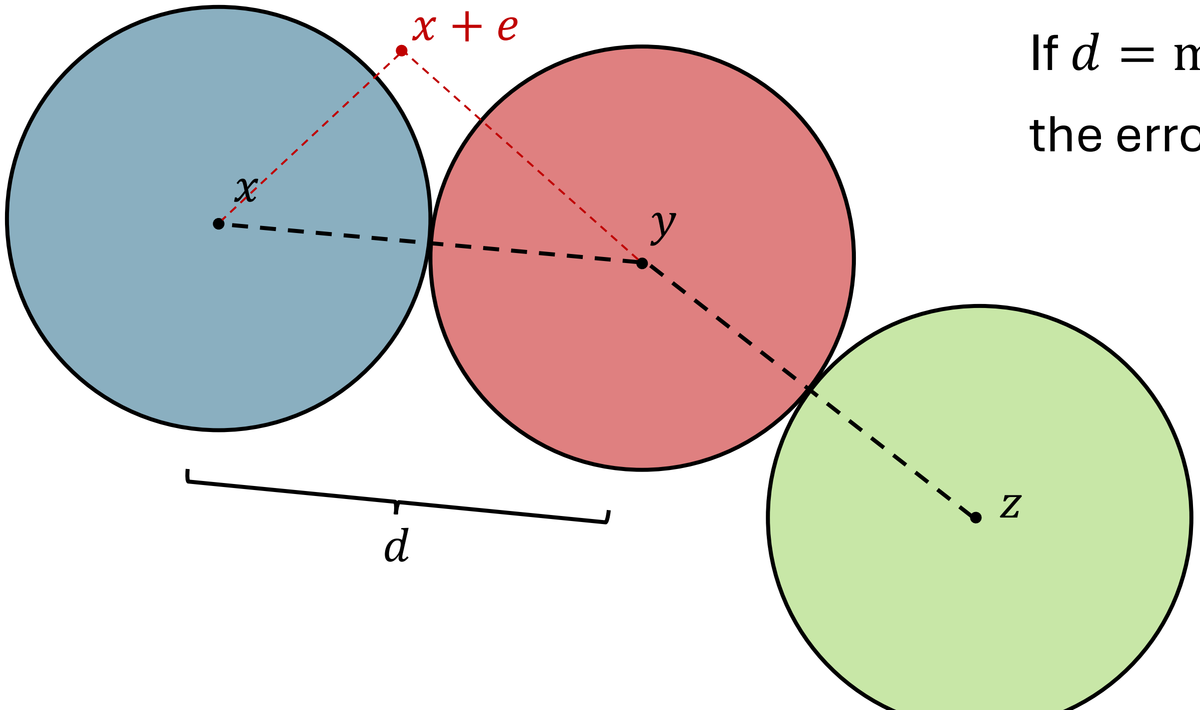
Nearest neighbour decoding.
Receive $x + e$, for $x \in C$ and e error.
Return the $c \in C$ closest to $x + e$.



If $d = \min\{d_H(x, y) : x, y \in C\}$,
the error correction capacity is $\left\lfloor \frac{d-1}{2} \right\rfloor$.

How to decode?

Nearest neighbour decoding.
Receive $x + e$, for $x \in C$ and e error.
Return the $c \in C$ closest to $x + e$.



If $d = \min\{d_H(x, y): x, y \in C\}$,
the error correction capacity is $\left\lfloor \frac{d-1}{2} \right\rfloor$.

How to decode?

Nearest neighbour decoding.
Receive $x + e$, for $x \in C$ and e error.
Return the $c \in C$ closest to $x + e$.

Theorem (Singleton Bound).
If d is the minimum distance of an $[n, k]$-code, $d \leq n - k + 1$.

How to decode?

Syndrome decoding.

Receive $x + e$, for $x \in C$ and e error. Then

$$H(x+e)^T = Hx^T + He^T = He^T$$

How to decode?

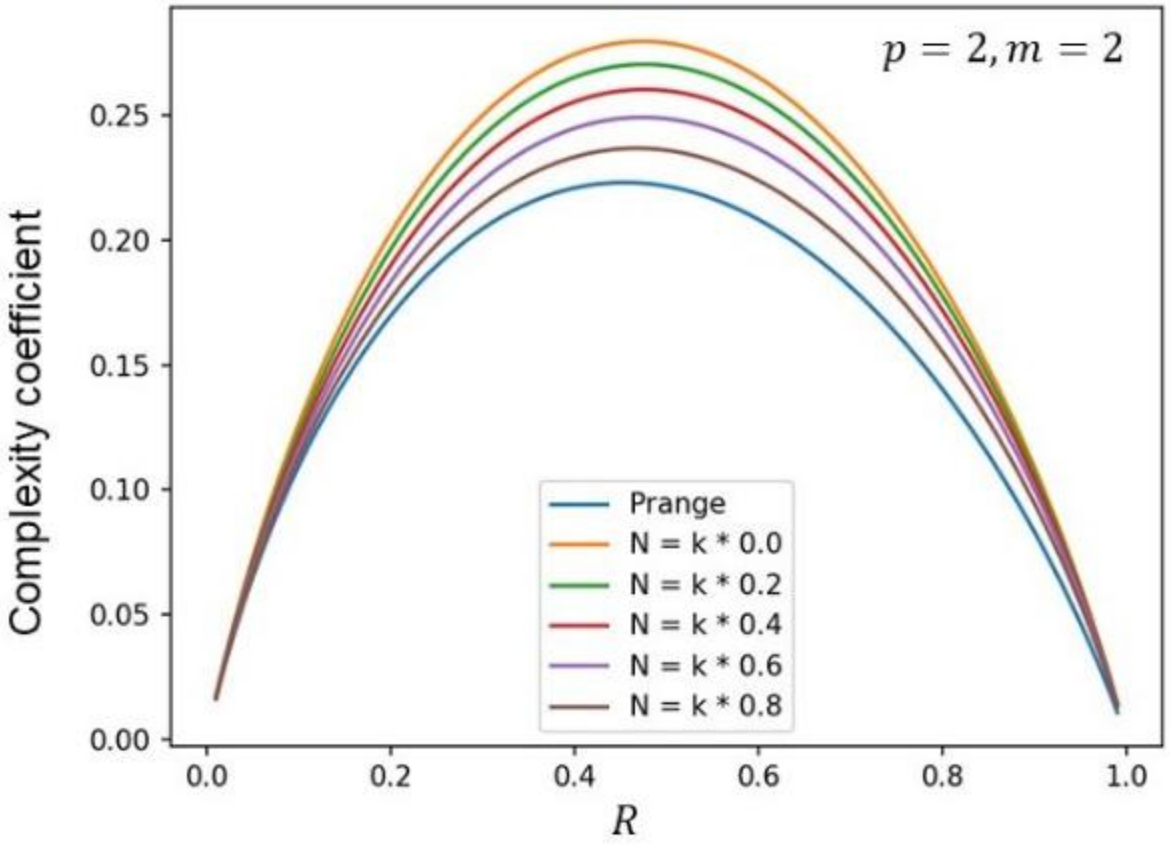
Build a table of all $n - k$ cosets of form $e + C$.

```
0000000 | 0000000  0001111  0010011  0011100  0100101  0101010  0110110  0111001  1000110  1001001  1010101  1011010  1100011  1101100  1110000  1111111
0000001 | 0000001  0001110  0010010  0011101  0100100  0101011  0110111  0111000  1000111  1001000  1010100  1011011  1100010  1101101  1110001  1111110
0000010 | 0000010  0001101  0010001  0011110  0100111  0101000  0110100  0111011  1000100  1001011  1010111  1011000  1100001  1101110  1110010  1111101
0000100 | 0000100  0001011  0010111  0011000  0100001  0101110  0110010  0111101  1000010  1001101  1010001  1011110  1100111  1101000  1110100  1111011
0001000 | 0001000  0000111  0011011  0010100  0101101  0100010  0111110  0110001  1001110  1000001  1011101  1010010  1101011  1100100  1111000  1110111
0010000 | 0010000  0011111  0000011  0001100  0110101  0111010  0100110  0101001  1010110  1011001  1000101  1001010  1110011  1111100  1100000  1101111
0100000 | 0100000  0101111  0110011  0111100  0000101  0001010  0010110  0011001  1100110  1101001  1110101  1111010  1000011  1001100  1010000  1011111
1000000 | 1000000  1001111  1010011  1011100  1100101  1101010  1110110  1111001  0000110  0001001  0010101  0011010  0100011  0101100  0110000  0111111
```

Find the coset of e via the syndrome.
Pick as candidate for e the element of lowest weight.

How to decode?

For a random code...



Behind the trick

Goal. Build codewords c such that

$$Hc^T = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} c^T = 0$$

Behind the trick

Goal. Build codewords c such that

$$Hc^T = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} c^T = 0$$

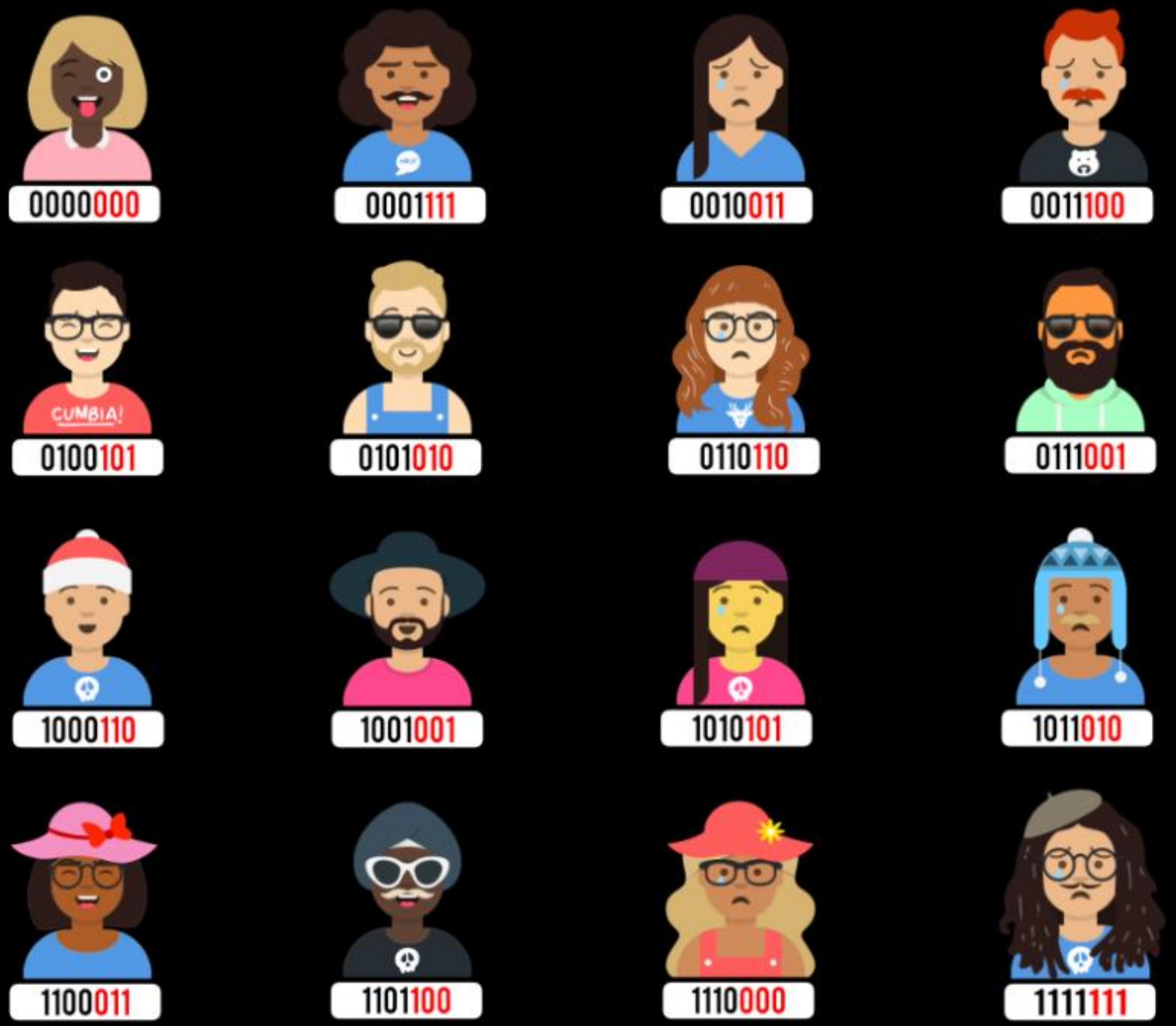
1. Identify 7 *independent* characteristics.

2. Use the generator matrix, it’s easier.

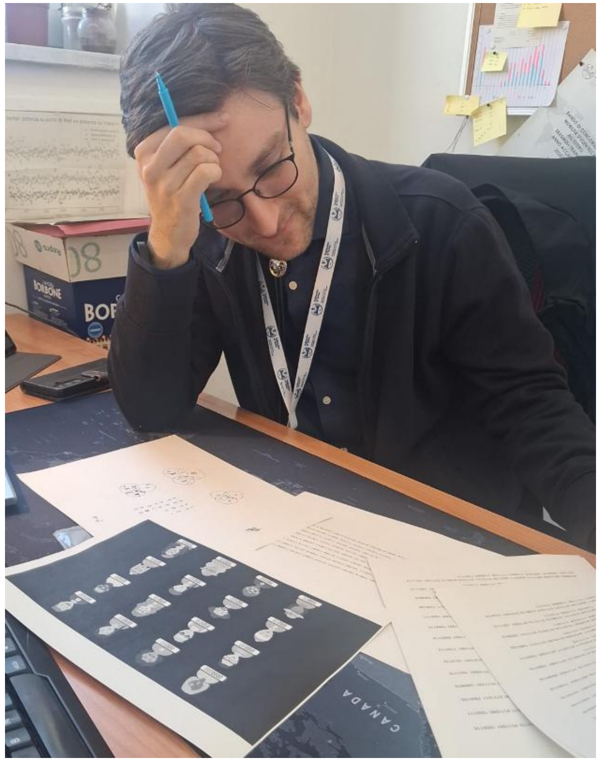
3. For every codeword c, generate the individual using them.


CAPUCINE
1010

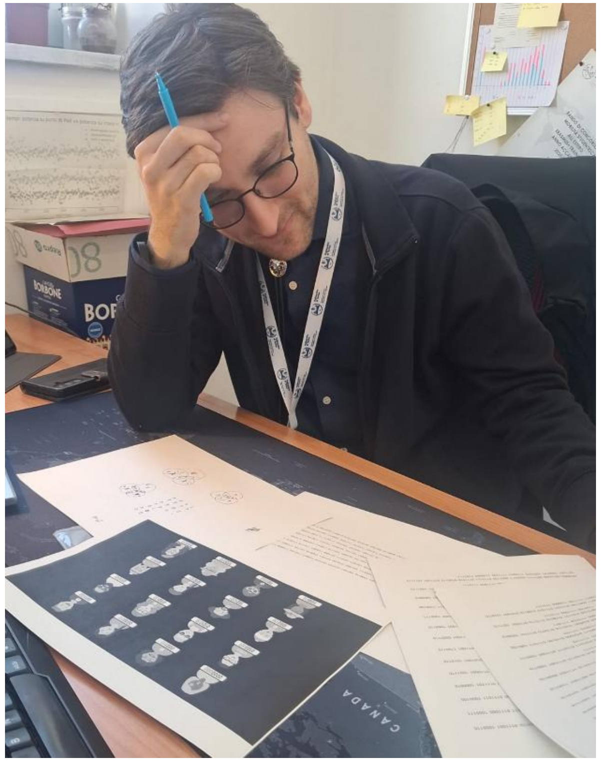

CAPUCINE
1010101


0000000
0001111
0010011
0011100
CUMBIA!
0100101
0101010
0110110
0111001
1000110
1001001
1010101
1011010
1100011
1101100
1110000
1111111

Behind the trick



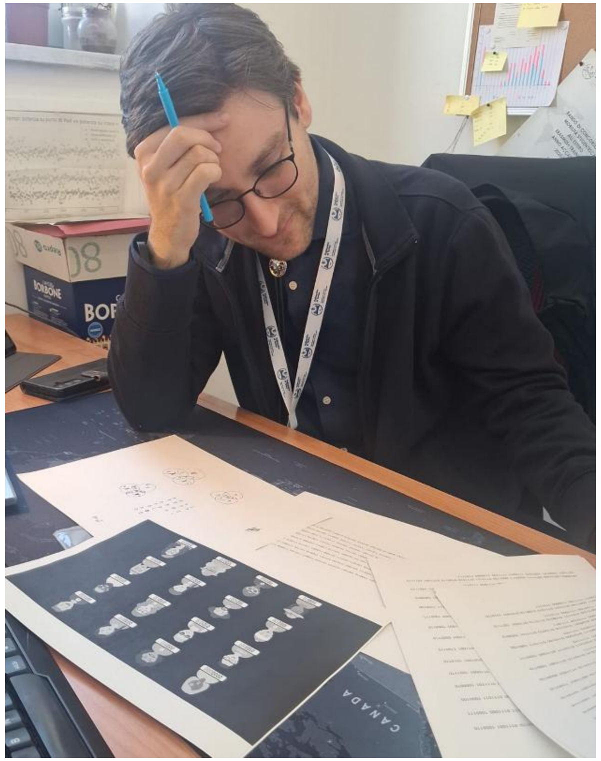
Behind the trick



```
answer with 'y' for Yes or 'n' for No:
1. Hat? 1
2. Glasses? 0
3. Is sad? 1
4. Moustache? 0
5. Drawing on shirt? 1
6. Blue shirt/garment? 1
7. Dark hair? 1
Lie detected at question 6!
Identified Person: CAPUCINE

Do you want to play again? (y/n):
```

Behind the trick



```
answer with 'y' for Yes or 'n' for No:
1. Hat? 1
2. Glasses? 0
3. Is sad? 1
4. Moustache? 0
5. Drawing on shirt? 1
6. Blue shirt/garment? 1
7. Dark hair? 1
Lie detected at question 6!
Identified Person: CAPUCINE

Do you want to play again? (y/n):
```

2: CRYPTOGRAPHY

Atbash cipher

*“And after all of them, the king of **Sheshak** will drink it too.”*

(Book of Jeremiah, 25:26)

Sheshak

shin-shin-kaf

22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
ת	ש	ר	ק	צ	פ	ע	ס	נ	מ	ל	כ	י	ט	ח	ז	ו	ה	ד	ג	ב	א
Tav	Shin	Reish	Qof	Tsadé	Pé	Ayin	Samekh	Noun	Meme	Lamed	Kaf	Yod	Teith	Heith	Zayin	Vav	Hé	Dalet	Guimel	Beith	Alpeh

Atbash cipher

“And after all of them, the king of ***Sheshak*** *will drink it too.”*

(Book of Jeremiah, 25:26)

Sheshak

shin-shin-kaf $\xrightarrow{\varphi_{atbash}}$ *bet-bet-lamed*

22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
ת	ש	ר	ק	צ	פ	ע	ס	נ	מ	ל	כ	י	ט	ח	ז	ו	ה	ד	ג	ב	א
Tav	Shin	Reish	Qof	Tsadé	Pé	Ayin	Samekh	Noun	Meme	Lamed	Kaf	Yod	Teith	Heith	Zayin	Vav	Hé	Dalet	Guimel	Beith	Alpeh

$$\varphi_{atbash}\colon \mathbb{Z}_{22} \to \mathbb{Z}_{22}$$
$$x \mapsto (23 - x) \bmod 22$$

Atbash cipher

*"And after all of them, the king of **Sheshak** will drink it too."*

(Book of Jeremiah, 25:26)

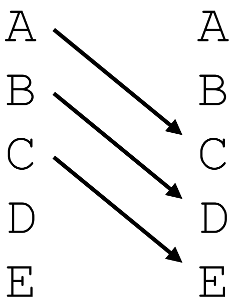
Sheshak		**Babylon**
shin-shin-kaf	$\xrightarrow{\varphi_{atbash}}$	*bet-bet-lamed*

22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
ת	ש	ר	ק	צ	פ	ע	ס	נ	מ	ל	כ	י	ט	ח	ז	ו	ה	ד	ג	ב	א
Tav	Shin	Reish	Qof	Tsadé	Pé	Ayin	Samekh	Noun	Meme	Lamed	Kaf	Yod	Teith	Heith	Zayin	Vav	Hé	Dalet	Guimel	Beith	Alpeh

$$\varphi_{atbash}: \mathbb{Z}_{22} \to \mathbb{Z}_{22}$$
$$x \mapsto (23 - x)\ mod\ 22$$

Caesar cipher

$$\varphi_{caesar}\colon \mathbb{Z}_{26} \to \mathbb{Z}_{26}$$
$$x \mapsto (x + k)\ mod\ 26$$




```
message:    ATTACKATDAWN
key:        C
ciphertext: DWWDFNDWGDZQ
```

$$H(K) = -\sum_{k \in K} P[k] \cdot \log_2 P[k]$$

$$H_{caesar} \simeq 4.7$$

Vigenere cipher

$$k = (k_0, \dots, k_{t-1})$$

$$\varphi_{vigenere}: \mathbb{Z}_{26} \to \mathbb{Z}_{26}$$
$$x \mapsto (x_i + k_{i \, mod \, t}) \, mod \, 26$$

```
message:    ATTACKATDAWN
key:        keykeykeykey
ciphertext: KXRKGIKXBYAL
```

$$H_{vigenere} \simeq 4.7 \cdot t$$

Vigenere cipher

$$k = (k_0, \dots, k_{t-1})$$

$$\varphi_{vigenere}: \mathbb{Z}_{26} \to \mathbb{Z}_{26}$$
$$x \mapsto (x_i + k_{i \bmod t}) \bmod 26$$

```
message:    ATTACKATDAWN
key:        keykeykeykey
ciphertext: KXRKGIKXBKAL
```

$$H_{vigenere} \simeq 4.7 \cdot t$$

Vernam cipher

$$k = (k_0, \dots, k_{n-1})$$

$$\varphi_{vernam}\colon \mathbb{Z}_{26} \to \mathbb{Z}_{26}$$
$$x_i \mapsto (x_i + k_{i \bmod n}) \bmod 26$$

```
message:    ATTACKATDAWN
key:        aezklwgrmali
ciphertext: AXSKNGGKPOHV
```

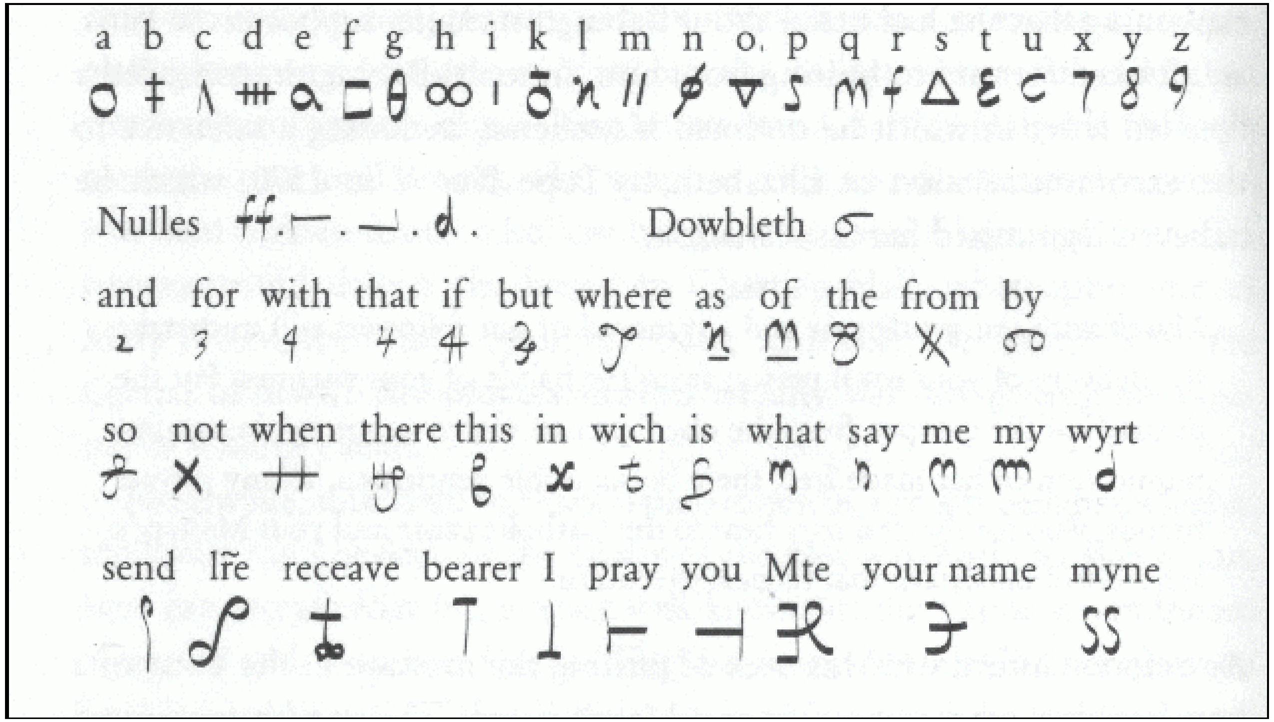
$$H_{vernam} \simeq 4.7 \cdot n$$

If k is uniformly distributed and independent on m this cipher is perfect, i.e.

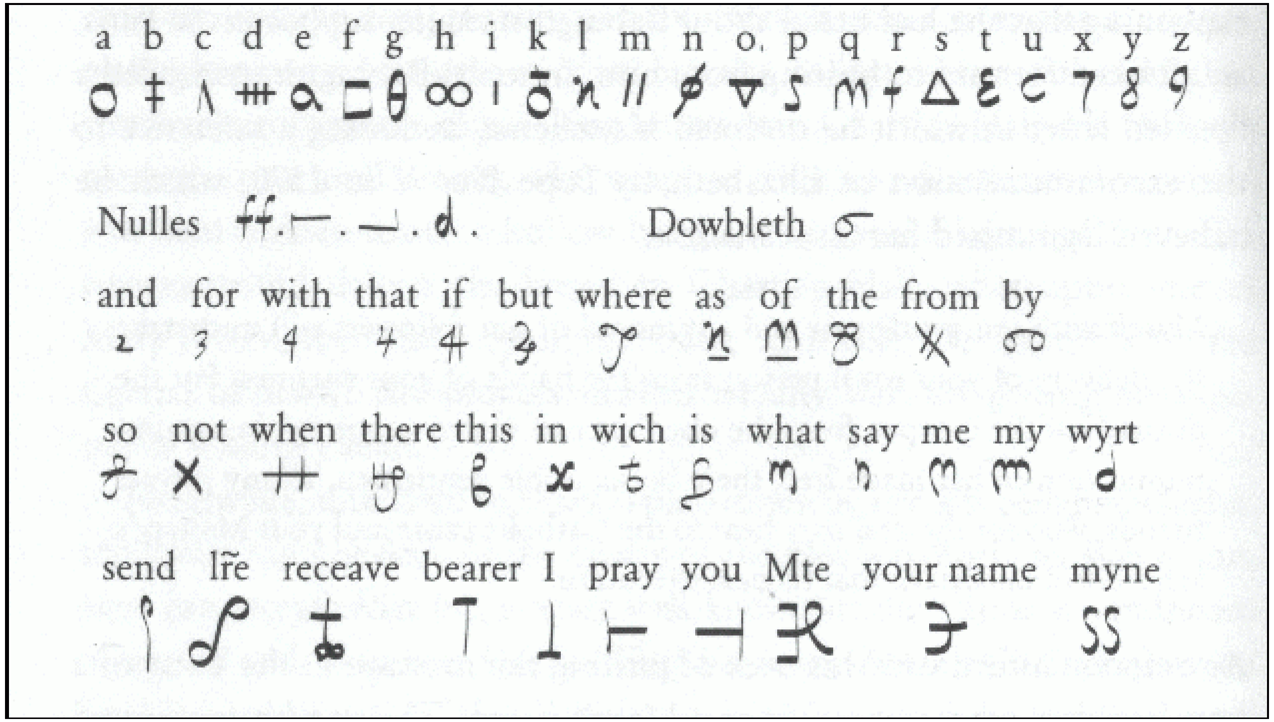
$$P[M = m \,|C = c] = P[M = m]$$

Nomenclators

Nomenclators

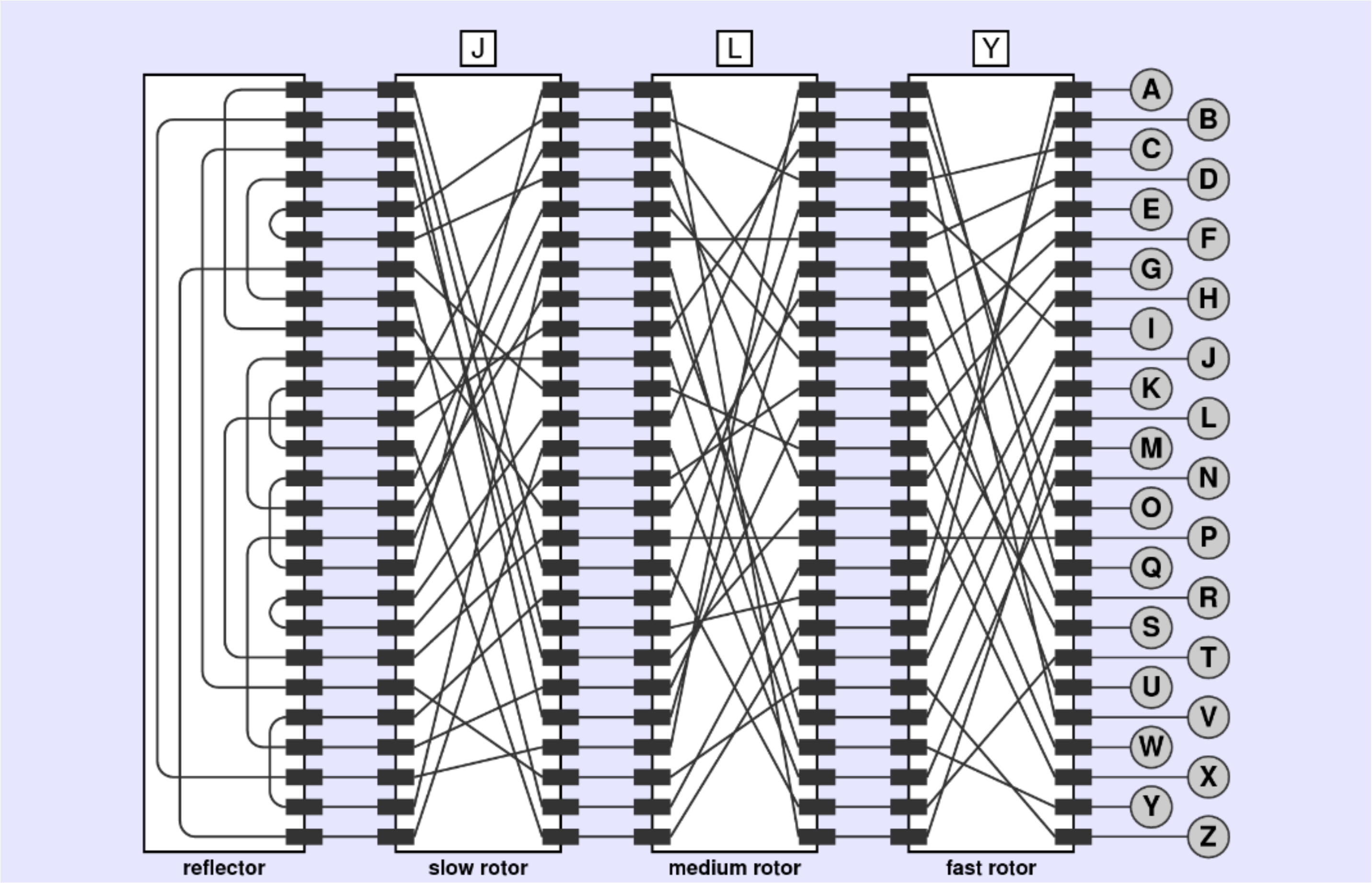


Nomenclators



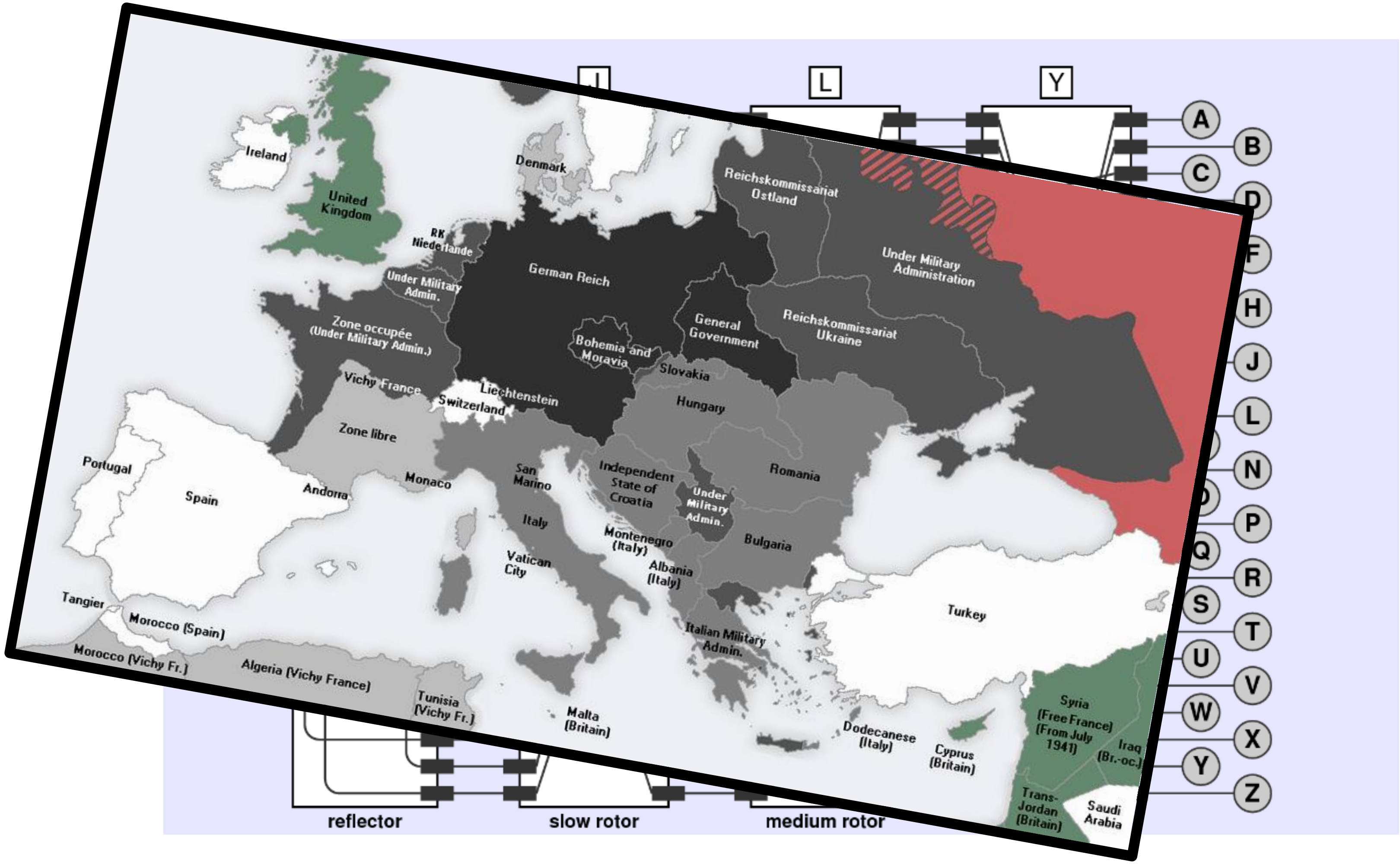
Machines to the rescue

Machines to the rescue



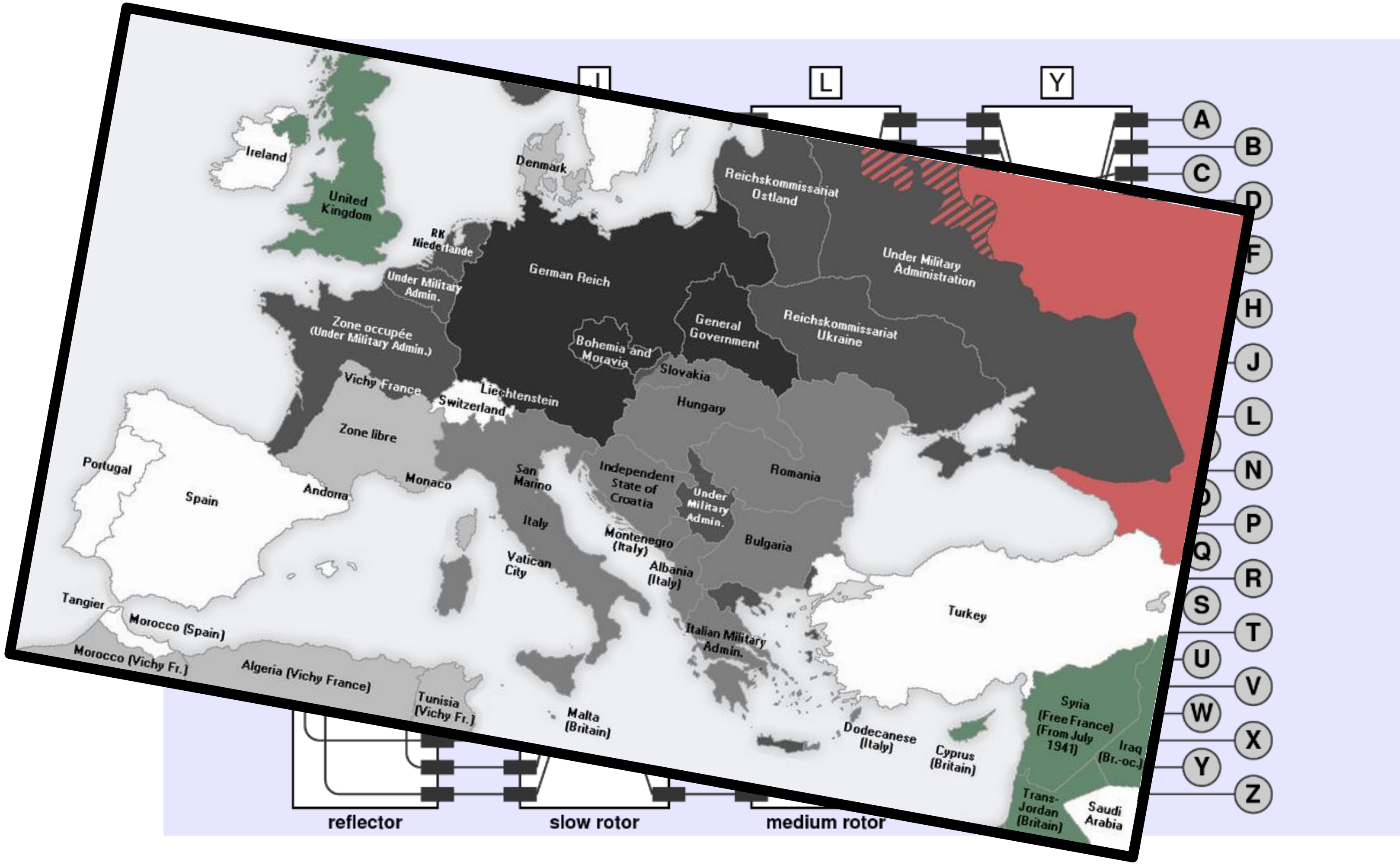
$$H_{enigma} = 65 \text{ or } 67$$

Machines "to the rescue"



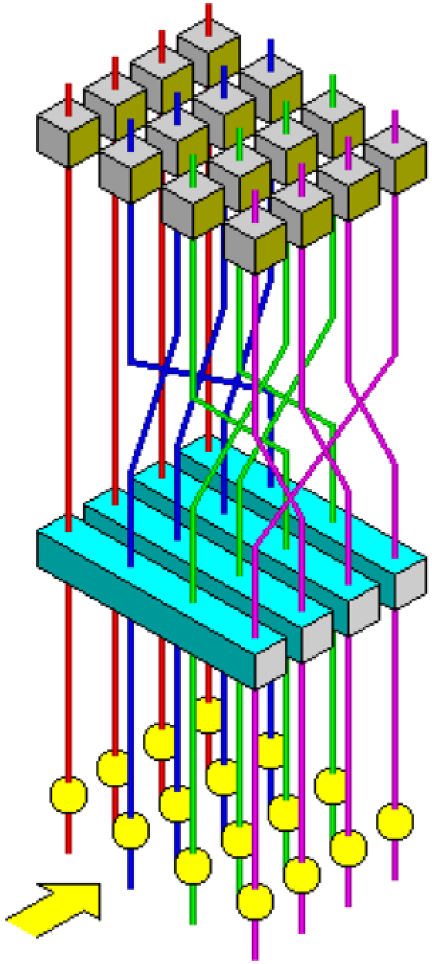
$$H_{enigma} = 65 \text{ or } 67$$

Machines “to the rescue”



$$H_{enigma} = 65 \text{ or } 67$$

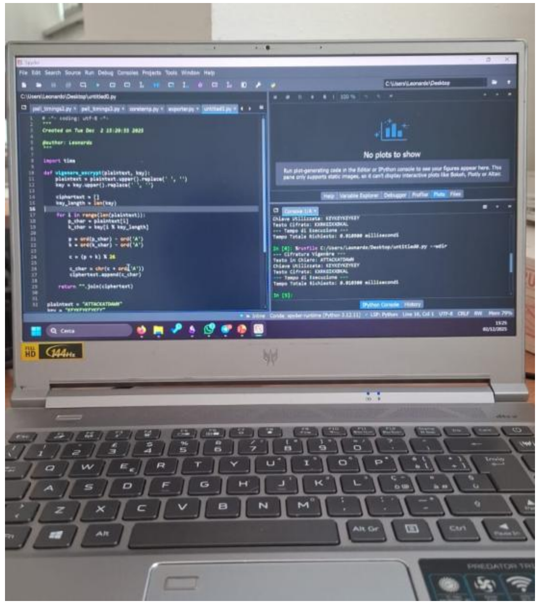
Machines to the rescue



$\times$ 10 (or 12 or 14)

$H_{AES} = 128 \text{ or } 192 \text{ or } 256$

Machines to the rescue

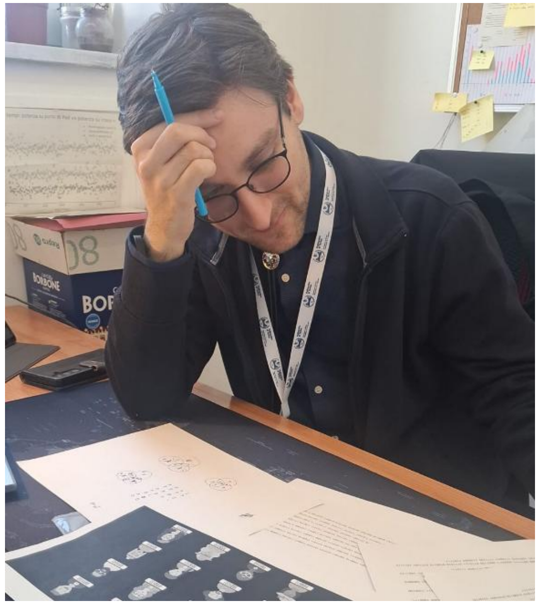


	me	**my laptop**
Vigenere	15 minutes (128 chars)	0.00001 seconds
AES-128	6 to 8 hours (128 chars)	0.00015 seconds

** mathematicians were hurt for this demonstration*

Machines to the rescue

Encrypting (Dec. 2025)



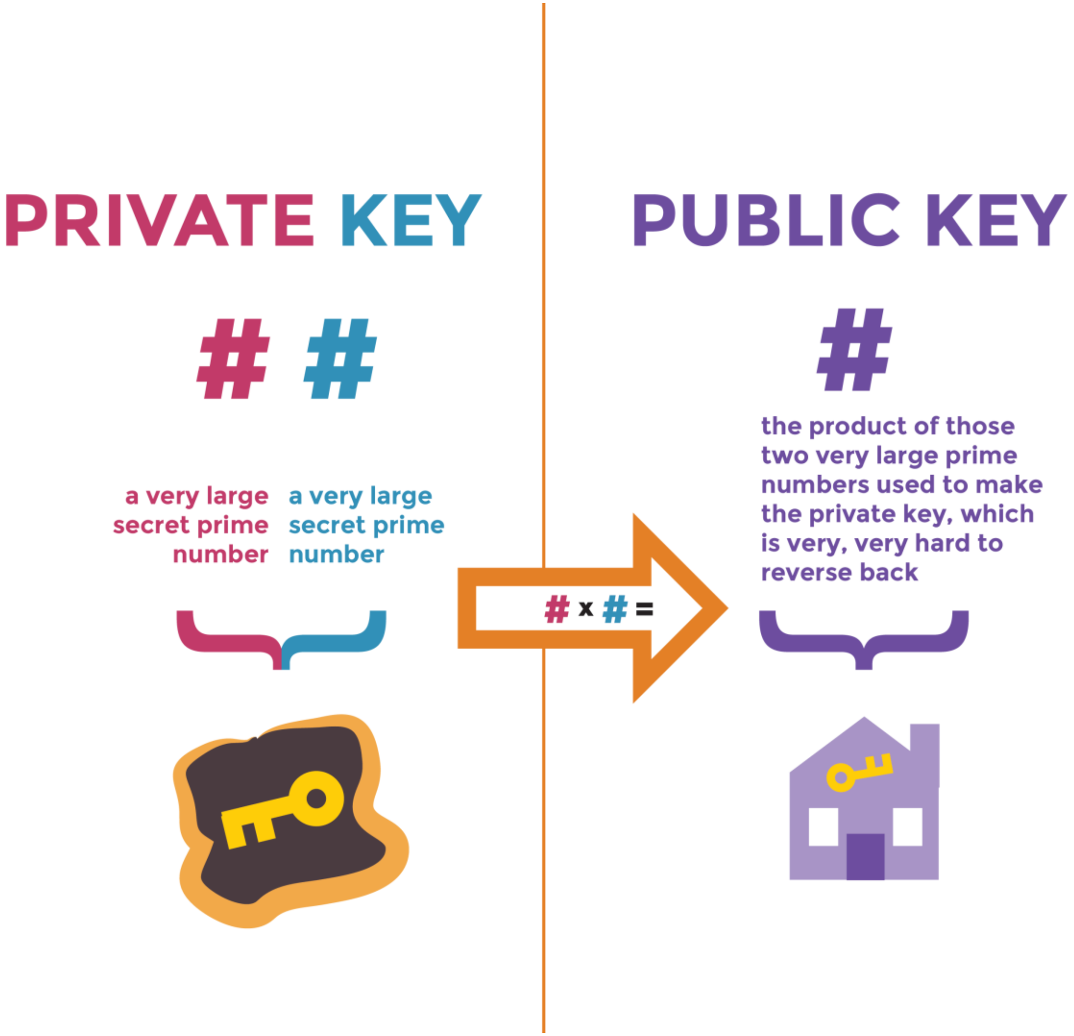
Decoding (Feb. 2026)

3: CODE-BASED CRYPTOGRAPHY

New paradigms

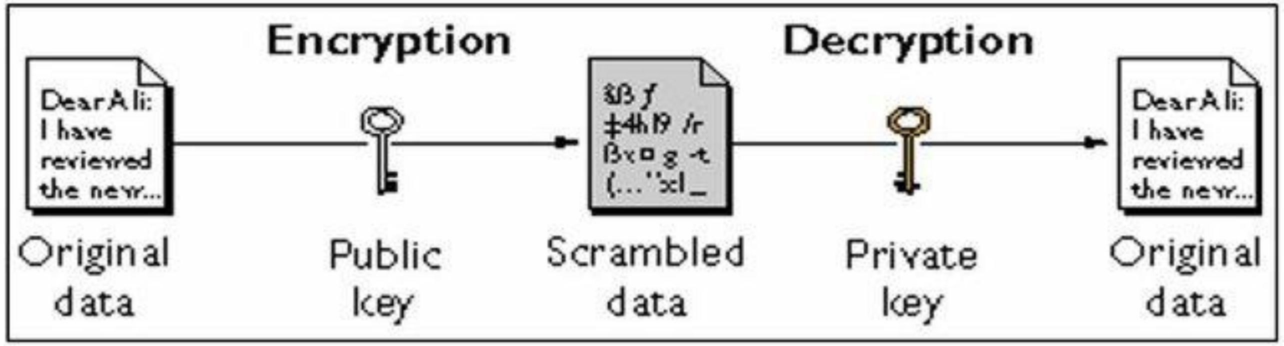
- Block ciphers
- Hash functions
- Stream ciphers
- Signature schemes
- Zero knowledge proofs
- Key encapsulation mechanisms
- ...

New paradigms



New paradigms

PRIVATE KEY | **PUBLIC KEY**



McEliece (1978)

KeyGen:

G generator matrix of a code C
Δ efficient description of C decoding t errors
$G' = SGP$ for S change of basis, P permutation
public key G'
private key S, G, P

Encrypt(x):

c $\leftarrow xG$ encoding
$e \leftarrow$ random noise
return y $= c + e$

Decrypt(y)

return $x = \text{decode}(yP^{-1}, \Delta)S^{-1}$

** very very high-level*

Thanks!