Mathématiques-Cryptographie

Anne Garcia-Sanchez

M2i cyber2 dev - CFA CCI Avignon

11 janvier 2024

Chiffrement par substitution poly-alphabétique

Chiffre de Vigenère

Chiffre de Beaufort

Chiffre de Hill

Enigma

Chiffre de Hill

mathématicien américain, Lester S. Hill: 1931

extension du chiffrement affine

Clé: matrice carrée inversible K de taille t dans $\mathbb{Z}/26\mathbb{Z}$.

Le bloc $(m_1, \dots, m_t)$ chiffré $(c_1, \dots, c_t)$ par multiplication matricielle:

$$\begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_t \end{pmatrix} = K \times \begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_t \end{pmatrix} \pmod{26}$$

Chiffre de Hill

Exemple: chiffrer le mot MESSAGE avec la matrice clé $\begin{pmatrix} 6 & 7 \\ 15 & 21 \end{pmatrix}$

nombre de lettres du message impair: rajouter X à la fin

message clair	M	E	S	S	A	G	E	X
rang lettre du message	12	4	18	18	0	6	4	23

on obtient la matrice: $\begin{pmatrix} 12 & 18 & 0 & 4 \\ 4 & 18 & 6 & 23 \end{pmatrix}$

produit matriciel $\begin{pmatrix} 6 & 7 \\ 15 & 21 \end{pmatrix}$

$$\begin{pmatrix} 100 & 234 & 42 & 185 \\ 264 & 648 & 126 & 543 \end{pmatrix} \bmod 26 = \begin{pmatrix} 22 & 0 & 16 & 3 \\ 4 & 24 & 22 & 23 \end{pmatrix}$$

On obtient le message chiffré: WEAYQWDX

Chiffre de Hill

Déchiffrement

$$\begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_t \end{pmatrix} = M^{-1} \times \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_t \end{pmatrix} \pmod{26}$$

M^{-1} inverse modulaire de la matrice M

Chiffre de Hill

cas où M est une matrice 2×2

$$\begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} m_1 \\ m_2 \end{pmatrix} \pmod{26}$$

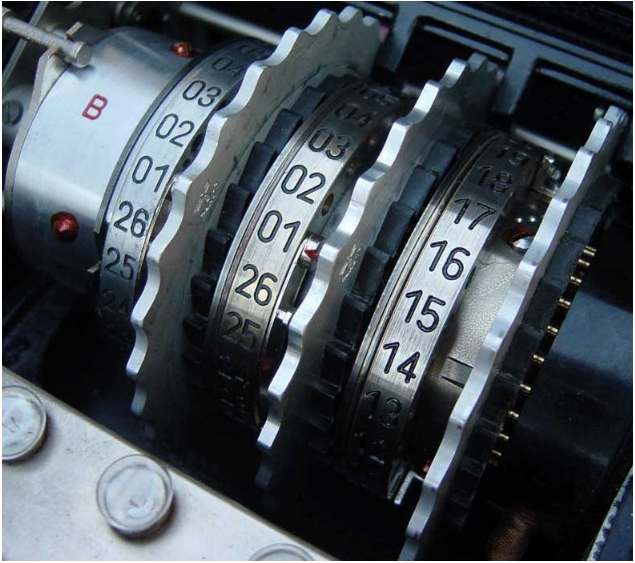
$$\begin{pmatrix} m_1 \\ m_2 \end{pmatrix} = (ad - bc)^{-1} * \begin{pmatrix} d & -b \\ -c & a \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \pmod{26}$$

Machine Enigma

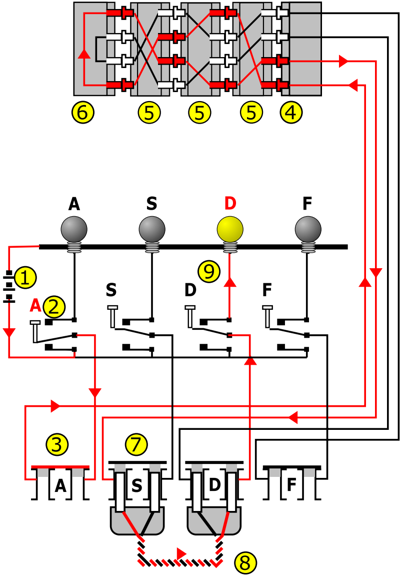
1918 Arthur Scherbius, ingénieur allemand

Machine Enigma

Machine Enigma: les rotors



Machine Enigma



Machine Enigma

Enigma M3 Code Book (UKW-B Reflector) - April 1940

Datum [Date]	*Walzenlage* [Rotors]	*Ringstellung* [Ring settings]	*Steckerverbindungen* [Plugboard settings]	*Grundstellung* [Initial rotor positions]
30	V III II	AKK	AO HI MU SN VX ZQ	FDV
29	IV III V	JHS	LW RH UQ VP YM ZA	OTO
28	IV I II	DIL	EM HL PZ RJ SV UQ	JJK
27	III I IV	ICC	AX CW FZ KT PO SQ	RXV
26	IV II III	ECW	GS JD MN OQ VF XH	GUB
25	V III I	MFO	DW GO HE UF YI ZJ	ZBY
24	V III I	UCO	GC JU KE MF OD XY	BDT
23	II V IV	RWQ	BN FK OS PW TA ZE	IYM
22	IV II I	TRK	BN DU JI OK TF XC	SFX
21	II V III	CTZ	AF BK GJ VQ XH YT	TQO
20	I V III	XOM	BX IS LY NF QO WA	DKV
19	IV V II	LDQ	CR FO LI NM PD XH	IAH
18	IV I III	NWL	HV IM JB OT QA UF	HSP
17	II IV III	HFZ	FE IB OQ VC YW ZM	GPZ
16	II I IV	UBJ	CO GV IH KD ML RB	PJU
15	I II IV	BCG	ES GD IZ JF LN YA	KFQ
14	II V IV	EAP	BT CO NE PK VY ZI	CCH
13	I V II	AOK	CA DZ HK LP RQ YV	DMF
12	III I II	CKU	CK IZ QT NP JY GW	VQN
11	II III I	BHN	FR LY OX IT BM GJ	XIO
10	I V II	QKP	AF HQ IJ OT PB YG	MSW
9	V I II	UTC	DE FT IP OB UC YL	EQL
8	V IV II	GDJ	GT HR JI OK QE UZ	PLE
7	I II III	WNM	HK CN IO FY JM LW	RAO
6	V I III	ETT	FT HC KD PM YO ZB	HXA
5	V I III	MHY	BZ HS JF MW NG PV	XXJ
4	IV V III	WXE	DG IN JT UC VB WZ	OFP
3	IV II III	LIQ	BJ HC PI RF UO ZQ	KTR
2	II I V	NQC	AV KZ MS QP XF YU	ZJR
1	V II I	IHQ	ET LD NP QS RA UW	UJJ

numpy

```
>>> values = [[1,2],[3,4],[5,6],[7,8]]
>>> np.array(values)
array([[1, 2],
       [3, 4],
       [5, 6],
       [7, 8]])
>>> np.array(values).transpose()
array([[1, 3, 5, 7],
       [2, 4, 6, 8]])
>>>
```

numpy

```
>>> import numpy as np
>>> ranks = [1,2,3,4,5,6,7,8]
>>> np.array(ranks)
array([1, 2, 3, 4, 5, 6, 7, 8])
>>> np.array(ranks).reshape((2, 4))
array([[1, 2, 3, 4],
       [5, 6, 7, 8]])
>>> np.array(ranks).reshape((4, 2))
array([[1, 2],
       [3, 4],
       [5, 6],
       [7, 8]])
>>> np.array(ranks).reshape((4, 2)).transpose()
array([[1, 3, 5, 7],
       [2, 4, 6, 8]])
>>>
```

numpy: produit de matrices avec @

```
>>>matA
array([[1, 2],
       [3, 4],
       [5, 6]])
>>>matB = np.array([[1, 1, 1], [1, 1, 1]])
>>>matB
array([[1, 1, 1],
       [1, 1, 1]])
>>>matA @ matB
array([[ 3,  3,  3],
       [ 7,  7,  7],
       [11, 11, 11]])
>>>matB @ matA
array([[ 9, 12],
       [ 9, 12]])
>>>
```