SUR LES TRACES D'ENIGMA

« LA » CÉLÈBRE MACHINE UTILISÉE PAR L'ARMÉE ALLEMANDE POUR CRYPTER SES COMMUNICATIONS PENDANT LA SECONDE GUERRE MONDIALE

PAR MARIE BERNIER (1ÈRE S), QUENTIN RIFFAULT (TERM. STI2D),
MATHYS GRANDET (TERM. S), MAXIME BAUSMAYER (1ÈRE S),
JULIEN GLEYZE (1ÈRE S), LUCAS JUDEAUX (1ÈRE S),
JULIEN LEBEAU (1ÈRE S) ET BENJAMIN ROJO (TERM. S)

INTRODUCTION

SUR LES TRACES D'ENIGMA

Nous sommes des élèves du lycée Maurice Genevoix, à Ingré.

Nous avons réalisé, dans le cadre de l’atelier Math.en.JEANS, un travail sur les méthodes de cryptage (codage) et de cryptanalyse (décodage) et surtout sur la machine Enigma.

La machine Enigma est la fameuse machine de cryptage qu’utilisait l’armée allemande pendant la seconde guerre mondiale.

Nous avions très envie d’en concevoir une. Pour y parvenir, il a fallu commencer par bien comprendre comment elle marchait ! C’est lors de ces recherches que deux groupes se sont constitués. L’un d’eux a travaillé sur le dénombrement des clés de la machine (ce qui la rend si difficile à « craquer ») pendant que l’autre groupe s’est attaqué à la conception des programmes informatiques simulant Enigma. Tout ça, pour tenter, de faire fonctionner une réplique électronique de cette machine !

PROBLÉMATIQUE

SUR LES TRACES D'ENIGMA

Voici les questions que nous nous sommes posées :

- Comment est constituée une machine Enigma ?
- Comment permet-elle de coder et de décoder des messages ?
- Est-il possible de réaliser une version de cette machine avec les outils numériques d'aujourd'hui ?

De nouvelles questions sont arrivées au fil de nos recherches :

- Pourquoi les messages cryptés par ces machines étaient-ils si difficiles à déchiffrer ?
- Combien existait-il de clés possibles pour régler, chaque jour, une machine Enigma ?

DÉVELOPPEMENT

0. Avant Enigma…
1. Les éléments d’une machine Enigma.
2. Comment permet-elle de coder et de décoder des messages ?
3. Pourquoi les messages cryptés par ces machines étaient-ils si difficiles à déchiffrer ?
4. Enigma et les outils numériques d’aujourd’hui.

0. Avant Enigma…

Nous avons commencé par réaliser des jeux de décryptage et de programmation pour connaître et comprendre, en premier, le code césar.

Ce code de cryptage est le plus connu. En effet, il est appliqué dans beaucoup d'exercices.

Celui-ci permet de crypter des messages en décalant les lettres de l'alphabet.

Le défi qu'on nous a lancé : trouver une méthode de décryptage et la programmer !

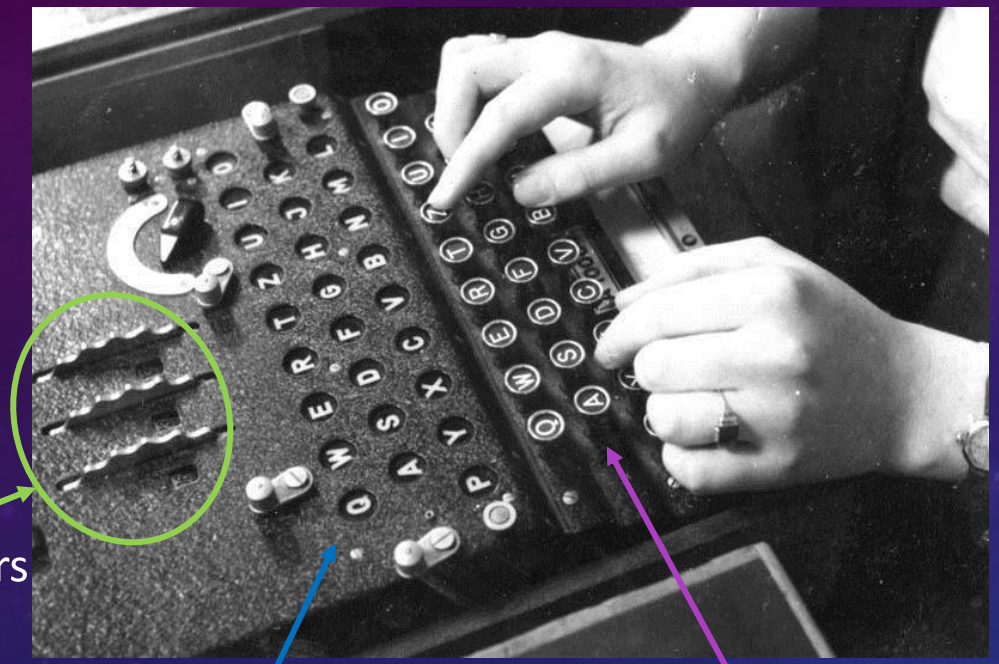
Pour cela, nous avons dû, d'abord, analyser ce code et nous nous sommes rendu compte que le calcul de fréquence d'apparition des lettres permettait de le « casser ».

En effet, dans le langage français, la lettre « E » apparaît le plus souvent.

Dans un message crypter en césar (initialement écrit en français et assez long pour que le calcul des fréquences soit fiable), la lettre apparaissant le plus souvent codera donc le « E ». La différence entre les places des deux lettres sera donc la valeur du décalage : E(=5) est codé en G(=7) – 7-5=2, la valeur du décalage est donc 2.

Nous avons alors écrit un programme permettant le codage et le décodage d'un message (voir annexe 1).

1. Les éléments d'une machine Enigma



Les rotors

Le tableau de lampes

Le clavier

Les rotors

Le tableau de lampes

Le « brouilleur » constitué des

trois rotors et du réflecteur

Le tableau de fiches

1. Les éléments d’une machine Enigma

Les rotors

Les rotors :

On aperçoit ici les fenêtres pour le réglage quotidien de leur orientation de départ.

Le tableau de fiches :
Chaque branchement échange deux lettres avant leur « passage » dans le brouilleur puis avant qu’une lampe ne s’allume.

Le « brouilleur » constitué des trois rotors et du réflecteur : chacun de ces éléments permute les lettres de l’alphabet.

1. Les éléments d’une machine Enigma

Le tableau de lampes

Le tableau de lampes :

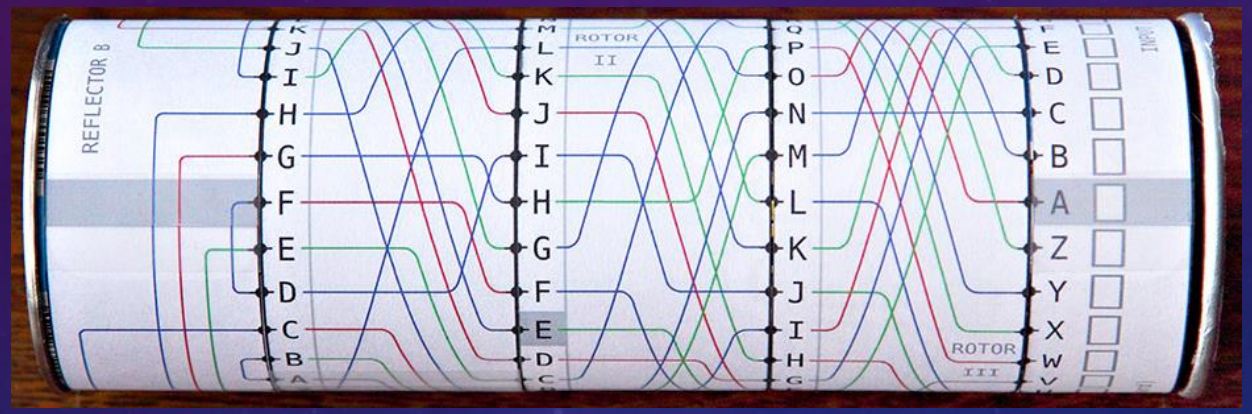
la lettre codée s’allume après chaque frappe.

Le clavier : Un opérateur tape les lettres une à une. Cela déclenche la rotation du rotor rapide (et parfois d’un (ou deux) autre(s) rotor(s)) puis un courant électrique traverse le « brouilleur » et provoque l’allumage d’une lampe…

1. Les éléments d’une machine Enigma

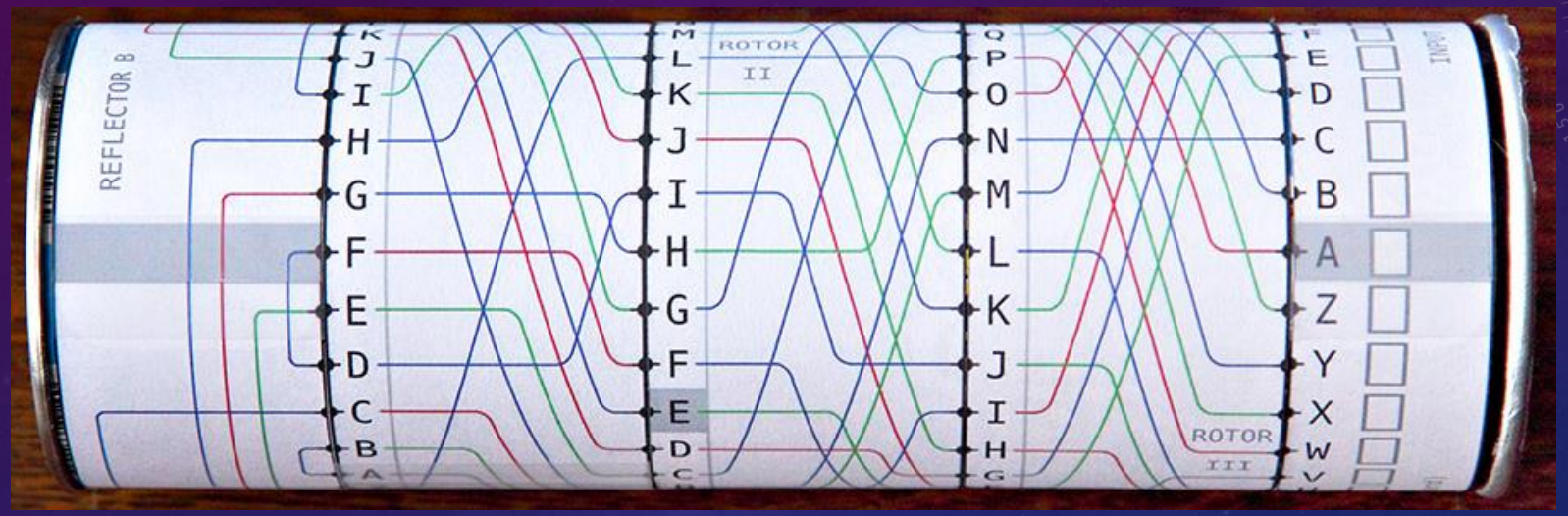
Nous avons fabriqué une version « papier » d’une machine Enigma grâce à des plans que nous ont donné nos professeurs avec une notice… en anglais !... Notre cadeau de Noël !!!

Elle nous sert beaucoup : d’abord pour comprendre comment ça marche, puis pour expliquer et faire des démonstrations. Mathys et Quentin s’en sont aussi beaucoup servi pour écrire leurs programmes !



- L’élément le plus à droite simule le clavier de la vraie machine : il est collé sur le cylindre.
- Viennent ensuite, de droite à gauche, trois bandes enroulées qui correspondent aux vrais rotors : le rotor III, puis le rotor II, puis le rotor I : ils peuvent tourner sur eux-mêmes.
- Enfin, tout à gauche, le réflecteur : comme le « clavier », il est collé sur le cylindre et ne peut donc pas tourner.
- Les traits de couleur représentent les circuits électriques authentiques des véritables rotors.

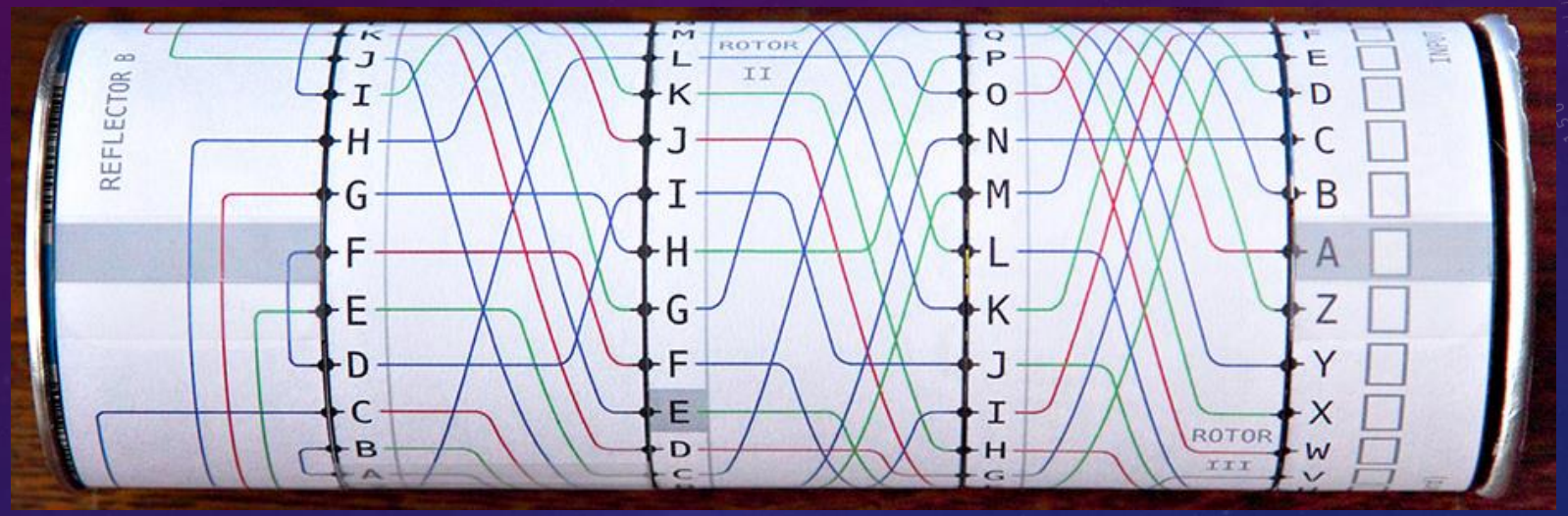
Une version « papier » d'une machine enigma



Avec les réglages visibles ici, on peut suivre le début du « trajet » de la lettre W dans la machine :

- Dans le rotor de droite, le W devient P…
- Dans le rotor central, ce P devient H… Qui devient G en traversant le rotor de gauche…
- Malheureusement, on ne voit plus la suite de ce trajet… Qui va jusqu'au « clavier » et nous donne la lettre codée.

1. Les éléments d’une machine Enigma



L’ingéniosité et la performance d’Enigma vient de la rotation de ses rotors.

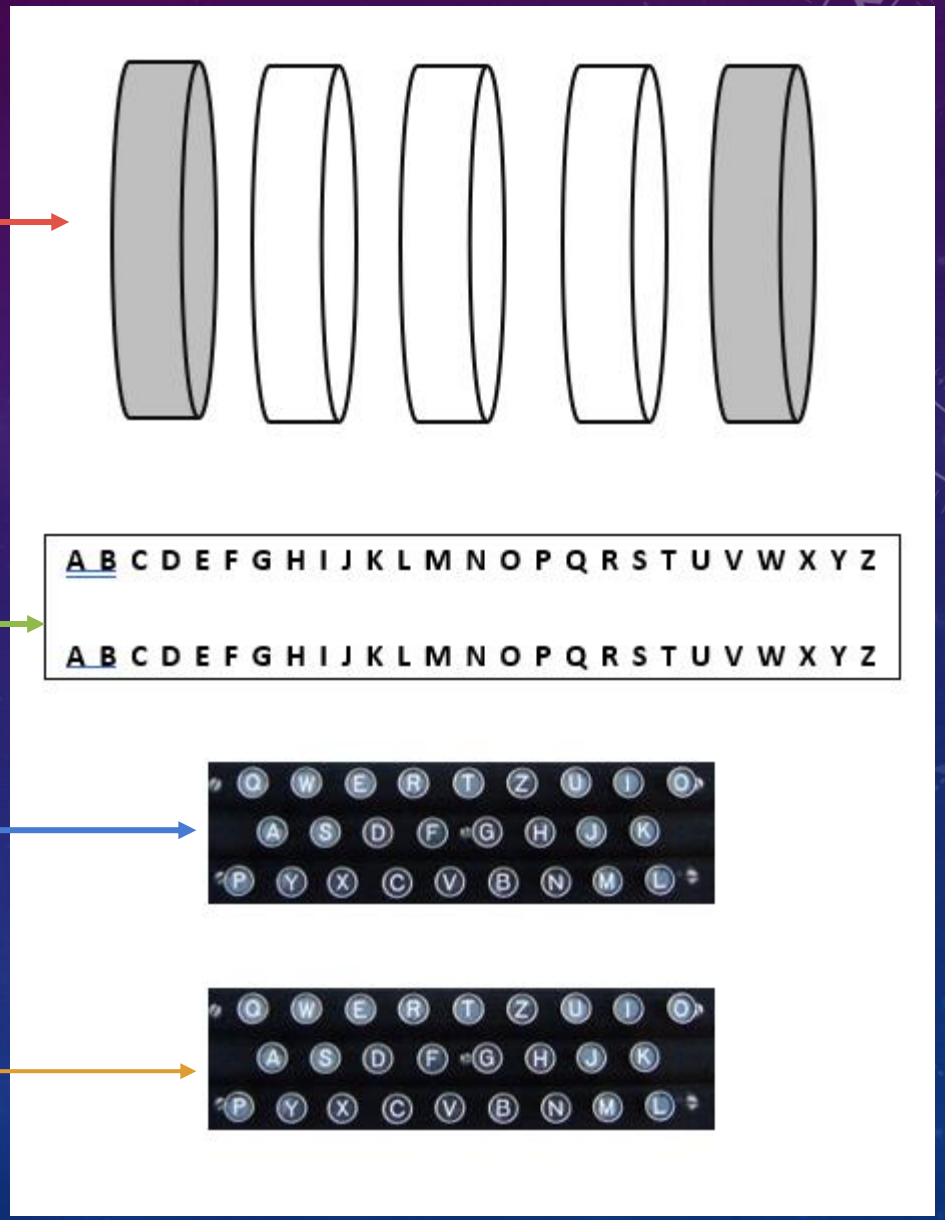
Chaque frappe provoque la rotation d’un cran du rotor situé à droite (le rotor dit « rapide ») : cela décale toutes les liaisons électriques à chaque frappe !

Sur notre machine en papier, on tourne d’un cran vers nous le rotor de droite AVANT d’observer le chemin parcouru par la lettre « frappée ».

2. Comment permet-elle de coder et de décoder des messages ?

SCHÉMA D'UNE MACHINE ENIGMA

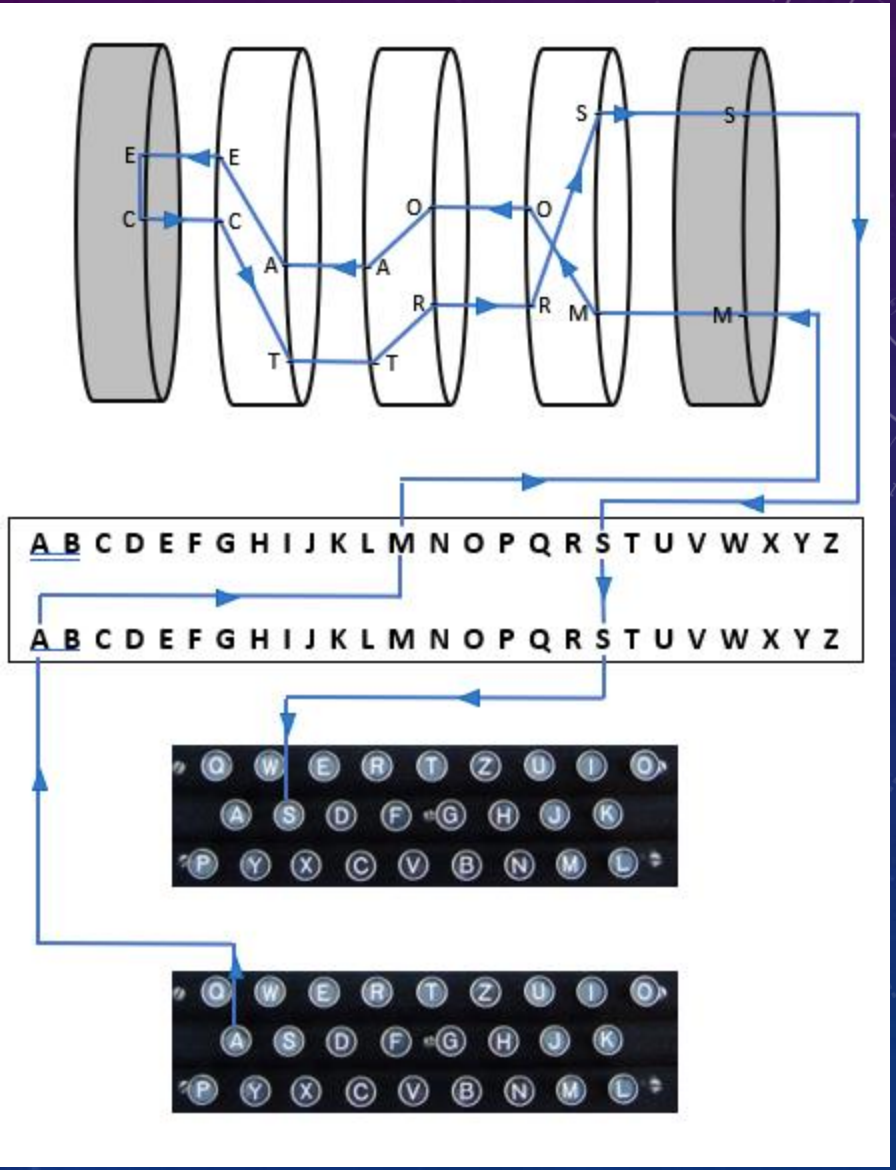
- Le clavier
- Le tableau de fiches
- Le « brouilleur » avec :
  - Un rotor fixe
  - 3 rotors mobiles
  - Un réflecteur
- Le tableau de lampes



2. Comment permet-elle de coder et de décoder des messages ?

- La lettre « A » est tapée au clavier.
- Le « A » devient « M » dans le tableau de fiches.
- Dans le « brouilleur »…
  - Le « M » devient « O », puis « A », puis « E »…
  - Dans le réflecteur devient « C ».
  - … Puis « T », puis « R », puis « S » dans les rotors.
- Le « S » n’est pas transformé par le tableau de fiches.
- La lettre « S » s ’allume sur le tableau de lampes.

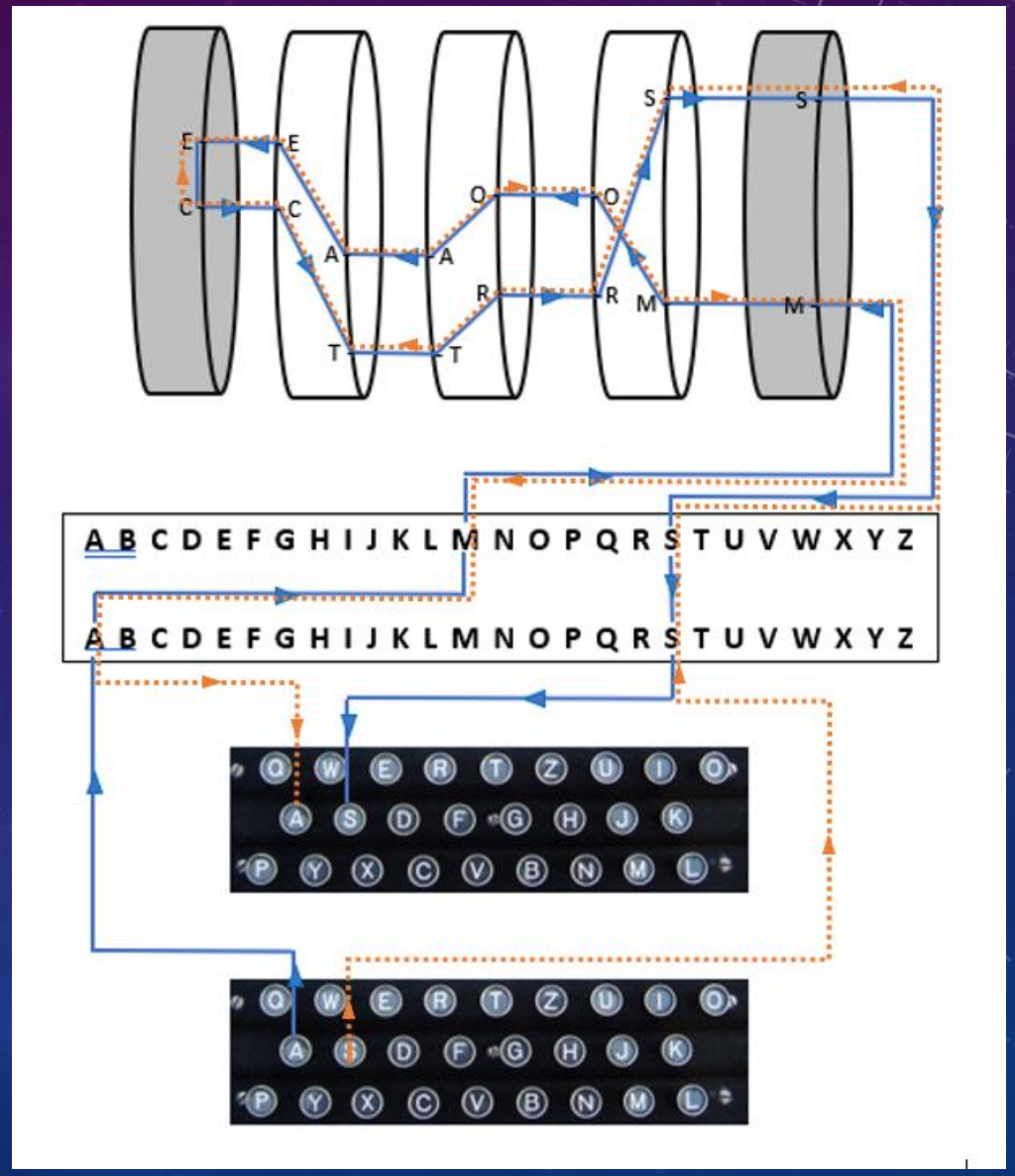
Le « A » est finalement codé en « S » lors de cette frappe.



2. Comment permet-elle de coder et de décoder des messages ?

Le « A » est finalement codé en « S » lors de cette frappe.

On observe, qu'avec les mêmes réglages, le « S » est transformé en « A » : c'est ce qui permet le décodage.



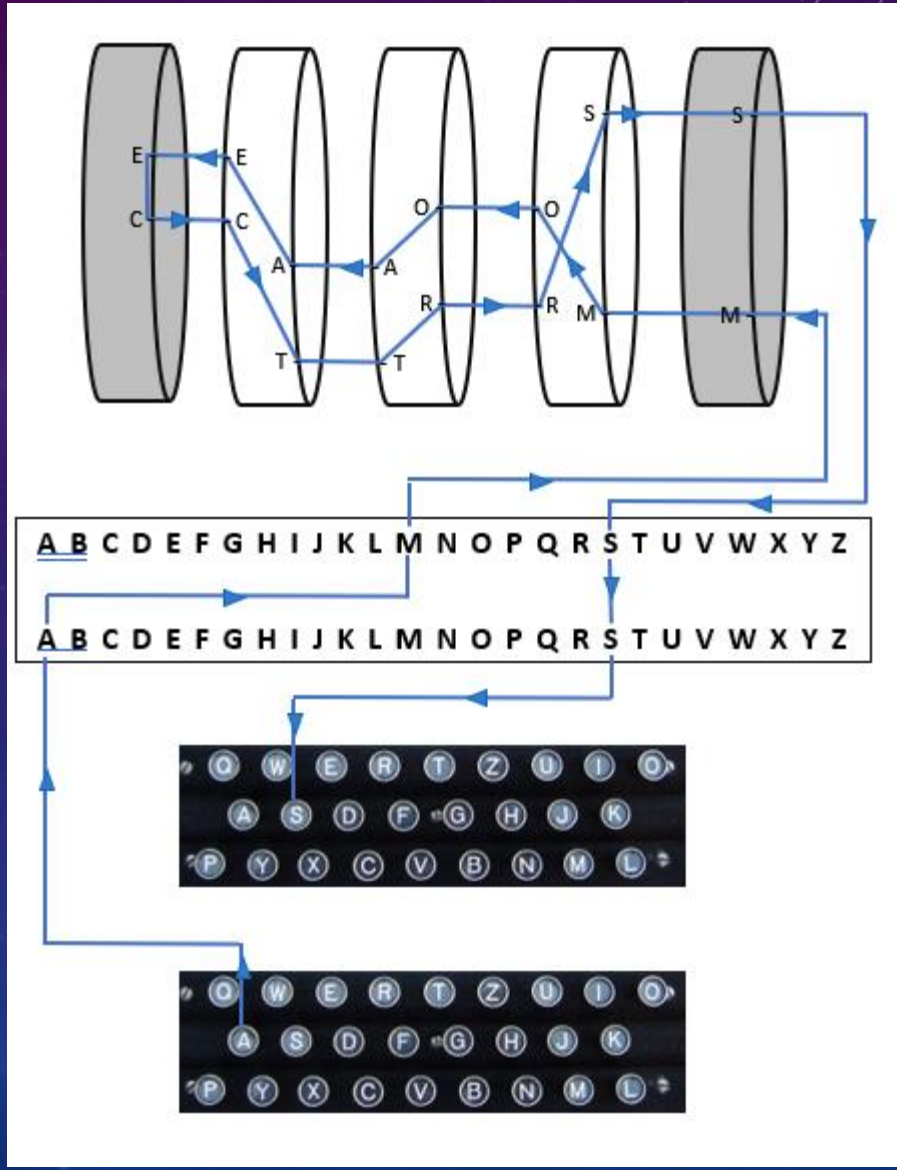
2. Comment permet-elle de coder et de décoder des messages ?

La rotation des rotors

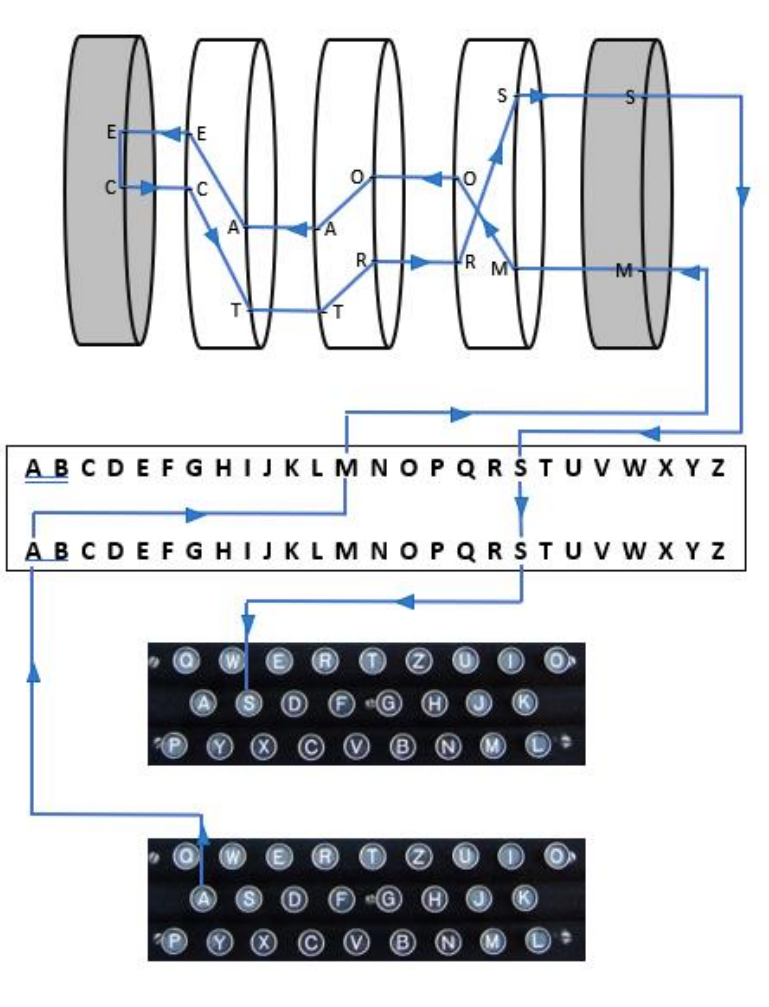
À chaque appui sur une touche, le rotor de droite, appelé « rotor rapide », tourne d'un cran.

Ainsi, si la deuxième lettre tapée est la même que la première, elle ne sera pourtant pas codée par la même lettre que la première fois.

Dans certaines configurations particulières, les deux autres rotors tournaient eux aussi, grâce à un astucieux système d'encoches.



2. Comment permet-elle de coder et de décoder des messages ?



Le rotor « rapide » a tourné d'un cran.

À chaque appui sur une touche,

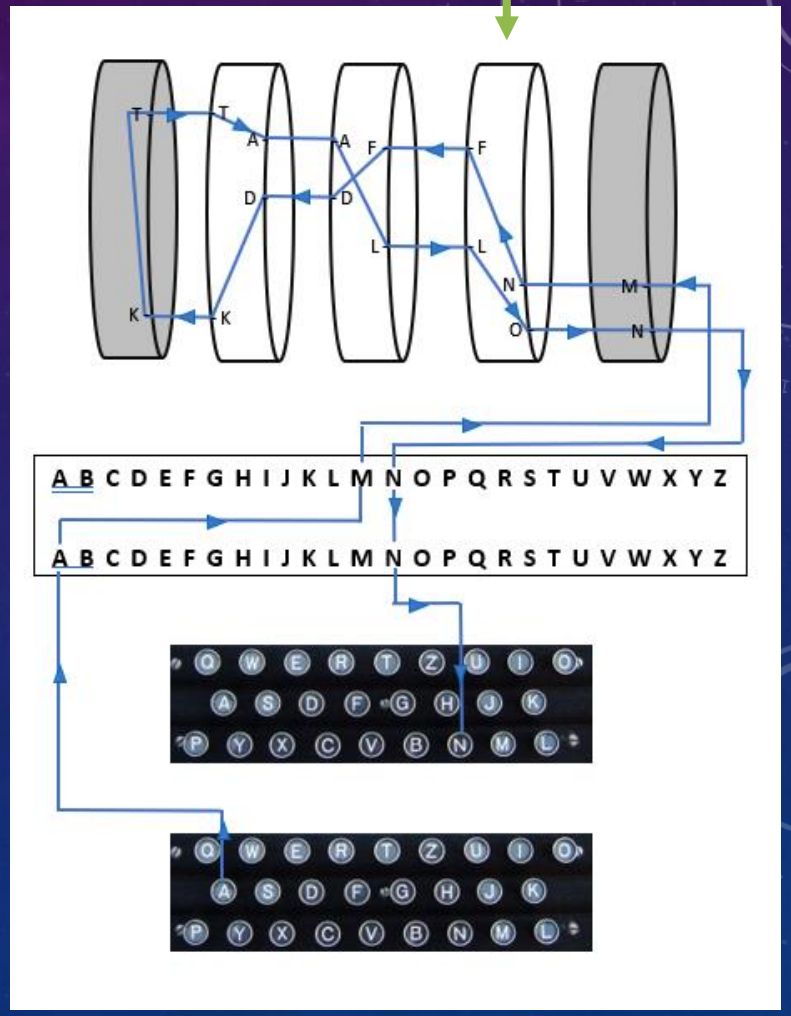
le rotor de droite, appelé « rotor rapide », tourne d'un cran :

ainsi, si la 2ème lettre tapée est la même que la 1ère, elle ne sera pourtant pas codée par la même lettre que la 1ère fois.

Dans certaines configurations particulières, les deux autres rotors tournaient eux aussi, grâce à un astucieux système d'encoches.

La rotation des rotors

Le rotor « rapide » a tourné d'un cran.



3. Pourquoi les messages cryptés par ces machines étaient-ils si difficiles à déchiffrer ?

Chaque jour, tous les opérateurs d'un même corps d'armée, réglaient leur machine Enigma à l'aide d'un carnet très… très… secret !
Ce sont ces réglages quotidiens qui multipliaient presque à l'infini les combinaisons possibles. Nous allons vous expliquer comment… Et compter pour vous toutes ces combinaisons !

3. Pourquoi les messages cryptés par ces machines étaient-ils si difficiles à déchiffrer ?

On peut lire…

- Le jour du mois (« Tag »)
- Le choix des rotors et leurs positions dans la machine (« Walzenlage »)
- Le réglage de l'orientation initiale des rotors (« Ringstellung »)
- Les couples de lettres échangées par le tableau de fiches (« Steckerverbindungen »)

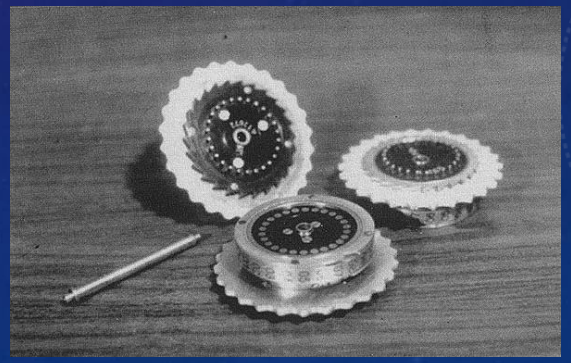
PAGE D'UN CARNET DE CODES MENSUELS

Tag	Walzenlage			Ringstellung	Steckerverbindungen
31	IV	V	I	24 17 13	AQ CT DV EN FW GP IX JS KR LO
30	II	V	III	22 11 01	AL BS EU FR GM IO PY QZ TW VX
29	V	III	I	09 25 21	AK BU CI DE GO HM LT OZ RV WY
28	IV	II	III	12 07 08	BW DZ EY FR GT HJ IV LS NQ OX
27	IV	III	II	02 03 15	AC BP EZ FX GV HK LT MR NO QW
26	IV	II	III	19 20 20	AK BX CQ DO EI HS LR PW TZ VY
25	III	I	IV	06 13 13	BX CE DR FG HV IN LY MT OS UW
24	V	I	IV	19 02 06	AL CM DO EP HN IK SZ TU VY WX
23	II	V	I	10 04 12	BE CK DX FG HI MO NP QU RS VZ
22	I	III	V	01 13 01	AS BC DU GY IK LZ MO NW QX RV
21	IV	V	I	22 01 01	AF CX DR GN HV JL KZ OW QS UY
20	I	III	V	25 11 10	AP BE FG IZ KT LN OR QX UW VY
19	IV	I	V	02 04 15	CS DW EF IN JQ KT OX PZ RV UY
18	III	IV	I	08 09 21	AY BK CS GQ HR JP LO MN UZ WX
17	II	I	V	26 09 05	AX CJ DY EW GP HO IN MS QR UV
16	III	II	IV	24 15 14	CV DJ EK FW GP HS IZ MT QX RY
15	IV	I	V	13 03 16	AE BV FK GO IZ JT LR MX NP WY
14	I	III	V	08 14 02	AL CP DG FY HK JW MS NV QZ TU
13	IV	I	V	10 06 10	CZ DR EF GT HS IU LO MV PQ WX
12	I	II	V	14 21 06	AK BD CL EJ FI GX OR PZ QT VW
11	III	V	I	03 03 18	AP BI CS DU EZ FN HQ KO LM TW
10	I	V	II	22 24 26	AN BQ DJ EI GU HV KR LP MS XY
09	III	II	I	05 19 12	BR CT DS EU HW IZ JP LX NO QV
08	I	IV	V	02 01 06	AG CV DH EL FR IT JY MW QU SZ
07	I	IV	V	06 15 10	AX BP CQ DR FI GY HJ KU MV SZ
06	V	II	IV	20 01 05	BG CW DI EF JV LZ NY OR PS UX
05	IV	V	I	07 06 12	AI CU DT GS HK JQ LM NV PZ XY
04	V	IV	III	25 15 09	AF EQ GW HX IO JN KZ MS PR UY
03	V	III	I	06 05 10	AV DR EX FY HI JM KZ LQ NS PU
02	II	V	I	23 09 21	AP CX DV EU FT GS HI KM LZ NR
01	III	I	II	16 12 02	AO BY CM DH GI KV LQ RW SZ TU

3. Pourquoi les messages cryptés par ces machines étaient-ils si difficiles à déchiffrer ?

PREMIÈRE ÉTAPE :

Choisir et agencer trois rotors parmi cinq dans la machine :



3. Pourquoi les messages cryptés par ces machines étaient-ils si difficiles à déchiffrer ?

PREMIÈRE ÉTAPE :

Choisir et agencer trois rotors parmi cinq dans la machine :

5 x 4 x 3 = 60 choix

5 choix pour le rotor de gauche
X
4 choix pour le rotor central
X
3 choix pour le rotor de droite

Ordre du jour :

Vert – Bleu - Rose

3. Pourquoi les messages cryptés par ces machines étaient-ils si difficiles à déchiffrer ?

DEUXIÈME ÉTAPE :

Choisir l'orientation initiale de chacun des trois rotors :

$$26 \times 26 \times 26 = 26^3 = 17\,576 \text{ choix}$$

1…2…3…4…5…6…7…8…9…10…11…12…13…14…15…16…17…18…19…20…21…22…23…24…25…26

3. Pourquoi les messages cryptés par ces machines étaient-ils si difficiles à déchiffrer ?

TROISIÈME ÉTAPE : LE TABLEAU DE FICHES

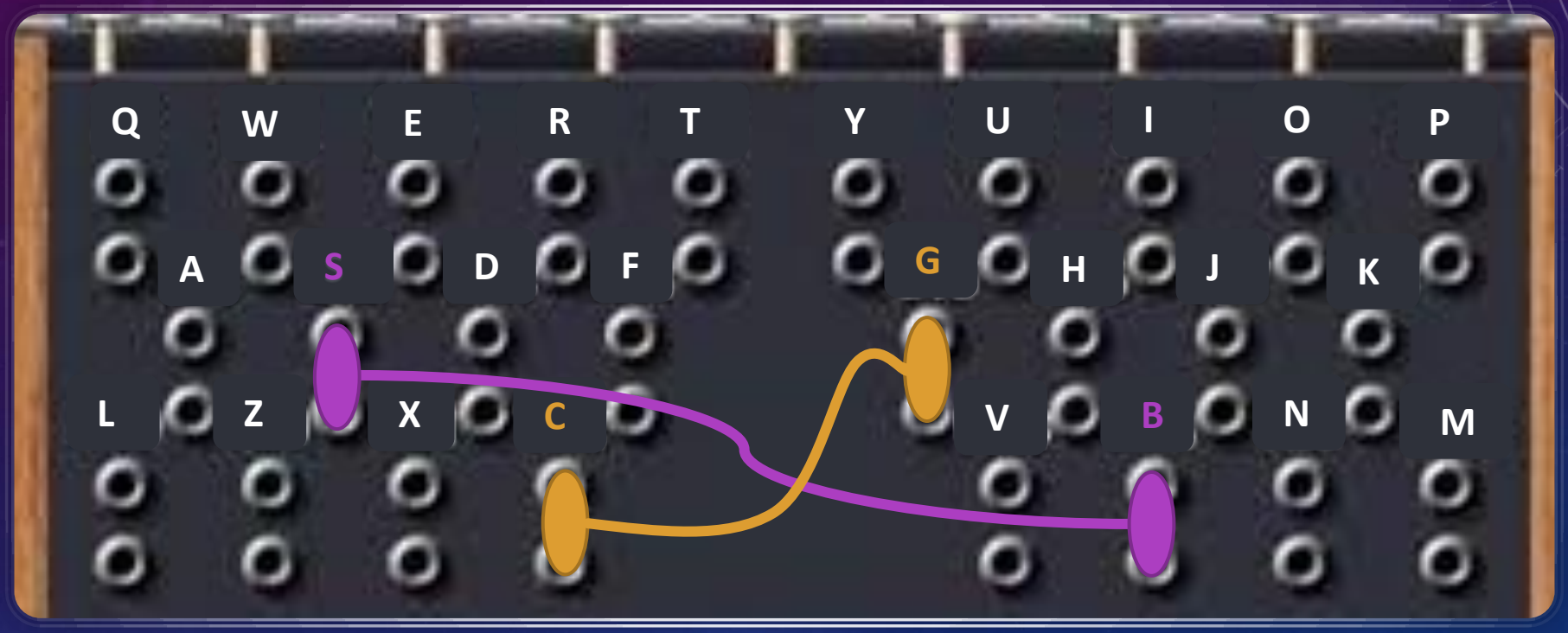
Pour bien comprendre cette étape,

on peut la décomposer comme ceci :

- ❑ D'abord, on choisit 20 lettres parmi 26.
- ❑ Ensuite, on fait 10 paires de lettres avec ces 20 lettres.
- ❑ Nous avons appris pour ce travail, la notation : $n\,! = n \times (n-1) \times (n-2) \ldots \times 4 \times 3 \times 2 \times 1$.

3. Pourquoi les messages cryptés par ces machines étaient-ils si difficiles à déchiffrer ?

TROISIÈME ÉTAPE : LE TABLEAU DE FICHES



Sur cet exemple, deux câbles sont branchés : le câble violet permute les lettres S et B ; le câble orange permute C et G.

Dans notre étude, 10 câbles sont branchés :c'était le cas des machines Enigma de la marine allemande.

3. Pourquoi les messages cryptés par ces machines étaient-ils si difficiles à déchiffrer ?

TROISIÈME ÉTAPE : LE TABLEAU DE FICHES

- **Choisir 20 lettres parmi 26 :**

Il y a : 26 x 25 x 24 x 23 x 22 x 21 x … x 9 x 8 x 7 = $\frac{26\,!}{6\,!}$

… façons d'écrire un mot de 20 lettres sans utiliser deux fois la même lettre.

Mais, ici, l'ordre des lettres n'a pas d'importance.

Ces 20 lettres peuvent être rangées de 20 x 19 x 18 x 17 x …x 3 x 2 x 1 = 20 ! façons différentes.

Ainsi, il y a… $\frac{26\,!}{20\,! \times 6\,!} = \mathbf{230\ 230}$**… façons de choisir les 20 lettres à connecter.**

3. Pourquoi les messages cryptés par ces machines étaient-ils si difficiles à déchiffrer ?

TROISIÈME ÉTAPE : LE TABLEAU DE FICHES

- **Faire 10 paires avec les 20 lettres choisies :**

  Si on choisit successivement 10 couples de lettres…

  Par exemple : (A,E) – (G,B) – (R,O) - … (S,Z)

  Il y a $(20 \times 19) \times (18 \times 17) \times (16 \times 15) \ldots \times (4 \times 3) \times (2 \times 1)$ choix possibles.

Mais on compte plusieurs fois les mêmes permutations. En effet, *chaque paire a été comptée 2 fois puisque :*

(A,E) – (G,B) – (R,O) - … (S,Z) revient à la même chose que (E,A) – (G,B) …

On est donc ramené à… $\frac{(20\times19)}{2} \times \frac{(18\times17)}{2} \times \frac{(16\times15)}{2} \ldots \times \frac{(4\times3)}{2} \times \frac{(2\times1)}{2} = \frac{20!}{2^{10}}$ choix possibles.

Mais, même ainsi, on compte encore trop de combinaisons car on peut les écrire dans n'importe quel ordre :

(A,E) – (G,B) – (R,O) - … revient à la même chose que (G,B)-(R,O)-…(A,E)

On peut organiser ces 10 paires de… 10 x 9 x 8 x … x 3 x 2 x 1 = 10 ! façons différentes.

Il faut donc encore diviser par 10 ! Le total précédent.

3. Pourquoi les messages cryptés par ces machines étaient-ils si difficiles à déchiffrer ?

TROISIÈME ÉTAPE : LE TABLEAU DE FICHES

- Faire 10 paires avec les 20 lettres choisies :

Finalement, il y a….

$$\frac{\frac{(20\times 19)}{2}\times\frac{(18\times 17)}{2}\times\frac{(16\times 15)}{2}\ldots\times\frac{(4\times 3)}{2}\times\frac{(2\times 1)}{2}}{10\times 9\times 8\times 7\times 6\times 5\times 4\times 3\times 2\times 1}$$

$$=\frac{20!}{2^{10}\times 10!}=654\ 729\ 075$$ façons de faire 10 paires avec les 20 lettres choisies.

3. Pourquoi les messages cryptés par ces machines étaient-ils si difficiles à déchiffrer ?

FINALEMENT, LE NOMBRE DE CLÉS POSSIBLES EST ÉGAL À…

$$5 \times 4 \times 3 \times 26^3 \times \frac{26\,!}{6\,! \times 20\,!} \times \frac{20\,!}{10\,! \times 2^{10}}$$

$$\approx 1{,}59 \times 10^{20}$$

3. Pourquoi les messages cryptés par ces machines étaient-ils si difficiles à déchiffrer ?

Imaginons qu'on arrive à tester une combinaison en une seconde…

- Trouver la bonne combinaison des rotors prendrait… $5 \times 4 \times 3$ = 60 s : 1 minute.
- Trouver l'orientation correcte des 3 rotors prendrait… 26^3 = 17 576 s : 4 h 56 min 05 s.
- Trouver les 20 lettres branchées sur le tableau de fiches prendrait… $\frac{26\,!}{6\,! \times 20\,!}$ = 230 230 s, soit 2 j 15 h 57 min 10 s.
- Retrouver les 10 couples de lettres échangées avec ces 20 lettres prendrait… $\frac{20\,!}{10\,! \times 2^{10}}$ s = 654 729 075 s, soit : 20 ans 277 j 21 h 11 min 15 s !

Finalement, pour tester les $5 \times 4 \times 3 \times 26^3 \times \frac{26\,!}{6\,! \times 20\,!} \times \frac{20\,!}{10\,! \times 2^{10}} \approx 1{,}59 \times 10^{20}$ clés quotidiennes possibles,

en prenant 1 s par clé, il faudrait… **environ 5 mille milliards d'années !!!**

4. Enigma et les outils numériques d'aujourd'hui.

Nous avons deux objectifs :

- Nous voulons concevoir un programme qui simule au mieux la machine Enigma. Pour cela, nous travaillons en langage Python que nous sommes plusieurs à connaître.

- L'aboutissement de notre projet sera enfin d'aménager notre programme pour qu'il soit compatible avec le système électronique que nous a préparé un de nos professeurs de sciences de l'ingénieur, M. Séguret, que nous tenons à remercier ici.

- Pour l'instant, notre programme simule la rotation du rotor rapide (celui de droite) ; nous sommes en train de finaliser la modélisation du tableau de fiches et de l'orientation initiale des rotors.

- Nous travaillons aussi à la simulation des (rares) rotations des deux autres rotors.

4. Enigma et les outils numériques d'aujourd'hui.

Dans notre programme (Voir Annexe 2) :

- Les lettres seront remplacées après leur saisie par un nombre allant de 0 (le « A ») à 25 (le « Z »).

- Nous avons choisi de modéliser les rotors par des listes de nombres qui correspondent aux décalages provoqués par les liaisons électriques avec un réglage initial par défaut immuable :

  Par exemple, si un rotor donné transforme le « A » (lettre n°0) en D (lettre n°3) :
  on définira ce décalage par le nombre 3 dans la liste associée à ce rotor.

- Nous avons choisi de faire une liste correspondant aux passages aller et une liste correspondant aux passages retour dans chaque rotor pour faciliter la modélisation des rotations.

- Pour préparer la version électronique de notre machine, notre professeur de SI nous a demandé de ne pas modifier trop souvent ces listes. Nous avons donc simulé la rotation des rotors par des appels décalés dans ces listes ce qui nous a valu quelques « prises de tête »… Mais on a quand même réussi…

Voici le résultat…

4. Enigma et les outils numériques d'aujourd'hui.

Dans notre programme (Voir Annexe 2) :

- Les lettres seront remplacées après leur saisie par un nombre allant de 0 (le « A ») à 25 (le « Z »).

- Nous avons choisi de modéliser les rotors par des listes de nombres qui correspondent aux décalages provoqués par les liaisons électriques avec un réglage initial par défaut immuable :

  Par exemple, si un rotor donné transforme le « A » (lettre n°0) en D (lettre n°3) :
  on définira ce décalage par le nombre 3 dans la liste associée à ce rotor.

- Nous avons choisi de faire une liste correspondant aux passages aller et une liste correspondant aux passages retour dans chaque rotor pour faciliter la modélisation des rotations.

- Pour préparer la version électronique de notre machine, notre professeur de SI nous a demandé de ne pas modifier trop souvent ces listes. Nous avons donc simulé la rotation des rotors par des appels décalés dans ces listes ce qui nous a valu quelques « prises de tête »… Mais on a quand même réussi…

Voici le résultat…

4. Enigma et les outils numériques d'aujourd'hui

PyScripter - C:\Users\caro2\Desktop\vidéos CGénial\21.02.18 enigma.py

Fichier Edition Rechercher Affichage Projet Exécuter Outils Aide Rechercher:

Nouveau Fichier...

```
A1=[4, 9, 10, 2, 7, 1, -3, 9, 13, 16, 3, 8, 2, 9, 10, -8, 7, 3, 0, -4, -20, -13, -21, -6, -22, -16]
A2=[0, 8, 1, 7, 14, 3, 11, 13, 15, -8, 1, -4, 10, 6, -2, -13, 0, -11, 7, -6, -5, 3, -17, -2, -10, -21]
A3=[1, 2, 3, 4, 5, 6, -4, 8, 9, 10, 13, 10, 13, 0, 10, -11, -8, 5, -12, -19, -10, -9, -2, -5, -8, -11]
R1=[20, 21, 22, 3, -4, -2, -1, 8, 13, 16, -9, -7, -10, -3, -2, 4, -9, 6, 0, -8, -3, -13, -9, -7, -10, -16]
R2=[0, 8, 13, -1, 21, 17, 11, 4, -3, -8, -7, -1, 2, 6, 10, 5, 0, -11, -14, -6, -13, 2, -10, -15, -3, -7]
R3=[19, -1, 4, -2, 11, -3, 12, -4, 8, -5, 10, -6, 9, 0, 11, -8, 8, -9, 5, -10, 2, -10, -5, -13, -10, -13]
reflecteurB=[24, 16, 18, 4, 12, 13, 5, -4, 7, 14, 3, -5, 2, -3, -2, -7, -12, -16, -13, 6, -18, 1, -1, -14, -24, -6]
```

21.02.18 enigma.py

Console Python

```
*** Console de processus distant Réinitialisée ***
>>>
```

Python input

Saisissez la phrase à décoder sans espace: QWCSDWU

OK Cancel

Nous avons saisi le message : QWCSDWU (pour la version électronique, cette saisie se fera lettre à lettre).

SUR LES TRACES D'ENIGMA

PyScripter - C:\Users\caro2\Desktop\vidéos CGénial\21.02.18 enigma.py

Fichier Edition Rechercher Affichage Projet Exécuter Outils Aide Rechercher:

Nouveau Fichier...

```
A1=[4, 9, 10, 2, 7, 1, -3, 9, 13, 16, 3, 8, 2, 9, 10, -8, 7, 3, 0, -4, -20, -13, -21, -6, -22, -16]
A2=[0, 8, 1, 7, 14, 3, 11, 13, 15, -8, 1, -4, 10, 6, -2, -13, 0, -11, 7, -6, -5, 3, -17, -2, -10, -21]
A3=[1, 2, 3, 4, 5, 6, -4, 8, 9, 10, 13, 10, 13, 0, 10, -11, -8, 5, -12, -19, -10, -9, -2, -5, -8, -11]
R1=[20, 21, 22, 3, -4, -2, -1, 8, 13, 16, -9, -7, -10, -3, -2, 4, -9, 6, 0, -8, -3, -13, -9, -7, -10, -16]
R2=[0, 8, 13, -1, 21, 17, 11, 4, -3, -8, -7, -1, 2, 6, 10, 5, 0, -11, -14, -6, -13, 2, -10, -15, -3, -7]
R3=[19, -1, 4, -2, 11, -3, 12, -4, 8, -5, 10, -6, 9, 0, 11, -8, 8, -9, 5, -10, 2, -10, -5, -13, -10, -13]
reflecteurB=[24, 16, 18, 4, 12, 13, 5, -4, 7, 14, 3, -5, 2, -3, -2, -7, -12, -16, -13, 6, -18, 1, -1, -14, -24, -6]
```

21.02.18 enigma.py

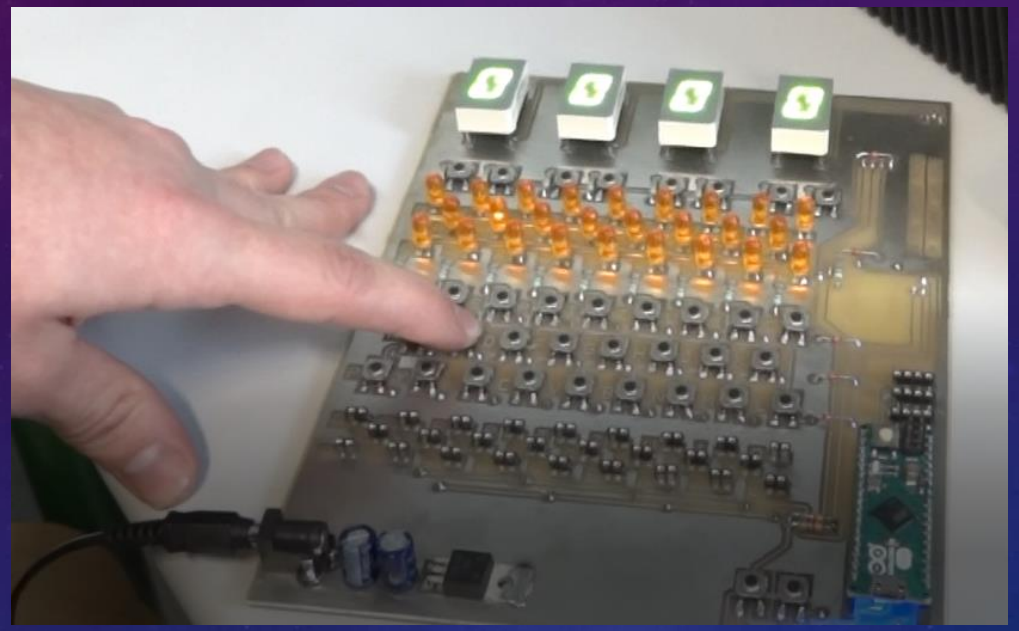
Console Python

```
*** Console de processus distant Réinitialisée ***
>>>
C
G
E
N
I
A
L
>>>
```

CGENIAL !!!

4. Enigma et les outils numériques d’aujourd’hui.

Et voici notre version électronique, préparée par M. Séguret :



Mathys appuie ici sur une touche pour faire une première démonstration : pour l’instant, lorsqu’on appuie sur une lettre, c’est la lampe de la même lettre qui s’allume. Voilà notre dernier défi à relever : finir d’adapter notre programme pour l’implanter dans la carte qu’on peut apercevoir en bas à droite de l’image.

CONCLUSION ET PERSPECTIVES

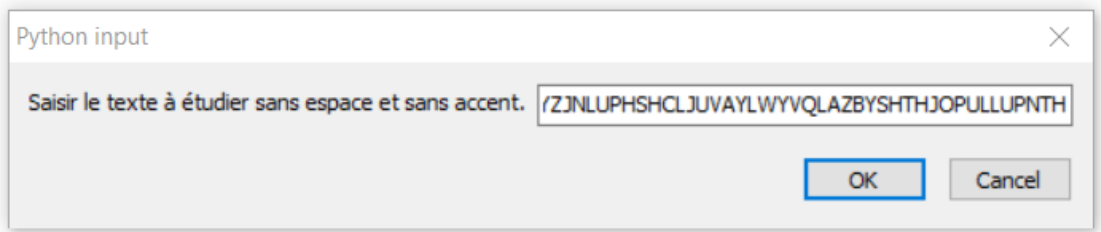
- Notre programme reste à compléter mais nous avons déjà bien avancé pour simuler tous les éléments des véritables machines. Nous avons utilisé des documents scientifiques et historiques pour avoir les décalages authentiques. Ce travail qui mélange des faits et des objets qui font partie de notre histoire et nos études d'aujourd'hui est vraiment très intéressant et les séances hebdomadaires nous paraissent bien trop courtes !!!

- Alan Turing, qui aura largement contribué à « casser » Enigma, est aussi considéré comme un des pères de l'ordinateur moderne. Aujourd'hui, la machine Enigma ne résisterait pas à une « attaque » par la « force brute » de nos ordinateurs, c'est-à-dire en essayant toutes les combinaisons possibles.

Côté dénombrement, nous allons chercher à estimer le temps qu'il fallait dans les années 1930 et le temps qu'il faudrait à un ordinateur d'aujourd'hui pour « casser » Enigma par la « force brute », c'est-à-dire en essayant toutes ces combinaisons.

ANNEXE 1 : NOTRE PROGRAMME POUR « CASSER » LE CODE CÉSAR PAR ANALYSE DES FRÉQUENCES

```python
#ce programme détermine le décalage probable grâce une fonction renvoyant le rang de la lettre apparaissant le plus grand nombre de fois dans un texte donné puis décode le texte donné
texte1=input("Saisir le texte à étudier sans espace et sans accent.")
texte1=texte1.upper()
print(texte1)
#partie 1 : analyse des fréquences et calcul de la clé
liste=[]
for i in range(25):
    lettre=chr(65+i)
    occurence=texte1.count(lettre)
    liste.append(occurence)
M=max(liste)
nb=liste.count(M)
if nb!=1:
    print("Attention, ce texte est peut-être trop court pour que cette analyse soit valable ou il ne s'agit peut-être pas d'un texte codé à l'aide d'un code César.")
else:
    rang=liste.index(M)
    cle=rang+65-ord("E")
    print("Le décalage identifié par analyse des fréquences est de ",cle)
#partie 2 : décodage avec la clé obtenue plus haut
texteclair=""
L=len(texte1)
print("La lettre ",chr(ord("E")+cle)," code vraisemblablement la lettre E.")
for i in range(L):
    rang=(ord(texte1[i])-ord("A"))
    rangclair=(rang-cle)%26
    lettreclair=chr(rangclair+ord("A"))
    texteclair=texteclair+lettreclair
print("Voici le texte décodé : ")
print(texteclair)
if nb!=1 or len(texte1)<20:
    reponse=input("Voulez-vous tenter la méthode brute ? Si oui, taper 1")
    if reponse=="1"or reponse=="oui":
        for c in range(24):
            texteclair=""
            for i in range(L):
                rang=(ord(texte1[i])-ord("A"))
                rangclair=(rang-c)%26
                lettreclair=chr(rangclair+ord("A"))
                texteclair=texteclair+lettreclair
            print("Avec une clé égale à ",c+1," le message deviendrait : ",texteclair)
    else:
        print("Comme vous voulez.")
```




```
Console Python
*** Console de processus distant Réinitialisée ***
>>>
JLAALHUULLUVBZWHYAPJWVUZHBJVUJVBYZJNLUPHSHCLJUVAYLWYVQLAZBYSHTHJOPULLUPNTH
Le décalage identifié par analyse des fréquences est de  7
La lettre  L  code vraisemblablement la lettre E.
Voici le texte décodé :
CETTEANNEENOUSPARTICPONSAUCONCOURSCGENIALAVECNOTREPROJETSURLAMACHINEENIGMA
>>>
```

ANNEXE 2 : LE SCRIPT DE NOTRE PROGRAMME

```python
A1=[4, 9, 10, 2, 7, 1, -3, 9, 13, 16, 3, 8, 2, 9, 10, -8, 7, 3, 0, -4, -20, -13, -21, -6, -22, -16]
A2=[0, 8, 1, 7, 14, 3, 11, 13, 15, -8, 1, -4, 10, 6, -2, -13, 0, -11, 7, -6, -5, 3, -17, -2, -10, -21]
A3=[1, 2, 3, 4, 5, 6, -4, 8, 9, 10, 13, 10, 13, 0, 10, -11, -8, 5, -12, -19, -10, -9, -2, -5, -8, -11]
R1=[20, 21, 22, 3, -4, -2, -1, 8, 13, 16, -9, -7, -10, -3, -2, 4, -9, 6, 0, -8, -3, -13, -9, -7, -10, -16]
R2=[0, 8, 13, -1, 21, 17, 11, 4, -3, -8, -7, -1, 2, 6, 10, 5, 0, -11, -14, -6, -13, 2, -10, -15, -3, -7]
R3=[19, -1, 4, -2, 11, -3, 12, -4, 8, -5, 10, -6, 9, 0, 11, -8, 8, -9, 5, -10, 2, -10, -5, -13, -10, -13]
reflecteurB=[24, 16, 18, 4, 12, 13, 5, -4, 7, 14, 3, -5, 2, -3, -2, -7, -12, -16, -13, 6, -18, 1, -1, -14, -24, -6]

# On a créé des listes Ai (pour passage Aller dans le rotor i et des liste Ri pour passage Retour dans rotor i.
# Ces listes correspondent au décalage numérique* associé à chaque liaison électrique sur un rotor (ou le réflecteur)
# lorsqu'il est en position initiale ; c'est-à-dire, quand les lettres "A" associées aux 1ers éléments de nos listes
# sont alignés sur les barres grises de la machine papier (correspondant aux fenêtres des véritables machines).
# On ne prend pas en compte les réglages (quotidiens) des bagues... pour l'instant...

# * Exemple : Pour le rotor 1 (sens aller), le A devient E, le B devient K, le C devient M, etc.
# En version "lettres", ça nous donnerait le début de liste : ["E","K","M",...]
# Ce qui donnera en "version numérique" :  [4 ; 9 ; 10 ; ...]
# car pour aller de A à E, on "avance" de 4 lettres, de B à K, on "avance" de 9 lettres, etc.

#Dans un premier temps, on fonctionnera sans proposer le choix des rotors et de leur agencement.

rotD=A3
rotM=A2
rotG=A1
ref=reflecteurB
rotRG=R1
rotRM=R2
rotRD=R3

phrase = input("Saisissez la phrase à décoder sans espace: ")
tours = 0
index = 0
NombreLettres = 0

while index != len(phrase):
    lettre = phrase[index]
    rang=ord(lettre)
    NombreLettres=NombreLettres+1
    N=NombreLettres%26
    lettre=lettre.upper()
    rang=ord(lettre)-65
    rang=(rang+N)%26
    # On utilise la ligne ci-dessus pour prendre en compte la rotation
    # du rotor "rapide" (celui de droite). Pour l'instant, on écrit un
    # premier programme SANS aucune rotation de rotor.
```

```python
    #Incrémentation index & tours
    index = (index+1)
    if index == 26:
        index = 0
        tours += 1
        #ensuite, on incrémente tours2

    # Passage aller dans le rotor de droite

    rangD=( rang+A3[ rang ] )%26
    #print(chr(65+rangD))

    # Passage aller dans le rotor du milieu
    rangM=( rangD - N + A2[ (rangD - N - tours)%26 ] )%26
    #print(chr(65+rangM))

    # Passage aller dans le rotor de gauche
    rangG=( rangM + A1[ rangM ])%26
    #print(chr(rangG+65))

    # Au 23 janvier, ça marche jusqu'ici...
    # On a affiché la lettre obtenue après passage aller dans les trois rotors.

    # Passage dans le réflecteur
    rangRef=( rangG + ref[ rangG ])%26
    #print(chr(rangRef+65))

    # Passage retour dans le rotor de gauche
    rangRG=( rangRef + R1[ rangRef ])%26
    #print(chr(rangRG+65))

    # Passage retour dans le rotor central
    rangRM=( rangRG + index + R2[ rangRG ])%26
    #print(chr(rangRM+65))

    # The last... But not least... Passage retour dans le rotor de droite
    rangcode=( rangRM - index + R3[ rangRM ])%26
    print(chr(rangcode+65))

    #Et c'est reparti pour un tour !...

#phrase = input("Phrase")
```

Sources (images et documentation) :

- http://wiki.franklinheath.co.uk
- http://users.telenet.be/d.rijmenants/en/enigmamenu.htm
- Le site Wikipedia
- Le site « Interstices » : https://interstices.info/jcms/int_70884/turing-a-l-assaut-d-enigma
- Site de M. Singh : https://simonsingh.net/cryptography/enigma-photos/