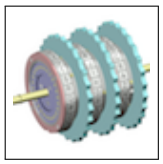

La machine Enigma

Fiche Élève

TS

La machine Enigma a été utilisée par l'armée allemande durant la seconde guerre mondiale. Elle servait à chiffrer et déchiffrer les messages secrets. Son intérêt par rapport aux systèmes de chiffrement antérieurs était le très grand nombre de clés possibles pour le chiffrement et le déchiffrement ce qui, pensait-on, la rendait particulièrement sûre. C'était sans compter sur le talent de mathématiciens polonais et anglais pour percer ses mystères.
Une machine Enigma était constituée de plusieurs dispositifs dont des rotors (de 4 à 6 selon les modèles). On se propose d'étudier le fonctionnement d'un de ces rotors.
On peut se représenter un rotor comme composé de deux anneaux. L'un, extérieur, fixe et comportant les 26 lettres de l'alphabet noté comme une chaîne de caractères :

`Abc="ABCDEFGHIJKLMNOPQRSTUVWXYZ"`,

l'autre mobile, tournant dans le premier et portant lui aussi les 26 lettres de l'alphabet, rangées dans un *ordre défini lors de la fabrication de la machine*, par exemple `SWILVAOUZHGERXBYQDCNFPTMJK` (la lettre qui suit `K` est `S`, les lettres sont lues dans le sens horaire). Chacune de ces lettres est positionnée en face d'une lettre de l'anneau fixe. Pour décrire entièrement la position du rotor au début du codage d'un message, il suffit d'indiquer quelle lettre de l'anneau mobile se trouve en face du A de l'anneau fixe. S'il s'agit de la lettre `G`, on notera

`Cle="GERXBYQDCNFPTMJKSWILVAOUZH"`

Figure 0.1.

qui s'appelle la *clé du chiffrement initialisée à* `G`. On représentera commodément la position initiale du rotor par

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
G	E	R	X	B	Y	Q	D	C	N	F	P	T	M	J	K	S	W	I	L	V	A	O	U	Z	H

Figure 0.2.

Comment Enigma crypte un message ?

Un message est une suite de lettres majuscules ; il ne comporte pas de blancs, pas de signes de ponctuation, pas d'accents. Soit à coder le mot `"BELLE"`.

Étape n°1 : L'opérateur d'Enigma tape d'abord `B` sur le clavier, ce qui produit en sortie un `E`, lettre située en face de `B`, cf. **Figure 0.2.** Aussitôt après, un mécanisme fait tourner l'anneau mobile d'un cran dans le sens rétrograde. Le `E` de l'anneau mobile vient se mettre sous le `A` de l'anneau fixe, le `R` sous le `B`, . . . , *etc*, ce qui donne une nouvelle disposition des anneaux :

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
E	R	X	B	Y	Q	D	C	N	F	P	T	M	J	K	S	W	I	L	V	A	O	U	Z	H	G

Figure 0.3.

`G` est passé de la première position à la dernière.

Étape n°2 : L'opérateur tape `E` sur le clavier (deuxième lettre du message à coder), ce qui produit `Y`, lettre située en face de `E` et l'anneau mobile tourne de nouveau d'un cran dans le sens rétrograde, ce qui donne la disposition :

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
R	X	B	Y	Q	D	C	N	F	P	T	M	J	K	S	W	I	L	V	A	O	U	Z	H	G	E

Figure 0.4.

Étapes suivantes : On continue de même. Il y aura 5 étapes puisque `"BELLE"` a 5 lettres.

1 - Terminer à la main le chiffrement de `"BELLE"`.

Ce codage à l'aide d'un seul rotor est difficile à percer car tout se passe comme si on changeait la grille de correspondance entre les lettres à chaque nouvelle lettre à coder. Ainsi, les 2 `L` de `"BELLE"` sont codés `M` et `J`.

Décryptage : Pour le décodage d'un message codé, on procèdera dans l'ordre inverse. Cela suppose que l'on dispose d'une machine identique à la machine qui a servi au codage et que l'on connaît la première lettre de la clé.

Cryptage et décryptage avec *scilab* :

2 - `X` désignant une liste de nombres notée, pour fixer les idées $X = (x_1, x_2, \ldots, x_{n-1}, x_n)$, décrire les listes `Y` et `Z` produits par les commandes suivantes :

```
Y=[X(2:$),X(1)];
Z=[X($),X(1:$-1)];
```

Utilisation des codes ASCII : *scilab* traite un message comme une chaîne de caractères, c'est à dire ici une liste de lettres majuscules encadrée par des guillemets. Dans le codage ASCII, les lettres majuscules de l'alphabet sont codées, dans l'ordre lexicographique, de 65 à 90. La commande scilab `asciimat("A")` renvoie `65`, `asciimat("TURNER")` renvoie `[84 85 82 78 69 82]` tandis que `asciimat([66 67])` renvoie `BC` (et non `"BC"`) (taper la commande `help asciimat` dans la console « scilab »). `asciimat` permet donc d'échanger un message et une liste d'entiers qui le caractérise. Le problème du chiffrement/déchiffrement devient un problème numérique.

3 - Cryptage : Dans le script téléchargeable `EbaucheC` ci-dessous visant à définir une fonction `Cryptage`, on a saisi l'alphabet et la clé donnée à la **Figure 0.1.** comme des chaînes de caractères, notées respectivement `Abc` et `Cle`. Le message à chiffrer, `Mo`, sera l'entrée de la fonction que l'on veut définir. On commence par remplacer ces trois chaînes de caractères par des listes d'entiers `AbcNu`, `CleNu` et `MoNu` (`Nu` pour "numérisé à l'aide de la commande `asciimat`"). On programmera le calcul (c'est l'objet de la question) de la forme numérisée `McNu` du message produit en cryptant `Mo`. La dernière commande du script met enfin `McNu` sous sa forme alphabétique `Mc`, qui sera la sortie de la fonction considérée. Au début du chiffrement, `McNu` est vide. `n` est le nombre de lettres de `Mo`.

```
function Mc=Cryptage(Mo)// Mo : message original (à coder).
    Abc="ABCDEFGHIJKLMNOPQRSTUVWXYZ";// Alphabet.
    AbcNu=asciimat(Abc);// Alphabet numérisé.
    Cle="GERXBYQDCNFPTMJKSWILVAOUZH";// Clé.
    CleNu=asciimat(Cle);// Clé numérisée.
    McNu=[];// Mc : message crypté.
    MoNu=asciimat(Mo);// Message original numérisé. Le problème est
    //maintenant entièrement numérisé.
    n=size(MoNu,"c");// Nombre de lettres du message à crypter.
    // à compléter //
    Mc=asciimat(McNu);
endfunction
```

3.a - Expliciter la valeur de `AbcNu`.

3.b - Compléter le script précédent par une boucle `for` qui calcule, pour `i` variant de 1 à `n` la $i^{\text{ème}}$ lettre de `Mc` et fait tourner d'un cran l'anneau mobile.

Indication : Utiliser la commande `find` (voir la documentation en ligne).

3.c - Application : coder `"RENDEZVOUSAPARIS"`.

Décryptage :

4.a - Compléter de même le script téléchargeable `EbaucheD` ci-dessous de manière à définir une fonction *scilab* notée `Decryptage` dont l'entrée est un message `Mc` à décrypter et dont la sortie sera ce message en clair.

```
function Md=Decryptage(Mc)// Mc message crypté (à déchiffrer) ;
    //Md : message déchiffré.
    Abc="ABCDEFGHIJKLMNOPQRSTUVWXYZ";// Alphabet.
    AbcNu=asciimat(Abc);// Alphabet numérisé.
    Cle="GERXBYQDCNFPTMJKSWILVAOUZH";// Clé.
    CleNu=asciimat(Cle);// Clé numérisée.
    McNu=asciimat(Mc);// Message original numérisé. Le problème est
    //maintenant entièrement numérisé.
    MdNu=[];
    n=size(McNu,"c");// Nombre de lettres de Mc.
    // àcompléter //
    Md=asciimat(MdNu);
endfunction
```

4.b - Application : décoder le message crypté `"NHVDAYJITSBLYHJU"` (on suppose toujours que la clé des machines utilisées est la clé de la **Figure 0.1.**).

5 - Extra Un message codé : `"NXBSGDUBBIBVUDEX"` a été reçu, mais la lettre qui initialise la clé n'a pas été envoyée (c'était `G` précédemment). Retrouver la clé et décoder le message.