R1.07 TP 4

Chiffrement par blocs

15 janvier 2026 – B. Colombel

Un compte-rendu de ce TP (réponses aux exercices et aux questions) est à rédiger sur le notebook *Jupyter* et votre compte rendu doit être déposé sous le nom

`prenom.nom-TP4.ipynb`

dans le dossier relatif à ce TP sur Ametice (R1.07, TP4).

De quoi s'agit-il ?

Le chiffrement que nous allons étudier a été publié par Lester S. Hill en 1929. C'est un chiffrement *polygraphique*, c'est-à-dire qu'on ne (dé)chiffre pas les lettres les unes après les autres, mais par paquets en utilisant des matrices à coefficients dans $\mathbb{Z}/26\mathbb{Z}$.

Cela a ouvert la voie à la cryptographie algébrique et de nos jours, c'est l'Aes (*Advanced Ecrypting Standart*), né en 2000, qui assure la majorité de la sécurité informatique.

On étudie ici la version *bigraphique*, c'est-à-dire que l'on groupe les lettres deux par deux, mais on peut envisager des paquets plus grands.

Dans la suite, nous noterons $\mathbb{Z}_n$ l'ensemble $\mathbb{Z}/n\mathbb{Z}$.

1 Méthode de chiffrement

Pour coder un message selon ce procédé, on suit les étapes :

- ➢ **Étape 1 :** On regroupe les lettres du message deux par deux ;
- ➢ **Étape 2 :** On remplace chaque lettre par un nombre, comme indiqué par le tableau suivant :

Lettre	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
Nombre	0	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

Exemple 1. On veut par exemple coder le message « j'adore les maths ».
On le décompose en ja-do-re-le-sm-at-hs puis on remplace par :

$$(9;0) - (3;14) - (17;4) - (11;4) - (18;12) - (0;19) - (7;18)$$

Remarque. Si le nombre de lettres du message avait été impair, on aurait ajouté une lettre arbitraire à la fin.

- ➢ **Étape 3 :** Chaque couple de nombres $(x;y)$ de la liste précédente est transformé en un nouveau couple $(x';y')$ de nombres entiers compris entre 0 et 25, à l'aide d'une matrice $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ via la relation

$$A\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x' \\ y' \end{pmatrix} \quad \text{c'est à dire} \quad \begin{cases} ax + by \equiv x' \pmod{26} \\ cx + dy \equiv y' \pmod{26} \end{cases}$$

Vocabulaire. La matrice A est appelé la *clé* du chiffrement.

- ➢ **Étape 4 :** Ces deux nombres x' et y' sont transformés en lettres en utilisant le tableau de correspondance.

2 Codage et décodage

Les première et dernière étapes consistent à faire correspondre lettres et nombres suivant le tableau donné page 1.
L'histoire de l'informatique nous donne une manière naturelle de procéder grâce au code ASCII :

— L'American Standard Code for Information Interchange (Code américain normalisé pour l'échange d'information), plus connu sous l'acronyme ASCII ([aski :]) est une norme de codage de caractères en informatique ancienne et connue pour son influence incontournable sur les codages de caractères qui lui ont succédé. Elle était la plus largement compatible pour ce qui est des caractères latins non accentués.
— ASCII contient les caractères nécessaires pour écrire en anglais. Les caractères accentués sont fournis par d'autres normes, comme aujourd'hui, l'Unicode. Le jeu de caractères codés ASCII est le principal système qui a permis l'échange de textes en anglais à un niveau mondial, limitant ainsi l'usage des langues locales au travers d'extensions régionales.

Sources : Wikipédia

La table 1 page 4 représente une partie de la table ascii.

Par exemple, : vaut 58, @ vaut 64, A vaut 65 , a vaut 97, b vaut 98, etc.
Les fonctions *sagemath* suivantes permettent de réaliser ces opérations de codage et décodage :

```
def lettreToEntier(lettre, alphabet = "abcdefghijklmnopqrstuvwxyz"):
    return alphabet.find(lettre)
def entierToLettre(a, alphabet = "abcdefghijklmnopqrstuvwxyz"):
    return alphabet[a]

print("Lettre -> entier : w ->", lettreToEntier('w'))
print("Entier -> Lettre : 22 ->", entierToLettre(22))
```

3 Chiffrement

Dans cette partie, on utilise comme clé de chiffrement la matrice : $A = \begin{pmatrix} 9 & 5 \\ 4 & 7 \end{pmatrix}$.

Exercice 1.

1. Par la méthode décrite ci-dessus, chiffrer les deux premières lettres du message « je ne suis pas un numero je suis un homme libre » [1].
2. Écrire avec *sagemath* une fonction `chiffre_hill(mess_clair)` qui prend en entrée la chaîne de caractère `mess_clair` [2] et qui retourne le message chiffré. Ne pas oublier d'enlever les espaces dans la chaîne de caractère.
3. Tester la fonction `chiffre_hill(chaine)` avec le message « je ne suis pas un numéro je suis un homme libre »

4 Déchiffrement

Exercice 2.

1. Montrer que la matrice A est inversible dans $\mathcal{M}_2(\mathbb{R})$ et déterminer son inverse.
   Écrire cette matrice sous la forme $\frac{1}{43} \times B$ où B est une matrice à coefficients entiers.

Or, la matrice A est à coefficient dans $\mathbb{Z}_{26}$, on doit déterminer son inverse dans $\mathcal{M}_2(\mathbb{Z}_{26})$. On ne peut donc pas utiliser le calcul $\frac{1}{43}B$ car la division par 43 est illicite dans $\mathbb{Z}_{26}$. Il faut ainsi déterminer l'inverse de 43 dans $\mathbb{Z}_{26}$ c'est-à-dire un élément $u \in \mathbb{Z}_{26}$ tel que

$$43 \times u \equiv u \times 43 \equiv 1 \pmod{26}$$

Si u existe alors on aura l'égalité :

$$\boxed{A^{-1} = u \times B}$$

1. *source* : Le prisonnier
2. On se limite dans cette partie aux lettres minuscules sans accents et sans espaces

Exercice 3.

1. Écrire en *sagemath* une fonction `inverse(n)` qui prend en entrée un entier `n` et qui retourne un entier `u` avec $0 \leqslant u \leqslant 25$ tel que
$$n \times u \equiv 1 \pmod{26}$$
s'il existe et qui retourne -1 sinon.
2. En utilisant la fonction précédente répondre aux questions :
— 43 est-il inversible dans $\mathbb{Z}_{26}$?
— Si oui, quel est son inverse [3] ?
3. Quelle est l'inverse de A dans $\mathbb{Z}_{26}$?
4. Écrire avec *sagemath* une fonction `dechiffre_hill(mess_chiffre)` qui permet de retrouver le message en clair à partir du texte chiffré.
5. Déchiffrer le message (on a ajouté une lettre à la fin du message en clair) :

pubilupymdlhmqeqzlzluesobiotxduejruyxxhrmwkofehefemlxdso
biwkwpqmguwaoaoegubrkimlxdsobiwkpagahuhpmkstpkkykr

5 Pour aller plus loin

On décide de travailler avec 27 symboles en ajoutant le caractère @ pour les espaces et de regrouper les lettres par paquets de 3 à la place de 2.

On travaille donc dans $\mathbb{Z}_{27}$ avec la matrice carrée d'ordre 3 à coefficients dans $\mathbb{Z}_{27}$:

$$A = \begin{pmatrix} 1 & 22 & 25 \\ 0 & 25 & 1 \\ 25 & 3 & 1 \end{pmatrix}$$

Exercice 4. Adapter la méthode vu précédemment et les fonctions écrites pour :

1. écrire une fonction de chiffrement et chiffrer le message
« je@ne@suis@pas@un@numero@je@suis@un@homme@libre » ;
2. déterminer la matrice de déchiffrement ;
3. écrire une fonction de déchiffrement.

3. Vérifier sur la table 2 page 4

Caractères		!	"	#	$	%	&	’	(	)	*	+	,	-	.
Code	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46

Caractères	/	0	1	2	3	4	5	6	7	8	9	:	;	<	=
Code	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61

Caractères	>	?	@	A	B	C	D	E	F	G	H	I	J	K	L
Code	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76

Caractères	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	[
Code	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91

Caractères	\	]	^	_	‘	a	b	c	d	e	f	g	h	i	j
Code	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106

Caractères	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y
Code	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121

Caractères	z	{	|	}	~
Code	122	123	124	125	126

Table 1 – Éléménts de la table ASCII

×	**0**	**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**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	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
2	0	2	4	6	8	10	12	14	16	18	20	22	24	0	2	4	6	8	10	12	14	16	18	20	22	24
3	0	3	6	9	12	15	18	21	24	1	4	7	10	13	16	19	22	25	2	5	8	11	14	17	20	23
4	0	4	8	12	16	20	24	2	6	10	14	18	22	0	4	8	12	16	20	24	2	6	10	14	18	22
5	0	5	10	15	20	25	4	9	14	19	24	3	8	13	18	23	2	7	12	17	22	1	6	11	16	21
6	0	6	12	18	24	4	10	16	22	2	8	14	20	0	6	12	18	24	4	10	16	22	2	8	14	20
7	0	7	14	21	2	9	16	23	4	11	18	25	6	13	20	1	8	15	22	3	10	17	24	5	12	19
8	0	8	16	24	6	14	22	4	12	20	2	10	18	0	8	16	24	6	14	22	4	12	20	2	10	18
9	0	9	18	1	10	19	2	11	20	3	12	21	4	13	22	5	14	23	6	15	24	7	16	25	8	17
10	0	10	20	4	14	24	8	18	2	12	22	6	16	0	10	20	4	14	24	8	18	2	12	22	6	16
11	0	11	22	7	18	3	14	25	10	21	6	17	2	13	24	9	20	5	16	1	12	23	8	19	4	15
12	0	12	24	10	22	8	20	6	18	4	16	2	14	0	12	24	10	22	8	20	6	18	4	16	2	14
13	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13
14	0	14	2	16	4	18	6	20	8	22	10	24	12	0	14	2	16	4	18	6	20	8	22	10	24	12
15	0	15	4	19	8	23	12	1	16	5	20	9	24	13	2	17	6	21	10	25	14	3	18	7	22	11
16	0	16	6	22	12	2	18	8	24	14	4	20	10	0	16	6	22	12	2	18	8	24	14	4	20	10
17	0	17	8	25	16	7	24	15	6	23	14	5	22	13	4	21	12	3	20	11	2	19	10	1	18	9
18	0	18	10	2	20	12	4	22	14	6	24	16	8	0	18	10	2	20	12	4	22	14	6	24	16	8
19	0	19	12	5	24	17	10	3	22	15	8	1	20	13	6	25	18	11	4	23	16	9	2	21	14	7
20	0	20	14	8	2	22	16	10	4	24	18	12	6	0	20	14	8	2	22	16	10	4	24	18	12	6
21	0	21	16	11	6	1	22	17	12	7	2	23	18	13	8	3	24	19	14	9	4	25	20	15	10	5
22	0	22	18	14	10	6	2	24	20	16	12	8	4	0	22	18	14	10	6	2	24	20	16	12	8	4
23	0	23	20	17	14	11	8	5	2	25	22	19	16	13	10	7	4	1	24	21	18	15	12	9	6	3
24	0	24	22	20	18	16	14	12	10	8	6	4	2	0	24	22	20	18	16	14	12	10	8	6	4	2
25	0	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

Table 2 – Table de multiplication modulo 26