Nom: Prénom:

L3 informatique. Sécurité. Partie J.-F. COUCHOT.

Seule une fiche manuscrite recto-verso de format A4 est autorisée. Tout moyen de communication est interdit. Toutes les réponses doivent être justifiées. Sans justification, une réponse est considérée comme fausse.

1 Analyse d'une fonction étrange

La figure 2 (dernière page) donne le code python d'une fonction étrange. L'objectif de cet exercice est de deviner ce qu'elle réalise (et de l'analyser).

1. Quel est l'objectif du code présenté aux lignes 1–3 ? Quel(s) package faut-il installer pour que ces lignes d'import soit interprétées sans erreur ?

2. Quel est le type de l'objet `m` passé en paramètre de cette fonction ? Sur combien de bits est usuellement encodé chaque élément constituant cet objet ?

3. Quel est l'objectif des lignes 7 et 8 ? On rappelle ci dessous la documentation de la fonction `pad`. Comment est implanté ce pading, à votre avis ?

```
"""Crypto.Util.Padding.pad(data_to_pad, block_size)
  Apply standard padding.
  Parameters:
    data_to_pad (byte string) - The data that needs to be padded.
    block_size (integer) - The block boundary to use for padding.
          The output length is guaranteed to be a multiple of block_size.
  Returns:
    the original data with the appropriate padding added at the end.
  Return type : byte string"""
```

4. Expliquer ce qui est réalisé entre les lignes 14 et 18. On pourra s'aider d'un schéma.

5. Expliquer ce qui est fait à la ligne 20.

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
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

TABLE 1 – Table de correspondance en lettres et entiers

6. Expliquer ce qui est proposé aux lignes 23–34. Détailler au maximum.

7. Quel est la finalité d'une telle fonction ? Quelles propriétées semblent établies ? Pour lesquelles n'a-t-on aucune garantie ?

2 Chiffre de bigraphique de Hill

Le chiffre de Hill ne code pas les lettres les unes après les autres, mais par paquets. Ici c'est la version bigraphique qui est étudiée, c'est-à-dire que les lettres sont groupées deux par deux.

Pour chiffrer un message, on commence par remplacer chaque lettre de celui-ci par un entier entre 0 et 25 (dont la correspondance est donnée à la table 1) et obtenir la séquence d'entiers $x = [x_1, x_2, x_3, \ldots, x_n]$. Tous les calculs se font alors modulo 26. On remplace celle-ci par une séquence $x' = [(x_1, x_2), (x_3, x_4), \ldots]$ des couples de x. Chaque couple (x_i, x_{i+1}) où i est impair va être chiffré en un couple (y_i, y_{i+1}) en calculant le produit matriciel $(y_i, y_{i+1}) = (x_i, x_{i+1}) \times \begin{pmatrix} 3 & 4 \\ 2 & 9 \end{pmatrix}$.

1. Chiffrer les deux premières lettres de votre nom de famille.

2. Pour déchiffrer, c'est chaque couple (y_i, y_{i+1}) qui est traité l'un après l'autre. Montrer qu'en calculant $(x'_i, x'_{i+1}) = (y_i, y_{i+1}) \times 11 \times \times \begin{pmatrix} 9 & 22 \\ 24 & 3 \end{pmatrix}$, on retrouve (x_i, x_{i+1}).

3. Que dire alors des matrices $\begin{pmatrix} 3 & 4 \\ 2 & 9 \end{pmatrix}$ et $11 \times \begin{pmatrix} 9 & 22 \\ 24 & 3 \end{pmatrix}$? En déduire que le chiffre est symétrique et

$$\begin{pmatrix} 1&0&0&0&0&0&0&0&0&0&0&0&0&1&1 \\ 0&1&0&0&0&0&0&0&0&0&0&0&1&0&1 \\ 0&0&1&0&0&0&0&0&0&0&0&0&1&1&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0&1&1&1 \\ 0&0&0&0&1&0&0&0&0&0&0&1&0&0&1 \\ 0&0&0&0&0&1&0&0&0&0&0&1&0&1&0 \\ 0&0&0&0&0&0&1&0&0&0&0&1&0&1&1 \\ 0&0&0&0&0&0&0&1&0&0&0&1&1&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&1&1&0&1 \\ 0&0&0&0&0&0&0&0&0&1&0&1&1&1&0 \\ 0&0&0&0&0&0&0&0&0&0&1&1&1&1&1 \end{pmatrix} \quad \begin{pmatrix} 0&0&1&1 \\ 0&1&0&1 \\ 0&1&1&0 \\ 0&1&1&1 \\ 1&0&0&1 \\ 1&0&1&0 \\ 1&0&1&1 \\ 1&1&0&0 \\ 1&1&0&1 \\ 1&1&1&0 \\ 1&1&1&1 \\ 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \end{pmatrix}$$

FIGURE 1 – Deux matrices pour un code de Hamming

donner la clef.

4. Supposions que vous obteniez un message chiffré par ce chiffre sans en connaître la clef. Par force brute, combien d'essais seraient suffisants pour le décrypter ? Est-ce réaliste sur un ordinateur récent ?

3 Codes correcteurs d'erreurs

On considère les deux matrices données à la figure 1.

1. Montrer que ces deux matrices pourraient être utilisées telles quelles dans un code de Hamming systématique. Utiliser les noms usuels de ces matrices dans ce code.

2. Dans ce code de Hamming, peut-on coder les mots $x_1 = (1, 0, 1, 0)$? $x_2 = (1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1)$? $x_3 = (1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1)$? A quoi correspondent alors les autres propositions ?

3. Trouver le mot de code associé au mot que l'on peut coder parmi x_1, x_2 et x_3.

4. Trouver le syndrome du mot de code parmi x_1, x_2 et x_3. En déduire le mot qui avait été codé.

```python
from Crypto.Util.Padding import pad
from Crypto.Cipher import AES
from Crypto.Util.strxor import strxor
import binascii

def aStrangeFunc(m):
  size = 16
  M = pad(m,size)
  nbblock =  int(len(M)/size)
  init = b''.join([(2**8-1).to_bytes(1, byteorder='big') for _ in range(size)])
  #init : séquence de 16 octets, chacun d'eux valant 1111 1111
  cipher = init

  for i in range(nbblock):
    message = M[i*size:(i+1)*size]
    aes_ecb_e = AES.new(cipher, AES.MODE_ECB)
    cipherp = aes_ecb_e.encrypt(message)
    cipher = strxor (cipher,cipherp)

  t = binascii.hexlify(cipher)
  return t

for j in range(10):
  print(aStrangeFunc(str(j).encode('utf-8')))

b'd095c66e1d070733564e95b8ed2ce77e'
b'ec7e8566972d24cfa75d68c92957e4e8'
b'8f71baa4de7c4c04d14ab9e394b1509d'
b'ed7bcd08816cf9cc845199afdd51ed08'
b'457ec6da58d2058cb8bfeced4c74e451'
b'6753931ee0e459e974cbaea7b50c1ddd'
b'1b0fcabf7ca30c5c28543b77056caea9'
b'fe1a53097e27111d537e947977735160'
b'85090819547366f418e02e183e3e6943'
b'd145b2c50954fa933eb9301ad9a5a7f6'
```

FIGURE 2 – Code d'une fonction étrange