# CyberSécurité
# Introduction Crypto (1)
# TD1 - 2023

***TD individuel***
Attention de bien lire le TD : 1-Objectif et 2-Réalisation pour le prochain TD

***« Dis-moi et j'oublie. Montre-moi et je me souviens. Implique-moi et je comprends. »***
*proverbe chinois*

- **Objectif du TD -**

**L'ensemble des exercices du TD doit être fait sans utiliser de librairie en Python3 et sur NoteBook (jupyter-lab)**

- **Installation Spyder et Notebook (jupyter-lab)**
  - **https://docs.spyder-ide.org/current/installation.html**
  - **https://jupyterlab.readthedocs.io/en/stable/getting_started/installation.html**
  - Installation de Jupyter-Lab : https://jupyter.org/install
  - **Documentation de Jupiter-lab : https://docs.jupyter.org/en/latest/**
- **.**

- **La remise du TD se fera obligatoirement déposer sur DVO**
  - Votre .zip contiendra obligatoirement :
    - L'ensemble des programmes Python dans jupyter-lab au format « ipynb »
      - Votre ipynb devra contenir obligatoirement:
        - Votre nom et prénom
        - N° du TD
        - N°de l'exercice
        - Question et réponse à l'exercice
        - Code Python, une fonction par cellule avec commentaires
        - **...**
    - **Toute non remise ou manquement aux consignes sera pénalisé**

## 1. Préparation et Réalisation Obligatoire pour le prochain TD

- Avec l'explosion des technologies de l'information et des communications, toutes les activités humaines peuvent être étroitement liées à la collecte et au traitement de données personnelles.
- Aujourd'hui, les frontières entre le monde réel et le monde virtuel ou numérique sont de plus en plus étroites.
- La numérisation de la société ne cesse de s'accélérer et transforme profondément nos manières de vivre, d'envisager nos relations, de nous comporter ou de travailler.
- Cette évolution comporte certes de nombreux avantages mais aussi porteuse de risques.

1. Pour le prochain TD vous devez obligatoirement faire une **présentation individuelle**
   - Cette présentation **individuelle** avec au minimum **un Slide par Item** et devra être remise sur Moodle avant la présentation du prochain TD au format (.PPTX) :
   - **Donner votre avis en répondant aux questions suivantes :**
     - Qu'est-ce qu'une donnée personnelle ? (donner des exemples)
     - Que sont les données sensibles ? (donner des exemples)
     - Qu'est-ce que les métadonnées ? (donner des exemples)
     - Qu'est-ce que le cyberespace ?
     - Pourquoi les données personnelles passionnent tant ? .
     - Internet, est-il un cimetière de données personnelles ?
     - Si je n'utilise pas ou très peu Internet, suis-je concerné par la sécurité des données personnelles ? Pourquoi ?
     - Le « sentiment de sécurité dans le cloud». .
     - « Pas grave si on prend mes données, je n'ai rien à cacher ».
     - Quels sont vos interrogations sur le monde digital ?

- **Présentation du code César**

- **Chiffre ou Code :**
  - il existe pourtant une différence subtile entre les deux termes. Un chiffre opère plutôt sur les unités élémentaires qui composent le message : lettres, syllabes, ou autres. En revanche, un code s'applique aux mots ou aux phrases du message, et sur leur signification.
  - Par exemple dans le code des émoticônes, ☺ signifie je suis content.
  - L'ensemble des codes est, en général, réuni dans un livre de codes, comme par exemple les dictionnaires chiffrés.
  -

- **Le chiffrement César**
  - **Jules César utilisait cette technique pour certaines de ses correspondances, notamment militaires, comme avec Cicéron (décalage de 3)**
- **Comment chiffrer avec le chiffre de César ? (Principe de chiffrement)**
  **Le code de César est en fait une simple substitution mono-alphabétique, c'est-à-dire qu'une lettre est remplacée par une seule autre. Le code de César a la particularité d'être basé sur un simple décalage circulaire de l'alphabet.**
- **On veut chiffrer « DCODEX » avec un décalage de 3**
  - **-Pour chiffrer D, on prend l'alphabet et on regarde trois lettres plus loin :**
    - **il y a G. Donc D se code G.**
  - **-Pour chiffrer X, on boucle l'alphabet : après X, il y a Y, après Y, il y a Z et après Z,**
    - **il y a A. Donc X se code A.**
  - **-Plus mathématique : si on note A = 0, B = 1, ..., Z = 25, qu'on ajoute une constante et conserve le résultat modulo 26 (longueur de l'alphabet). On obtient le texte codé.**
    - **D = 4, 4+3 = 7 et 7 = G, donc on code D par G**
  - **X = 23, 23+3 = 26 et 26 mod 26 = 0, 0 = A, donc on code X par A, etc.**

- **DCODEX devient GFRGHA**

- **Comment déchiffrer le chiffrement de César ? (Principe de déchiffrement)**
  - **Le déchiffrement remplace une lettre par une autre par décalage dans l'alphabet. On veut déchiffrer GFRGHA avec un décalage de 3.**
    - **Pour déchiffrer G, on prend l'alphabet et on regarde trois lettres avant : il y a D.**
    - **Pour déchiffrer A, on boucle l'alphabet : avant A : Z, avant Z : Y et avant Y, il y a X.**
  - **Une autre façon de déchiffrer, plus mathématique : si on note A = 0, B = 1, ..., Z = 25, qu'on soustrait une constante et conserve le résultat modulo 26 (longueur de l'alphabet). On obtient le texte codé.**
    - **G = 7, 7-3 = 4 et 4 = D, donc on déchiffrer G par D**
    - **A = 0, 0-3 = -3 et -3 mod 26 = 23, 23 = X, donc on décode A par X, etc.**
    - **GFRGHA devient DCODEX .**

- **Les différents Noms des types de chiffre du code césar**
  - **Le chiffre d'Auguste**
    - **Le chiffre d'Auguste est le nom donné pour un décalage de 1.**

Dans le code Hélène, L vaut N (Hélène LN), le décalage est de 2
Dans le code Œufs Pourris, E vaut I (Œufs pourris, E pour I), le décalage est de 4
Dans le code KO, K vaut O (KO : Knock Out), le décalage est de 4
Dans le code Pété, P vaut T (Pété PT), le décalage est de 4
Dans le code Hervé, R vaut V (Hervé RV), le décalage est de 4
Dans le code Jeux Olympiques, J vaut O (JO : jeux olympiques), le décalage est de 5
Dans le code Agé, A vaut G (Agé AG), le décalage est de 6
**Dans le code Eiffel, F vaut L (Eiffel FL), le décalage est de 6**
Dans le code PV, P vaut V (Procès Verbal PV), le décalage est de 6
Dans le code WC, W vaut C (WC : Water Closet), le décalage est de 6
Dans le code Avocat, A vaut K (Avocat), le décalage est de 10
Dans le code Acheté, H vaut T (Acheté HT), le décalage est de 12
Dans le code J'y vais, J vaut V (J'y vais JV), le décalage est de 12
Dans le code ROT13, l'alphabet devient réversible, le décalage est de 13
Très utilisé chez les Hackers
Dans le code Hergé/RG, R vaut G (Hergé Renseignements Généraux RG),
le décalage est de 15
Dans le code Déesse, D vaut S (Déesse DS), le décalage est de 15
Dans le code Happé, A vaut P (happé), le décalage est de 15
**Dans le code Cassé, K vaut C (Cassé KC), le décalage est de 18**
Dans le code Meuh, M vaut E (Meuuuh ME), le décalage est de 18
**Dans le code A voté, A vaut T (a voté), le décalage est de 19**

- **Messages Codés pour les exercices**

1. **Cesar_FR.message 1 : « WP NZOLRP PDE FY LCE »**
2. **Cesar_FR.message 2 : « HA YKZWCA AOP QJ WNP »**

3. **Cesar_FR.message 3** : "hwfvsfl ds kwugfvw ymwjjw egfvasdw dwk sewjausafk wehdgqwjwfl vwk afvawfk fsnsbgk hgmj ujqhlwj vwk ewkksywk. u'wkl d'mf vwk jsjwk ugvwk vw d'zaklgajw s f'sngaj bsesak wlw tjakw. d'aehwfwljstadalw vm ugvw fsnsbg nawfl wf hsjlaumdawj vm xsal imw uwllw dsfymw f's smumf dawf snwu mfw imwdugfimw dsfymw wmjghwwffw gm skaslaimw."

4. **Cesar_FR.message 4** : « Ghpdlq, ghv o'dxeh, d o'khxuh rx eodqfklw od fdpsdjqh, Mh sduwludl. Yrlv-wx, mh vdlv txh wx p'dwwhqgv. M'ludl sdu od iruhw, m'ludl sdu od prqwdjqh. Mh qh sxlv ghphxuhu orlq gh wrl soxv orqjwhpsv. Mh pdufkhudl ohv bhxa ilahv vxu phv shqvhhv, Vdqv ulhq yrlu dx ghkruv, vdqv hqwhqguh dxfxq euxlw, Vhxo, lqfrqqx, oh grv frxueh, ohv pdlqv furlvhhv, Wulvwh, hw oh mrxu srxu prl vhud frpph od qxlw. Mh qh uhjdughudl ql o'ru gx vrlu txl wrpeh, Ql ohv yrlohv dx orlq ghvfhqgdqw yhuv Kduiohxu, Hw txdqg m'duulyhudl, mh phwwudl vxu wd wrpeh Xq erxtxhw gh krxa yhuw hw gh euxbhuh hq iohxu. Ylfwru Kxjr - Ohv Frqwhpsodwlrqv »

5. **Message5** :

   « Yjaxt ti Yjath dci xcktcit jc bdntc st rdbbjcxrpixdc htrgti. a'tmetsxitjg trgxi at itmit axvct epg axvct spch jc gtripcvat st 6 axvcth ti 3 rdadccth. tchjxit xa at gtrdext rdadcct epg rdadcct. Yjath, etcspci at rdcigdat st gthtpjm, p djqaxt hp rparjatiit. xa tckdxt at bthhpvt hjxkpci: QFIUONAIZTSERQ ?EU. fjtaat sdxi tigt ap gtedcht st Yjaxt? »

## 1 Exercice – Code César

**L'ensemble des exercices devront être programmés en python sur Jupyter-lab**

### 1.1 Exo-1-1 : Dans cet exercice seulement les Lettres seront codées

- **Écrire une fonction Python « def cesar_Code_mot(….) » qui permettra de chiffrer un message en code césar en déterminant la clé du message par exemple à « 11 » le mot étant « cybersecurite » .**
- **Écrire une fonction Python « def cesar_Decode_mot(….) » qui permettra de déchiffrer le message en code césar de l'exercice précédemment en déterminant la clé du message par exemple à « 11 »**

### 1.2 Exo-1-2 : Dans cet exercice seulement les Lettres seront codées

- **Écrire une fonction Python qui permettra de déterminer la clé des messages (messages 1 et 2) codés par le chiffrement de César.et déchiffrera automatiquement de type Brut Force les messages 1 et 2 .**
- **Concernant les messages 1 et 2, que remarquez-vous ? Et pourquoi ?**

1.3 **Exo-1-3 : Dans cet exercice les Lettres majuscules et minuscules sont codés. La ponctuation n'est pas codée**

- **Écrire une fonction Python qui permettra de déchiffrer les messages interceptés 3 et 4 en tenant compte**
- **Écrire une fonction Python qui permettra de re-chiffrer les messages 3 et 4**
  - **vous devez obtenir le mème résultat**

**1.4 Exo-1-3 :**

- **Déchiffrer le Message 5 en python**
- **Donner la réponse à la question posée**
- **Écrire une fonction Python qui permettra de déchiffrer facile la question**
- **Écrire une fonction Python qui permettra de répondre en sur-chiffrant à la question avec la même méthode.**

## 2 Présentation – Indice de Coïncidence

L'**indice de coïncidence** est une technique de cryptanalyse inventée par William F. Friedman en 1920 (publiée dans *The Index of Coincidence and its Applications in Cryptography*) et améliorée par son collaborateur Solomon Kullback (en).

L'indice permet de savoir si un texte a été chiffré avec un chiffre mono-alphabétique ou un chiffre poly-alphabétique en étudiant la probabilité de répétition des lettres du message chiffré. Il donne également une indication sur la longueur de la clé probable.

L'indice se calcule avec la formule suivante :

$$IC = \sum_{q=A}^{q=Z} \frac{n_q(n_q-1)}{n(n-1)}$$

avec *n* le nombre de lettres total du message, $n_A$ le nombre de *A*, $n_B$ le nombre de *B*, etc.

En français, l'indice de coïncidence vaut environ 0,0746. Dans le cas de lettres uniformément distribuées (contenu aléatoire sans biais), l'indice se monte à 0,0385. L'indice ne varie pas si une substitution mono-alphabétique des lettres a été opérée au préalable. C'est-à-dire que si l'on remplace par exemple 'a' par 'z' et 'z' par 'a', l'indice ne changera pas.

| Langue | Indice |
|---|---|
| Suédois | 0,0644 |
| Serbe | 0,0643 |
| Russe | 0,0529 |
| Portugais | 0,0745 |
| Néerlandais | 0,0798 |
| Norvégien | 0,0694 |
| Malaysien | 0,0852 |
| Japonais | 0,0772 |
| Italien | 0,0738 |
| Hébreu | 0,0768 |
| Grec | 0,0691 |
| Français | 0,0778 |
| Finnois | 0,0737 |

| | |
|---|---|
| Esperanto | 0,069 |
| Espagnol | 0,077 |
| Danois | 0,0707 |
| Arabe | 0,0758 |
| Anglais | 0,0667 |
| Allemand | 0,0762 |

2.1 **Exo-2-1 : Écrire une fonction python qui calcule l'indice de coïncidence pour les messages 3,4,5, et 6 et qui retourne la langue du message. Vérifier si cela correspond au tableau**

## 3 Présentation de L'Analyse Fréquentielle

### ◦ L'Analyse Fréquentielle

- Lien : http://fr.wikipedia.org/wiki/Analyse_fr%C3%A9quentielle

L'**analyse fréquentielle**, ou *analyse d e fréquences*, est une méthode de cryptanalyse dont la description la plus ancienne est réalisée par Al-Kindi au IXe siècle. Elle consiste à examiner la fréquence des lettres employées dans un message chiffré. Cette méthode est fréquemment utilisée pour décoder des messages chiffrés par substitution, dont un exemple très simple est le chiffre de César.

L'analyse fréquentielle est basée sur le fait que, dans chaque langue, certaines lettres ou combinaisons de lettres apparaissent avec une certaine fréquence. Par exemple, en français, le *e* est la lettre la plus utilisée, suivie du *a* et du *s*. Inversement, le *w* est peu utilisé.

Ces informations permettent aux cryptanalystes de faire des hypothèses sur le texte clair, à condition que l'algorithme de chiffrement conserve la répartition des fréquences, ce qui est le cas pour des substitutions mono-alphabétiques et poly-alphabétiques.

Une deuxième condition nécessaire pour appliquer cette technique est la longueur du message à décrypter (Cryptogramme). En effet, un texte trop court ne reflète pas obligatoirement la répartition générale des fréquences des lettres. De plus, si la clé est de la même longueur que le message, il ne pourra y avoir de répétition de lettre et l'analyse fréquentielle deviendra impossible.

On peut constater que selon la langue, un texte comportera une répartition particulière des fréquences de lettres. Par exemple en français, les lettres les plus fréquentes, c'est-à-dire les lettres que l'on retrouve le plus souvent, sont le *E*, suivi du *A*, du *I* et du *S* ... On obtient ainsi la répartition de fréquences des lettres suivante (en %) :

| | A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Français** | 8.4 | 1.1 | 3.0 | 4.2 | 17.3 | 1.1 | 1.3 | 0.9 | 7.3 | 0.3 | 0.1 | 6.0 | 3.0 |
| **Anglais** | 8.2 | 1.5 | 2.8 | 4.3 | 12.7 | 2.2 | 2.0 | 6.1 | 7.0 | 0.2 | 0.8 | 4.0 | 2.4 |

| | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Français** | 7.1 | 5.3 | 3.0 | 1.0 | 6.6 | 8.1 | 7.1 | 5.7 | 1.3 | 0.1 | 0.4 | 0.3 | 0.1 |
| **Anglais** | 6.7 | 7.5 | 1.9 | 0.1 | 6.0 | 6.3 | 9.1 | 2.8 | 1.0 | 2.4 | 0.2 | 2.0 | 0.1 |

- **Ce qui donne l'ordre suivant pour la langue française par fréquence** :

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

# frequency taken from http://en.wikipedia.org/wiki/Letter_frequency

englishLetterFreq = {'E': 12.70, 'T': 9.06, 'A': 8.17, 'O': 7.51, 'I': 6.97, 'N': 6.75, 'S': 6.33, 'H': 6.09, 'R': 5.99, 'D': 4.25, 'L': 4.03, 'C': 2.78, 'U': 2.76, 'M': 2.41, 'W': 2.36, 'F': 2.23, 'G': 2.02, 'Y': 1.97, 'P': 1.93, 'B': 1.29, 'V': 0.98, 'K': 0.77, 'J': 0.15, 'X': 0.15, 'Q': 0.10, 'Z': 0.07}

ETAOIN = 'ETAOINSHRDLCUMWFGYPBVKJXQZ'
LETTERS = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'

On peut aussi analyser la fréquence dans un texte des digrammes, c'est-à-dire des groupes de deux lettres. Cela amènera des indices importants pour décrypter un texte chiffré car on sait que l'on ne pourra trouver des digrammes tels que *XK* ou *WX* dans le texte clair.

Les 20 digrammes les plus fréquents en français

| Bigrammes | ES | DE | LE | EN | RE | NT | ON | ER | TE | EL |
|---|---|---|---|---|---|---|---|---|---|---|
| Nombres | 3318 | 2409 | 2366 | 2121 | 1885 | 1694 | 1646 | 1514 | 1484 | 1382 |
| Bigrammes | AN | SE | ET | LA | AI | IT | ME | OU | EM | IE |
| Nombres | 1378 | 1377 | 1307 | 1270 | 1255 | 1243 | 1099 | 1086 | 1056 | 1030 |

Les 20 trigrammes les plus fréquents en français

| Trigrammes | ENT | LES | EDE | DES | QUE | AIT | LLE | SDE | ION | EME |
|---|---|---|---|---|---|---|---|---|---|---|
| Nombres | 900 | 801 | 630 | 609 | 607 | 542 | 509 | 508 | 477 | 472 |
| Trigrammes | ELA | RES | MEN | ESE | DEL | ANT | TIO | PAR | ESD | TDE |
| Nombres | 437 | 432 | 425 | 416 | 404 | 397 | 383 | 360 | 351 | 350 |

**C'est avec la fréquence d'apparition de chacune des lettres que vous trouverez la langue utilisée.**

3.1 **Exo-3-1 : En utilisant la méthode d'analyse fréquentielles pour décoder les messages chiffrés par substitution et en vous servant du tableau des fréquences d'apparition de référence des lettres de l'alphabet en français et anglais.**

- **Écrire une fonction Python qui permet de déchiffrer les messages 4 et 5 par la fréquence des lettres (pour déterminer la clé), en vous aidant de votre fonction du calcul de coïncidence (qui détermine la langue du message)**

- **Message supplémentaire pour les exercices :**
- **Message 6 :**

Dpsspht Ishrl Aol Apnly APNLY, apnly, ibyupun iypnoa Pu aol mvylzaz vm aol upnoa, Doha pttvyahs ohuk vy lfl Jvbsk myhtl aof mlhymbs zfttlayf? Pu doha kpzahua kllwz vy zrplz Ibyua aol mpyl vm aopul lflz? Vu doha dpunz khyl ol hzwpyl? Doha aol ohuk khyl zlpgl aol mpyl? Huk doha zovbskly huk doha hya Jvbsk adpza aol zpuldz vm aof olhya? Huk dolu aof olhya ilnhu av ilha, Doha kylhk ohuk huk doha kylhk mlla? Doha aol ohttly? doha aol johpu? Pu doha mbyuhjl dhz aof iyhpu? Doha aol hucps? Doha kylhk nyhzw Khyl paz klhksf alyyvyz jshzw? Dolu aol zahyz aoyld kvdu aolpy zwlhyz, Huk dhaly'k olhclu dpao aolpy alhyz, Kpk Ol ztpsl Opz dvyr av zll? Kpk Ol dov thkl aol shti thrl aoll? Apnly, apnly, ibyupun iypnoa Pu aol mvylzaz vm aol upnoa, Doha pttvyahs ohuk vy lfl Khyl myhtl aof mlhymbs zfttlayf ?

### 3.2 Exo-3-2 : Valider votre fonction python par la fréquence des lettres (pour déterminer la clé), avec le message 6

### 3.3 Exo-3-3 : En utilisant la méthode des digrammes (fréquence d'apparition par groupes de deux « digrammes » ou par groupes de trois « trigrammes » ) cela vous permettra de valider si le message est crypté ou codés . Lien de la liste des digrammes et trigrammes : https://fr.wikipedia.org/wiki/Liste_de_digrammes_et_trigrammes

- **Écrire une fonction Python qui compte l'ensemble des digrammes (fréquence d'apparition par groupes de deux)**
- **Écrire une fonction Python qui permet de savoir si un message est en clair en utilisant fréquence d'apparition par groupes de deux ou trois**

### 3.4 BONUS - Exo-2-3 : Pour améliorer la reconnaissance de la langue.

- **Écrire une fonction Python qui compte l'ensemble des lettres répétées 2 fois (RR, LL,MM…), 3 fois dans un tableau**

## 4 Présentation base64

**Base64 est un codage de l'information utilisant 64 caractères, sélectionnés pour être disponibles sur la majorité des tables de caractères. Il est utilisé pour les e-mails par exemple.**

### 4.1 Comment chiffrer avec Base64 ? (Principe de codage)

Le chiffrement utilise les valeurs binaires texte (dépendant du codage de celui-ci).
DCODE s'écrit 01100100 01000011 01101111 01100100 01100101 en binaire (code ASCII)
Le codage consiste à découper le message en groupe de 6 bits, on complète avec des 0 si besoin.
011001 000100 001101 101111 011001 000110 0101 (+00)

**Chaque groupe de 6 bits a une valeur en base 10,** on y associe le caractère de même rang dans l'alphabet Base64 (départ à 0) :
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz123456789+/'

25 4 13 47 25 6 20
Dans l'alphabet, 25 correspond à Z, 4 correspond à E, etc. On obtient **ZENvZGU.**

La Base64 ne fonctionne que par groupes de 4 caractères simples, il faut au besoin compléter avec le caractère =. **ZENvZGU devient ZENvZGU='**

### 4.2 Comment décoder par Base64 ? (Principe de décodage)

Le déchiffrement consiste à retrouver les valeurs des lettres dans l'alphabet Base64 :
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz123456789+/'

Soit le message ZENvZGU=
Les rangs sont : 25 4 13 47 25 6 20 (le signe égal = n'existe pas, il est ignoré)
Les valeurs sont converties en Binaire sur 6 bits.
011001 000100 001101 101111 011001 000110 010100
Le message binaire est alors découpé tous les 8 bits.
01100100 01000011 01101111 01100100 01100101 00 (les bits restants, ici les deux 0 finaux sont ignorés)
Le message binaire est alors ré-encodé selon la table de caractères désirée (ASCII, Unicode, etc.)
Le message clair est DCODE (en ASCII).

### 4.3 . Le chiffre Base64 :

Le message est composé de 65 caractères maximum. Par défaut il s'agit de :
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz123456789+/=

- **La taille des données augmente :**

En Base64, on utilise 4 caractères ASCII pour coder 3 octets, le volume des données est donc augmenté de 33%.

- **Base64URL :**

Les caractères 62 +' et 63 / peuvent poser des problèmes dans les URL, on les remplace alors par respectivement - et _. Le = est quant à lui supprimé.