# Cryptographie

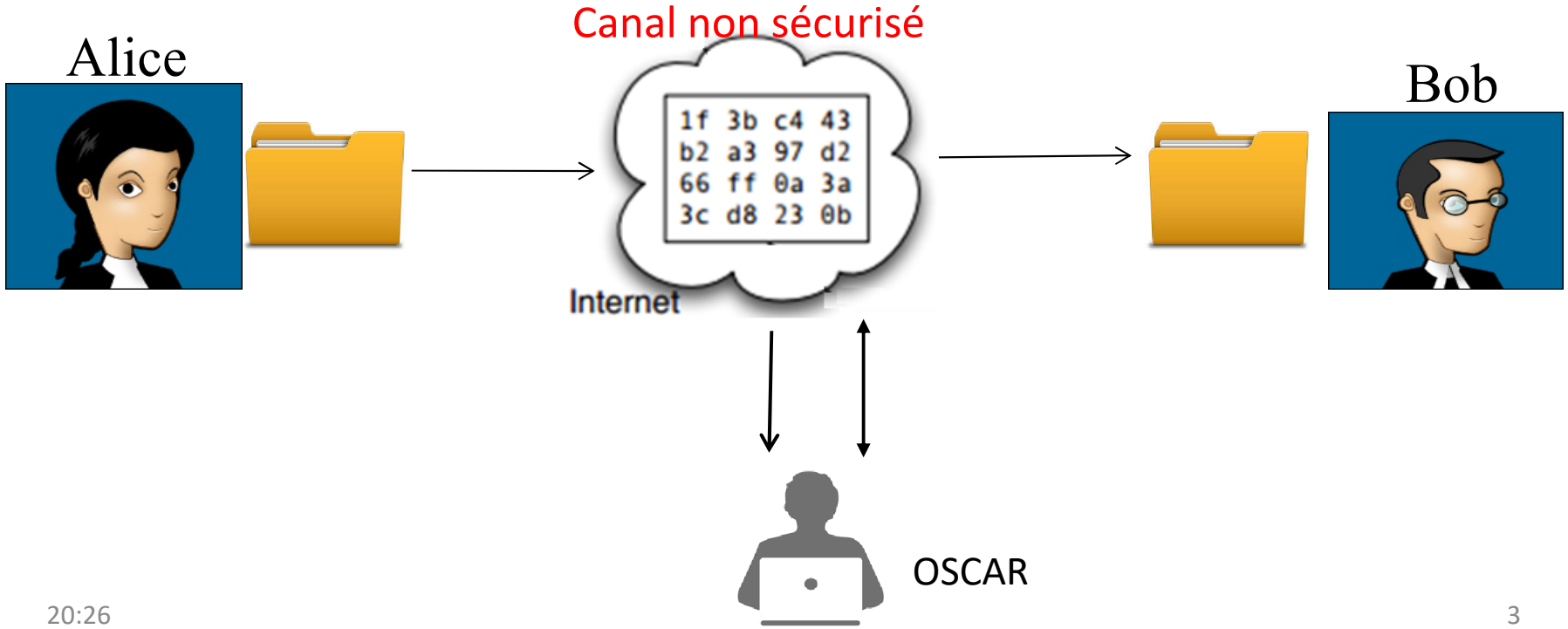
Pr. Mustapha JOHRI

mu.johri@gmail.com

# Introduction
# &
# Algorithmes classiques

# Terminologie

On se placera dans la problématique d'un émetteur et d'un récepteur désirant s'envoyer un message sur un canal de transmission public (donc ouvert à la possibilité qu'une tierce personne intercepte le message).

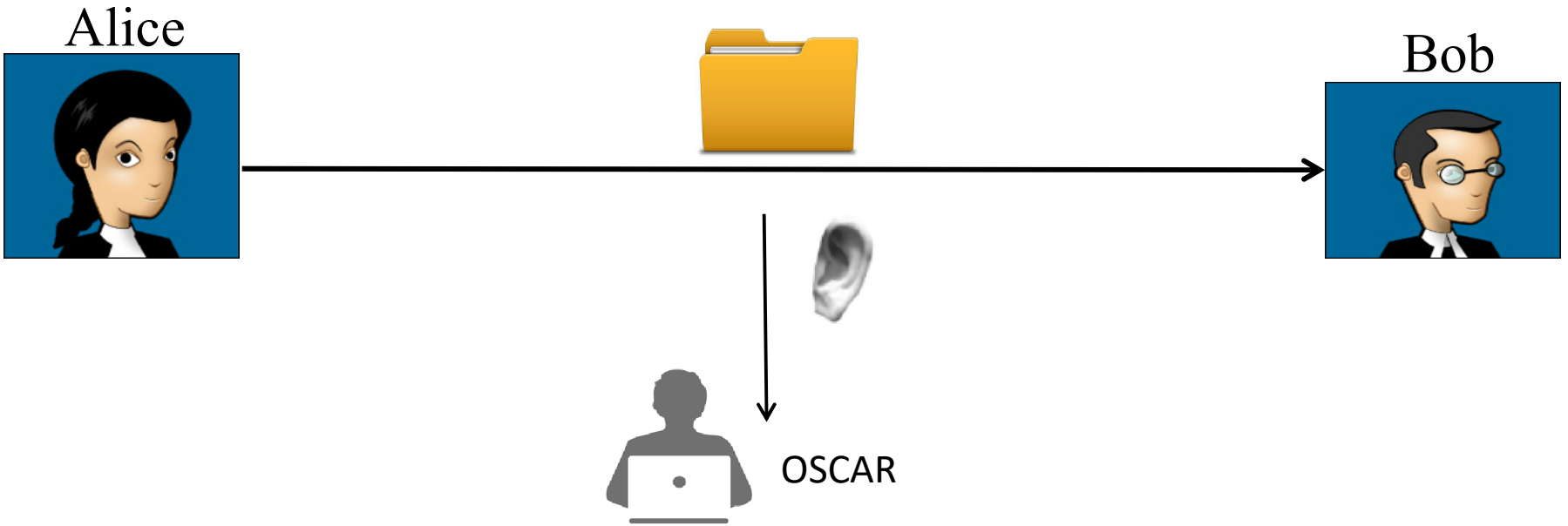


# Terminologie

Le but est de décrire et d'analyser des procédés permettant de transformer le message original, que l'on désignera par message clair (ou texte clair), en un message équivalent dont le contenu initial sera dissimulé (par la transformation). Ce procédé sera appelé chiffrement (ou cryptage).

# Terminologie

- **Attaques passives**

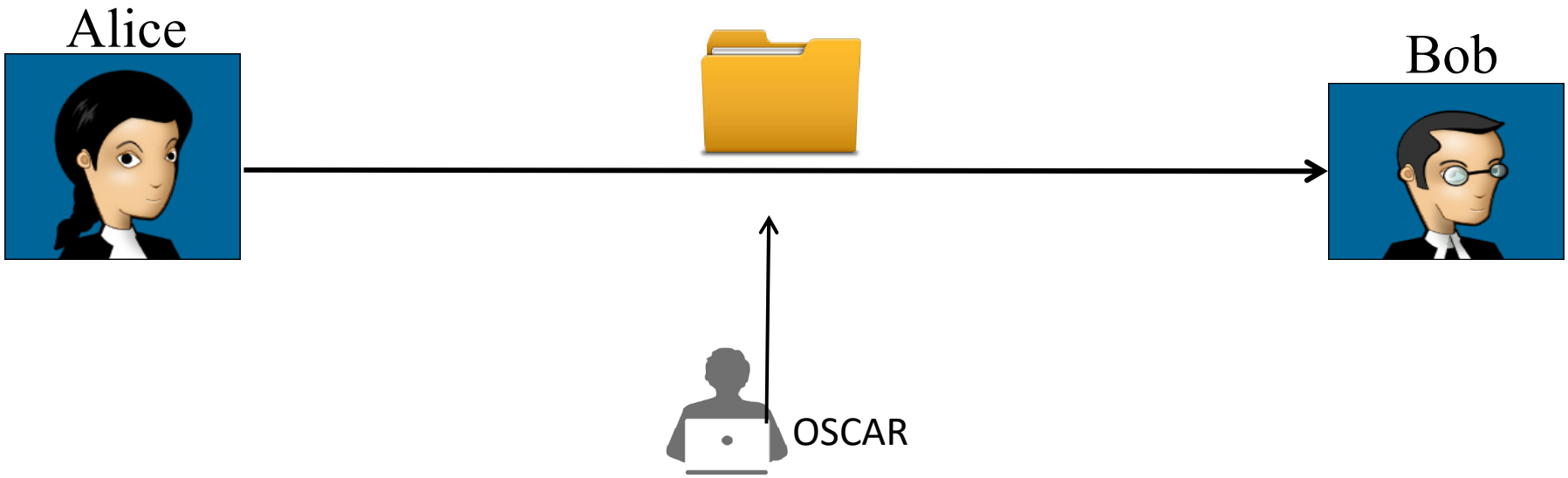



- **Menace contre la confidentialité de l'information :**

une information sensible parvient à une personne autre que son destinataire légitime

# Terminologie

- **Attaques actives : interventions sur la ligne**




**Menace contre l'intégrité et l'authenticité de l'information :**

- **Impersonification** : modification de l'identité de l'émetteur / récepteur
- **Altération des données** (modification du contenu)
- **Destruction des données**
- **Retardement de la transmission**

# Terminologie

- La cryptologie est la science du secret. Elle se divise en deux disciplines :
  - La cryptographie qui est l'étude des algorithmes permettant de protéger de l'information. Ces algorithmes sont appelés cryptosystèmes ;
  - la cryptanalyse qui est l'étude du niveau de sécurité des cryptosystèmes fournis par les cryptographes (art de "casser" des cryptosystèmes).

# Terminologie

- la stéganographie est une forme de dissimulation d'information dans le but de **transmettre un message de manière inaperçue au sein d'un autre message**. L'information utile est cachée au premier abord et à l'œil nu, mais non protégée pour qui sait où regarder.

- En -600, Nabuchodonosor, roi de Babylone, utilisait la stéganographie :
  - Écrire sur le crâne rasé de ses esclaves,
  - Attendre que leurs cheveux aient repoussé.
  - Pour lire le message : raser la tête de l'esclave.

# Terminologie

- Chiffrer : l'action de rendre un message en clair M (plaintext) à l'aide d'une clé K en un message illisible C appelé (ciphertext) cryptogramme ou message chiffré.

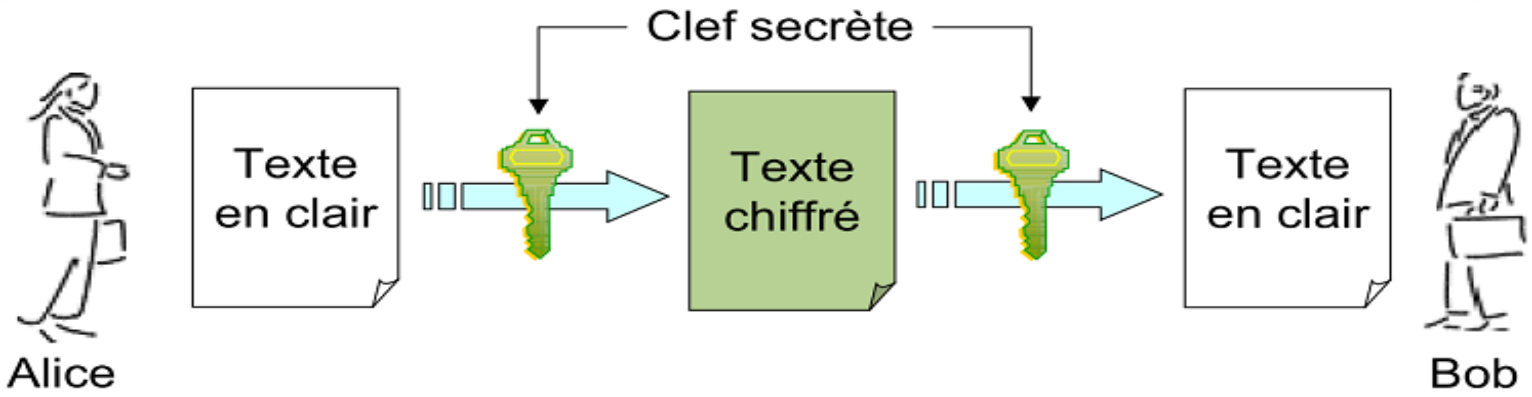
- Déchiffrer : Action inverse du chiffrement.

- Cryptosystème : L'algorithme (ou le dispositif physique) permettant de chiffrer des données.

- Attaquer, Casser : Mettre à mal la sécurité d'un cryptosystème (retrouver M à partir de C sans connaitre la clé, retrouver la clé).

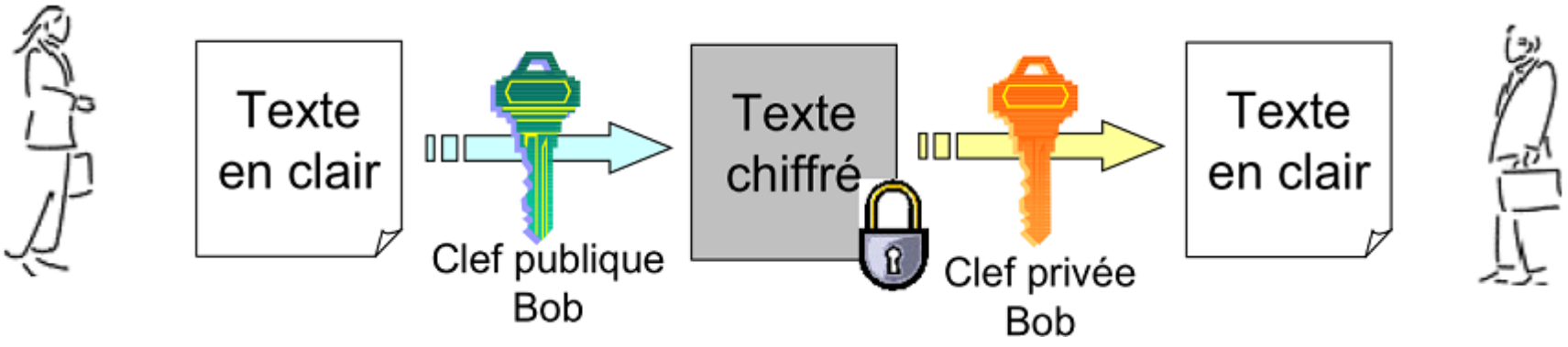
# Terminologie

- Il existe 2 types de chiffrement:
  - Le chiffrement symétrique (ou chiffrement à clé privée) consiste à utiliser la même clé pour le chiffrement et le déchiffrement.



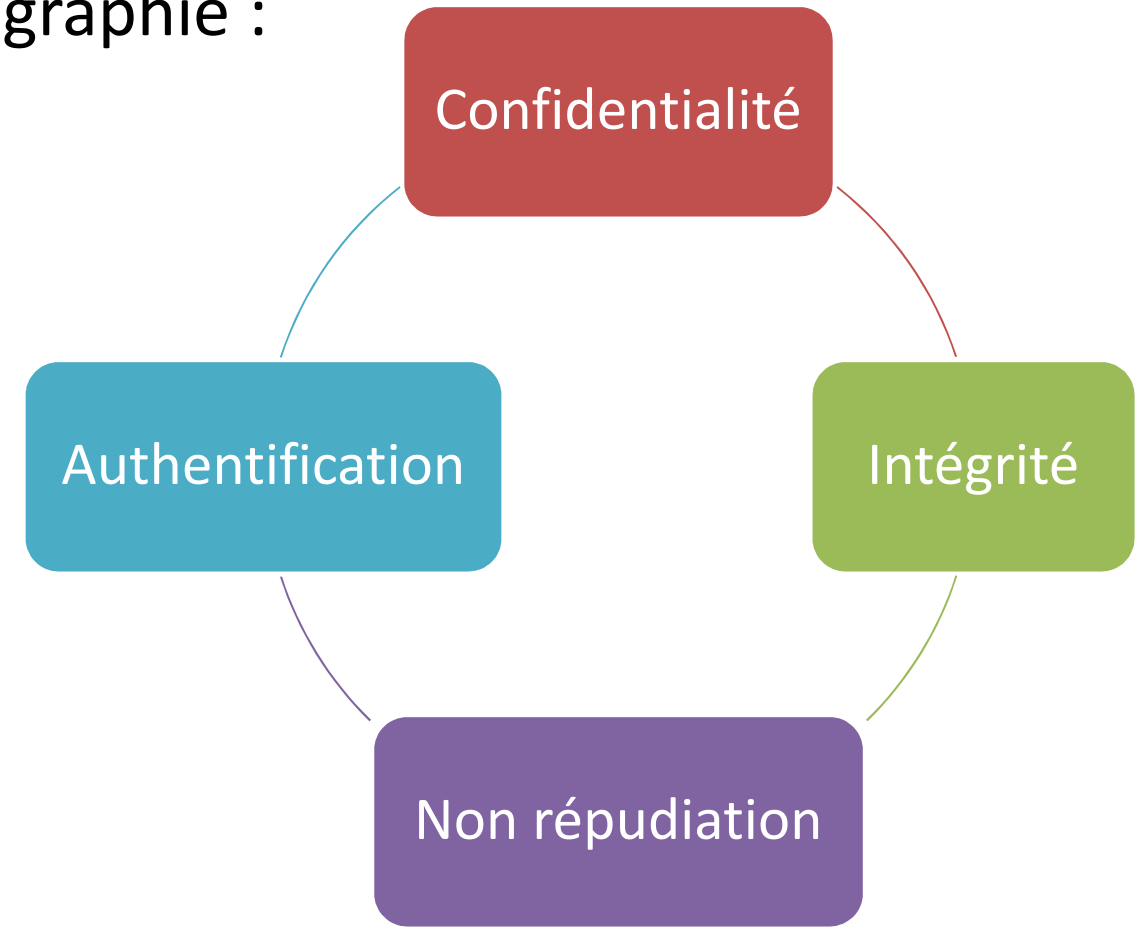
# Terminologie

– Le chiffrement asymétrique (ou chiffrement à clés publiques) consiste à utiliser une clé publique pour le chiffrement et une clé privée pour le déchiffrement.



# Problème à résoudre par cryptographie

Voici les problèmes fondamentaux que doit résoudre la cryptographie :



# Problème à résoudre par cryptographie

## La confidentialité

– Il s'agit de garantir le secret de l'information transmise ou archivée.

– Seuls les utilisateurs autorisés doivent y avoir accès.

# Problème à résoudre par cryptographie

## L'authentification:

- l'émetteur est sûr de l'identité du destinataire c'est à dire que seul le destinataire pourra prendre connaissance du message car il est le seul à disposer de la clef de déchiffrement.

- le destinataire est sûr de l'identité de l'émetteur

- L'authentification Consiste à vérifier qu'une personne possède bien l'identité, ou les droits, qu'elle affirme avoir.

# Problème à résoudre par cryptographie

## L'intégrité

- Il s'agit de préserver les informations contre les modifications.
- "L'intégrité est la prévention d'une modification non autorisée de l'information "
- Avec les techniques actuelles, cette fonction est réalisée par la signature numérique.

# Problème à résoudre par cryptographie

## La non répudiation

Impossibilité, pour une personne ou pour toute autre entité engagée dans une communication par voie informatique, de nier avoir reçu ou émis un message.

# Cryptanalyse

## Définition :

Reconstruction d'un message chiffré en clair à l'aide de méthodes mathématiques.

## Pourquoi ?

- Tout cryptosystème doit nécessairement être résistant aux méthodes de cryptanalyse.

- Lorsqu'une méthode de cryptanalyse permet de déchiffrer un message chiffré à l'aide d'un cryptosystème, on dit alors que l'algorithme de chiffrement a été « cassé ».

# Cryptographie classique

# Quelques cryptosystèmes classiques

- Chiffrement par substitution
- Chiffrement par transposition

# Chiffrement par substitution

## Définition:

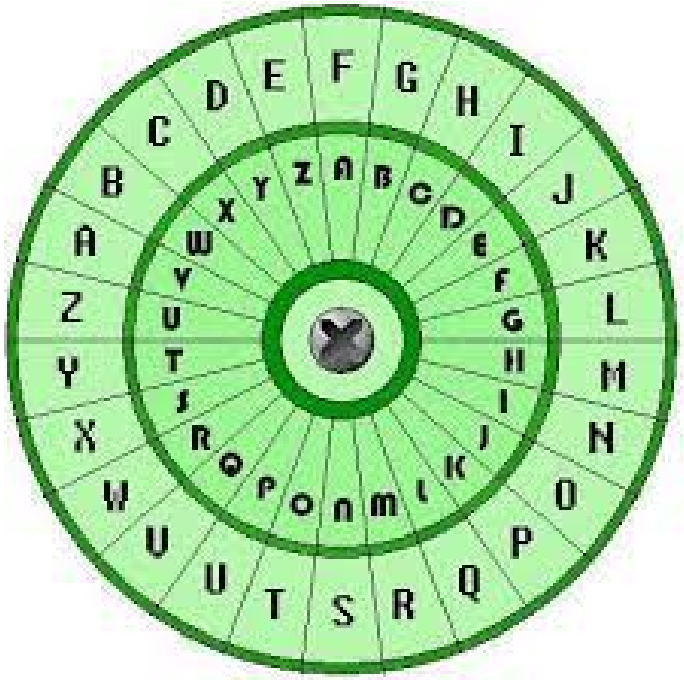
- Le chiffrement par substitution consiste à remplacer dans un message une ou plusieurs entités (généralement des lettres) par une ou plusieurs autres entités.

- Toutes les substitutions simples sont vulnérables à une analyse des fréquences d'apparition des lettres.

# Chiffrement par substitution

## Chiffre de César:

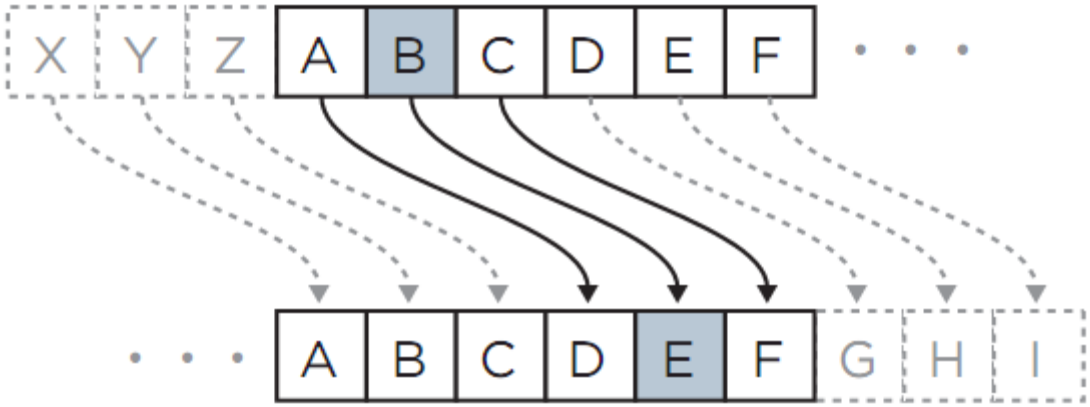
- Substituer chaque lettre du message en clair par une autre située à distance fixe dans l'alphabet. Cette distance devait être connue de l'expéditeur comme du destinataire.



# Chiffrement par substitution

## Chiffre de César:

- décalage de trois lettres (K=3) :




- Par exemple décalage de trois lettres vers la droite :

ECOLEDHIVER devient HFROHLYHU

- Exercice : Crypter le texte « ESTBM » avec la clé k =-1 :

ESTBM devient DRSAL

# Chiffrement par substitution

## Chiffrement de César:

| A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |

| N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ |
| 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

### ❑ Principe :

- Soit **m** (resp. **c**) l'indice de la lettre du message en clair (resp. chiffré) et **k** le décalage (la clé):
- **Chiffrement (Encryption) :**

$$c = E_k(m) = (m + k) \bmod 26$$

- **Déchiffrement (Decryption) :**

$$m = D_k(c) = (c - k) \bmod 26$$

# Chiffrement par substitution

## Chiffrement de César:

- Exercice :
  a) Chiffrez le message « bonjour tout le monde » en utilisant le cryptosystème de César(k=3).

     ERQMRXU WRXW OH PRQGH

  b) Décryptez le message chiffré suivant et donnez la clé de chiffrement :

     HWDUYTLWFUMNJ

  – Texte clair : Cryptographie
  – K=-5 ou K=21

# Chiffrement par substitution

## Chiffrement de César:

- **L'espace de clés est:|K|=26.**
- **Cette méthode est vulnérable aux attaques de type:**
  - **Force brute (26 clés possibles !)**
  - **Analyse de fréquences :**

Le principe de cette cryptanalyse consiste à deviner les lettres d'un texte clair sur la base de leur fréquence d'apparition

# Chiffrement de César:

- Créer un programme qui reçoit en entrée une clé de codage **n** (entier) et un message **M** (tableau de caractère) à coder et qui doit afficher le message crypté correspondant.
- ***Entrée*** : La clé **n** (qui peut être négative) et un message **M** en majuscule sans accent.
- ***Sortie*** : **C** Le message crypté par le chiffrement de César avec la clé **n**. La ponctuation et espaces ne doivent pas changer.

```c
#include <stdio.h>
void cesar(char M[], int n) {
  int i = 0;
  while (M[i] != '\0') {
    if (M[i] >= 'A' && M[i]<= 'Z') {
        char c = M[i] - 'A';
        c += n;
        c = c % 26;
        if (c<0) c+=26;
        M[i] = c +'A';
    }
    i++;
  }
  printf("%s", M);
}
```

# Chiffrement de César:

- Réaliser une fonction ***Cesar*** qui permet de réaliser le chiffrement et le déchiffrement selon un ***argument en entrée e*** (0 : chiffrement, 1 : déchiffrement)

- ***Entrée :*** La clé **n**, le message **M** en majuscule sans accent et le paramètre **e**.
- ***Sortie*** : Le message correspondant.

```c
#include <stdio.h>
void cesar(char M[], int n,int e) {
    int d,i = 0;
    if (e)   d=-1;   else  d=1;
    while (M[i] != '\0') {
        if (M[i] >= 'A' && M[i]<= 'Z') {
            char c = M[i] - 'A';
            c += d*n;
            c = c % 26;
            if (c<0) c+=26;
            M[i] = c +'A';
        }
        i++;
    }
    printf("%s", M);
}
```

# Chiffrement par substitution

## Analyse fréquentielle:

- **Fréquences d'apparition des lettres(français)**

| Lettre | A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Fréquence | 8,11 | 0,81 | 3,38 | 4,28 | 17,69 | 1,13 | 1,19 | 0,74 | 7,24 | 0,18 | 0,02 | 5,99 | 2,29 |

| Lettre | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Fréquence | 7,68 | 5,2 | 2,92 | 0,83 | 6,53 | 8,87 | 7,44 | 5,23 | 1,28 | 0,06 | 0,53 | 0,26 | 0,12 |

• Analyse des fréquences des lettres : E, S et A sont les plus fréquentes en français, le moins fréquent est W.

• Cette technique ne fonctionne bien que si le message chiffré est suffisamment long pour avoir des moyennes significatives.

# Chiffrement par substitution

## Analyse fréquentielle(di/tri-grammes):

- Digrammes les plus utilisés en langue française : ES, LE, EN …
- Trigrammes : ENT, LES, EDE…

| Digrammes | Pourcentages |
|---|---|
| ES | 3,15 % |
| LE | 2,46 % |
| EN | 2,42 % |
| DE | 2,15 % |
| RE | 2,09 % |
| NT | 1,97 % |
| ON | 1,64 % |
| TE | 1,63 % |
| ER | 1,63 % |
| SE | 1,55 % |

# Analyse fréquentielle

- Ecrire une fonction qui reçoit en paramètre une chaine de caractère **M** et qui permet de renvoyer un tableau d'entiers de 26 valeurs représentant le nombre d'occurrences de chaque caractère de **M** (*case[0]=nombre d'occurrence de 'A', case[1]=nombre d'occurrence de 'B'…*).

```c
#include <stdio.h>
void Statistique(char M[], int n,int e) {
    int d,i = 0;
    if (e)   d=-1;   else  d=1;
    while (M[i] != '\0') {
        if (M[i] >= 'A' && M[i]<= 'Z') {
            char c = M[i] - 'A';
            c += d*n;
            c = c % 26;
            if (c<0) c+=26;
            M[i] = c +'A';
        }
        i++;
    }
    printf("%s", M);
}
```

- Exemple : Si ***M = AABBBCDDXYZ.*** Alors la fonction renvoie le tableau suivant :

2 3 1 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1

# Chiffrement par substitution

## Le chiffrement affine:

- L'idée est d'utiliser comme fonction de chiffrement une fonction affine du type $y=(k_1.x + k_2) \bmod 26$, où $k_1$ et $k_2$ sont des constantes, et où $x$ et $y$ sont des nombres correspondant aux lettres de l'alphabet (A=0,B=1,…,Z=25).
- On peut remarquer que si $k_1=1$, alors on retrouve le chiffre de César et $k_2$ est le décalage.

# Chiffrement par substitution

## Le chiffrement affine : fonctionnement

- **Message**

$$M=m_1m_2\ldots m_{n-1}m_n$$

- **Clé** :

$$(k_1, k_2) \in \{0,25\} \text{ et } \mathbf{pgcd(k_1,26)=1}$$

- **Chiffrement**:

$$c_i= f(m_i) = ( k_1*m_i + k_2 ) \bmod 26$$

- **Déchiffrement** :

$$m_i= f^{-1}(c_i) = (u*(c_i–k_2)) \bmod 26$$

Où u est l'inverse de $k_1$ mod 26 (càd $k_1 *u=1 \bmod 26$)

- **Nombres de clés possibles** :

Il n'existe que 12 entiers compris entre 0 et 26 et premiers avec 26 [1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23 et 25]. Il n'existe donc que 12 × 26 = 312 clés de chiffrement possible. Cette méthode est vulnérable par force brute.

# Chiffrement par substitution

## Le chiffrement affine:

**Identité de Bézout :**

Soient a et b deux entiers relatifs. Si d est le PGCD de a et b, alors il existe deux entiers relatifs x et y tels que ax + by = d.

- On a pgcd($k_1$,26)=1, alors il existe x et y tels que :

$$k_1.x+26.y=1$$

- Si On prend cette équation mod 26, on trouve que:

$$k_1.x=1 \bmod 26 \text{ (car } 26 =0 \bmod 26)$$

D'où u=x

# Chiffrement par substitution

## Le chiffrement affine:

- Exemple:
  - Clé = $(k_1, k_2) = (17,3)$
  - Message : C O D E → 2 ; 14 ; 3 ; 4
  - Chiffrement :

$$c_i = f(m_i) = (17 * m_i + 3) \bmod 26$$

**C= L H C T**

  - Déchiffrement :

17.x+26.y=1 → u=-3 et y=2, donc u=-3 mod 26 = 23

$$m_i = f^{-1}(c_i) = 23 * (c_i - 3) \bmod 26$$

On trouve 2 ; 14 ; 3 ; 4 - > M = C O D E

# Le chiffrement affine : Recherche de l'inverse

- **Algorithme d'Euclide étendu:**

Trouver les coefficients de Bézout pour 255 et 141 en utilisant l'algorithme d'Euclide étendu.

$$
\begin{aligned}
255 &= 1 \times 141 + \boxed{114} \\
141 &= 1 \times 114 + \boxed{27} \\
114 &= 4 \times 27 + \boxed{6} \\
27 &= 4 \times 6 + \boxed{3} \\
6 &= 2 \times 3 + 0
\end{aligned}
$$

Maintenant on élimine les restes :

$$
\begin{aligned}
3 &= 27 - 4 \times \boxed{6} \\
&= \boxed{27} - 4 \times (114 - 4 \times \boxed{27}) \\
&= -4 \times 114 + 17 \times \boxed{27} \\
&= -4 \times \boxed{114} + 17 \times (141 - 1 \times \boxed{114}) \\
&= 17 \times 141 - 21 \times \boxed{114}) \\
&= 17 \times 141 - 21 \times (255 - 1 \times 141) \\
&= 38 \times 141 - 21 \times 255
\end{aligned}
$$

Les coefficients Bachet-Bézout sont 38 et $-21$.

# Le chiffrement affine : Recherche de l'inverse

- **Algorithme d'Euclide étendu:**

**Exercice : Calculer $12^{-1}$ mod 67.**

On applique l'algorithme d'Euclide étendu pour trouver le pgcd de 12 et 67.

$$\begin{aligned} 67 &= 5 \cdot 12 + 7 \\ 12 &= 1 \cdot 7 + 5 \\ 7 &= 1 \cdot 5 + 2 \\ 5 &= 2 \cdot 2 + 1 \\ 2 &= 2 \cdot 1 + 0 \end{aligned}$$

Le pgcd de 12 et 67 est alors bien 1.

On cherche les coefficients de Bézout.

$$\begin{aligned} 1 &= 5 - 2 \cdot 2 \\ &= 5 - 2(7 - 5) \\ &= -2 \cdot 7 + 3 \cdot 5 \\ &= -2 \cdot 7 + 3 \cdot (12 - 7) \\ &= 3 \cdot 12 - 5 \cdot 7 \\ &= 3 \cdot 12 - 5 \cdot (67 - 5 \cdot 12) \\ &= -5 \cdot 67 + 28 \cdot 12. \end{aligned}$$

# Le chiffrement affine : Recherche de l'inverse

- **Algorithme d'Euclide étendu:**

**Exercice : Calculer $12^{-1}$ mod 67.**

Nous avons alors

$$
\begin{aligned}
-5 \cdot 67 + 28 \cdot 12 &= 1 \\
28 \cdot 12 &\equiv 1 \mod 67 \\
28 &\equiv 12^{-1} \mod 67.
\end{aligned}
$$

# Chiffrement par substitution

## Le chiffrement affine : Recherche de l'inverse

- **Algorithme d'Euclide étendu:**

**Entrée** : a, b entiers (naturels)

**Sortie** : d entier (naturel) et x, y entiers relatifs tels que :

d = pgcd(a, b) et d = a*x+b*y

```
Fonction Euclide_etendu(a,b)
    Si b>a Alors Echange(a,b) FinSI
    Si b=0 Alors retourne (a,1,0)
    Sinon
        (d,x,y) <- Euclide_etendu(b,a mod b)
        (d,x,y)=(d,y,x-⌊a/b⌋y)
        retourne (d,y,x)
    FinSi
Fin
```

# Chiffrement par substitution

## Cryptanalyse: chiffre affine

- message chiffré : HGAHY RAEFT GAGRH DGAGM OEHIY RAAOT ZGAGJ GKFDG AZGSB INNTG KGRHE NNIRG
- Sachant que le message a été chiffré par le chiffre affine, trouvez le message en clair (utiliser une analyse de fréquence).

- Solution:
  - ✓ On remarque que G apparait 13 fois et A 8 fois.
  - ✓ E, S, A, I sont les lettres les plus fréquentes, donc

E → G et S → A

Indice 4

Indice 18

# Chiffrement par substitution

## Cryptanalyse: chiffre affine

Trouver $(k_1; k_2)$ ??

On a

$$E_{k1;\,k2}(E)=G \quad \text{et} \quad E_{k1;\,k2}(S)=A$$

$$4k_1 + k_2 = 6 \bmod 26 \quad \text{et} \quad 18k_1 + k_2 = 0 \bmod 26$$

Alors

$$14k_1=-6 \bmod 26 \rightarrow 7k_1=20 \bmod 26 \rightarrow 7k_1=10 \bmod 13.$$

D'où

$$\mathbf{k_1=7 \text{ et } k_2=4.}$$

Le message déchiffré est:

TESTONS A PRESENT LES EQUATIONS SUR DES EXEMPLES DE CHIFFREMENT AFFINE

# Chiffrement par permutation

- Exemple d'une permutation

$$\begin{pmatrix} A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z \\ D,X,Y,W,V,H,N,A,I,B,T,U,G,S,C,R,O,J,Q,P,E,Z,K,F,M,L \end{pmatrix}$$

- Nombre des clés et le nombre de permutation possibles :

$$|K| = 26x25x24x\ldots x1 = |K| = 26!$$

- Soit **π** une permutation:
  - Alors $E_k(m) = \pi(m) = c$, $D_k(c) = \pi^{-1}(c) = m$.
  - M="CE TEXTE EST CHIFFRE PAR SUBSTITUTION"
  - C="YV PVFPV VQP YAIHHJV RDJ QEXQPIPEPICS"

# Chiffrement par permutation

1) Ecrire une fonction **generer_permutation(int K[])** qui reçoit en paramètre un tableau d'entiers K de taille 26 qui représente l'alphabet et qui génère d'une manière aléatoire les nombres entre 0 et 25 afin de créer une permutation.

- **Exemple** : generer_permutation(K) renvoie le tableau K suivant

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

2) Ecrire la fonction **coder(char M[], int K[])** qui reçoit en paramètre une chaine de caractère M et une permutation K et qui renvoie le cryptogramme C

# Chiffrement par permutation

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
void generer_permutation(int K[]){
    int i,j,temp;
    srand(time(NULL));
    for(i=0;i<26;i++)    K[i]=i;
    for(i=0;i<26;i++){
        j = (i + rand())%26;
        temp=K[i];
        K[i]=K[j];
        K[j]=temp;
    }
}
```

```c
void coder(char M[],int K[]){
    int i=0;
    while(M[i]!='\0'){
        M[i]=K[M[i]-65]+65;
        i++;
    }
    puts(M);
}
```

```c
int main()
{   int K[26],i;
    char M[20]="ESTBM";
    generer_permutation(K);
    //for(i=0;i<26;i++) printf("%d ",K[i]);
    coder(M,K);
    return 0;
}
```

# Chiffrement par substitution

## Subst. polyalphabétique: chiffre de Vigenère

- Le **chiffre de Vigenère** est une amélioration décisive du chiffre de César.
- Sa force réside dans le fait que ce chiffre utilise une **clef** qui définit le décalage pour chaque lettre du message.

# Chiffrement de Vigenère

## Exemple: Chiffrement de Vigenère

chiffrons le texte "CHIFFRE DE VIGENERE" avec la clef "FPBM"

(cette clef est éventuellement répétée plusieurs fois pour être aussi longue que le texte en clair).

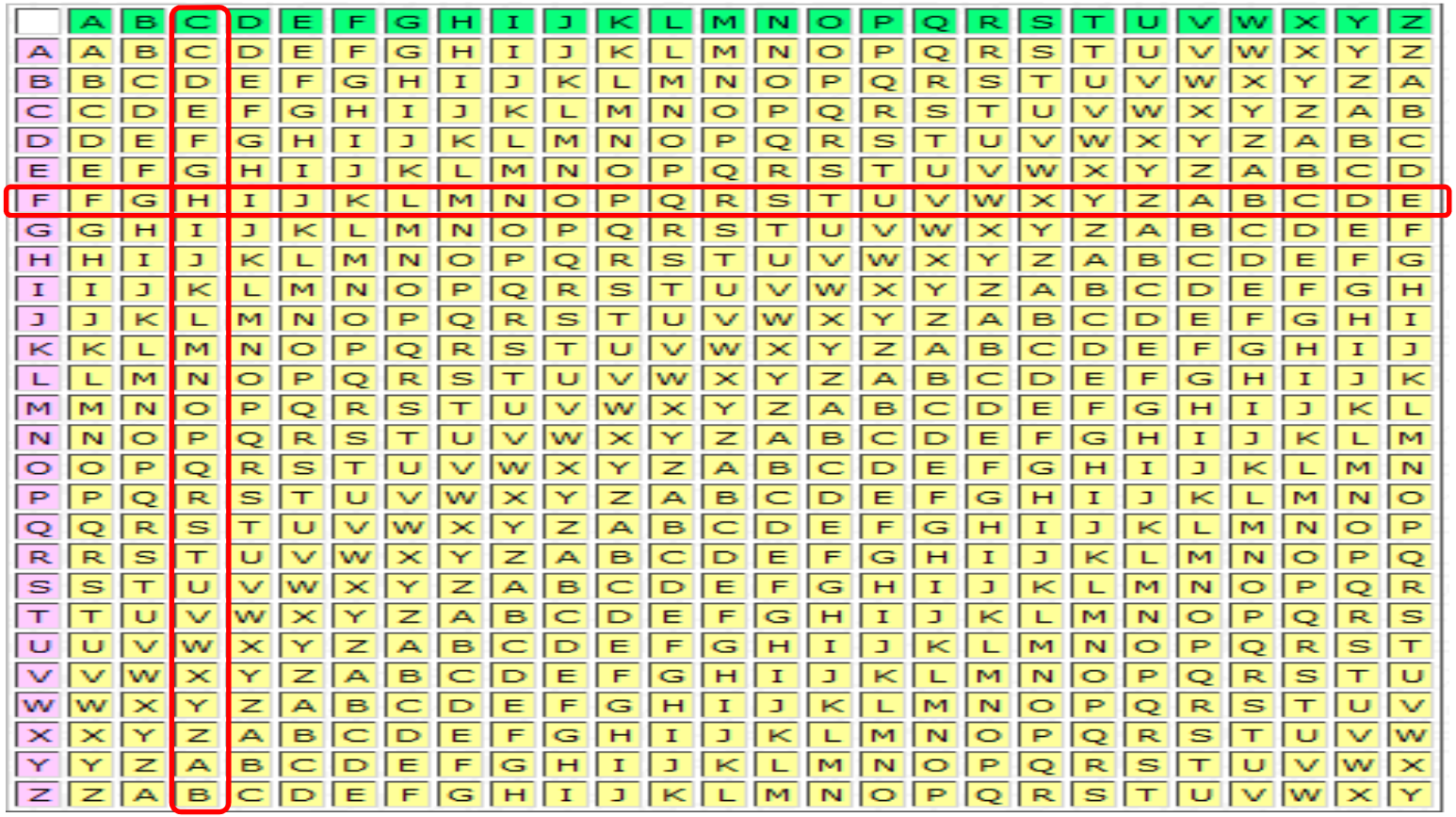
| A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |

| N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ |
| 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

| clair | C | H | I | F | F | R | E | D | E | V | I | G | E | N | E | R | E |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| clef | **F** | P | B | M | **F** | P | B | M | **F** | P | B | M | **F** | P | B | M | F |
| décalage | 5 | 15 | 1 | 12 | 5 | 15 | 1 | 12 | 5 | 15 | 1 | 12 | 5 | 15 | 1 | 12 | 5 |
| chiffré | H | W | J | R | K | G | F | P | J | K | J | S | J | C | F | D | J |

# Chiffrement de Vigenère

## Carré de Vigenère

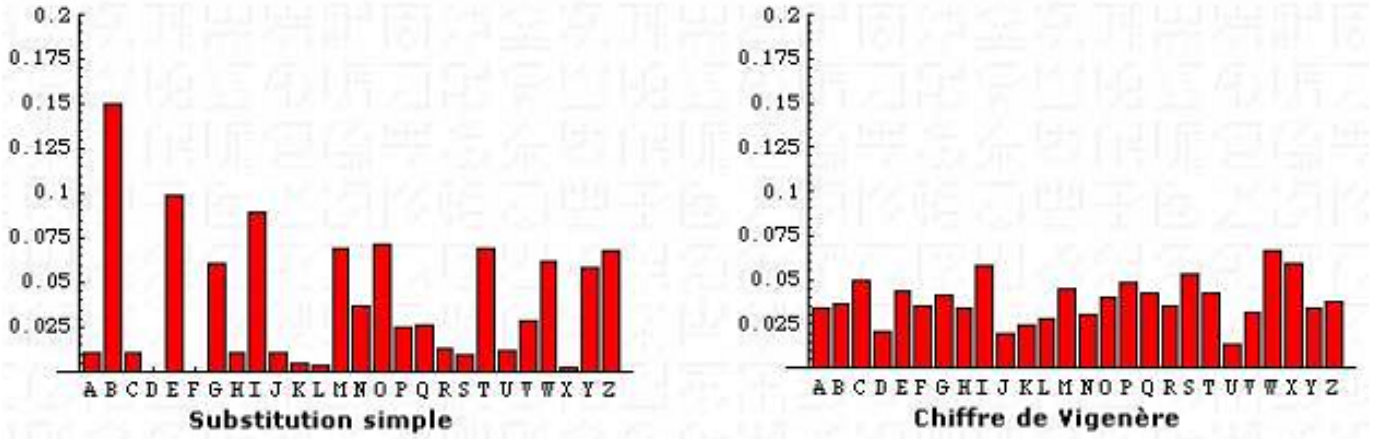


| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T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

- ✓ La lettre de la clef est dans la colonne la plus à gauche,
- ✓ la lettre du message clair est dans la ligne tout en haut.
- ✓ La lettre chiffrée est à l'intersection des deux.

# Chiffrement de Vigenère

- La grande force du chiffre de Vigenère est que la même lettre sera chiffrée de différentes manières d'où perte de la fréquence des lettres, ce qui rend inutilisable l'analyse de fréquence classique.




Perte de la fréquence des lettres

- Dans la littérature, pour faire une **cryptanalyse** du chiffre de Vigenère on utiliser l'un des deux tests "Test de Kasiski" ou "Test de Friedman" pour déterminer la **longueur de la clé**.

# Chiffrement de Vigenère

## Carré de Vigenère

Soit la déclaration suivante :

*typedef int matrice[26][26] ;*

Qui permet de définir une matrice.

1) Écrire une fonction ***void CarreVignere(matrice V)*** qui permet de générer le carré Vigenère dans la matrice V de dimension 26x26.

2) Écrire la fonction ***coder(char M[],char K[],matrice V)*** qui permet de coder le message M suivant la clef K en utilisant le carré vigenère V

# Chiffrement de Vigenère

```
typedef int matrice[26][26];
void CarreVignere(matrice V){
     int i,j;
     for(i=0;i<26;i++){
          for(j=0;j<26;j++){
               V[i][j]=(i+j)%26;
     }
}
}
```

```
void afficher(matrice M){
     int i,j;
     for(i=0;i<26;i++){
          for(j=0;j<26;j++){
               printf("%d ",M[i][j]);
          }
          printf("\n");
     }
}
```

```
void coder(char M[],char K[],matrice V){
     int i=0,l,a,b;
     for(l=0;K[l]!='\0';l++);
     while(M[i]!='\0'){
          a=M[i]-65;
          b=K[i%l]-65;
          //printf("%d %d",b,a);
          //system("pause");
          M[i]=V[b][a]+65;
          i++;
     }
     puts(M);
}
```

```
int main()
{    matrice V;
     char K[10]="FPBM";
     char M[20]="CHIFFREDEVIGENERE";
     CarreVignere(V);
     //afficher(V);
     coder(M,K,V);
     return 0;
}
```

# Chiffrement par substitution

## Chiffre Playfair:

- Le chiffre de Playfair utilise un tableau de 5×5 lettres, contenant un mot clé.
- La mémorisation du mot clé et de 4 règles à suivre suffisent pour utiliser ce chiffrement.
- Remplir le tableau avec les lettres du mot clé (en ignorant les doublons), puis le compléter avec les autres lettres manquantes de l'alphabet en respectant l'ordre d'apparaition.
- Pour un texte en langue française, la lettre **W exclu** car inutile, on utilise **V à la place**
- La variante anglaise consiste à **garder le W** et à fusionner I et J.

# Chiffrement par substitution

## Chiffre Playfair:

### Exemple

- Soit le mot clé = GALOIS, Donner le carré Playfair correspondant :

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

# Chiffrement par substitution

- Chiffre Playfair : Chiffrement :

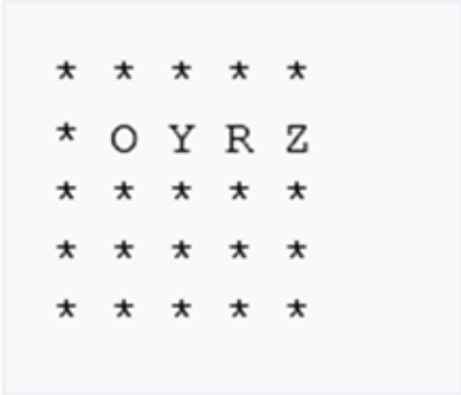
Pour chiffrer un message, il faut prendre les lettres ***2 par 2*** et appliquer les règles suivantes :

1. Si les **2 lettres sont identiques** (ou s'il n'en **reste qu'une**) **mettre un 'X' après la première lettre**. Chiffrer la nouvelle paire ainsi constituée et continuer avec la suivante.

Par exemple pour chiffrer le message **'OR'.**

1. Si les lettres se trouvent sur **la même ligne de la table**, il faut les remplacer par celles se trouvant immédiatement à leur droite (en bouclant sur la gauche si le bord est atteint),

sur la même ligne




**OR → YZ**

**YR → RZ**

# Chiffrement par substitution

- Chiffre Playfair : Chiffrement :

3. Si les lettres apparaissent sur la **même colonne**, les **remplacer** par celles qui sont juste **en dessous** (en bouclant par le haut si le bas de la table est atteint),

sur la même colonne


**OR → BY**

**RY → YO**

4. **Sinon**, **remplacer** les lettres par celles se trouvant **sur la même ligne**, mais dans le **coin opposé** du **rectangle** défini par la paire originale.

forment un rectangle


**OR → ZX**

# Chiffrement par substitution

- Chiffre Playfair **:** Déchiffrement :
- Utiliser la **méthode inverse** en prenant les **lettres à gauche** dans le cas d'une même **ligne**, vers le **haut** dans le cas d'une même **colonne**, et toujours les **coins opposés** dans le cas d'un rectangle.
- Ignorer les 'X' qui n'ont pas leur place dans le message final.

# Chiffrement par substitution

## Exercice :

1) Soit la clé « exemple playfair », remplissez le tableau et chiffrez le message M=« Cache l'or dans la souche de l'arbre » :

**CA CH EL OR DA NS LA SO UC HE DE LA RB RE**

**BY DB XE QI BF JU ER VJ TD BL BM ER AH AL**

2) Déchiffrer le message « **AE SC PX**» :

**AE SC PX**

**ES TB ME**

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

# Chiffrement de Hill

Ce chiffre est publié en 1929 par Lester S. Hill, est un chiffre polygraphique (comme le chiffre Playfair), c'est-à-dire qu'on ne (dé)chiffre pas les lettres les unes après les autres, **mais par paquets**.

Afin de simplifier le choses, nous grouperons les lettres **deux par deux,** mais on peut imaginer des paquets plus grands, par exemple des paquets de trois lettres.

## ❑ Chiffrement

Les lettres sont d'abord remplacées par leur rang dans l'alphabet. Les lettres $P_k$ et $P_{k+1}$ du texte clair seront chiffrées $C_k$ et $C_{k+1}$ avec la formule ci-dessous:

$$\begin{pmatrix} C_k \\ C_{k+1} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} P_k \\ P_{k+1} \end{pmatrix} \pmod{26}$$

# Chiffrement de Hill

Ce qui signifie, pour fixer les idées, que les deux premières lettres du message clair ($P_1$ et $P_2$) seront chiffrées ($C_1$ et $C_2$) selon les deux équations suivantes:

$$C_1 \equiv aP_1 + bP_2 \pmod{26}$$
$$C_2 \equiv cP_1 + dP_2 \pmod{26}$$

## ❑ Exemple de chiffrement

Alice prend comme clef de cryptage la matrice $K=\begin{pmatrix} 9 & 4 \\ 5 & 7 \end{pmatrix}$

pour chiffrer le message P="je parle" . Après avoir remplacé les lettres par leur rang dans l'alphabet (a=1, b=2, etc.), elle obtiendra:

$$C_1 \equiv 9 \cdot 10 + 4 \cdot 5 \pmod{26} = 110 \pmod{26} = 6$$
$$C_2 \equiv 5 \cdot 10 + 7 \cdot 5 \pmod{26} = 85 \pmod{26} = 7$$

Elle fera de même avec les 3e et 4e lettres, 5e et 6e, etc.

# Chiffrement de Hill

## ❑ Déchiffrement

Pour déchiffrer, le principe est le même que pour le chiffrement: on prend les lettres deux par deux, puis on les multiplie par une matrice.

$$\begin{pmatrix} P_1 \\ P_2 \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} \begin{pmatrix} C_1 \\ C_2 \end{pmatrix} \pmod{26}$$

Cette matrice **doit être l'inverse** de matrice de chiffrement (**modulo 26**).

**Remarque**

*la matrice $K$ à une matrice inverse modulo 26 si et seulement si $PGCD(\det\ K, 26) = 1$*

Ordinairement l'inverse de la matrice $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ est:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

# Chiffrement de Hill

## ❑ Exemple de déchiffrement

Pour déchiffrer le message d'**Alice**, **Bob** doit calculer :

$$\begin{pmatrix} 9 & 4 \\ 5 & 7 \end{pmatrix}^{-1} = \frac{1}{43}\begin{pmatrix} 7 & -4 \\ -5 & 9 \end{pmatrix} \pmod{26} = (43)^{-1}\begin{pmatrix} 7 & -4 \\ -5 & 9 \end{pmatrix} \pmod{26}$$

Comme ***pgdc(43,26)=1***, alors $(43)^{-1}$ existe et égale à 23 . Bob a donc maintenant la matrice de déchiffrement:

$$\begin{pmatrix} 9 & 4 \\ 5 & 7 \end{pmatrix}^{-1} = 23\begin{pmatrix} 7 & -4 \\ -5 & 9 \end{pmatrix} \pmod{26} = \begin{pmatrix} 161 & -92 \\ -115 & 207 \end{pmatrix} \pmod{26} = \begin{pmatrix} 5 & 12 \\ 15 & 25 \end{pmatrix} \pmod{26}$$

Bob prend donc la matrice pour déchiffrer le message Il obtiendra:

$$P_1 \equiv 5\cdot 6 + 12\cdot 7 \pmod{26} = 114 \pmod{26} = 10$$
$$P_2 \equiv 15\cdot 6 + 25\cdot 7 \pmod{26} = 265 \pmod{26} = 5$$

# FIN