Chiffrement symétrique

Christophe Viroulaud

Terminale - NSI

Archi 20

La communication sur internet est organisée en couches.

Couche application (Navigateur)
Couche TCP (Transport)
Couche IP (Internet)
Couche réseau (Matérielle)

Tableau 1 – Protocole TCP/IP

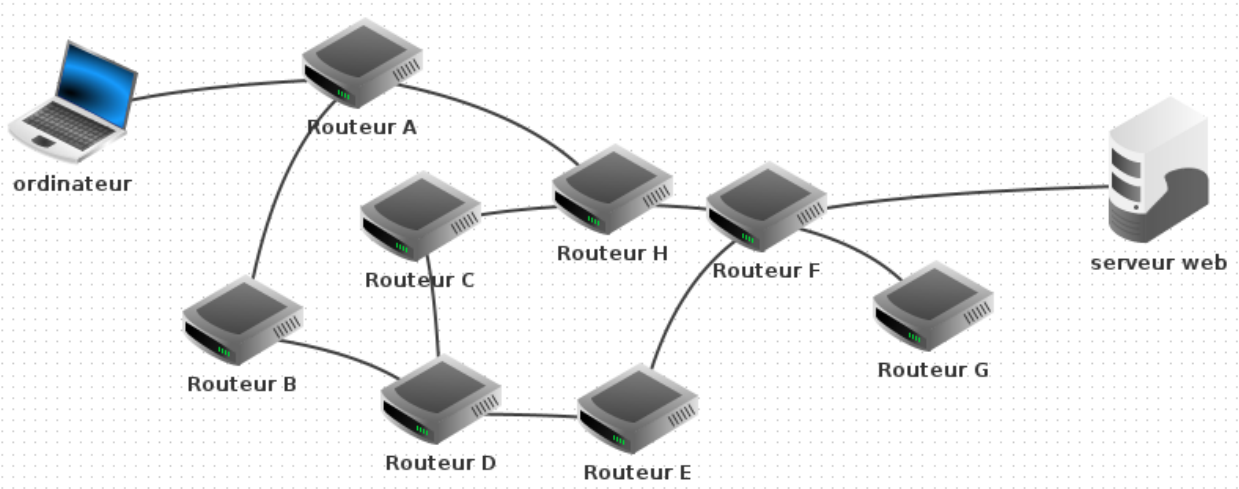


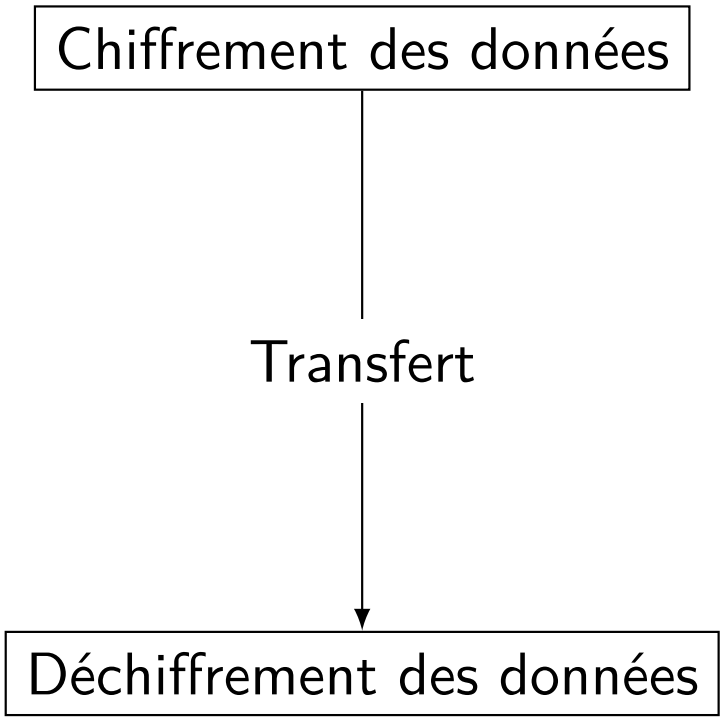
FIGURE 1 – Les paquets IP transitent sur le réseau internet en circulant de routeurs en routeurs.

En théorie, rien n'interdit à un routeur d'inspecter un paquet et donc d'en connaître son contenu.

Comment chiffrer le contenu des communications ?

Sommaire

Chiffrement symétrique



- La source utilise une *fonction de chiffrement* pour coder un message `m` avec une clé de chiffrement `k`. La fonction produit en sortie un message chiffré `s`.

$$\texttt{chiffrement(m, k)} \rightarrow \texttt{s}$$

- La source utilise une *fonction de chiffrement* pour coder un message `m` avec une clé de chiffrement `k`. La fonction produit en sortie un message chiffré `s`.

$$\texttt{chiffrement(m, k)} \rightarrow \texttt{s}$$

- Le destinataire utilise une *fonction de déchiffrement* pour décoder le message `s` avec la clé de chiffrement `k`. La fonction produit en sortie le message clair `m`.

$$\texttt{déchiffrement(s, k)} \rightarrow \texttt{m}$$

À retenir

Dans un chiffrement symétrique on utilise la même clé pour chiffrer et déchiffrer le message.

Sommaire

Chiffrement de César

Le chiffrement de César utilise un décalage alphabétique comme clé de chiffrement. Par exemple, avec la clé **+2** :

- A devient C
- B devient D
- ...
- Z devient B

Activité 1 : Écrire la fonction `chiffrement(message: str, cle: int)` $\rightarrow$ `str` qui code `message`.
On n'utilisera que des caractères majuscules ASCII dans le message et on supprimera les espaces. Dans un premier temps, on ne s'occupera pas du *débordement de l'alphabet*. Ainsi l'appel

```
1 >>> chiffrement("Z", 1)
```

renverra le caractère `[` situé à la 91° position du code ASCII.

Correction

```
1  def chiffrement(message: str, cle: int) -> str:
2      sortie = ""
3      for lettre in message:
4          # code ASCII de la lettre chiffrée
5          code = ord(lettre) + cle
6          # ajout
7          sortie = sortie+chr(code)
8      return sortie
```

Activité 2 : Modifier la fonction pour que l'appel

```
1 >>> chiffrement("Z", 1)
```

renvoie la lettre **A**

Correction

```
1  def chiffrement(message: str, cle: int) -> str:
2      sortie = ""
3      for lettre in message:
4          # code ASCII de la lettre chiffrée
5          code = ord(lettre) + cle
6          # ajustement du code ASCII
7          if code > ord("Z"):
8              code = code-26
9          # ajout
10         sortie = sortie+chr(code)
11     return sortie
```

Activité 3 : Écrire la fonction `dechiffrement(secret: str, cle: int)` $\rightarrow$ `str` qui déchiffre `secret` en prenant en compte le *débordement de l'alphabet*.

Correction

```
1  def dechiffrement(secret: str, cle: int) -> str:
2      sortie = ""
3      for lettre in secret:
4          # code ASCII de la lettre chiffrée
5          code = ord(lettre) - cle
6          # ajustement du code ASCII
7          if code < ord("A"):
8              code = code+26
9          # ajout
10         sortie = sortie+chr(code)
11     return sortie
```

```
1  >>> secret = chiffrement("NSI",15) # CHX
2  >>> dechiffrement(secret, 15) # NSI
```

Remarque

Si la clé vaut **13**, la fonction de chiffrement permet également de déchiffrer.

```
1 >>> chiffrement("NSI", 13) # AFV
2 >>> chiffrement("AFV", 13) # NSI
```

Sommaire

Un chiffrement faible

- Le chiffrement de César n'offre que 25 clés.

Un chiffrement faible

- Le chiffrement de César n'offre que 25 clés.
- La fréquence d'apparition des lettres est une méthode simple à mettre en place pour décrypter un message.

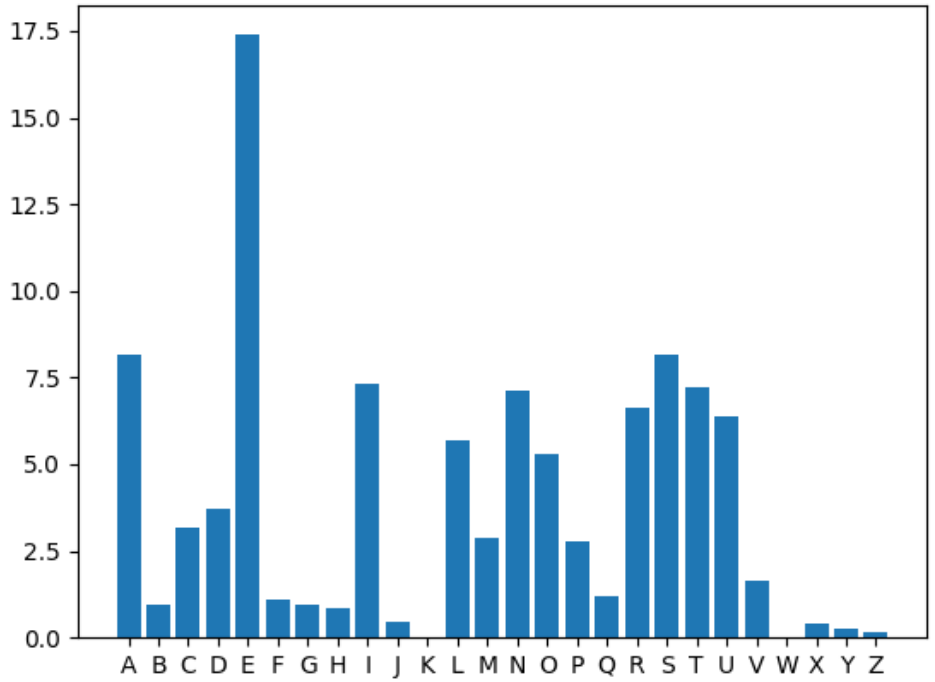


FIGURE 2 – Fréquences d'apparition des lettres

Sommaire

Chiffrement polyalphabétique - Principe

Il consiste à :

- utiliser une clé de chiffrement composée de plusieurs lettres,
- recopier la clé de façon à obtenir une chaîne de la longueur du message.

B	R	A	V	O
N	S	I	N	S

Remarques :

- Une même lettre ne sera plus forcément codée par le même symbole.

Remarques :

- Une même lettre ne sera plus forcément codée par le même symbole.
- Une clé de la même taille que le message garantit une protection sûre (téléphone rouge).

Remarques :

- Une même lettre ne sera plus forcément codée par le même symbole.
- Une clé de la même taille que le message garantit une protection sûre (téléphone rouge).
- Cette méthode est utilisée dans le code de Vigenère ou la machine Énigma.

Sommaire

Porte XOR

À retenir

La porte XOR est réversible :

$$\text{Si } A \oplus B = C \text{ alors } A \oplus C = B \text{ et } B \oplus C = A$$

B	R	A	V	O
66	82	65	86	79
N	S	I	N	S
78	83	73	78	83

Tableau 2 – Conversion en ASCII

B	R	A	V	O
66	82	65	86	79
1000010	1010010	1000001	1010110	1001111
N	S	I	N	S
78	83	73	78	83
1001110	1010011	1001001	1001110	1010011

Tableau 3 – Conversion en binaire

Message	1000010	1010010	1000001	1010110	1001111
$\oplus$	1001110	1010011	1001001	1001110	1010011
Chiffré	0001100	0000001	0001000	0011000	0011100

Tableau 4 – Application de la porte XOR

Remarque

Le message chiffré est envoyé, puis une application de la porte XOR avec la même clé permet de retrouver le message d'origine.

Chiffré	0001100	0000001	0001000	0011000	0011100
$\oplus$	1001110	1010011	1001001	1001110	1010011
Message	1000010	1010010	1000001	1010110	1001111

Tableau 5 – Application de la porte XOR

Sommaire

Utilisations

La fonction `xor` est implémentée dans les processeurs : il est possible de chiffrer en temps réel :

- chiffrement d'un disque dur,
- chiffrement des données d'un smartphone.

La méthode est utilisée dans plusieurs algorithmes de chiffrement :

- algorithme DES (Data Encryption Standard)
  - obsolète à cause d'une clé maximale de 56 bits (2^{56} possibilités),
  - lenteur pendant le chiffrage.

La méthode est utilisée dans plusieurs algorithmes de chiffrement :

- algorithme DES (Data Encryption Standard)
  - obsolète à cause d'une clé maximale de 56 bits (2^{56} possibilités),
  - lenteur pendant le chiffrage.
- algorithme AES (Advanced Encryption Standard) :
  - utilise une clé 128 bits,
  - choisi par l'institut de standardisation américain NIST (National Institute of Standards and Technology) en décembre 2001.

La méthode est utilisée dans plusieurs algorithmes de chiffrement :

- algorithme DES (Data Encryption Standard)
  - obsolète à cause d'une clé maximale de 56 bits (2^{56} possibilités),
  - lenteur pendant le chiffrage.
- algorithme AES (Advanced Encryption Standard) :
  - utilise une clé 128 bits,
  - choisi par l'institut de standardisation américain NIST (National Institute of Standards and Technology) en décembre 2001.
- Chacha20 :
  - date de 2008,
  - améliore les performances d'un autre algorithme (Salsa20),
  - 20 étapes de mélange.