Sécurisation des communications

J. Boucher

Lycée Pierre-Paul RIQUET, Terminale NSI

21 mars 2025

Plan

Problématique

Lorsqu'un utilisateur utilise son navigateur pour accéder à la page Web `http://snt-nsi.info/tnsi/`,

1. le navigateur isole le protocole (`http`), le nom du serveur (`snt-nsi.info`) et la ressource demandée (`/tnsi/`) ;
2. le navigateur effectue une requête DNS pour obtenir l'adresse IP du serveur (IPv4 : 51.178.82.21) ;
   → couche Internet : déterminer la route
3. le navigateur se connecte à la machine dont l'adresse IP est 51.178.82.21, en utilisant le *protocole TCP* sur le port 80 ;
   → couche transport : garantir l'intégrité des données
4. une fois la connexion établie, client et serveur échange des données en utilisant le protocole HTTP.
   → couche d'applications : protocoles de haut niveau

Comment garantir que le contenu échanger ne soit connu que de la source et de la destination ? Que le serveur auquel on se connecte est bien celui qu'il dit être ?

Les techniques de chiffrement, toute une Histoire !

Dès l'Antiquité, généraux et souverains ont rivalisé d'ingéniosité pour imaginer [...] des procédés sophistiqués pour protéger le secret de leurs communications.

Les mathématiciens [1] ont alors régulièrement été mis à contribution [...] pour décrypter les messages chiffrés ennemis, œuvrant à l'apparition d'une science nouvelle, la « cryptanalyse ».

Code de César

machine Enigma

clé de chiffrement RSA

1. un des plus célèbres fut Alan Turing

Glossaire de la cryptographie

Coder : représenter de l'information par un ensemble de signes prédéfinis. On utilise parfois le verbe *encoder*.

Décoder : interpréter un ensemble de signes pour extraire l'information qu'ils représentent.

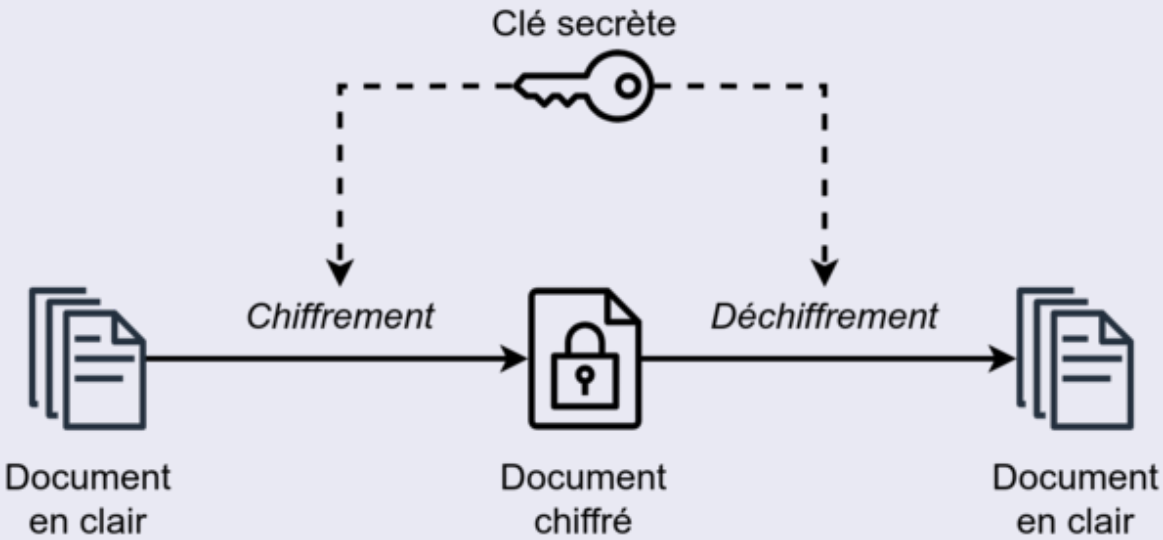
Chiffrer : rendre un suite de symboles incompréhensibles au moyen d'une **clé de chiffrement**.

Déchiffrer/décrypter : retrouver la suite de symboles originale à partir du message chiffré.
$\rightarrow$ On utilise le terme *déchiffrer lorsque l'on utilise une* **clé de chiffrement** pour récupérer le texte initial et le terme *décrypter* lorsque l'on arrive à déterminer le message original sans utiliser la clé.

1. Cryptographie symétrique

Chiffrement symétrique

Le chiffrement symétrique est une techinque basée sur l'utilisation d'une même *clé secrète* ou clé de chiffrement pour à la fois chiffrer et déchiffrer un message.



Application : Codage de César

Principe : Choisir un entier n puis décaler chaque lettre du message initial de n lettres dans l'alphabet. Recommencer à « A » si le décalage fait dépasser « Z ».

1. Décrypter le message suivant :

`OLQIRUPDWLTXHFHVWIDQWDVWLTXH`

2. En déduire la valeur de la clé de chiffrement utilisée.

Indication : On peut calculer la fréquence d'apparition des lettres en français sur le corpus de Wikipédia en français.

En 2008, le laboratoire CLLE-ERSS de l'Université de Toulouse en a tiré la table de fréquence suivante :

Rang	Caractère	Nombre d'occurrences	Pourcentage
1	e	115 024 205	12.10%
2	a	67 563 628	7.11%
3	i	62 672 992	6.59%
4	s	61 882 785	6.51%
5	n	60 728 196	6.39%
6	r	57 656 209	6.07%

...

Script Python

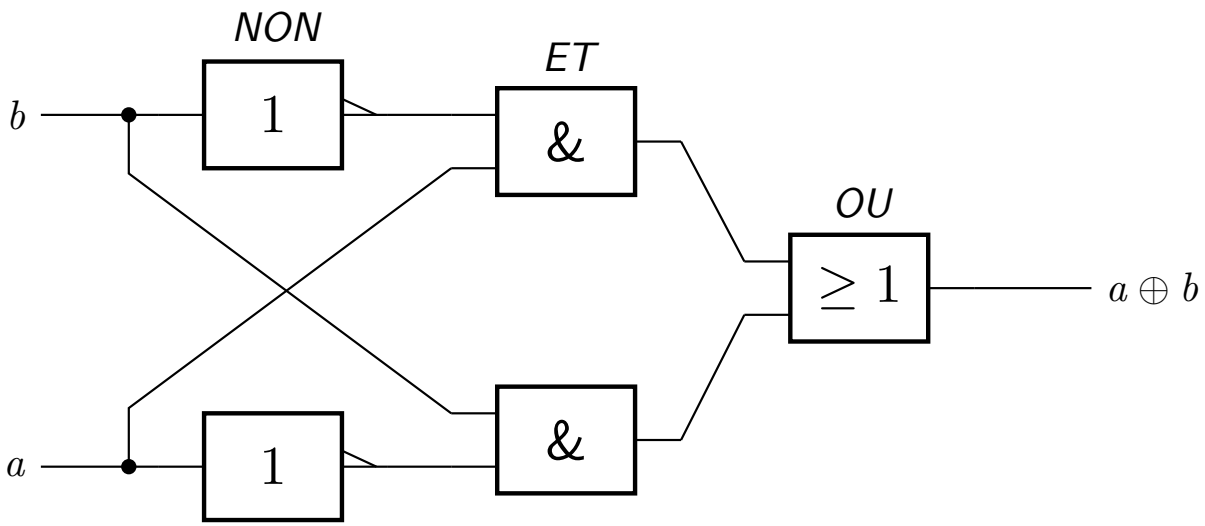
```
1  def chiffrement(mes, cle):
2      mes_chiffre = ''
3      for c in mes:
4          c_chiffre = (ord(c) - 65 + cle)%26
5          mes_chiffre += chr(c_chiffre+65)
6      return mes_chiffre

7  m = chiffrement('LINFORMATIQUECESTFANTASTIQUE', 3)
8  print(m)
9  > > > OLQIRUPDWLTXHFHVWIDQWDVWLTXH

10 m = chiffrement(m, -3)
11 print(m)
12 > > > LINFORMATIQUECESTFANTASTIQUE
```

Chiffrement par *OU exclusif* ou *XOR*

Chiffrement par *OU exclusif*

Le chiffrement par *OU exclusif* repose sur l'utilisation de l'opérateur *ou exclusif* noté $\oplus$.

Opérateur *OU exclusif*



À partir de son circuit logique, dresse la table de vérité de l'opérateur *OU exclusif*.

a	b	$a \oplus b$
0	0	
1	0	
0	1	
1	1	

Chiffrement par *OU exclusif* ou *XOR*

Chiffrement par *OU exclusif*

Le chiffrement par *OU exclusif* repose sur l'utilisation de l'opérateur *ou exclusif* noté $\oplus$.
L'algorithme est le suivant :

- Choisir une clé de chiffrement sous la forme d'une chaîne de caractères.
- Répéter la clé de façon à obtenir une chaîne de la même longueur que le message.
- Convertir le message et la clé étendue en nombre avec une table d'encodage.
- Effectuer l'opération $\oplus$, bit à bit, entre les chiffres du message et ceux de la clé étendue.

Déchiffrement : L'opération $\oplus$ étant réversible, il suffit d'effectuer à nouveau l'opération $\oplus$ entre le message chiffré et la clé pour le déchiffrer.
En effet, l'opération $\oplus$ vérifie la propriété suivante : $(a \oplus b) \oplus b = a$.

Exemple de chiffrement par *OU exclusif*

Message	I	N	F	O
Clé étendue	n	s	i	n
Message (en binaire)	1001001	1001110	1000110	1001111
Clé étendu (en binaire)	1101110	1110011	1101001	1101110
Message chiffré (en binaire)	0100111	0111101	0101111	0100001
Message chiffré (en ASCII)	'	=	/	!

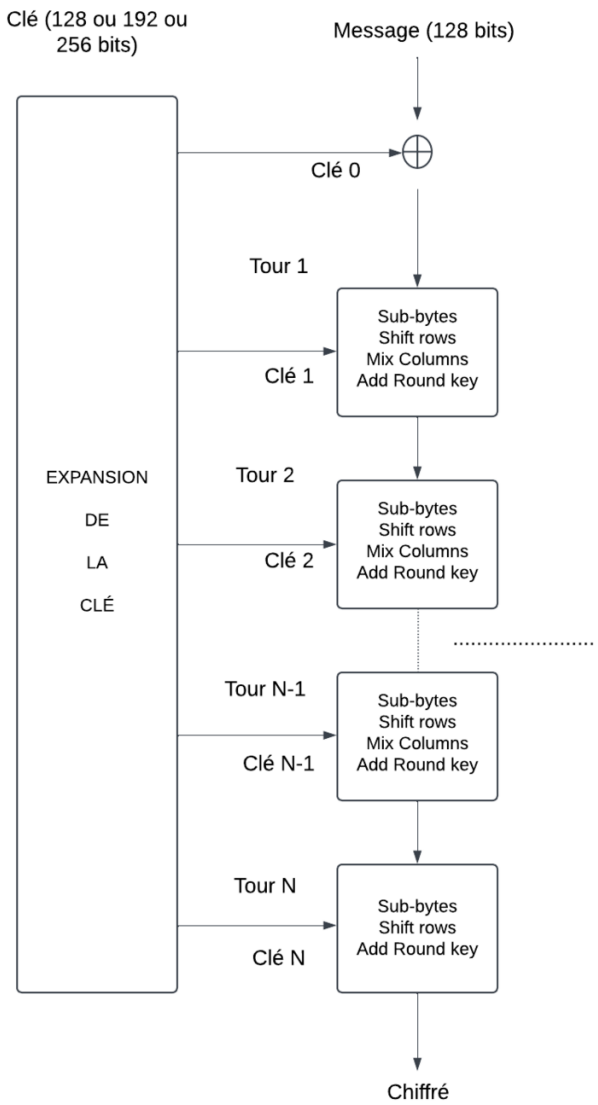
Chiffrement AES (Advanced Encryption Standard)

Principe du chiffrement AES

Le chiffrement AES (Advanced Encryption Standard) ou *norme de chiffrement avancé* est un des chiffrement actuellement le plus utilisé. Il repose sur un prinicpe similaire au cihffrement par *OU exclusif* :

- une clé initiale est étendue ;
- la clé et le message sont mélangés en utilisant des opérations $\oplus$, de façon réversible.

Chiffrement AES (Advanced Encryption Standard)



Remarques :

- À ce jour, il n'existe aucune attaque connue efficace pour casser le chiffrement AES.
- Les algorithmes utilsés sont très efficace et permettent de chiffrer des messages en temps réel (texte, audio, vidéo...).
- Ce chiffrement est très répandu, notamment utilisé par le protocole `https`, par les services de messagerie ou pour chiffrer les données écrites sur un disque dur ou dans une base de données...

Applications

Chiffrement *OU exclusif*

1. Écris en Python une fonction `chiffre_xor(msg, cle)` qui prend en arguments deux chaînes de caractères et qui renvoie le chiffrement par *OU exclusif* du message avec la clé, sous la forme d'une chaîne de caractère.
   Tu pourras utiliser :
   - `bin(n)` : renvoie une chaîne de caractère contenant la représentation binaire de l'entier `n` ;
   - `ord(c)` : renvoie le code entier du carctère `c` ;
   - `chr(n)` : renvoie le carctère `c` correspondant au code entier `n` ;
   - les opérateurs logiques `and`, `or` et `not`.
2. Sers-toi maintenant de l'opérateur `operator.xor` et du type `bytes` pour écrire une fonction `chiffre_xor_bin(msg, cle)`, qui prend en argument deux chaîne d'octets et qui renvoie le résultat sous la forme d'une chaîne d'octets.

Applications

Codage de César

Tu veux attaquer par force brute le message chiffré par la méthode de César.

Écris une fonction force_brute_cesar qui prend en paramètre une chaîne de caractères texte_chiffre qui a été chiffrée par la méthode de César, et qui décrypte cette chaîne par force brute.

On suppose que le message initial ne contient que des carcatères en majuscule, non accentuées et sans aucun espace, ni symbole de ponctuation.