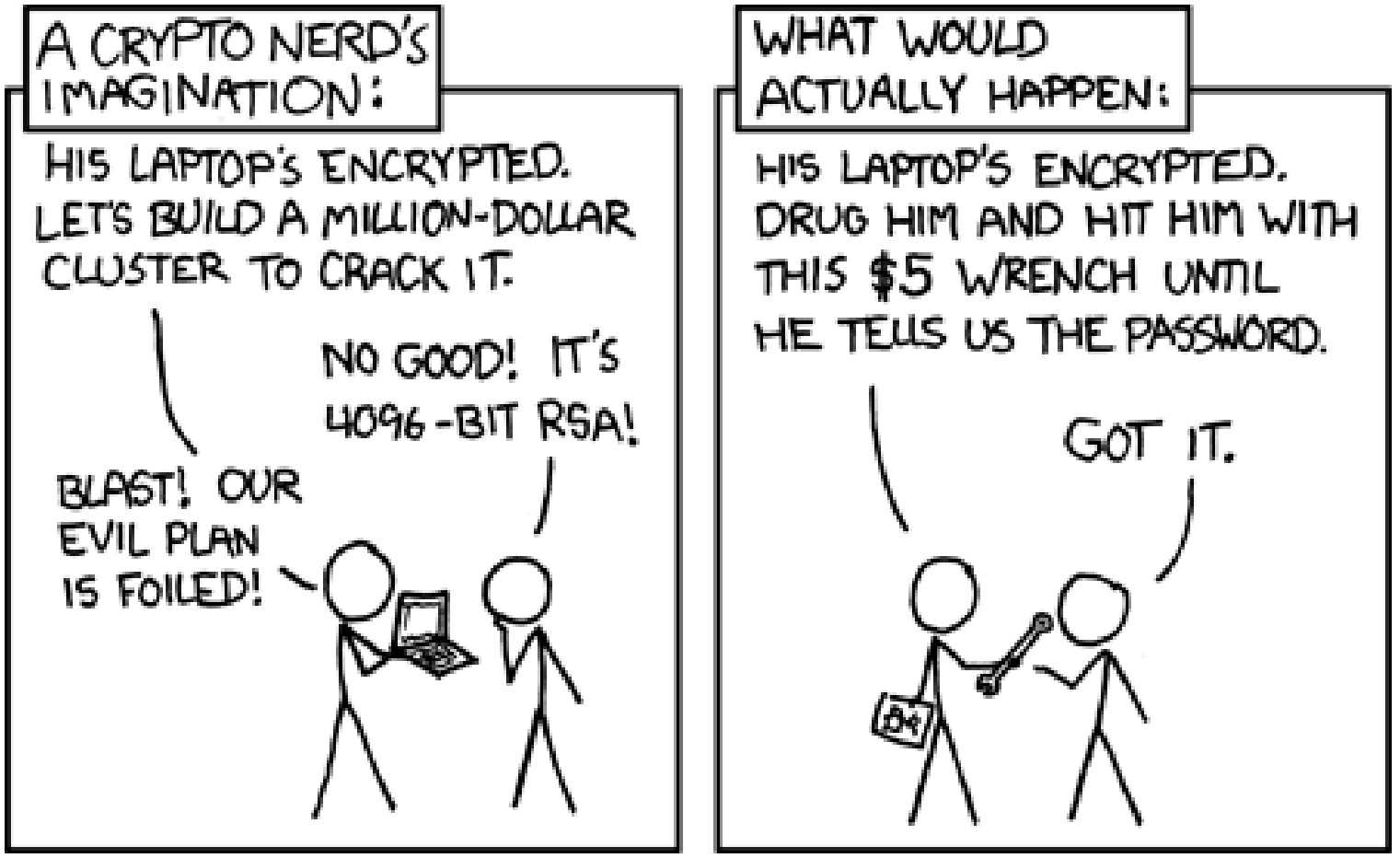
Cryptographie



Objectifs de ce cours

- Compréhension des **concepts centraux**
- Déploiement de **solutions fonctionnelles** dans la pratique informatique quotidienne (e-mailing, données, authentification de sources)
- (un peu) de **maths** et de théorie de l'information
- Mise en évidence de la faiblesse de certains algorithmes par le **crack** (approche expérimentale et exploratoire)

Pourquoi crypter

Assurer la sauvegarde et le partage du secret lorsque

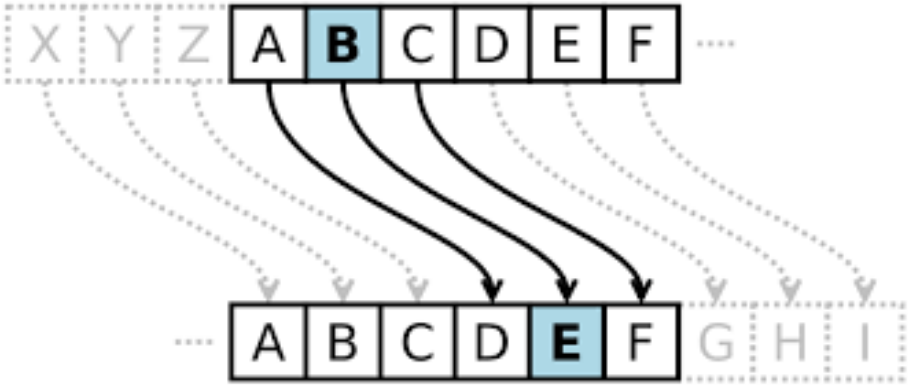
- l’identité du destinataire n’est pas indubitable
- Il existe des tiers dans la chaîne de communication
- Lorsqu’une source est accessible de l’extérieur

→ nous distinguerons ces différents usages par la suite

Commençons par un peu d’histoire

Cypher de César

- Décalage du rang des lettres d’un intervalle fixe



Initiation à la cryptographie

Exercice : décodez ce message (rang +3)

orqjwhpsv#mh#ph#vxlv#frxfkh#gh#erqqh#khxuh

Implémentation d'un décodage en python

```
e = 'orqjwhpsv#mh#ph#vxlv#frxfkh#gh#erqqh#khxuh'
e = [ord(x) for x in e]  # liste du rang des lettres
d = [x-3 for x in e]
d = [chr(x) for x in d]  # conversion du rang en lettres
d = ''.join(d)
```

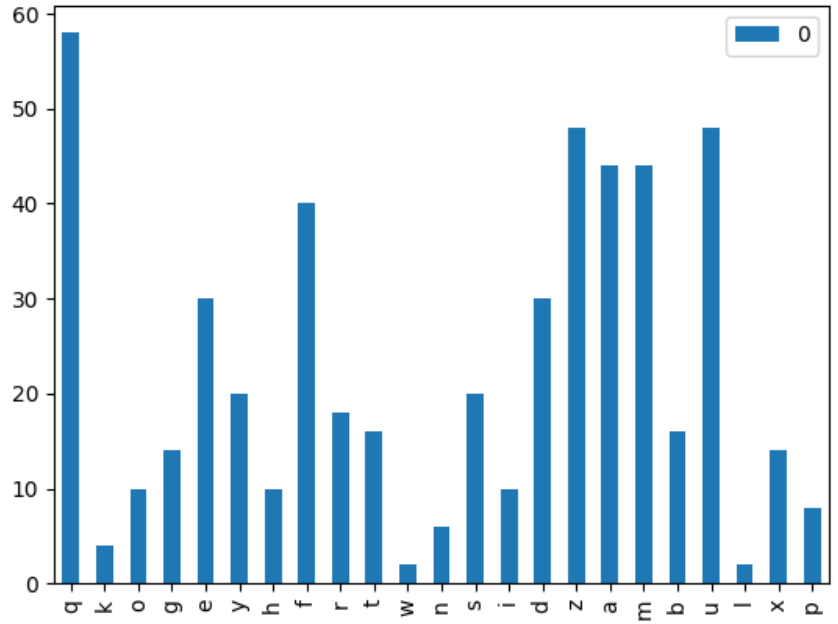
Initiation à la cryptographie

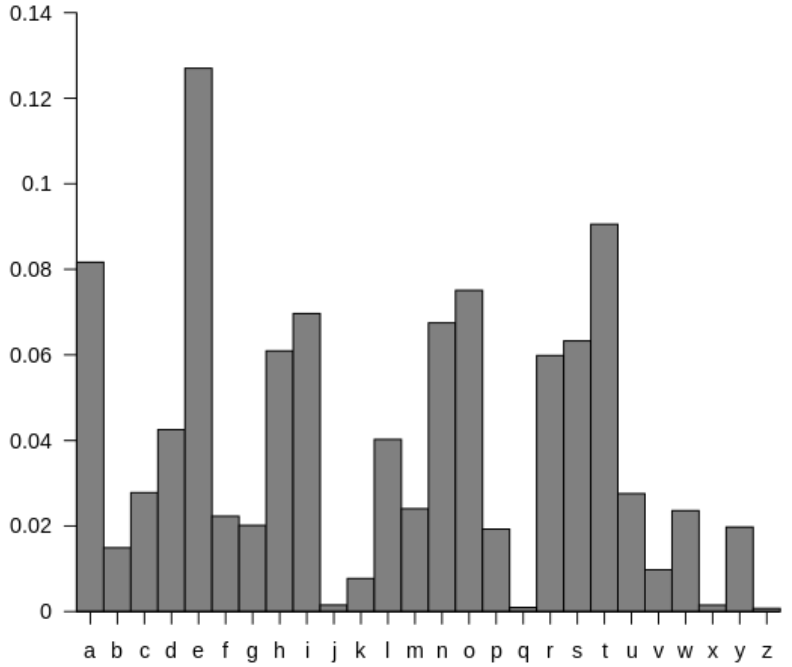
Craquez ce message :

fiayazftemrfqdoayuzsuzfaarruoqbduyqyuzuefqdymfqgelyadmi
uqowuarbaxmzpmzzagzoqpmeiqqbuzsdqetgrrxqarftqsahqdzy
qzfazfgqepmkmyahqeqqzmemzqrradffabgzuetbaxufuomxduh
mxeituxquybdahuzsftqsahqdzuzsbmdfkeuymsqmndampmeuf
qzsmsqeuzmndgueuzsnmffxqiuftaftqdqgdabqmzzmfuazefiaya
zftemrfqdoayuzsuzfaarruoqbduyqyuzuefqdymfqgelyadmiuqo
wuarbaxmzpmzzagzoqpmeiqqbuzsdqetgrrxqarftqsahqdzyqzf
azfgqepmkmyahqeqqzmemzqrradffabgzuetbaxufuomxduhmx
eituxquybdahuzsftqsahqdzuzsbmdfkeuymsqmndampmeufqzs
msqeuzmndgueuzsnmffxqiuftaftqdqgdabqmzzmfuaze

En fait, plus c’est long, plus c’est simple (si vous connaissez la langue encodée)

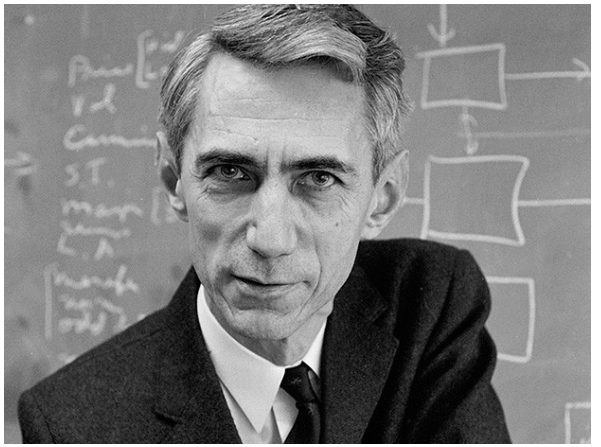
Initiation à la cryptographie





Il suffit de comparer la distribution des lettres du message encodé avec celle de la langue du message source

défaut d'uniformité = entropie = information



Prenons quelques minutes pour comprendre ce point

Un petit détour par les maths : dénombrement, signal et entropie

Commençons simple

quelle est la probabilité d’obtenir trois fois de suite pile ?

Quelle est la probabilité d’obtenir 2 pile et 3 face sur 5 lancers ?

Un petit détour par les maths : dénombrement, signal et entropie

Commençons simple

quelle est la probabilité d'obtenir trois fois de suite pile ?

- 1/2*1/2*1/2 = **1/8**

  Quelle est la probabilité d'obtenir 2 pile et 3 face sur 5 lancers ?

- On appelle cela un tirage avec remise (plus compliqué, passons)

Un petit détour par les maths : dénombrement, signal et entropie

Plus tricky maintenant

En mettant 1€ sur le vert, combien puis-je espérer gagner « en moyenne »?

(roulette française : 37 cases, cote 1→ 35)

Un petit détour par les maths : dénombrement, signal et entropie

Plus tricky maintenant

En mettant 1€ sur le (roulette française : 37 cases), combien puis-je espérer gagner « en moyenne »?

Reformulé en d'autres termes, cela revient à faire la somme des gains de chaque issue possible

E[X] = - 1 * 36/37 + 35 * 1/37 = -0.027027

On appelle cela l'utilité espérée

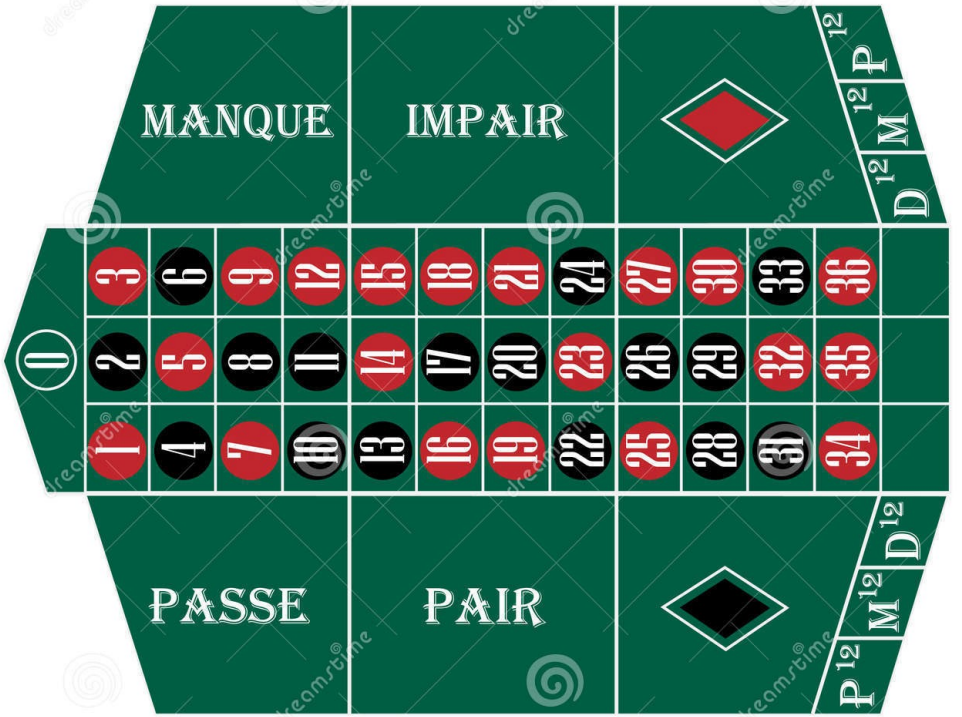
Utilité espérée, un peu d’histoire

- Jacques Bernoulli, *Ars Conjectandi* 1713, Bâle
- John von Neumann Oskar Morgenstern, *Theory of Games and Economic Behavior*, Princeton University Press, 1944

Un petit détour par les maths : dénombrement, signal et entropie

A l'aide des cotes calculez l'utilité espérée de

- 1€ sur tous les pairs
- 1€ sur les rouges et 2€ sur la ligne 1
- 1€ sur la première douzaine et 1€ sur les noirs
- 1€ sur la deuxième douzaine et 1€ sur les noirs



Un petit détour par les maths : dénombrement, signal et entropie

A l'aide des cotes calculez l'utilité espérée de

- 1€ sur tous les pairs (cote 1)

  → -1*19/37 + 1*18/37 = -0.02702702

- 1€ sur les rouges et 2€ sur la ligne 1
- 1€ sur la première douzaine et 1€ sur les noirs
- 1€ sur la deuxième douzaine et 1€ sur les noirs

Un petit détour par les maths : dénombrement, signal et entropie

A l'aide des cotes calculez l'utilité espérée de

- 1€ sur tous les pairs (cote 1)

  → -1*19/37 + 1*18/37 = -0.02702702

- 1€ sur les rouges et 2€ sur la ligne 1 (cote 1 et 2)

  → -3*12/37+(1-2) * 12/37+(4-1)*6/37 +(4+1)*6/37 =0

- 1€ sur la première douzaine et 1€ sur les noirs
- 1€ sur la deuxième douzaine et 1€ sur les noirs

Un petit détour par les maths : dénombrement, signal et entropie

A l'aide des cotes calculez l'utilité espérée de

- 1€ sur tous les pairs (cote 1)

  → -1*19/37 + 1*18/37 = -0.02702702

- 1€ sur les rouges et 2€ sur la ligne 1 (cote 1 et 2)

  → -3*12/37+(1-2) * 12/37+(4-1)*6/37 +(4+1)*6/37 =0

- 1€ sur la première douzaine et 1€ sur les noirs (cote 2 et 1)

  → -2*12/37+(1-1)*12/37+ (2-1)*6/37+(2+1)*6/37=0

- 1€ sur la deuxième douzaine et 1€ sur les noirs

Un petit détour par les maths : dénombrement, signal et entropie

Pourquoi ce détour par les probas ?

Parce que la définition formelle de l'I**nformation** repose sur l'**utilité espérée**

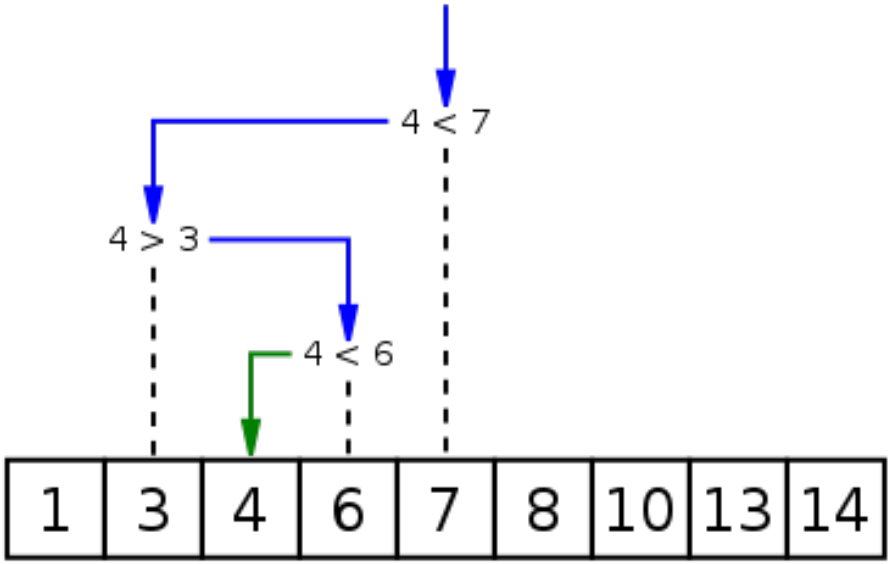
Claude Shannon, *A Mathematical Theory of Communication*, 1948

$$H_b(X) = -\mathbb{E}[\log_b P(X)] = \sum_{i=1}^{n} P_i \log_b \left(\frac{1}{P_i}\right) = -\sum_{i=1}^{n} P_i \log_b P_i.$$

Un petit détour par les maths : dénombrement, signal et entropie

Un illustration intuitive (base 2 = mesure en bits)

Combien de questions oui/non avez vous besoin de poser pour trouver un nombre compris de de 1 à 14 ?



Un petit détour par les maths : dénombrement, signal et entropie

1 jet de pièce
1 jets de dés
2 jets de dés
une clé hexa-décimale de 8 caractères produite aléatoirement

Un petit détour par les maths : dénombrement, signal et entropie

1 jet de pièce (b = 2)
→ -[1/2* log2(1/2) + 1/2* log2(1/2)] = 1 bit
1 jets de dés
2 jets de dés
une clé hexa-décimale de 8 caractères produite aléatoirement

Un petit détour par les maths : dénombrement, signal et entropie

1 jet de pièce (b = 2)
→ -[1/2* log2(1/2) + 1/2* log2(1/2)] = 1 bit
1 jets de dés
- -[1/6* log2(1/6) + ….+ 1/6* log2(1/6)] = 2,5849
2 jets de dés (NB:événements indépendants)
une clé hexa-décimale de 8 caractères produite aléatoirement

Un petit détour par les maths : dénombrement, signal et entropie

1 jet de pièce (b = 2)
→ -[1/2* log2(1/2) + 1/2* log2(1/2)] = 1 bit
1 jets de dés
→ -[1/6* log2(1/6) + ….+ 1/6* log2(1/6)] = 2,5849
2 jets de dés (NB:événements indépendants)
→ 2*H(1 jet de dés) = -5,1699
une clé hexa-décimale de 8 caractères produite aléatoirement

Un petit détour par les maths : dénombrement, signal et entropie

Maintenant, imaginez un pièce faussée qui renvoie **pile** à tous les coups ?

…. qui renvoie pile dans 75 % des cas. Comparez :

- Pile-Pile-Pile-Pile
- Pile-Pile-face-Pile

NB : nous nous plaçons dans un cas de figure où nous connaissons l'issue des tirages (n=1)

Un petit détour par les maths : dénombrement, signal et entropie

Maintenant, imaginez un pièce faussée qui renvoie **pile** à tous les coups ?

…. qui renvoie pile dans 75 % des cas. Comparez :

- Pile-Pile-Pile-Pile

  0,3112+0,3112+0,3112+0,3112 = 1,2451

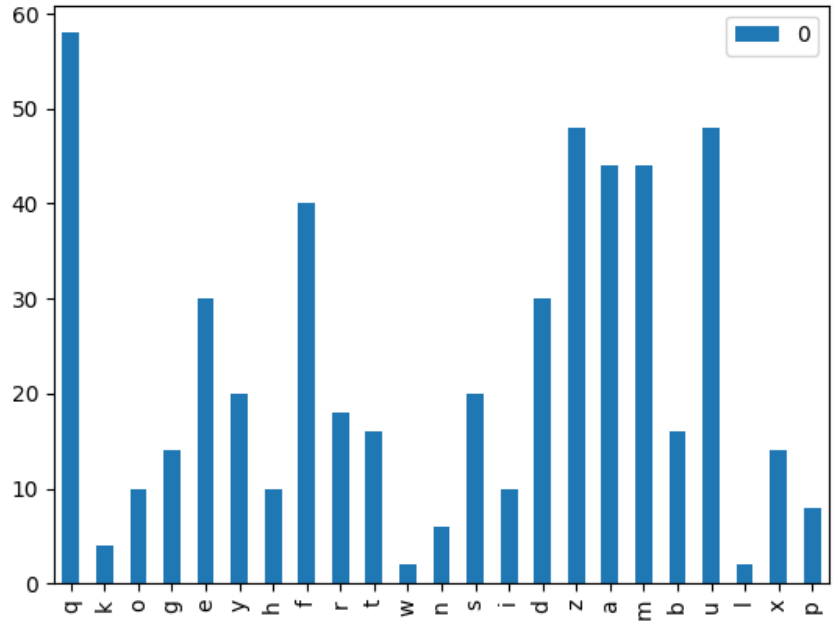
- Pile-Pile-face-Pile

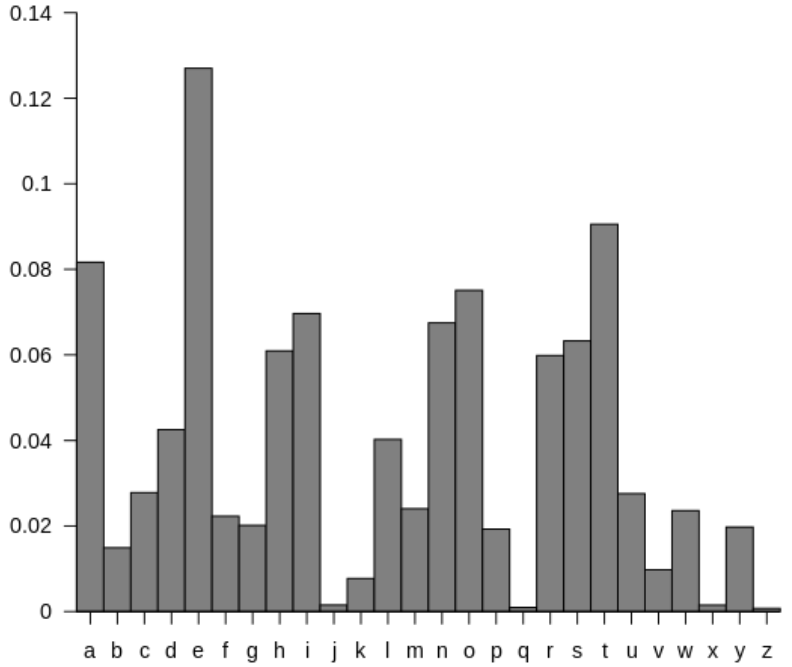
  0,3112+0,3112+0,5+0,3112 = 1,4338

Cela nous apprend que l'information est intrisèquement lié à

- l'absence d'uniformité
- l'entropie

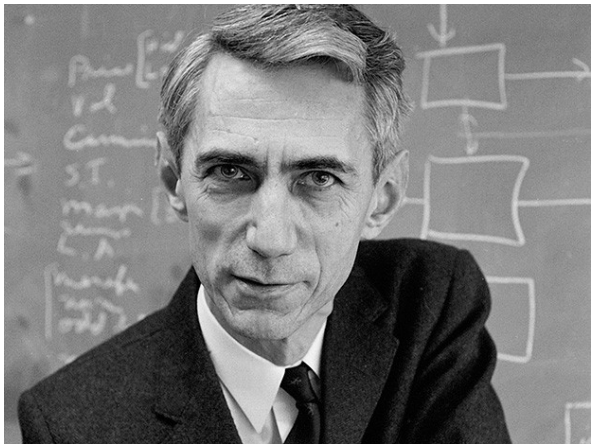
Revenons au problème de César





Il suffit de comparer la distribution des lettres du message encodé avec celle de la langue du message source

défaut d'uniformité = entropie = information



Comment fait-on, dans ce cas ?

Un bon algorithme de cryptographie est un algorithme qui **uniformise** (randomise) le message encodé

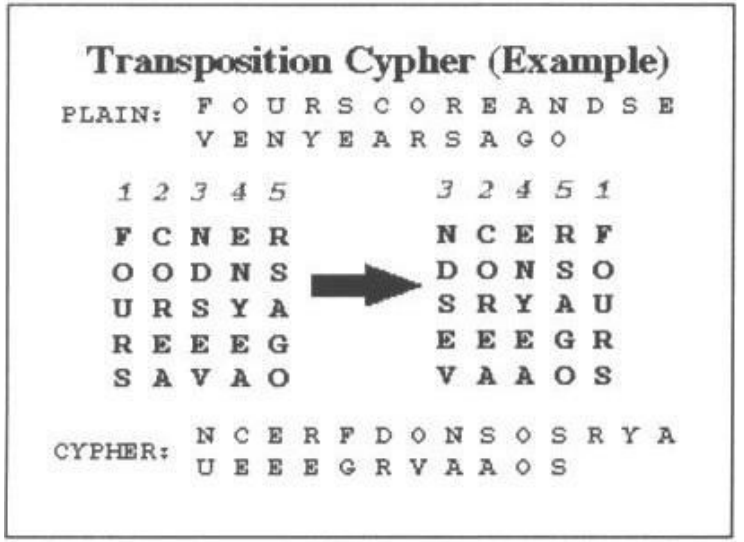
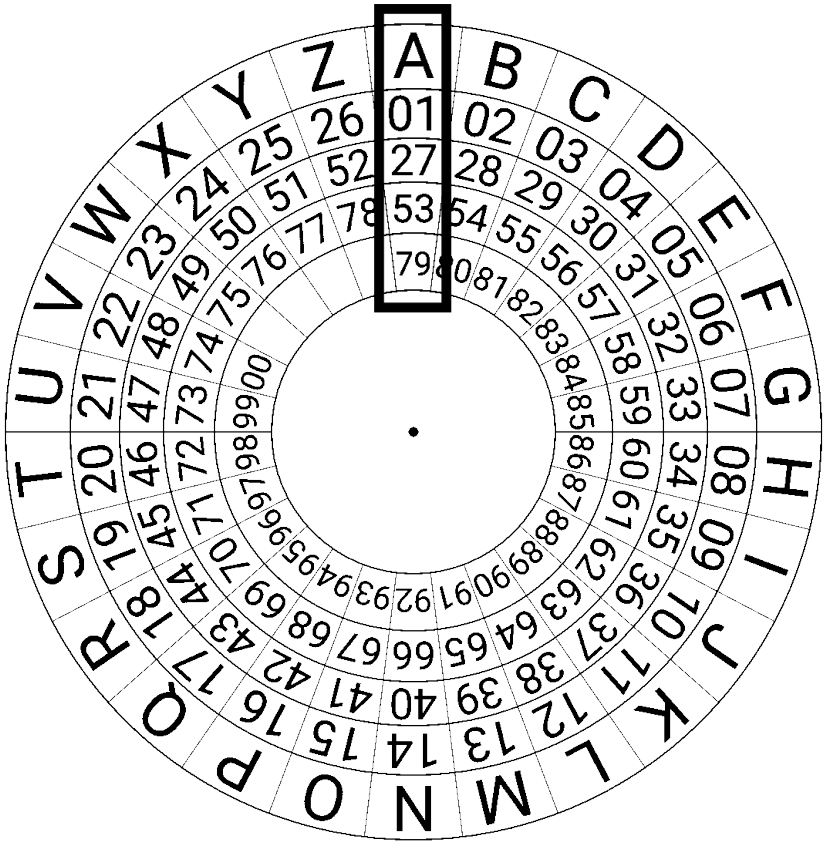
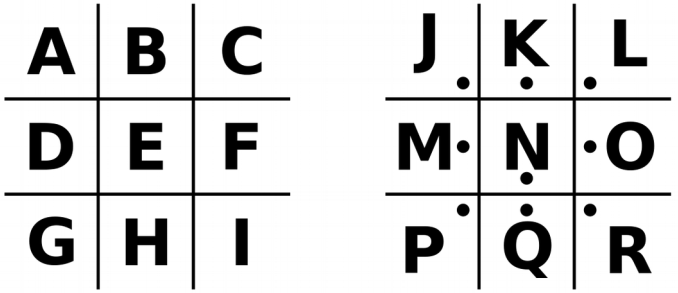
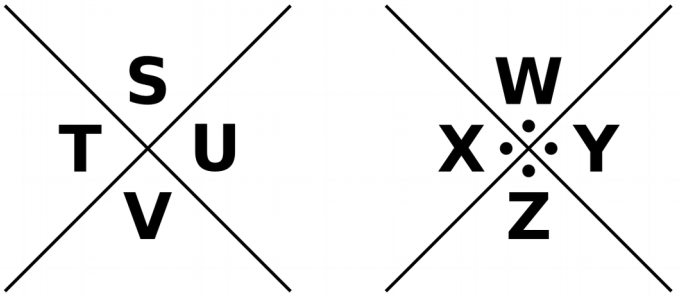


Figure 21. Example of a basic transposi... download...







Comment fait-on, dans ce cas ?

Ces systèmes récursifs ou spatialisés s'apparentent à des **proto-algorithmes**

En soi, entre ces systèmes et nos algorithmes de cryptographie modernes, la différence tient surtout :

- Au fait d'être implémentés électroniquement
- À leur degré de complexité (**différence de degré** plus qu'une **différence de nature**)

Implémentez un algorithme de transposition à clé

m	o	n		m	e	s	s	a	g	e
m	a		c	l	e	m	a		c	l
y	o	m	b	x	i	d	s		i	p

NB : ‘ ‘ est aussi un caractère

(12 + 12) % 27 = 24 = y

(14 + 0) % 27 = 14 = 0

(13+ 26)%27 = 12 = m

Remarques historiques

- Origines byzantines, redécouvert par Blaise de Vigenère au 16ème
- Guerre de Sécession et campagnes Napoléoniennes

Implémentez un algorithme de transposition à clé

Exercice 2

- A l’aide de votre programme, assurez vous de la (relative) uniformité d’un long message encodé

Passons à l’age moderne

Sous le nom ‘cryptage’ se cache divers types d’usages/algos

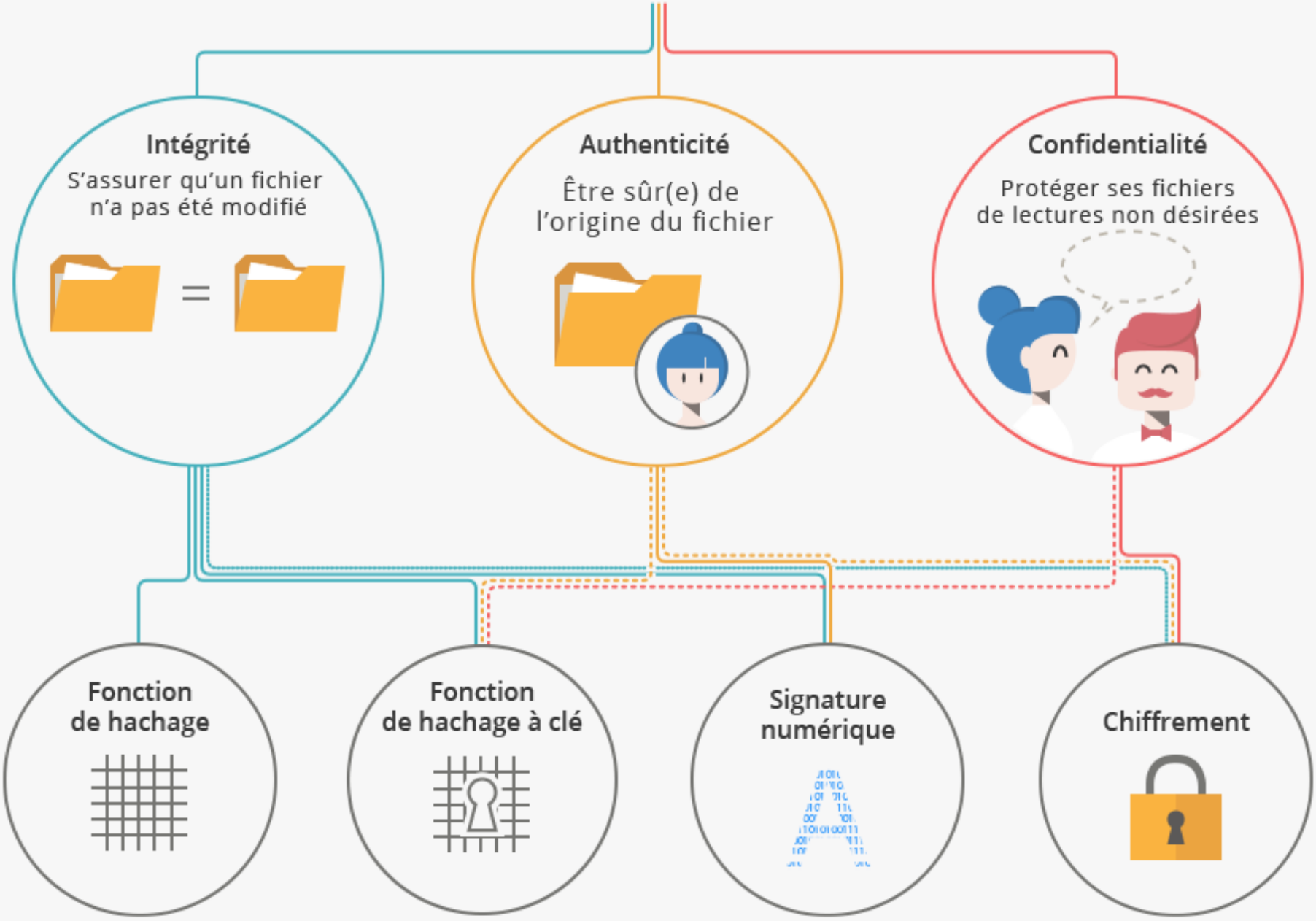
Algos « réversibles »

- Encryption, chiffrement
- Confidentialité

Algos « irréversibles »

- Fonction de Hash, signature numérique
- Vérification de l’intégrité d’un fichier, de l’identité de son expéditeur

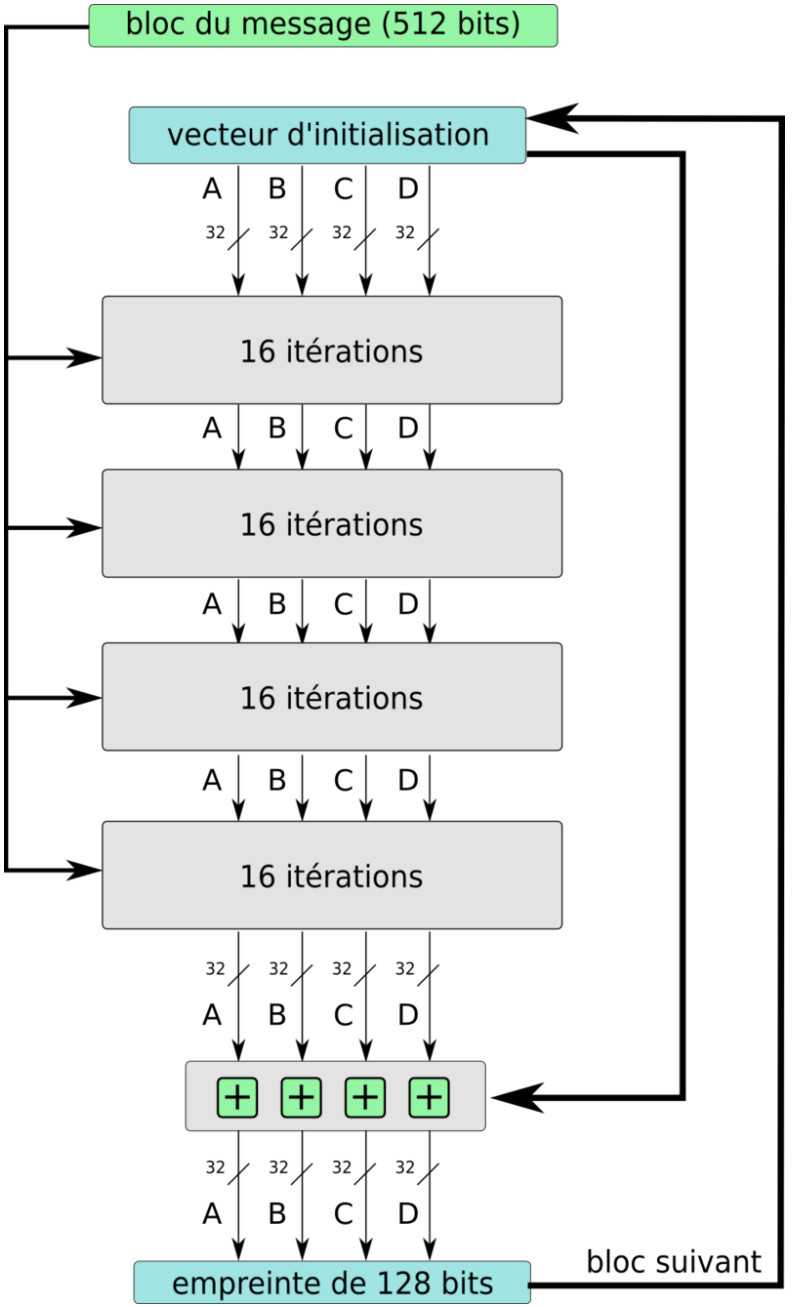
Les usages de la CRYPTOGRAPHIE



Commençons par les fonctions de hash

Md5

- Empreinte numérique:128bits → 32 caractères
- Plus considéré comme sur
- Encore pas mal utilisé pour s'assurer de l'intégrité d'une source logiciel
- **Mdp sur BDD SQL** (app web)



Exercices

Quelle est la probabilité pour d'un programme quelconque ait le même hash md5 que votre programme ?

Assurez vous de l'authenticité de **cette ISO**

une aide

Hashez les deux textes suivants

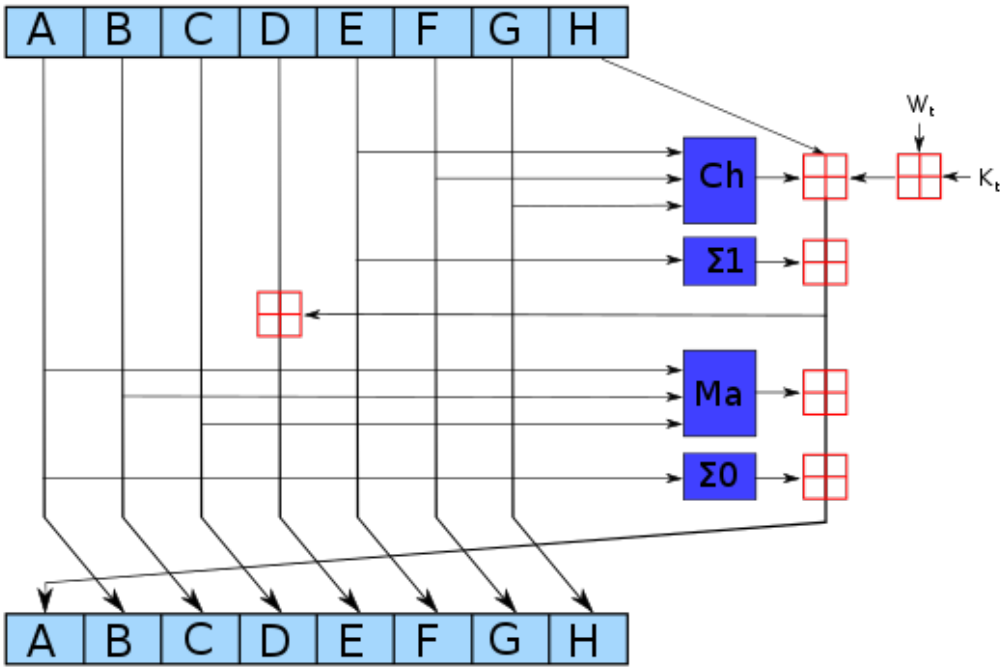
- Longtemps, je le suis couché de bonne heure
- Longtemps, je le suis couché de bonne **H**eure

A l'aide d'Hashcat, craquez ce mdp hashé

- 2c9341ca4cf3d87b9e4eb905d6a3ec45

une aide

Plus solide, SHA 256



A, B,C : mots de 32 bits (SHA 256)

\+ addition modulo 2**32

Ch Ma, fonction bits à bits non linéaires

Sigma 1 et Sigma 2 : décallage circulaire et XOR

kt : constante dépendant du numéro de tour

- Wt : mot (32 bits) dépendant du numéro de tour

Exercices

Générez la signature SHA 256 d’un fichier ou d’une string de votre choix (ligne de commande ou app, à votre guise)

(Essayez) de craquer ce mot de passe hashé :

75c5bb41bed00043c50d5bec3ed60a77a2ef77f0731b71dfefa187a639268785

Le chiffrement par l'exemple

Assurer la confidentialité des données transmises à un serveur HTTPS : **TLS/SSL**

Attention à certains contre-sens :

- TLS et SSL ne sont pas de **algorithmes de chiffrement** mais des **protocoles** qui norment la manière dont client et serveur recours à des algorithmes spécifiques
- Il en existe **plusieurs versions** → différents algorithmes
- Plusieurs étapes durant le **handshake** → plusieurs algorithmes

Une distinction préalable

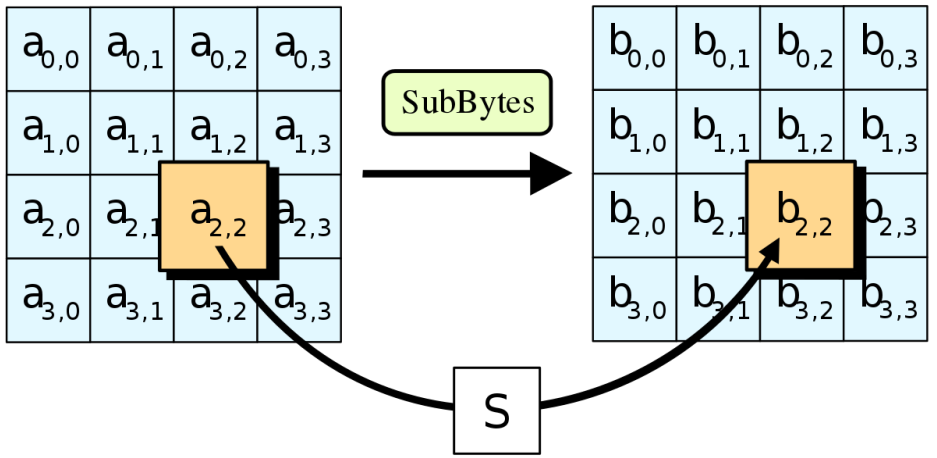
Chiffrement symétrique

- Une seule clé crypte et décrypte
- eg. AES 256
- Comment envoyer la clé au destinataire pour qu'il puisse décrypter ?
- Rapide

Chiffrement asymétrique

- Clé publique VS clé privé
- eg. RSA
- Très solide + identité unique
- Très énergivore

AES 256



Principes généraux

- Substitution / permutation matricielle
- Taille de block:128 bits
- Longueur de la clé:256 bits

Incassable au jour d'aujourd'hui (sauf par force brute)

standard moderne de chiffrement

« Oui mais pour déchiffrer, il faut avoir la clé »

Il faut donc que A la transmettre à B

(avec le risque d'interception que cela représente)

Il faut donc la transmettre :

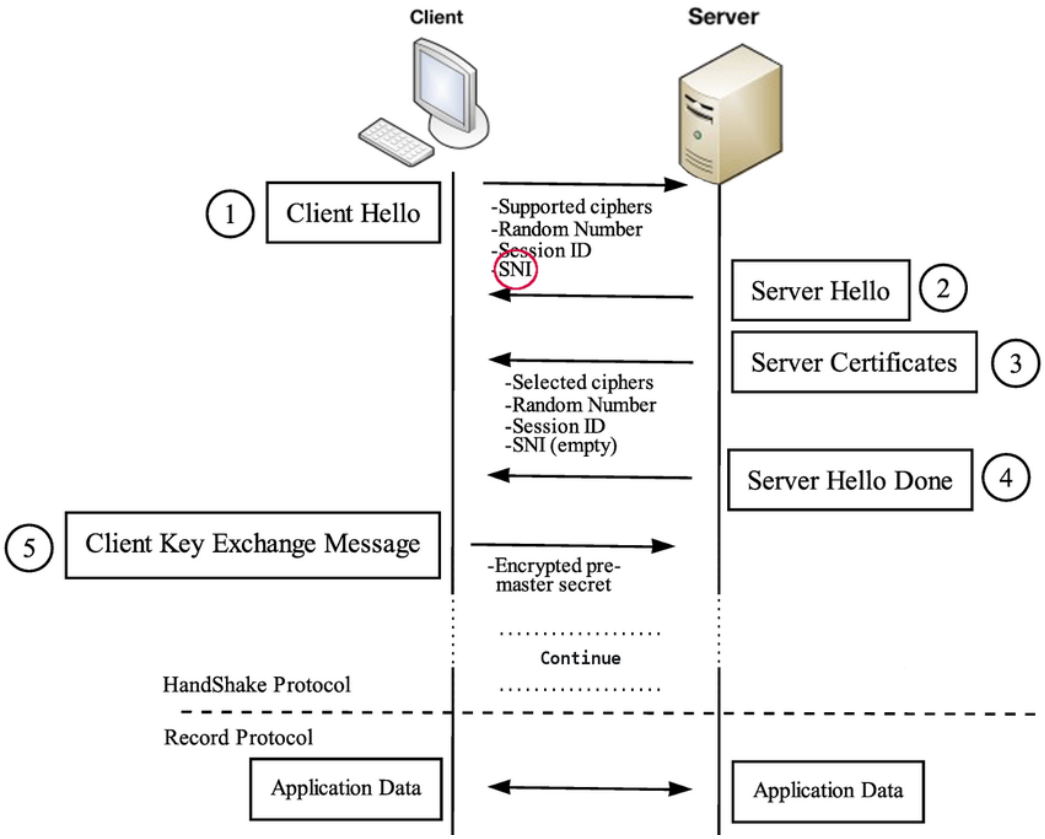
- De manière à ce qu'elle ne puisse pas être **interceptée**
- En s'assurant de l'**identité** i) de l'émetteur et ii) du récepteur

Voilà la fonction du **TLS Handshake**

Détails du TLS Handshake

Étape 1

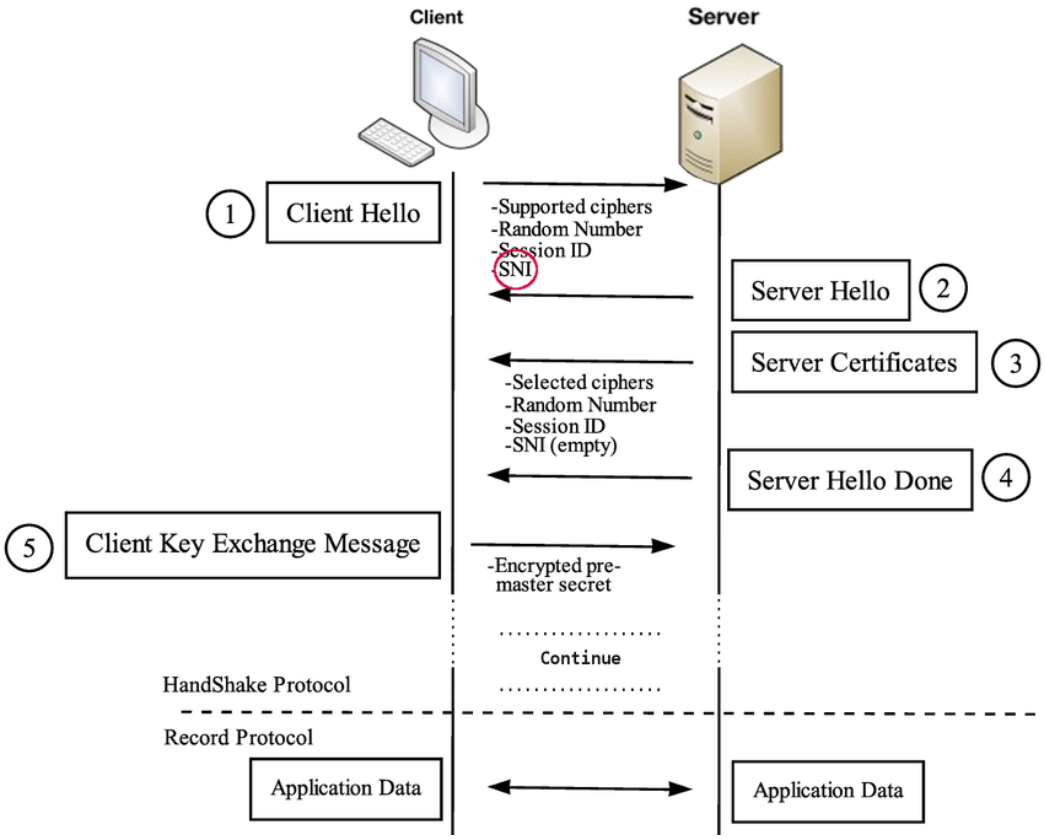
« salut Serveur, moi c’est Client. Je peux encrypter dans telle et telle langue. Voici un **nombre** random et l’**ID** de notre conversation (session). Voici le nom que je vais te donner (SNI) »



Détails du TLS Handshake

Étape 2-3

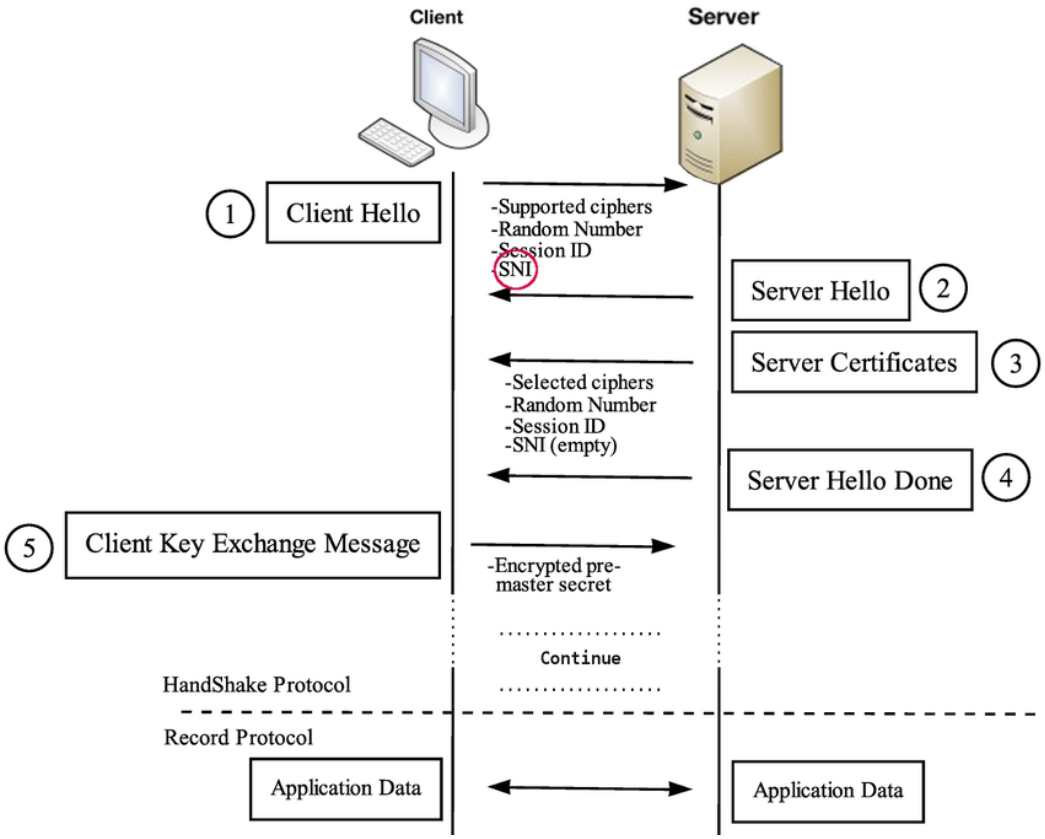
« Salut Client, bah écoute, je peux te parler dans telle langue. Tiens, vérifie que le **nombre** et l'**ID** que je te renvois correspondent à ce que tu m'as transmis »



Détails du TLS Handshake

Étape 4

« c’est bon, j’ai terminé »

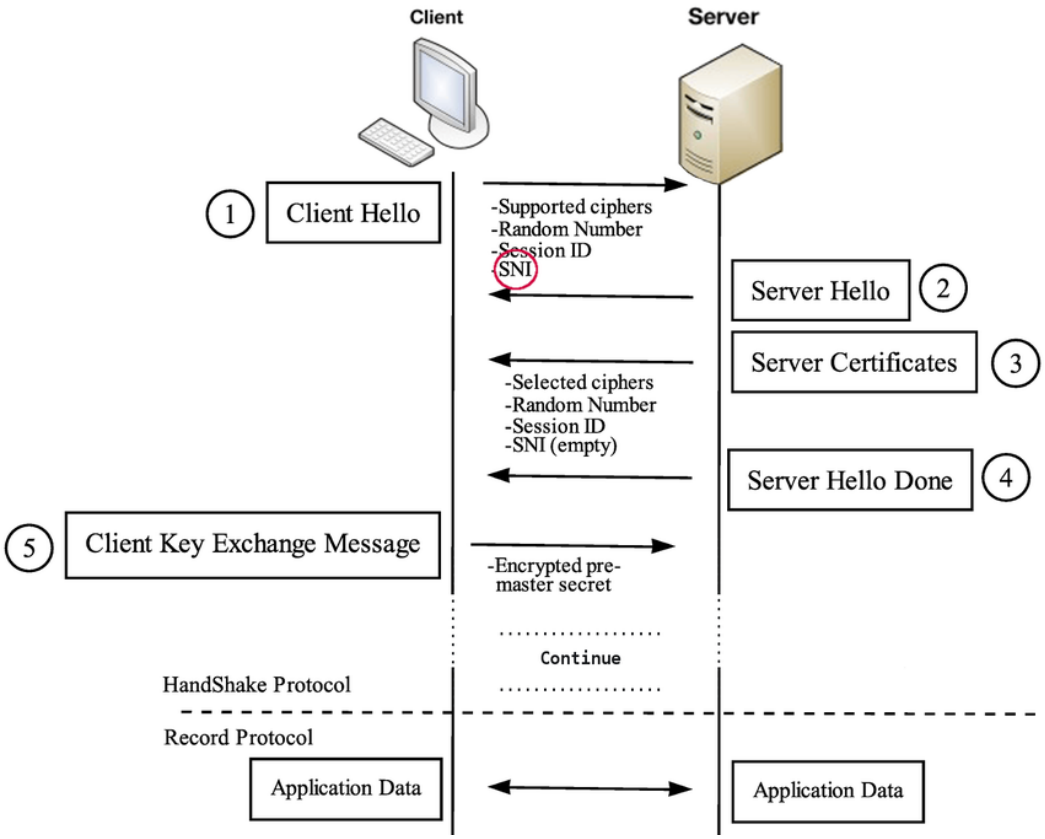


Détails du TLS Handshake

Étape 5

« Tiens, voilà ma clé » (encryptée avec la clé publique du serveur)

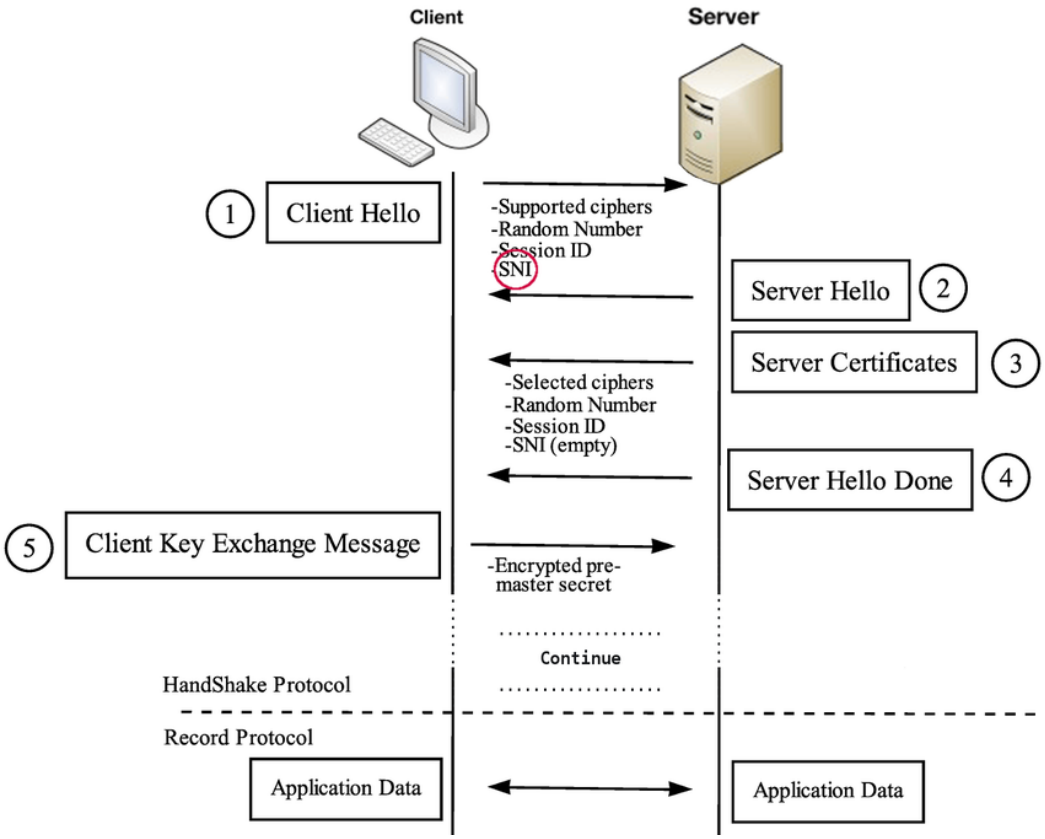
→ c'est ici que le **chiffrement asymétrique** intervient



Détails du TLS Handshake

Étapes optionnelles

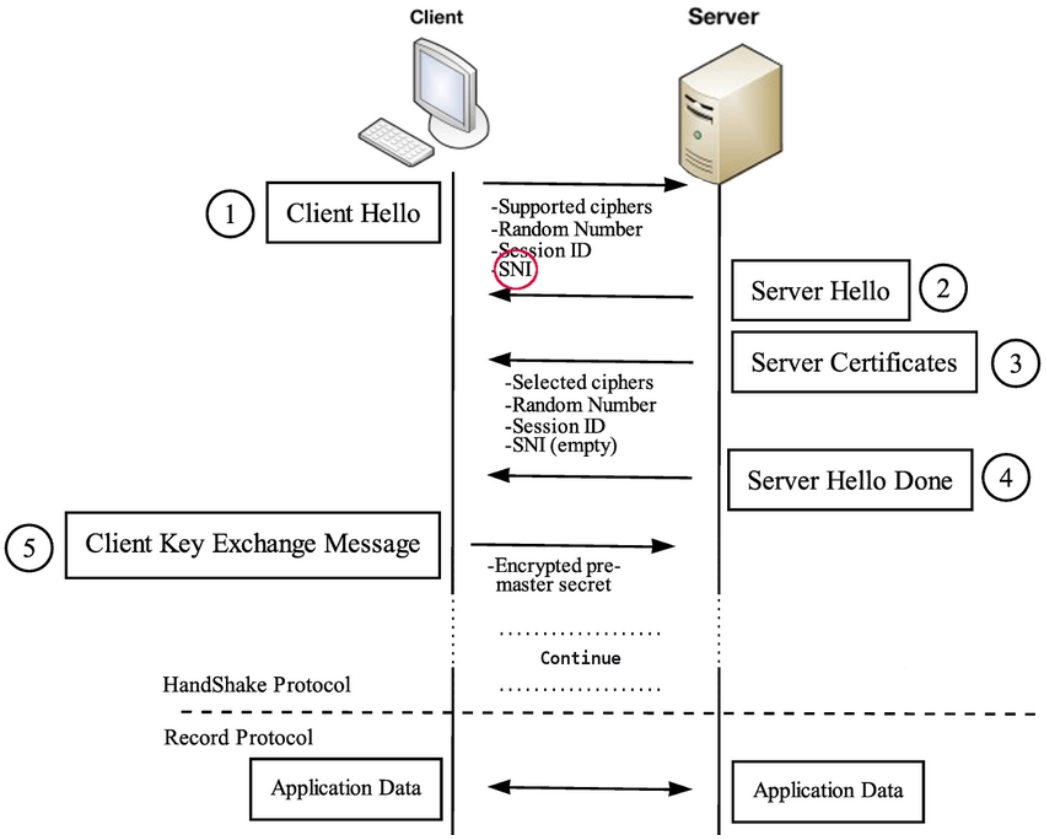
variables selon la version



Détails du TLS Handshake

La communication chiffrée en AES 256 commence

chiffrement symétrique



Atelier pratique

Nous allons mettre en place un protocole de chiffrement de point à point pour encrypter nos mails.

L’idée : rendre le contenu illisible au

- FAI
- Serveur mail
- Serveur mail du destinataire

Nous allons nous servir d’openPGP (Pretty Good Privacy)

dans sa version asymétrique