CHAPITRE 4 : MÉTHODES DE PROTECTION DES DONNÉES

MÉTHODES DE PROTECTION DES DONNÉES

CRYPTOGRAPHIE

- Chiffrement
- Hachage
- Signature numérique

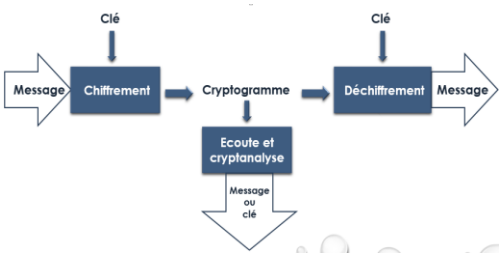
CRYPTOGRAPHIE - INTRODUCTION

- la cryptographie (**KRYPTOS** = CACHÉ, **GRAPHOS** = ECRITURE) est la science qui étudie le chiffrement et le déchiffrement de messages.
- Son but est d'échanger des informations de façon sécurisée.

Cryptographie : Etude des algorithmes et méthodes permettant d'envoyer des données de manière confidentielle.

Cryptanalyse : Technique qui permet de retrouver le message clair À partir d'un texte chiffré.

Crypto-système : triplet (algorithmes de **chiffrement**, algorithmes de **déchiffrement**, l'ensemble de **clefs** utilisées).



CRYPTOGRAPHIE - INTRODUCTION

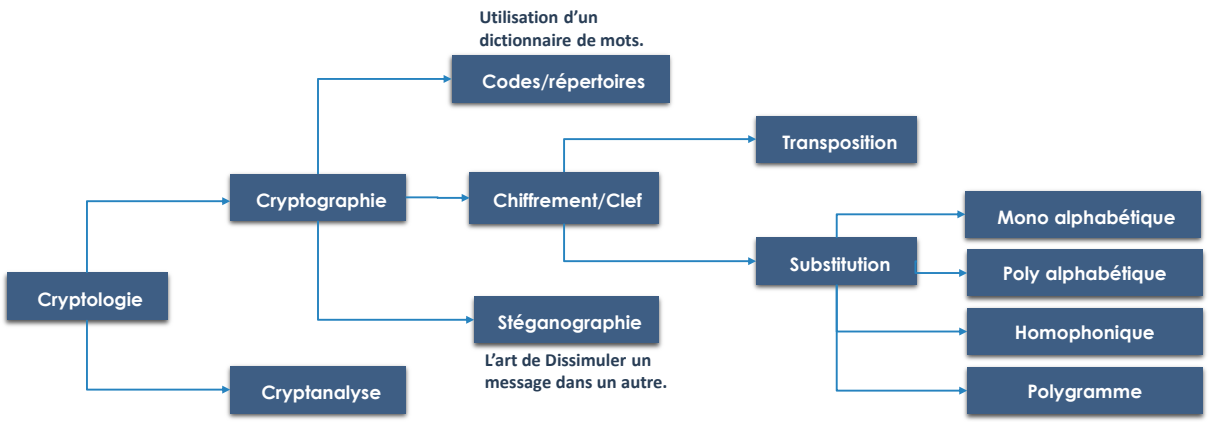
Il y a essentiellement deux types de cryptographie :

- La cryptographie à clé secrète ou cryptographie symétrique. C'est la plus ancienne.
- La cryptographie à clé publique ou cryptographie asymétrique. C'est la plus récente : on considère généralement qu'elle est née en 1976 avec l'article de Diffie et Hellman : "New directions in cryptography".
- Le chiffrement symétrique est beaucoup plus rapide que le chiffrement asymétrique mais a l'inconvénient de nécessiter le partage au préalable d'une clé secrète.
- En pratique, on utilise d'abord un chiffrement asymétrique pour échanger la clé secrète et ensuite un chiffrement symétrique pour l'échange des données.

CRYPTOGRAPHIE - INTRODUCTION

- **Chiffrement** : transforme une donnée afin de la rendre inintelligible par une personne autre que le destinataire.
- **Déchiffrement** : permet d'obtenir la version originale d'un message précédemment chiffré (connaissant l'algorithme de chiffrement et les clefs).
- **Cryptogramme** : c'est le texte chiffré, le résultat du chiffrement d'un texte clair.
- **Clef** : paramètre indispensable à l'application des opérations de chiffrement - déchiffrement.
- **Décrypter/casser le code** : récupérer le texte en clair ou la clef sans disposer des paramètres nécessaires.

CRYPTOGRAPHIE - INTRODUCTION



CRYPTOGRAPHIE - INTRODUCTION

STÉGANOGRAPHIE

l'art de la dissimulation ou cacher un message dans un autre message.

❑Encre invisible

Écrire avec du lait ou du jus de citron, ensuite chauffer le support sur lequel on écrit pour faire réapparaître le message (1er siècle av. J.C). À l'heure, D'autres substances sont utilisées (aspirine ou pyramidon).

❑Crâne rasé

On tatouait des messages dans la tête des esclaves. Ils faisaient un long voyage avant d'arriver chez le destinataire qui rasait leurs têtes pour lire le message.

❑Micro point

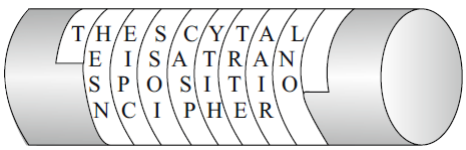
Dissimuler du texte ou une image dans un point de ponctuation d'une lettre. Il suffisait alors d'agrandir le point pour voir le message.

CRYPTOGRAPHIE - INTRODUCTION

CHIFFREMENT PAR TRANSPOSITION – MÉTHODE ASSYRIENNE

Les assyriens ont utilisé un outil appelé la Scytale, un cylindre sur lequel on enroulait le papyrus avant d'écrire le message.

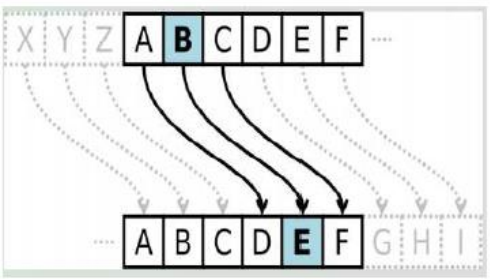
Le diamètre de la Scytale représentait la clé de chiffrement.

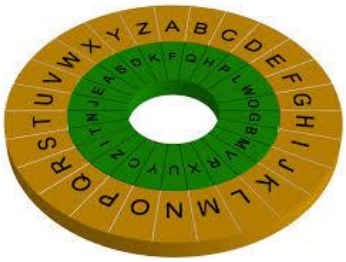


CRYPTOGRAPHIE - INTRODUCTION

CHIFFREMENT DE CÉSAR

Consiste à décaler chaque lettre de l'alphabet un nombre précis de fois. Le message chiffré est généré par la correspondance des deux alphabet.





Décalage circulaire

CRYPTOGRAPHIE - INTRODUCTION

CHIFFREMENT DE CÉSAR

Exemple :

Ici le décalage est réalisé 3 fois.

Le message '**BONJOUR**' donnera '**ERQMRXU**'.

La clé de cryptage est le nombre de décalage.

CRYPTOGRAPHIE - INTRODUCTION

CHIFFREMENT DE CÉSAR

La fonction de chiffrement de décalage k=4

$$C_k : \begin{cases} Z/26Z \rightarrow Z/26Z \\ x \rightarrow x+4 \end{cases}$$

La fonction de déchiffrement de décalage k est

$$D_k : \begin{cases} Z/26Z \rightarrow Z/26Z \\ x \rightarrow x-4 \end{cases}$$

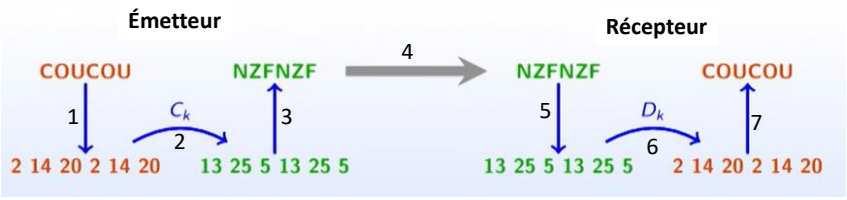
D_k est la bijection réciproque de $C_k \rightarrow D_k(C_k(x)) = x$

CRYPTOGRAPHIE - INTRODUCTION

CHIFFREMENT DE CÉSAR

Principe de chiffrement

Ils se mettent sur une clé secrète k (par exemple k=11)



CRYPTOGRAPHIE - INTRODUCTION

CHIFFREMENT DE CÉSAR

- Il y a 26 fonctions de chiffrements C_k différentes, k=0, 1, 2,…,25
- Les fonctions C_{29} et C_3 sont identiques
- Sécurité faible

CRYPTOGRAPHIE - INTRODUCTION

CHIFFREMENT PAR SUBSTITUTION

On associe à chaque lettre une autre lettre

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

ÊTRE OU NE PAS ÊTRE TELLE EST LA QUESTION

E → X T → G R → K

XGKX DR SX OFV GXWWX XVG WF ZRXVGPDS

Pour le décrypter on fait les substitutions inverses

Inconvénient :

Sécurité faible

CRYPTOGRAPHIE - INTRODUCTION

CHIFFREMENT PAR SUBSTITUTION

ESPACE DE CLÉS

Mathématiquement, choisir une clé de chiffrement revient à choisir une Bijection de :

E: {A, B,…,Z} → {A, B,…,Z}

Il y a 26! choix possibles

- Pour décrypter la lettre A il y a 26 choix
- Pour décrypter la lettre B il reste 25 choix
- …
- Pour Z une reste une seule possibilité

26x25x24x..x1 = 26! Choix de clés = $4x10^{26}$ clés

CRYPTOGRAPHIE - INTRODUCTION

CHIFFREMENT PAR SUBSTITUTION

ATTAQUE STATISTIQUE

Une même lettre est toujours cryptée de la même façon

E → X

Les lettres n'apparaissent pas avec la même fréquence

En français, les lettres les plus fréquents sont :

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

E	S	A	I	N	T	R	U	L
14.6%	8.0%	7.5%	7.1%	6.8%	6.8%	6.4%	6.1%	5.6%

CRYPTOGRAPHIE - INTRODUCTION

CHIFFREMENT PAR SUBSTITUTION

ATTAQUE STATISTIQUE

Dans le texte, on cherche la lettre qui apparait le plus souvent

C(E) – C(S) – C(A) – …

Par exemple : LHLZ HFQ BC HFFPZ WH YOUPFH MUPZH

On compte apparitions des lettres : H:6, F:4, P:3, Z:3

On suppose que H crypte E te F crypte S

→Le texte à trou suivant : *E** ES* ** ESS** *E ***SE ****E

Dans les statistiques P et Z devraient se décrypter en A et I (ou I et A)

- HFFPZ → ESSAI ou ESSIA
- En réfléchissant un peu -> CECI EST UN ESSAI DE PHRASE VRAIE

CRYPTOGRAPHIE - INTRODUCTION

MÉTHODE DE VIGENÈRE

Est une technique de chiffrement poly-alphabétique qui améliore le chiffrement de César en utilisant une clé sous forme de mot ou de phrase répétée tout au long du texte à chiffrer.

Principe

- On associe chaque lettre du texte clair avec une lettre de la clé.
- On applique un décalage correspondant à la position de la lettre de la clé dans l'alphabet.

Formellement, chaque lettre C_i du texte chiffré est obtenue par :

$$C_i = (P_i + K_i) \mod 26$$

où :

- P_i est la position de la lettre du texte clair (A = 0, B = 1, ..., Z = 25),
- K_i est la position de la lettre de la clé,
- C_i est la position de la lettre chiffrée.

CRYPTOGRAPHIE - INTRODUCTION

MÉTHODE DE VIGENÈRE

On regroupe les lettres par blocs de longueur k

Exemple :

K=4

CETTE PHRASE NE VAUT RIEN DIRE → CETT EPHR ASEN EVAU TRIE NDIR E

Une clé est constituée de k nombres de 0 à 25 (n1, n2, .. ,n_k)

→Décalage de n1 de la 1ère lettre de chaque bloc

→Décalage de n1 de la 2ème lettre de chaque bloc

→…

Exemple : considérons la clé (3,1,5,2) . CETT → FFYV

CRYPTOGRAPHIE - INTRODUCTION

MÉTHODE DE VIGENÈRE

L'élément de base est un Bloc

$C_{n_1,n_2,..,n_k}$:

- $Z/26Z \times \dots Z/26Z \rightarrow Z/26Z \times \dots Z/26Z$
- $(x_1,x_2,\dots,x_k) \rightarrow x_1+n_1,\ x_2+n_2,\ \dots\ x_k+n_k$

Chacun de ces composants est un chiffrement de César

La fonction de déchiffrement :

$C_{-n_1,-n_2,..,-n_k}$

CRYPTOGRAPHIE - INTRODUCTION

MÉTHODE DE VIGENÈRE

Fonction de chiffrement

Soit Σ un alphabet fini de cardinal q que l'on identifie à l'anneau Z/qZ et soient k et x deux mots de longueur n de Σ^*.

La *fonction de chiffrement de Vigenère* $e_k:\Sigma^n \rightarrow (Z/qZ)^n$ est définie par :

$$e_k(x) = x+k$$

CRYPTOGRAPHIE - INTRODUCTION

MÉTHODE DE VIGENÈRE

ESPACES DES CLÉS ET ATTAQUES

Pour une clé de longueur k, il y a 26^k choix possibles

Exemple : pour k=4 → 26^4 = 456 976 clés

Même faiblesse que le chiffrement par substitution

ALPH ABET → DMUJ DCJV : les deux A sont cryptés par la même lettre

Attaque :

- ❑ On découpe le texte en plusieurs listes
- ❑ Les premières lettres de chaque bloc, les deuxièmes lettres de chaque bloc…
- ❑ Attaque statistique sur chacun de ces regroupements
- ❑ La taille des blocs doit être petite devant la longueur du texte

CRYPTOGRAPHIE - INTRODUCTION

CRYPTANALYSE

1. *chiffré connu*, Oscar connaît des chiffrés y.
2. *clair connu*, Oscar connaît des couples (x,y).
3. *clair choisi*, Oscar peut chiffrer des clairs x de son choix (mais ne dispose pas de la clef secrète).
4. *chiffré choisi*, Oscar peut déchiffrer des chiffrés y (mais ne dispose pas de la clef secrète.)

CRYPTOGRAPHIE - INTRODUCTION

CRYPTANALYSE

La cryptanalyse du système de chiffrement de Vigenère est une attaque à chiffré connu

1. Estimation de la longueur de la clef : Méthode de Kasiski

- La première étape de la cryptanalyse de ce système est de déterminer la longueur *l* de la clef secrète. Pour cela on applique le *test de* ***Kasiski*** qui consiste à chercher des motifs identiques de longueur au moins 2 dans le texte chiffré y.
- Deux raisons peuvent expliquer plusieurs occurrences d'un même motif :
  - Une pure coïncidence.
  - Les motifs correspondant dans le message clair x sont identiques.

CRYPTOGRAPHIE - INTRODUCTION

CRYPTANALYSE

Exemple :

Considérons la clef k=SECRETS et le message clair suivant :

x=SUDESTRHODESAPPLIQUERCODESX

La table ci-suivante résume l'opération de chiffrement.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Message x	S	U	D	E	S	T	R	H	O	D	E	S	A	P	P	L	I	Q	U	E	R	C	O	D	E	S	X
Clef k	S	E	C	R	E	T	S	S	E	C	R	E	T	S	S	E	C	R	E	T	S	S	E	C	R	E	T

CRYPTOGRAPHIE - INTRODUCTION

CRYPTANALYSE

- On calcule donc le **PGCD** des distances entre les différentes occurrences du motif dans le message chiffré;
- Comme le motif DES ici aux positions 2, 9 et 23 , et que ces répétitions sont distantes d'un multiple de la longueur de la clef (ici $9-2=7$ et $23-2=21=3\times7$), il est évident qu'il y aura une répétition du motif chiffré, ici FVW.

  → La clé k=7

CRYPTOGRAPHIE - INTRODUCTION

CRYPTANALYSE

Si **l** désigne la longueur de la clef et i∈[0, l−1], alors tous les symboles du message chiffré y dont la position modulo **l** est égale à **i** ont été chiffrés avec le même symbole k_i de la clef k par un simple décalage circulaire (César)

CRYPTOGRAPHIE - INTRODUCTION

CRYPTANALYSE

L'indice de coïncidence : IC

- L'indice de Coïncidence (IC) est la probabilité que deux lettres choisies aléatoirement dans un texte soient identiques.
- Il fut inventé par William Friedman et publié en 1920, dans l'article "The Index of Coincidence and its Applications in Cryptography", Riverbank Publications Number 22.

$$\mathbf{IC} = \frac{\sum_{k=1}^{26} n_i(ni\ \_\ 1)}{\mathbf{N(N-1)}}$$

- **IC du français = 0.0746**
- **IC(M)=IC(C)**

CRYPTOGRAPHIE - INTRODUCTION

LIMITATIONS

- Pour pallier ses faiblesses, on peut utiliser une clé aussi longue que le message (One-Time Pad), rendant le chiffrement théoriquement incassable
- La méthode de Vigenère a été historiquement utilisée mais est aujourd'hui dépassée par des algorithmes modernes comme AES.

CRYPTOGRAPHIE - INTRODUCTION

Une liste non exhaustive d'algorithme de chiffrements symétriques :

I- Algorithmes de chiffrement par blocs

- **DES** (Data Encryption Standard) : Blocs de 64 bits, clé de 56 bits (désormais considéré comme obsolète).
- **3DES** (Triple DES) : Version améliorée de DES, appliquant trois passes de chiffrement.
- **AES** (Advanced Encryption Standard) : Standard actuel, utilisant des blocs de 128 bits avec des clés de 128, 192 ou 256 bits.
- **IDEA** (International Data Encryption Algorithm) : Blocs de 64 bits, clé de 128 bits, utilisé dans PGP.
- **RC5** : Algorithme à blocs avec tailles de clé et de bloc variables.
- **RC6** : Version améliorée de RC5, finaliste d'AES.
- …

CRYPTOGRAPHIE - INTRODUCTION

Une liste non exhaustive d'algorithme de chiffrements symétriques :

II- Algorithmes de chiffrement par flux

Les données sont chiffrées bit par bit ou octet par octet, souvent utilisés pour le chiffrement des communications en temps réel.

- **RC4 (Rivest Cipher 4)** : Chiffrement rapide utilisé dans SSL/TLS (désormais considéré comme vulnérable).
- **Salsa20** : Chiffrement rapide et sécurisé, utilisé dans des protocoles modernes.
- **ChaCha20** : Variante améliorée de Salsa20, utilisé dans TLS 1.3 et WireGuard.
- **A5/1 et A5/2** : Utilisés dans le chiffrement des communications GSM (considérés comme cassés).
- **Grain** : Algorithme léger pour les dispositifs contraints (IoT).
- **Trivium** : Algorithme de chiffrement par flux très efficace, souvent utilisé en cryptographie légère.
- …

CRYPTOGRAPHIE - INTRODUCTION

Une liste non exhaustive d'algorithme de chiffrements symétriques :

III- Algorithmes spécialisés et légers

Optimisés pour les environnements contraints (IoT, cartes à puce, etc.).

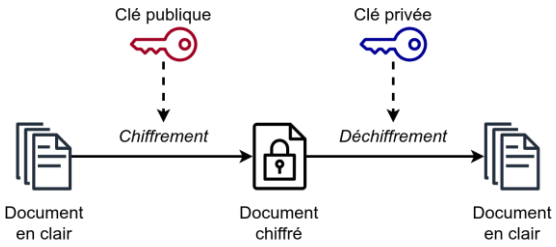
- **LEA (Lightweight Encryption Algorithm)** : Utilisé pour les systèmes embarqués.
- **PRESENT** : Algorithme ultra-léger pour des applications à faible puissance.
- **Simon & Speck** : Conçus par la NSA pour des applications à faible consommation d'énergie

CRYPTOGRAPHIE - INTRODUCTION

CHIFFREMENT ASYMÉTRIQUE

Les algorithmes asymétriques ont besoin de deux clés :

i. Une clé publique qui sert au chiffrement ou parfois aussi à la vérification de signature,
ii. Une clé privée qui sert au déchiffrement ou parfois aussi à la signature.



CRYPTOGRAPHIE - INTRODUCTION

CHIFFREMENT ASYMÉTRIQUE

Les plus connus d'entre eux :

- **RSA** (**R**ivest, **S**hamir et **A**dleman en 1977) : Il permet le chiffrement et la signature.
- Le protocole d'échanges de clés Diffie-Hellman (en 1976), protocole à l'origine de la cryptographie moderne.
- La cryptographie sur les courbes elliptiques (Koblitz et Miller en 1985) : ECC (Elliptic Curve Cryptography).
- L'algorithme de signature numérique DSA (Digital Signature Algorithm) proposé par le NIST (National Institute of Standards and Technology) en 1991.
- Et bien d'autres encore...

CRYPTOGRAPHIE - INTRODUCTION

NOTIONS MATHÉMATIQUES

ALGORITHME D'EUCLIDE

L'algorithme d'Euclide est une méthode efficace pour calculer le PGCD de deux nombres entiers.

L'algorithme repose sur la propriété mathématique suivante :

PGCD(a, b) = PGCD(b, a mod b)

- On remplace a par b, et b par a mod b, jusqu'à ce que b = 0.
- Le PGCD est alors la dernière valeur non nulle de a

EXEMPLE :

- $252 \bmod 105 = 42 \implies PGCD(252,105) = PGCD(105,42)$
- $105 \bmod 42 = 21 \implies PGCD(105,42) = PGCD(42,21)$
- $42 \bmod 21 = 0 \implies PGCD(42,21) = 21$

→ $PGCD(252,105) = 21$

CRYPTOGRAPHIE - INTRODUCTION

NOTIONS MATHÉMATIQUES

ALGORITHME D'EUCLIDE ÉTENDU

Une version améliorée de l'algorithme d'Euclide qui permet non seulement de calculer le PGCD de deux nombres a et b, mais aussi de trouver les coefficients x et y tels que :

$$ax+by=PGCD(a,b)$$

Ces coefficients x et y sont appelés **coefficients de Bézout,** et ils sont très utiles en cryptographie RSA pour calculer l'inverse modulaire.

CRYPTOGRAPHIE - INTRODUCTION

NOTIONS MATHÉMATIQUES

EXPONENTIATIONS MODULAIRES

- L'**exponentiation modulaire** (ou powmod, ou modpow) est un calcul sur des nombres entiers composé d'une puissance suivie d'un modulo.
- Ce type de calcul est très utilisé en cryptographie moderne comme RSA ou Diffie-Hellman.
- Consiste à calculer :

$$C=A^B \bmod M$$

Avec :
- **A** : la base
- **B** : l'exposant
- **M** : le modulo
- **C** : le résultat.

CRYPTOGRAPHIE - INTRODUCTION

NOTIONS MATHÉMATIQUES

EXPONENTIATIONS MODULAIRES - ALGORITHME D'EXPONENTIATION MODULAIRE RAPIDE

i. On utilise la **décomposition binaire** de B pour réduire le nombre de multiplications

ii. On applique la récurrence suivante :

- B pair : $C = A^B \bmod M = (A^{B/2} \bmod M)^2 \bmod M$
- B impair : $C = A^B \bmod M = (A^{(B-1)} \bmod M) \times A \bmod M$

CRYPTOGRAPHIE - INTRODUCTION

NOTIONS MATHÉMATIQUES

EXPONENTIATIONS MODULAIRES - ALGORITHME D'EXPONENTIATION MODULAIRE RAPIDE

Exemple

$$c = 3^{13} \bmod 7$$

a. Décomposition binaire de 13 :

$13 = 1101_2 \rightarrow 3^{13} = 3^8 \times 3^4 \times 3^1$

b. Calculs successifs avec réduction modulo 7 :

$3^1 = 3 \bmod 7$

$3^2 = 9 = 2 \bmod 7$

$3^4 = (3^2)^2 = 2^2 = 4 \bmod 7$

$3^8 = (3^4)^2 = 4^2 = 16 = 2 \bmod 7$

$3^{13} = 3^8 \times 3^4 \times 3^1\ 4^2 = (2 \times 4 \times 3) \bmod 7 = 24 = 3 \bmod 7$

Donc $3^{13} \bmod 7 = 3$

CRYPTOGRAPHIE - INTRODUCTION

CHIFFREMENT RSA

- Calcul de la clé publique et de la clé privée
- Chiffrement du message
- Déchiffrement du message
- Attaques possibles sur RSA et contre-mesures

CRYPTOGRAPHIE - INTRODUCTION

CHIFFREMENT RSA

PRINCIPE

CRYPTOGRAPHIE - INTRODUCTION

PRINCIPE DU CHIFFREMENT RSA

RSA repose sur trois étapes :

i. **Génération des clés**

- Choisir deux nombres premiers p et q
- Calculer $n = p \times q$ (modulo public).
- Calculer $\phi(n) = (p-1) \times (q-1)$
- Choisir un exposant e tel que $1 < e < \phi(n)$ et e est premier avec $\phi(n)$.
- Calculer d tel que $d \times e = 1 \bmod \phi(n)$ ← à laide le l'algo d'Euclide etendu

ii. **Chiffrement**

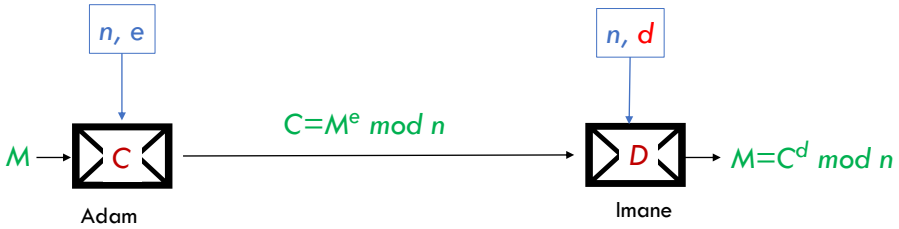
M est chiffré en C avec : $C = M^e \bmod n$

iii. **Déchiffrement**

Le message est récupéré avec : $M = C^d \bmod n$

CRYPTOGRAPHIE - INTRODUCTION

PRINCIPE DU CHIFFREMENT RSA



LEMME

Soit ***d*** l'inverse de modulo ϕ (n) avec n=p x q (p # q)

Si $x = m^e \bmod(n)$ alors $m = x^d \bmod(n)$

CRYPTOGRAPHIE - INTRODUCTION

CHIFFREMENT RSA

Exemple 1: Chiffrement du nombre 7 (M = 7) avec RSA

i. **Génération des clés**

Choisissons p=3 et q=11

- $n = 3 \times 11 = 33$
- $\phi(n)=2 \times 10 = 20$
- Choisissons ***e* = 3** (est un premier avec 20).
- Trouver **d** tel que $d \times e = 1 \bmod 20$ -> **prenons d=7, $7 \times 3 = 21 = 1 \bmod 20$**
  - **Clé publique** : (e=3, n=33)
  - **Clé privée** : (d=7, n=33)

CRYPTOGRAPHIE - INTRODUCTION

CHIFFREMENT RSA

Exemple1: Chiffrement du nombre 7 (M = 7) avec RSA.

ii. Chiffrement du message *M=7*

$C=M^e \bmod n = 7^3 \bmod 33$

On a $7^3 \bmod 33 = 343 \bmod 33 = 13$

→ **C = 13**

ii. Déchiffrement

$M=C^d \bmod n=13^7 \bmod 33$

- $13^2 = 169 = 4 \bmod 33$
- $13^4 = (13^2)^2 = 4^2 = 16 \bmod 33$
- $13^7 = 13^4 \times 13^2 \times 13 = 16 \times 4 \times 13 \bmod 33$
  - $16 \times 4 = 64 = 31 \bmod 33$
  - $31 \times 13 = 403 = 7 \bmod 33$

→ M déchiffré est 7

CRYPTOGRAPHIE - INTRODUCTION

CHIFFREMENT RSA

Exemple 2: Chiffrement du message « DSBD » avec RSA.

i. Génération des clés

Choisissons p=61 et q=53

- **n = 61 x 53** = 3233
- **ϕ(n)=60 x 52 = 3120**
- Choisissons ***e* = 17** (est un premier avec 3120).
- Trouver **d** tel que **d×e = 1 mod 3120 -> prenons d=2753**
  - **Clé publique** : (e=15, n=3233)
  - **Clé privée** : (d=2753, n=3233)

CRYPTOGRAPHIE - INTRODUCTION

CHIFFREMENT RSA

Exemple 2: Chiffrement du message « DSBD » avec RSA.

ii. Chiffrement

Chiffrement de D → $C=68^{17} \bmod 3233=1509$

Chiffrement de S → $C=83^{17} \bmod 3233=2592$

Chiffrement de B → $C=66^{17} \bmod 3233=131$

Message chiffré : **1509 2592 131 1509**

iii. Déchiffrement

$M=1509^{2753} \bmod 3233=68$ → D

$M=2592^{2753} \bmod 3233=83$ → S

$M=131^{2753} \bmod 3233=66$ → B

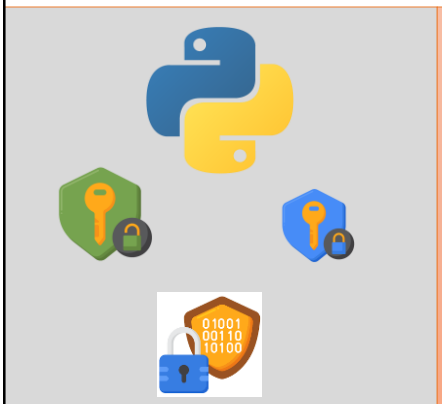
1509 2592 131 1509 → DSBD

CRYPTOGRAPHIE - INTRODUCTION

CHIFFREMENT RSA – ATTAQUES ET CONTRE-MESURES

Attaque	Principe	Contre-mesures
Factorisation de N	Trouver p,q	Clés RSA ≥ 2048 bits, p et q bien choisis
Exposant faible	Si e est petit, $M^e<N$	e=65537, utiliser un padding (OAEP)
Chosen Ciphertext (CCA)	Forcer le déchiffrement	Utiliser OAEP, éviter PKCS#1 v1.5
Timing Attack	Mesurer le temps de déchiffrement	Blinding, exécution constante
Key Substitution	Modifier la clé publique	Vérifier les certificats (X.509)
Common Modulus Attack	Même N, exposants différents	Ne pas réutiliser NNN
…		

CRYPTOGRAPHIE – RSA_APPLICATION



OBJECTIFS

- **AJOUT DE LA BIBLIOTHÈQUE RSA / PYCRYPTODOME**
- **GÉNÉRATION D'UNE CLÉ PUBLIQUE**
- **GÉNÉRATION D'UNE CLÉ PRIVÉE**
- **CHIFFREMENT D'UN MESSAGE**
- **DÉCHIFFREMENT D'UN MESSAGE CRYPTÉ**

CRYPTOGRAPHIE – RSA_APPLICATION

- ✓ L'extension **.pem** (Privacy Enhanced Mail) est un format utilisé pour stocker **des clés cryptographiques, des certificats, et d'autres données sécurisées** sous forme de **texte encodé en Base64**. Ce format est très utilisé en cryptographie, notamment avec **RSA, TLS/SSL et OpenSSL**
- ✓ load_pkcs1 est une fonction de la bibliothèque **rsa** (Python-RSA) qui permet de charger une **clé RSA** (publique ou privée) au format **PKCS#1.**
- ✓ **PKCS#1 (Public Key Cryptography Standards #1)** est un standard utilisé pour stocker et échanger des clés RSA.

- ✓ La méthode **.encode()** convertit une chaîne de caractères (de type str) en octets (bytes).
- ✓ En cryptographie RSA (et en général dans la plupart des bibliothèques cryptographiques), les fonctions de chiffrement attendent des données binaires (bytes) plutôt que des chaînes de caractères.

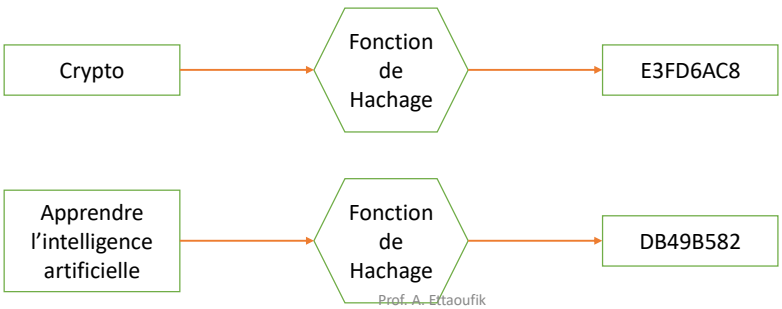
```python
import rsa
#public_key, private_key=rsa.newkeys(1024)
#with open("public.pem","wb") as f:
    #f.write(public_key.save_pkcs1("PEM"))
#with open("private.pem","wb") as f:
   # f.write(private_key.save_pkcs1("PEM"))

with open("public.pem","rb") as f:
    public_key=rsa.PublicKey.load_pkcs1(f.read())
with open("private.pem","rb") as f:
    private_key=rsa.PrivateKey.load_pkcs1(f.read())
#print (private_key)

message="Mabrouk Ramadan"
message_crypte=rsa.encrypt(message.encode(),public_key)
#print (message_crypte)
with open("crypte.message","wb") as f:
    f.write(message_crypte)

message_crypte=open("crypte.message","rb").read()
message_clair=rsa.decrypt(message_crypte,private_key)
print(message_clair.decode())

message2="Mabrouk l3id"
#signature=rsa.sign(message2.encode(),private_key,"SHA-256")
#with open("signature","wb") as f:
#    f.write(signature)
with open("signature","rb") as f:
    signature=f.read()
print(rsa.verify(message2.encode(),signature,public_key))
```


CRYPTOGRAPHIE - INTRODUCTION

HACHAGE - FONCTION DE HACHAGE

Une fonction de hachage consiste en une fonction qui transforme une donnée quelconque en une donnée de taille fixée.

Les fonctions de hachage sont des fonctions a sens unique



CRYPTOGRAPHIE - INTRODUCTION

HACHAGE – COLLISION

On parle de problème de collision lorsqu'il existe deux entrées différentes M1 et M2 qui possèdent le même code haché:

H(M1)=H(M2) / M1#M2

CRYPTOGRAPHIE - INTRODUCTION

HACHAGE – CAS D'USAGE

- **Stockage de mots de passe -** Le hachage garantit que les données sont stockées sous une forme brouillée, ce qui complique leur vol.
- **Signatures numériques -** Un petit bit de données permet de démontrer qu'un message n'a pas été modifié entre le moment où il quitte la boîte d'envoi d'un utilisateur et arrive dans la boîte de réception du destinataire.
- **Gestion des documents -** Les algorithmes de hachage permettent d'authentifier des données. L'auteur utilise un hachage pour sécuriser un document une fois ce dernier terminé.
  Un destinataire peut générer un hachage et le comparer à l'original.
- **Gestion des fichiers -** Certaines sociétés utilisent également des hachages pour indexer les données, identifier les fichiers et supprimer les doublons. Si un système héberge des milliers de fichiers, l'utilisation de hachages permet de gagner un temps précieux.

CRYPTOGRAPHIE - INTRODUCTION

FONCTIONS DE HACHAGE POPULAIRES

Les algorithmes de hachage les plus couramment utilisés sont les suivants :

- SHA-1 (Secure Hash Algorithm 1) ;
- SHA-2 ;
- SHA-3 ;
- MD4 (Message Digest 4) ;
- MD5.
- Whirlpool : cet algorithme a été créé en 2000, sur la base de la norme AES (Advanced Encryption Standard). Il est également jugé très sûr.

❑ La fonction **MD5** est toujours très répandue. Cependant, des failles de sécurité sous la forme de séries de collision ont été trouvées

❑ **SHA-3** est la version la plus récente de l'algorithme SHA et présente actuellement le plus haut niveau de sécurité.

CRYPTOGRAPHIE - INTRODUCTION

FONCTIONS DE HACHAGE POPULAIRES – MD5

```
import hashlib
print(hashlib.md5("Bonjour tout le monde".encode('UTF-8')).hexdigest())
print(hashlib.md5("Blockchain".encode('UTF-8')).hexdigest())
print(hashlib.md5("Web3.0".encode('UTF-8')).hexdigest())
print (len(hashlib.md5("Web3.0".encode('UTF-8')).hexdigest())* 4)
```

OUTPUT

```
d26dddcf80d724b031e58a7cfec0cd27
3cc377f79bda308c750459a2caf7fc38
07b68b153dc0207f74c16ad9a1ea5ba9
128
```

CRYPTOGRAPHIE - INTRODUCTION

FONCTIONS DE HACHAGE POPULAIRES – MD5

d26dddcf80d724b031e58a7cfec0cd27

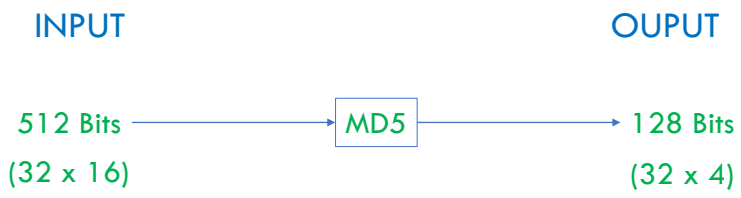
32 Caractères Hex

01000110001.....
.....010

128 Bits

CRYPTOGRAPHIE - INTRODUCTION

FONCTIONS DE HACHAGE POPULAIRES – MD5



CRYPTOGRAPHIE - INTRODUCTION

FONCTIONS DE HACHAGE POPULAIRES – MD5

Utilisation des opérateurs logiques pour garantir une taille fixe des outputs

- OR
- AND
- XOR
- NOT
- ROTATION
- ADDITION

CRYPTOGRAPHIE - INTRODUCTION

SHA-3

- SHA-3 (Secure Hash Algorithm 3) est la dernière famille de fonctions de hachage normalisées par le NIST (National Institute of Standards and Technology des États-Unis).
- Il succède aux algorithmes SHA-1 et SHA-2, en offrant une sécurité renforcée grâce à une structure totalement différente.
- SHA-3 est basé sur une primitive cryptographique appelée ***Keccak***, une construction dite ***éponge***.
- **Flexible** : Peut générer des hachages de différentes tailles (224, 256, 384, 512 bits).

CRYPTOGRAPHIE - INTRODUCTION

SHA-3

Les fonctions éponge calculent les hachages en deux phases :

(1) la phase d'absorption :

- Le message est découpé en blocs et absorbé dans un état interne de 1600 bits.
- Un processus de permutation est appliqué à chaque itération.

(2) la phase d'essorage :

- Une fois le message entièrement absorbé, la sortie est générée en extrayant des bits de l'état interne.
- Cela se poursuit jusqu'à obtenir la longueur de sortie désirée.

CRYPTOGRAPHIE - INTRODUCTION

SHA-3 - EXEMPLE

```
import hashlib
message = "Algorithme, SHA-3!"
# Calcul du hash SHA3-256
hash_sha3 = hashlib.sha3_256(message.encode('UTF-8')).hexdigest()
print("SHA3-256 :", hash_sha3)
```

```
SHA3-256 : 0746393da3ee530d2ea17853b693851596cc4a98104f93319a9909efea39b2c9
```

CRYPTOGRAPHIE - INTRODUCTION

SIGNATURES NUMÉRIQUES ET INTÉGRITÉ DES DONNÉES

- Principe de signature numérique
- Algorithmes : RSA, DSA, ECDSA
- Vérification de l'authenticité d'un message
- **Exercice pratique** :

Générer et vérifier une signature numérique avec Python.

CRYPTOGRAPHIE - INTRODUCTION

SIGNATURES NUMÉRIQUES ET INTÉGRITÉ DES DONNÉES

Une signature numérique est un mécanisme cryptographique permettant d'assurer l'authenticité, l'intégrité et la non-répudiation d'un message ou d'un document électronique.

Elle fonctionne de manière similaire à une signature manuscrite, mais avec un niveau de sécurité bien plus élevé.

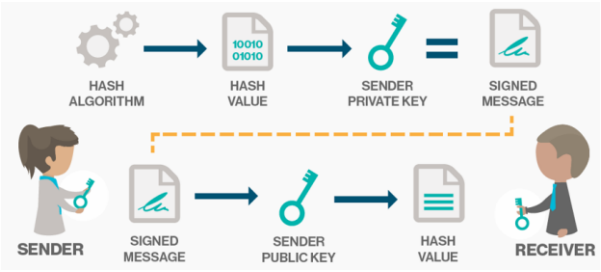
Elle repose sur la cryptographie asymétrique

CRYPTOGRAPHIE - INTRODUCTION

SIGNATURES NUMÉRIQUES - PRINCIPE

Expéditeur : **Condenser** le document (fonction de **hashage**) puis le "chiffre" avec sa clé privée, cela représente une **signature**.

Récepteur : possède la clé publique de l'émetteur. Il peut donc vérifier la signature. Si le message est altéré en cours de route, la signature ne correspondra pas.



CRYPTOGRAPHIE - INTRODUCTION

PROPRIÉTÉS DE LA SIGNATURE NUMÉRIQUE

LES SIGNATURES DOIVENT ÊTRE CARACTÉRISÉES PAS LES PROPRIÉTÉS SUIVANTES :

- AUTHENTIQUE
- INFALSIFIABLE
- NON-RÉUTILISABLE
- INALTÉRABLE
- NON-RÉPUDIABLE

CRYPTOGRAPHIE - INTRODUCTION

ALGORITHMES DE SIGNATURES

DIGITAL SIGNATURE ALGORITHM **DSA** :

- UN ALGORITHME À CLÉ PUBLIQUE
- REPOSE SUR LE PROBLÈME DES LOGARITHMES DISCRETS
- NE RÉALISE QUE DES SIGNATURES (PAS DE CHIFFREMENT)

DSA VS **RSA** :

- LA GÉNÉRATION DE LA SIGNATURE EN **DSA** EST PLUS RAPIDE QUE SA VÉRIFICATION.
- LE **RSA** PEUT SERVIR AU CHIFFREMENT ET AUX SIGNATURES.

CRYPTOGRAPHIE - INTRODUCTION

AUTHENTIFICATION

PERMET D'AUTHENTIFIER LES PARTICIPANTS DANS UNE COMMUNICATION (*CLIENT - SERVEUR*). PLUSIEURS TECHNIQUES PEUVENT ÊTRE EMPLOYÉES :

- CRYPTO-SYSTÈME SYMÉTRIQUE AVEC UTILISATION DE ***NONCE***.
- CRYPTO-SYSTÈME ASYMÉTRIQUE COMBINANT MESSAGE EN CLAIR ET CHIFFRÉ
- ***MAC*** (MESSAGE AUTHENTIFICATION CODE) + FONCTION DE HACHAGE

CRYPTOGRAPHIE - INTRODUCTION

AUTHENTIFICATION

CRYPTO-SYSTÈME SYMÉTRIQUE AVEC UTILISATION DE ***NONCE*.**

- LE **NONCE** (ABRÉVIATION DE *NUMBER USED ONCE*, "NOMBRE UTILISÉ UNE SEULE FOIS") EST UNE VALEUR UNIQUE UTILISÉE EN CRYPTOGRAPHIE POUR GARANTIR LA SÉCURITÉ DES COMMUNICATIONS CHIFFRÉES.
- IL EMPÊCHE LES ATTAQUES PAR RELECTURE ET ASSURE L'UNICITÉ DES MESSAGES CHIFFRÉS.

EXEMPLE :

- Imaginons une transaction bancaire où un utilisateur envoie un message « **Transférer 1000 DH à Amine** ».
- Un attaquant pourrait intercepter ce message chiffré et le rejouer plusieurs fois pour forcer des transferts multiples.
- Avec un Nonce unique ajouté à chaque message, chaque requête devient différente, rendant impossible une attaque par relecture.

CRYPTOGRAPHIE - INTRODUCTION

AUTHENTIFICATION

CRYPTO-SYSTÈME ASYMÉTRIQUE COMBINANT MESSAGE EN CLAIR ET CHIFFRÉ

- UNE MÉTHODE DE CHIFFREMENT HYBRIDE OÙ UNE PARTIE DU MESSAGE EST ENVOYÉE EN CLAIR TANDIS QUE L'AUTRE EST CHIFFRÉE À L'AIDE D'UN ALGORITHME ASYMÉTRIQUE
- GÉNÉRALEMENT, SEULS LES DONNÉES SENSIBLES OU CRITIQUES SONT CHIFFRÉES POUR LIMITER LE COÛT COMPUTATIONNEL DU CHIFFREMENT.
- CE TYPE DE SYSTÈME PEUT ÊTRE UTILISÉ POUR DIVERS OBJECTIFS, NOTAMMENT L'OPTIMISATION DES PERFORMANCES ET LA SÉCURISATION PROGRESSIVE D'UN ÉCHANGE D'INFORMATIONS.

CRYPTOGRAPHIE - INTRODUCTION

AUTHENTIFICATION

MAC (MESSAGE AUTHENTIFICATION CODE) + FONCTION DE HACHAGE

Le **MAC** est une valeur d'authentification calculée à partir d'un message et d'une clé secrète. Il est utilisé pour garantir à la fois l'intégrité et l'authenticité du message :

- L'expéditeur et le destinataire partagent une clé secrète **K**.
- L'expéditeur applique une fonction MAC sur le message **M** avec la clé **K**, produisant un code d'authentification **MAC(M, K)**.
- Le MAC est envoyé avec le message.
- Le destinataire, connaissant **K**, recalcule le MAC et vérifie s'il correspond.

CRYPTOGRAPHIE - INTRODUCTION

SIGNATURES NUMÉRIQUES - EXEMPLE

```
message2="Mabrouk l3id"
signature=rsa.sign(message2.encode(),private_key,"SHA-256")
with open("signature","wb") as f:
    f.write(signature)
with open("signature","rb") as f:
    signature=f.read()
print(rsa.verify(message2.encode(),signature,public_key))
```

GESTION DES CLÉS

PROBLÈME

CES PROTOCOLES SUPPOSENT QUE LES PROTAGONISTES DISPOSENT DES CLÉS NÉCESSAIRES.

COMMENT S'ASSURER QUE LES CLÉS NE SONT PAS DIVULGUÉES AU MAUVAIS DESTINATAIRE ?

- GÉNÉRATION DES CLÉS : UTILISANT DES GÉNÉRATEURS SOLIDES
- DISTRIBUTION : PAR CANAL SÉCURISÉ OU UN TIERS DE CONFIANCE
- VÉRIFICATION : PAR HACHAGE OU CERTIFICAT
- STOCKAGE : SUPPORT LOCAL OU DISTANT, SOUS SCELLÉ…

GESTION DES CLÉS

CLÉ SYMÉTRIQUE

L'OBJECTIF EST DE PARTAGER UNE CLÉ SECRÈTE COMMUNE

- PHYSIQUEMENT
- VIA UN TIERS DE CONFIANCE **KDC*** (GÉNÈRE ET ENVOIE LA CLÉ)
- PAR LE BIAIS D'UNE ANCIENNE CLÉ COMMUNE UTILISÉE POUR CHIFFRER LA NOUVELLE (PRÉSENTE DES RISQUES)

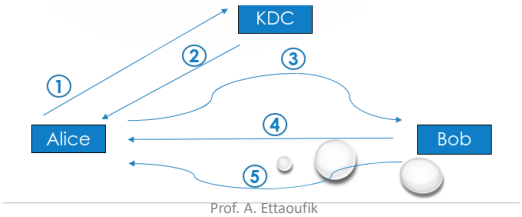
* : KEY DISTRIBUTION CENTER

GESTION DES CLÉS

CLÉ SYMÉTRIQUE

LE PROTOCOLE DE *NEEDHAM-SCHROEDER (TRIPARTITE)* SUPPOSE QUE CHAQUE ENTITÉ (ALICE ET BOB) POSSÈDE UNE PROPRE CLÉ SECRÈTE PARTAGÉE AVEC LE **KDC**.

1- $A.B.N_A$
2- **EK(**$B.N_A.K_{AB}$**.EK(**$K_{AB}.A$, K_{BK}**)**, K_{AK}**)**
3- **EK(**$K_{AB}.A$, K_{BK}**)**
4- **EK(**N_B , K_{AB}**)**
5- **EK(**N_B-1 , K_{AB}**)**



N_i Nonce généré par i
K_{ij} clé secrète entre i et j
A identité
. regroupement
EK(M,K) chiffrement de M avec K

GESTION DES CLÉS

CLÉ ASYMÉTRIQUE

COMMENT PARTAGER LA CLÉ PUBLIQUE. PLUSIEURS TECHNIQUES :

- ❑ ANNUAIRE PUBLIQUE
- ❑ AUTORITÉ PUBLIQUE
- ❑ CERTIFICAT

- ❑ Enregistre nom et clé publique
- ❑ Sécurise les inscriptions
- ❑ Met à jour les clefs
- ❑ Publie de façon régulière

Vulnérable, sujet aux contrefaçons..

GESTION DES CLÉS

CLÉ ASYMÉTRIQUE

COMMENT PARTAGER LA CLÉ PUBLIQUE. PLUSIEURS TECHNIQUES :

- ANNUAIRE PUBLIQUE
- AUTORITÉ PUBLIQUE
- CERTIFICAT

  - Renforce le concept précédent
  - Attribue une clé publique à l'annuaire.
  - Utilise des marqueurs timeStamp

  Passage obligatoire par l'autorité à chaque demande

GESTION DES CLÉS

CLÉ ASYMÉTRIQUE

LES PARTICIPANTS UTILISENT DES CERTIFICATS POUR ÉCHANGER DES CLÉS APPROUVÉES SANS PASSER PAR L'AUTORITÉ DE CERTIFICATION **CA** (CERTIFICATE AUTHORITY).

PKI – PUBLIC KEY INFRASTRUCTURE

OBJECTIF

PROUVER LA CORRESPONDANCE ENTRE UNE CLÉ PUBLIQUE ET UNE PERSONNE OU ENTITÉ.

- EMETTRE DES CERTIFICATS À DES UTILISATEURS AUTHENTIFIÉS
- RÉVOQUER OU MAINTENIR DES CERTIFICATS
- PUBLIER LES CERTIFICATS VIA DES SERVICES D'ANNUAIRES
- GÉRER LES CLÉS, ARCHIVER…

PKI – PUBLIC KEY INFRASTRUCTURE

ACTEURS

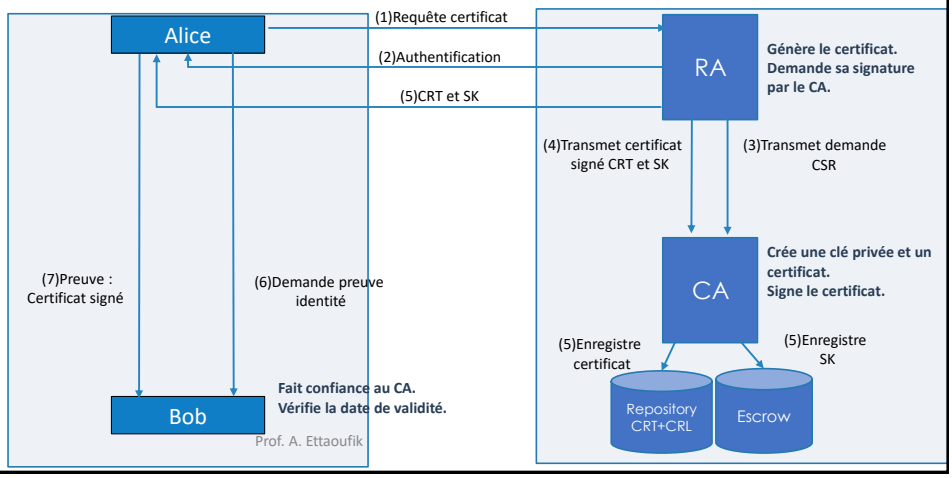
- ENTITÉ FINALE (***EE*** : CLIENT, SERVEUR…)
  - UTILISATEUR OU DÉTENTEUR DU CERTIFICAT
- AUTORITÉ D'ENREGISTREMENT (***RA***)
  - AUTHENTIFIE ET VÉRIFIE L'IDENTITÉ DU DEMANDEUR
  - EFFECTUE LA DEMANDE ET REMET LE CERTIFICAT SIGNÉ À EE
- AUTORITÉ DE CERTIFICATION (***CA***)
  - GÉNÈRE ET SIGNE LES CERTIFICATS
  - GÈRE LA ***CRL*** (CERTIFICATION REVOCATION LIST)
- AUTORITÉ DE DÉPÔT (***REPOSITORY***)
  - STOCKE LES CERTIFICATS ET LES CRL
- AUTORITÉ DE SÉQUESTRE (***KEY ESCROW***)
  - MET SOUS SCELLÉ LES CLÉS (SPÉCIALEMENT CLÉS SECRÈTES)

- VeriSign
- Visa
- La poste
- Symantec
- Google Trust
- AOL Time Warner

PKI – PUBLIC KEY INFRASTRUCTURE

PRINCIPE

Alice Détenteur
Bob utilisateur
RA Registration Authority
CA Certification Authority
SK Secret Key
CRT Signed Certificate
CSR Certificate Signing Request
CRL Certification Revocation List



PKI – PUBLIC KEY INFRASTRUCTURE

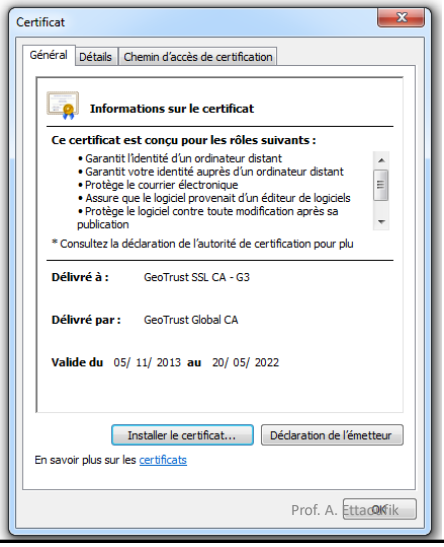
CONTENU – SELON LE STANDARD **X.509**

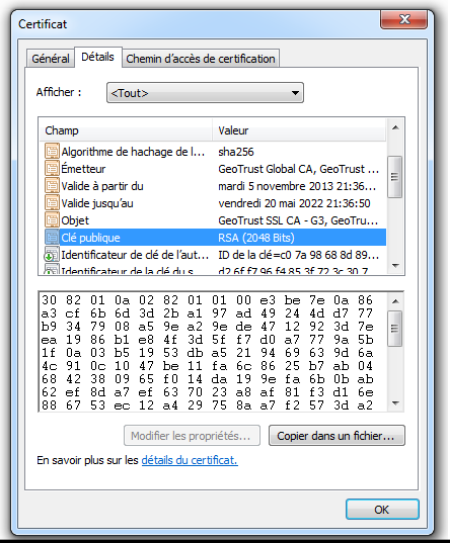
LA NORME SPÉCIFIANT LES FORMATS ET LE CONTENU DES CERTIFICATS À CLÉ PUBLIQUE.

- ❑Version
- ❑Numéro de série
- ❑Algorithme de signature du certificat : *MD5, SHA-1*
- ❑Nom du signataire
- ❑Validité
- ❑Détenteur du certificat
- ❑Algorithme de la clé publique : *RSA, ELGAMAL, EC…*
- ❑Clé publique
- ❑Identifiant unique du signataire
- ❑Identifiant unique du détenteur

PKI – PUBLIC KEY INFRASTRUCTURE

CONTENU





PKI – PUBLIC KEY INFRASTRUCTURE

FORMATS DES CERTIFICATS X.509

- ENCODAGE DER (DISTINGUISHED ENCODING RULES) : .DER, .CER, .CRT, .CERT
- ENCODAGE PEM (PRIVACY ENHANCED MAIL) ENCODÉ **BASE64** AVEC EN-TÊTE ET PIED DE PAGE : .PEM, .CER, .CRT, .CERT

  -----BEGIN X509 CRL-----

  -----END X509 CRL-----

PKI – PUBLIC KEY INFRASTRUCTURE

LES CLASSES DE CERTIFICATS

- CLASSE 1 : ADRESSE E-MAIL DU DEMANDEUR
- CLASSE 2 : PREUVE DE L'IDENTITÉ REQUISE
- CLASSE 3 : PRÉSENTATION PHYSIQUE
- CLASSE 3+ : CERTIFICAT STOCKÉ SUR UN SUPPORT PHYSIQUE (CARTE À PUCE, CLÉ USB…)

PKI – PUBLIC KEY INFRASTRUCTURE

RÉVOCATION DE CERTIFICAT

- PERTE OU COMPROMISSION DE LA CLEF PRIVÉE
- DISPARITION DU DÉTENTEUR DU CERTIFICAT

DANS CES CAS LE CERTIFICAT EST AJOUTÉ À LA **CRL**

- LA LISTE EST (NUMÉRO DE SÉRIE + MOTIF) ENVOYÉE SOUS FORMAT DER OU PEM
- ÉCHANGE LOURD (GROSSE LISTE)

SOLUTION : OCSP

PKI – PUBLIC KEY INFRASTRUCTURE

OCSP

LA CONSULTATION PAR LE CLIENT DE LA CRL EST COMPLEXE. ELLE PRÉSENTE UNE CONTRAINTE (DIVULGATION DE TOUS LES CERTIFICATS COMPROMIS).

ONLINE **C**ERTIFICATE **S**TATUS **P**ROTOCOL EST UN PROTOCOLE DE VÉRIFICATION DE CERTIFICAT EN LIGNE. L'UTILISATEUR UTILISE UN SERVEUR **OCSP** APPELÉ AUTORITÉ DE VALIDATION (**VA**).

- L'EMPREINTE DU CERTIFICAT DU VENDEUR EST TRANSMISE (REQUÊTE **OCSP**) AU **VA**
- LE **VA** CONSULTE **CA** (VÉRIFIE LA VALIDITÉ ET LE **CRL**)
- LA RÉPONSE **OCSP** EST ENVOYÉE AU CLIENT
- LE SERVICE EST FACTURÉ AU PROPRIÉTAIRE DU CERTIFICAT

CE PROTOCOLE ALLÈGE LA PROCÉDURE.

CRYPTOGRAPHIE - INTRODUCTION

APPLICATIONS ET SÉCURITÉ

- Protocoles de sécurité (SSL/TLS, PGP, SSH)
- Utilisation des certificats numériques
- Attaques courantes et protections :
- Attaque par facteur premier
- Attaque de l'homme du milieu (MitM)
- Timing attack
- **Démonstration** :
  - Vérification d'un certificat SSL avec openssl.
  - Simulation d'une attaque MitM avec un proxy.

CRYPTOGRAPHIE - INTRODUCTION

PROTOCOLES DE SÉCURITÉ

Les **protocoles de sécurité** permettent de protéger les communications et les données échangées sur les réseaux contre les attaques

- SSL (obsolète) et son successeur TLS:
  - Protocoles permettant de sécuriser les communications sur Internet via le chiffrement, l'authentification et l'intégrité des données.
  - Largement utilisés pour sécuriser le Web (HTTPS), les emails (SMTP, IMAP, POP3S) et la VoIP.

CRYPTOGRAPHIE - INTRODUCTION

PROTOCOLES DE SÉCURITÉ

- PGP (Pretty Good Privacy)
  - Protocole de chiffrement asymétrique permettant de sécuriser les emails et fichiers.
  - Il garantit la confidentialité, l'intégrité et l'authenticité grâce à un système de clé publique / clé privée.

CRYPTOGRAPHIE - INTRODUCTION

PROTOCOLES DE SÉCURITÉ

- SSH (Secure Shell)
  - Protocole permettant d'établir une connexion sécurisée à distance entre un client et un serveur, utilisé pour l'administration des serveurs et le transfert de fichiers.
  - Il remplace les protocoles non sécurisés comme Telnet et FTP..