

CRYPTOGRAPHIE

1. Objectifs de la sécurité informatique

- **Confidentialité (2)** : Empêcher l'accès aux données par des tiers non autorisés.
- **Authentification (2)** : Vérifier l'identité de l'utilisateur ou de la source des données.
- **Intégrité (2)** : Garantir que les données n'ont pas été altérées.
- **Disponibilité** : Assurer l'accès aux données et services en temps voulu.
- **Non-répudiation (2)** : Empêcher une partie de nier une action précédemment effectuée (envoi de message, signature numérique).

(2) : Services attendus du chiffrement asymétrique

2. Chiffrement symétrique vs asymétrique

CONFIDENTIALITÉ

Comment fonctionne le CHIFFREMENT ?

CHIFFREMENT SYMÉTRIQUE

Le chiffrement symétrique permet de chiffrer et déchiffrer un fichier avec la même clé, dite secrète. Pour s'échanger un message il faut donc que les deux parties partagent la même clé.

MISE EN PRATIQUE

Alice vient d'enregistrer la liste des cadeaux de Noël de sa famille sur l'ordinateur familial. Elle souhaite être la seule à pouvoir y accéder.

1. Pour ce faire, Alice chiffre la liste en utilisant sa clé secrète.
2. Plus tard dans la journée, Bob trouve la liste et cherche à l'ouvrir.
3. Malheureusement pour lui, Bob est incapable de lire la liste car il ne possède pas la clé secrète.
4. La liste est donc bien protégée. Seule Alice peut réussir à la déchiffrer et la lire !

CHIFFREMENT ASYMÉTRIQUE

Le chiffrement asymétrique repose sur l'utilisation d'une paire de clés : une publique et une privée.

La clé publique, accessible à tous, est utilisée pour chiffrer les fichiers. Seule la clé privée permet de déchiffrer ces fichiers, celle-ci étant connue que d'un seul individu.

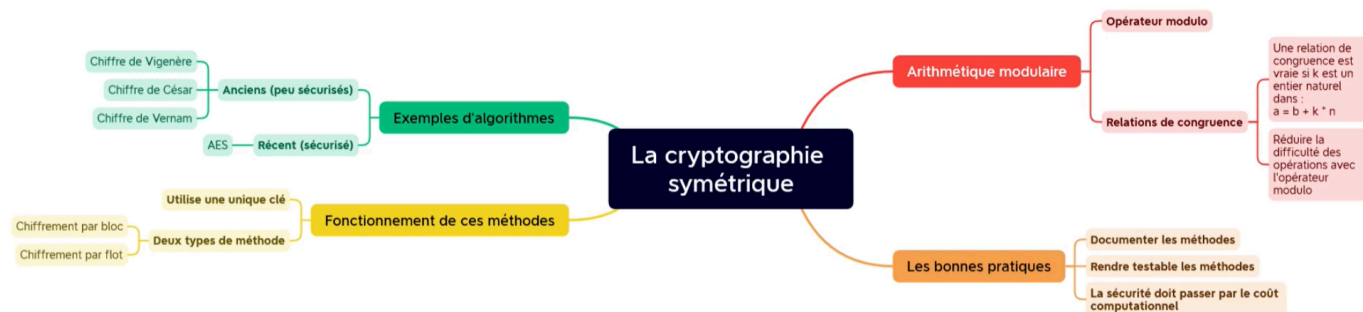
MISE EN PRATIQUE

Alice, hackeuse, vient de découvrir des informations d'intérêt public. Elle veut les transmettre à Bob, journaliste, pour qu'il enquête.

1. Alice vient de récupérer la clé publique de Bob. Elle l'utilise pour chiffrer son document.
2. Elle l'envoie à Bob.
3. Bob reçoit le document et le déchiffre à l'aide de sa clé privée.
4. Une fois le document déchiffré, il rédige un article puis le publie dans son journal.

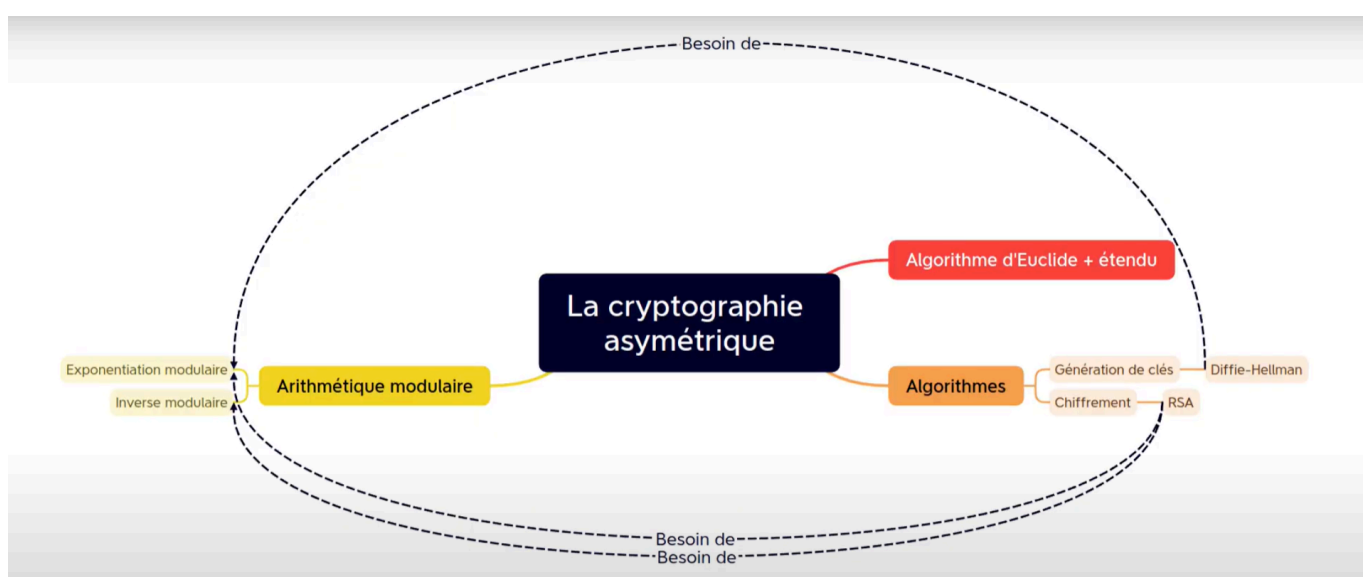
Chiffrement symétrique

- **Principe** : Une seule clé pour chiffrer et déchiffrer.
- **Exemples** : AES (Chiffrement par blocs de 128 bits), DES.
- **Avantage** : Rapide pour chiffrer des grandes quantités de données.
- **Inconvénient** : La clé doit rester secrète et être partagée de manière sécurisée.



Chiffrement asymétrique

- **Principe** : Utilise une paire de clés : une clé publique pour chiffrer et une clé privée pour déchiffrer.
- **Exemples** : RSA (Basé sur la factorisation de grands nombres premiers), ECC.
- **Avantages** : Pas besoin de partager une clé secrète, la clé publique peut être diffusée.
- **Inconvénients** : Plus lent que le chiffrement symétrique.



3. Algorithmes et techniques de chiffrement

Chiffrement Symétrique

- **Chiffre de César** : $C = (M + k) \bmod 26$
- **Carré de Polybe** :

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	I/J	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

- **Chiffre de Vigenère** : $C_i = (M_i + K_i) \bmod 26$, où M_i est le texte clair, K_i est la clé répétée, et C_i est le texte chiffré.

	Lettre en clair																									
	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
Lettre de la clé	Lettres chiffrées (au croisement de la colonne <i>Lettre en clair</i> et de la ligne <i>Lettre de la clé</i>)																									
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

- **Chiffre de Vernam (One Time Pad/Codage à masque jetable)** : $C = M \oplus K$ où M est le message, K est la clé, et C est le résultat.
- **AES (Advanced Encryption Standard)** : Utilise un algorithme de chiffrement par blocs avec des clés de 128, 192, ou 256 bits. Ses principales étapes sont :
 1. **SubBytes** : Chaque octet du bloc est remplacé par un autre selon une table de substitution fixe (S-Box).
 2. **ShiftRows** : Les lignes du bloc sont décalées vers la gauche.
 3. **MixColumns** : Chaque colonne du bloc est mélangée via une opération linéaire (multiplication dans un corps fini).
 4. **AddRoundKey** : Un XOR est appliqué entre le bloc et la clé de tour.
- **Chiffrement par flot / par flux** :
 - Chiffrement qui traite les données sous forme de flux continu, plutôt qu'en blocs.
 - **RC4** est un exemple de chiffrement par flot, où les bits du message sont combinés avec une séquence de bits pseudo-aléatoires.

Chiffrement Asymétrique

- **RSA** :
 - Clé publique : (e, n) , clé privée : (d, n)
 - Clé privée : $d = e^{-1} \bmod \phi(n)$, avec $\phi(n) = (p - 1)(q - 1)$
 - Chiffrement : $C = M^e \bmod n$
 - Déchiffrement : $M = C^d \bmod n$
- **Exponentiation modulaire** :
 - Utilisée dans des algorithmes asymétriques comme RSA ou Diffie-Hellman.
 - Le calcul : $C = M^e \bmod n$, permet de chiffrer des messages en grande partie grâce à des opérations modulaires.
 - **Méthode de l'exponentiation rapide** : Calcule efficacement des puissances modulaires en réduisant à chaque étape.
- **Algorithme d'Euclide Étendu** :
 - Permet de calculer $d = \text{pgcd}(a, b)$ ainsi que x et y tels que $ax + by = d$.
 - Utile pour trouver l'inverse modulaire $d \equiv e^{-1} \bmod \phi(n)d$.
- **Théorème de Bézout** :
 - Le théorème affirme que pour deux entiers a et b , il existe des entiers x et y tels que $ax + by = d$, où d est le plus grand commun diviseur (PGCD) de a et b .
 - Ceci est utile dans la cryptographie pour trouver des inverses modulaires.
- **Calcul de l'inverse modulaire** :
 - Pour un entier a et un modulo n , l'inverse modulaire de a est un entier x tel que $a \cdot x \equiv 1 \bmod n$.
 - **Algorithme d'Euclide Étendu** : Permet de résoudre l'équation $ax + by = 1$ pour calculer l'inverse de $a \bmod n$.
- **Échange de clé Diffie-Hellman** :
 - Clés partagées : $A = g^a \bmod p, B = g^b \bmod p$
 - Clé commune : $K = B^a \bmod p = A^b \bmod p$

Alice			Bob		
Secret	Calcul	Public	Public	Calcul	Secret
		p, g	p, g		
a					b
	$A = g^a[p]$	A	(reçoit)		
		(reçoit)	B	$B = g^b[p]$	
	$B^a[p] = (g^b[p])^a[p] = g^{ab}[p]$			$A^b[p] = (g^a[p])^b[p] = g^{ab}[p]$	

4. Fonctions de hachage

- **SHA-256**: Algorithme de hachage produisant une sortie de 256 bits.
 - **Propriétés importantes**:
 - **Résistance aux collisions**: $H(x) = H(y)$ implique $x = y$
 - **Résistance à la préimage**: Il est difficile de trouver x tel que $H(x) = y$.
-