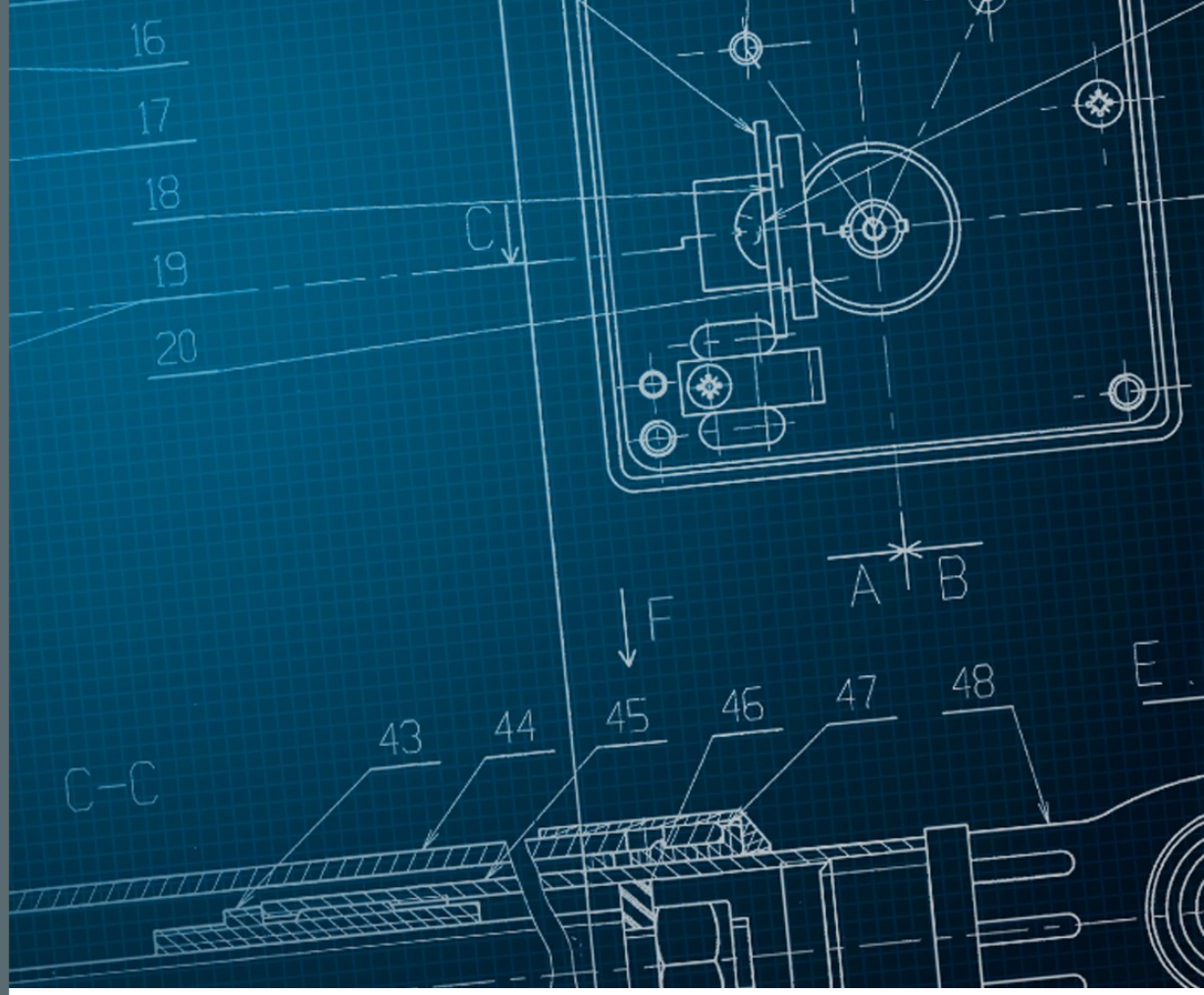
THIERRY MEYER
SÉCURITÉ INFORMATIQUE

Bienvenue




# Les concepts fondamentaux de la cryptographie appliquée

Partie 1

# INTRODUCTION À LA CRYPTOGRAPHIE APPLIQUÉE

# PLAN DU COURS

## I. Introduction à la cryptographie appliquée

- Qu'est ce que la cryptographie ?
  - *Étymologiquement*
- A quoi sert la cryptographie ?
- Vocabulaire et concepts de base de la cryptographie
  - *Les acteurs de la cryptographie*
  - *La notion de Message et de chiffrement*
  - *Les algorithmes cryptographiques …*
  - *… et les clefs de chiffrement*
  - *Les algorithmes de substitution …*
  - *… et de transposition*
  - *La fonction XOR*

# INTRODUCTION À LA CRYPTOGRAPHIE APPLIQUÉE

## Quelques ressources (très) utiles ...

> Bruce Schneier - @schneierblog

- Applied Cryptography *(ISBN : 978-1-119-09672-6)*
- https://www.schneier.com/

> Jean-Philippe Aumasson - @veorq

- Serious Cryptography *(ISBN-13: 978-1-59327-826-7)*
- https://131002.net/

> Renaud Lifchitz - @nono2357

- Cryptographie quantique – podcast Nolimitsecu
- https://www.nolimitsecu.fr/informatique-quantique/

# Quelques ressources (très) utiles ...

## > Voir aussi :

- Contenu détaillé sur les concepts de la cryptographie :
  - http://etutorials.org/Programming/Programming+.net+security/Part+III+.NET+Cryptography/
- A propos de la taille des clefs :
  - https://www.keylength.com/fr/
  - https://www.keylength.com/fr/5/
- A propos des collisions de hash :
  - https://www.nolimitsecu.fr/collisions-de-hash/
- A propos du stockage des mots de passe :
  - https://news.sophos.com/fr-fr/2013/11/21/stocker-mots-de-passe-en-securite/
- A propos de la vulnérabilité Curveball :
  - https://www.nolimitsecu.fr/curveball/

## Quelques ressources (très) utiles ...

> A propos de PKZIP

- http://www.cs.sjsu.edu/~stamp/crypto/PowerPoint_PDF/8_PKZIP.pdf

> RFC 2104 – HMAC : https://tools.ietf.org/html/rfc2104

> Stream Cipher : M.J.B. Robshaw – RSA Laboratories

- ftp://ftp.rsasecurity.com/pub/pdfs/tr701.pdf

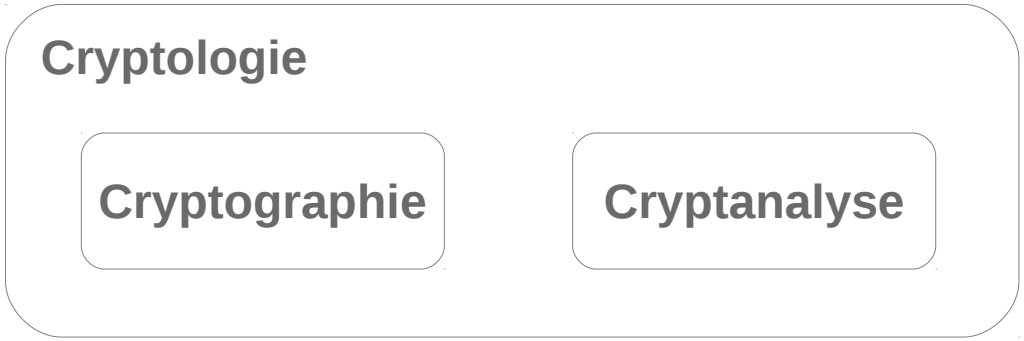
# Qu'est-ce que la cryptographie ?

## > Étymologiquement ...

- C'est la science des écritures secrètes,
- C'est l'art (et la science) de garder les messages secrets.

# Qu'est-ce que la cryptographie ?

> C'est un sous-ensemble de la cryptologie ...



# A quoi sert la cryptographie ?

## > Les 4 services de sécurité fournis par la cryptographie

- Authentification
- Intégrité
- Confidentialité
- Non-répudiation

## > Les acteurs de la cryptographie

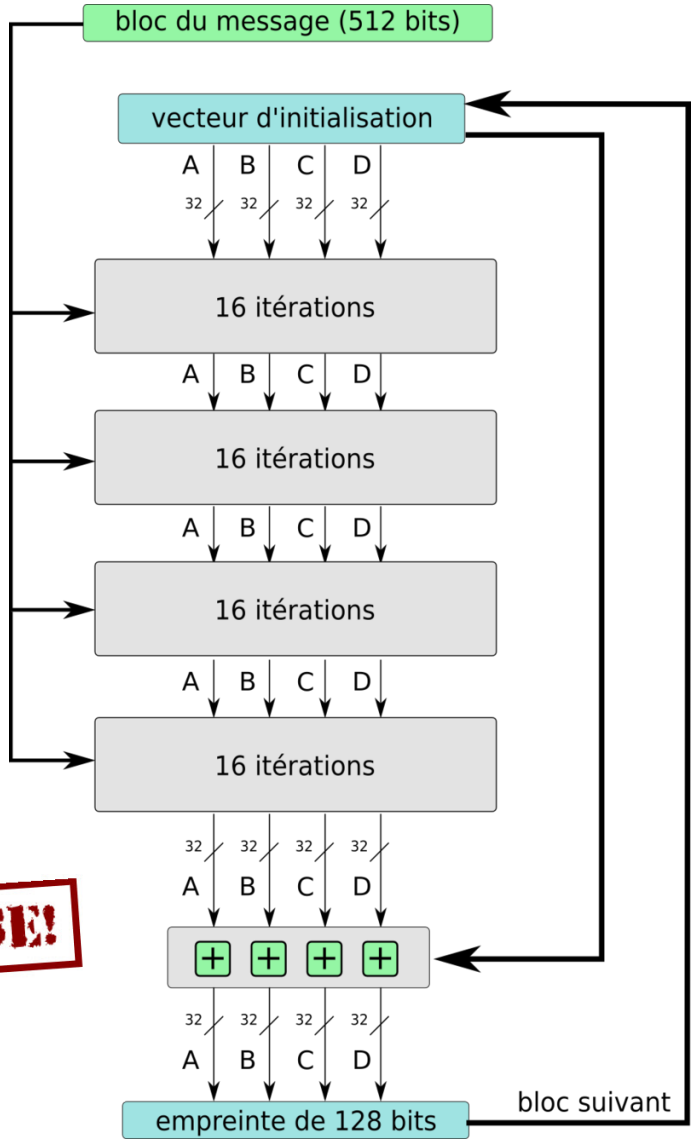
- L'utilisateur (Users) :
  - Expéditeur (Sender)
  - Destinataire (Receiver)
- Le cryptographe (Cryptographers),
- Le cryptanalyste (Cryptanalysts).

## > La notion de message et de chiffrement

- Message clair (Plaintext ou cleartext)
- Message chiffré (Ciphertext)
- Chiffrement (Encryption)
- Déchiffrement (Decryption) :
  attention != Décryptage

## > Les algo. cryptographiques

- Cryptographic algorithm
- Cipher





Vue générale des opérations de l'algorithme MD5 →

## > Les clefs cryptographiques

- Cryptographic keys

Exemple d'une clef de chiffrement RSA 2048 bits →

```
#
# 2048 bit OpenVPN static key
#
-----BEGIN OpenVPN Static key V1-----
599b28148d93279646374407 96b6ab23
e6b9ba0b0c7ad4d0ea950ed38da37014
63f03f629ce529c07a986efd6e5b8e48
de56f59d54966329662fe20e35b73b50
fb9528f017e7db367a05bdeb2c1ecb6d
4666a86b7e8bb1055fc698fb6f6ea320
90b555aea1d2c05cb8fdd584f7329a92
df48f77dbf44b63f6361beda6155d466
2cc0def1358ad2eccab817b5c8ac8e3d
e2459ba76cfff7ce63b6f7599b8e9042
57d56f940f1c398e1936e17e19ae274f
a104673d3fac868fa345c9fd6bb4d9fd
6e23efd604e1836ee396e3938cf1cf50
f576233f8ba6d3283b6d738135c284d7
84c5543671f711fc501169affe8f7dfd
ab599fcbc1b51720fe52bf7e82c1d96d
-----END OpenVPN Static key V1-----
```

## > One-time Pad

- Clefs cryptographiques à une seule utilisation.
- Clef de même longueur que le message.
- Technique de chiffrement parfaite si :
  - La clef n'est effectivement **utilisée qu'une seule fois ET**,
  - Qu'elle est générée de façon **parfaitement aléatoire**.

## > One-time Pad (exemple)

Soit :

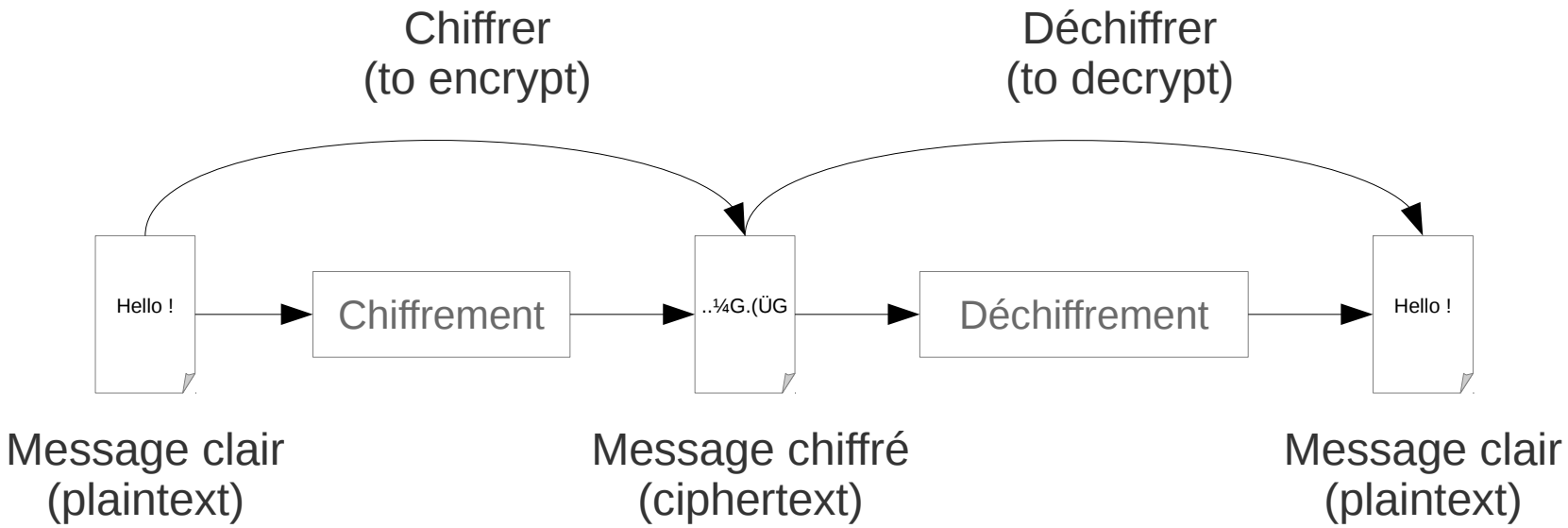
- `ONETIMEPAD` le message,
- `TBFRGFARFM` la clef de chiffrement.

Alors : **`IPKLPSFHGQ`** est le message chiffré.

Car :

- `O + T mod 26 = I`
- `N + B mod 26 = P`
- ...

## > Principe de fonctionnement général



## > D'un point de vue plus mathématique :

Soit :

- M le message clair,
- C le message chiffré,
- E la fonction de chiffrement,
- D la fonction de déchiffrement

Alors :

$E(M)=C$ et $D(C)=M$

et $D(E(M))=M$

Si :

$D(E(M))=M$ et $D=E$ alors l'algorithme de chiffrement est dit **réversible**

## > La fonction Booléenne XOR (« ou exclusif »)

Soit :

- A et B deux opérandes,
- 1 signifie VRAI,
- 0 signifie FAUX

La table de vérité de A⊕B est :

| A | B | A⊕B |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

## > Quelques propriétés importantes de XOR :

Soit :

- A et B deux opérandes,
- 1 signifie VRAI,
- 0 signifie FAUX

Alors :

- $A \oplus A = 0$
- $A \oplus \bar{A} = 1$
- $A \oplus 1 = \bar{A}$
- $A \oplus 0 = A$
- $\oplus$ dispose de la commutativité et de l'associativité.

Partie 2

# LES PRINCIPAUX SYSTÈMES CRYPTOGRAPHIQUES

# PLAN DU COURS

**II.Les principaux systèmes cryptographiques :**

- Les fonctions de hachage (One Way Hash Functions)
  - *Principe de fonctionnement*
  - *Principaux algorithmes*
  - *A quoi servent les fonctions de hachage ?*
- La cryptographie symétrique :
  - *Principe de fonctionnement*
  - *Les différents types d'algorithmes*
  - *Différents types de modes d'opération*
  - *A quoi sert la cryptographie symétrique ?*
- Le cryptographie asymétrique, ou à clef publique
  - *Principe de fonctionnement*
  - *Le chiffrement avec la cryptographie asymétrique*
  - *La signature électronique avec la cryptographie asymétrique*
  - *Principaux algorithmes actuelles*

# Les fonctions de hachage (One Way Hash Functions)

## > Principe de fonctionnement

- Fonction de prise d'empreinte numérique,
- A partir d'une donnée quelconque, l'algorithme génère une donnée :
  - De **longueur fixe**,
  - **Représentative de la donnée initiale.**
  - On appelle cette donnée `h` **empreinte** (**hash**).
- L'algorithme ne permet pas retrouver la donnée initiale à partir de l'empreinte. (irréversibilité)

## > Principe de fonctionnement (d'un point de vue plus mathématique)

Soit :

- `M` le message clair,
- `H` la fonction de hachage,
- `h` l'empreinte de `M`.

Alors :

$$H(M) = h$$

Sachant `M` **il est très difficile de trouver** `M'!=M` tel que :

$$H(M) = H(M')$$

sinon on appelle cela **une collision.**

## > Principe de fonctionnement

- Renforcement par l'utilisation d'un sel (salt)
- L’idée est de concaténer la donnée avec une chaîne de caractère « aléatoire » pour complexifier le cassage de l’empreinte.

# LES PRINCIPAUX SYSTÈMES CRYPTOGRAPHIQUES
## Les fonctions de hachage (One Way Hash Functions)

### > Principaux algorithmes

- MD5 (Message Digest Algorithm) :

```
user1@Workstation1:~/Bureau$openssl md5 report.pdf
MD5(report.pdf)= 29f0f85d1297c2e3af9eff6147bd9a0f
```

- SHA (Secure Hash Algorithm) :

```
user1@Workstation1:~/Bureau$openssl sha1 report.pdf
SHA1(report.pdf)= b54a7ef09140ebcb7b7ba487e730ad12cd749d5e
```

```
user1@Workstation1:~/Bureau$openssl sha256 report.pdf
SHA256(report.pdf)= fc24163e3b599ac1795b72c3fda8b798fb0c0d9d8c72741e1048627837ba469c
```

**Donc il ne faut plus utiliser MD5 et SHA1**

## > A quoi servent les fonctions de hachage ?

- Contrôle d'Intégrité.

Aperçu de la zone de téléchargement d'une distribution linux :

| Koozali SME Server : STABLE releases | | | | |
|---|---|---|---|---|
| Version | DVD ISO | Checksum | | Netinstall ISO |
| **SME Server 9.2**<br>Home<br>Release notes<br>Addendum<br>EOL November 30th, 2020 | **i386** | MD5 | SHA1 | **i386** |
| | **x86_64** | MD5 | SHA1 | **x86_64** |

- Confidentialité

Extrait d'un fichier /etc/shadow :

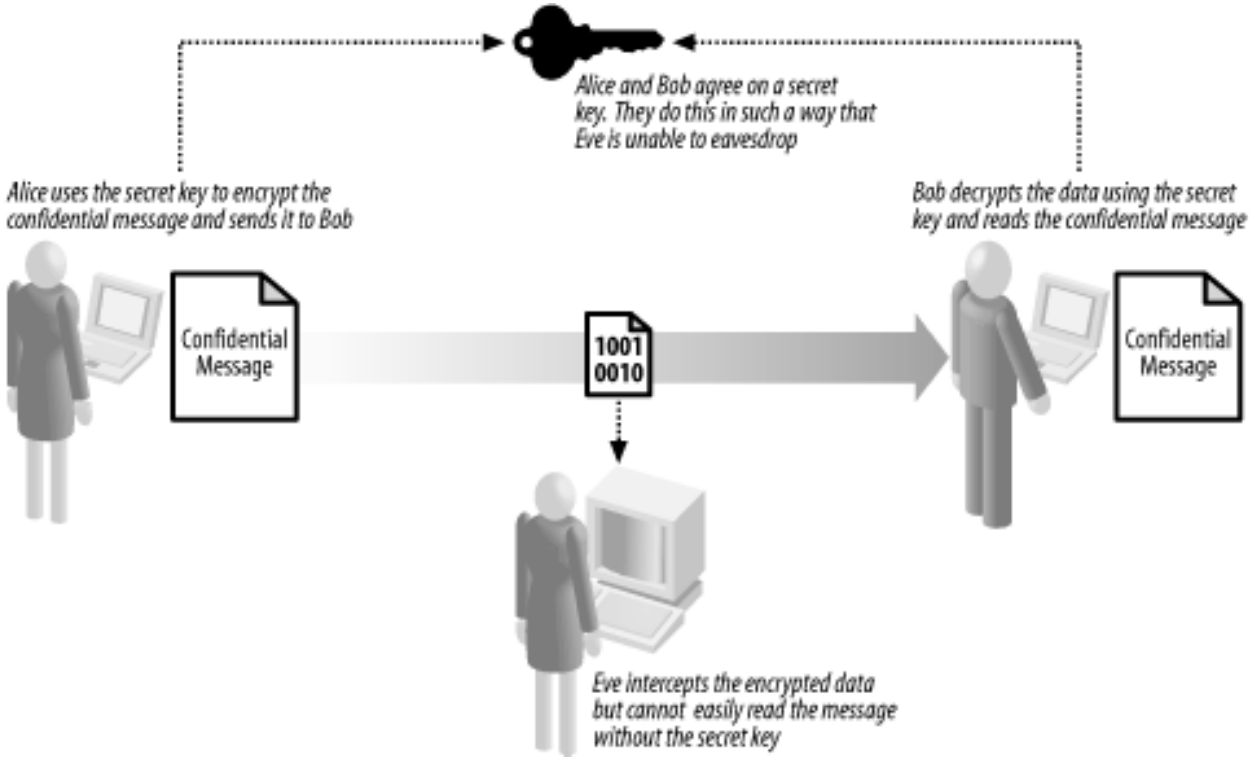
```
root:$1$934b4a210c17493f68bf6bfe74bff77a:16749:0:99999:7:::
fred:$1$9ebf8e708dcb3f28cb43d5d52655ab14:16561:0:99999:7:::
mysql:!:16939:0:99999:7:::
uuidd:*:16940:0:99999:7:::
giselle:$1$6e5fa4d9c48ca921c0a2ce1e64c9ae6f:17078:0:99999:7:::
libvirt-qemu:!:17105:0:99999:7:::
```

# La cryptographie symétrique

## > Principe de fonctionnement

- Consiste à chiffrer et déchiffrer un message via :
  - La même clef **ET**,
  - Le même algorithme.
- Les partenaires partagent donc une clef dite **secrète**.

## > Principe de fonctionnement (en image)




source : http://etutorials.org

## > Principe de fonctionnement (d'un point de vue plus mathématique)

Soit :

- $M$ le message clair,
- $C$ le message chiffré,
- $E$ la fonction de chiffrement,
- $D$ la fonction de déchiffrement
- $K$ la clef secrète

Alors :

$$E_K(M) = C \quad \text{et} \quad D_K(C) = M$$

$$\text{et } D_K(E_K(M)) = M$$

## > Les différents types d'algorithmes :

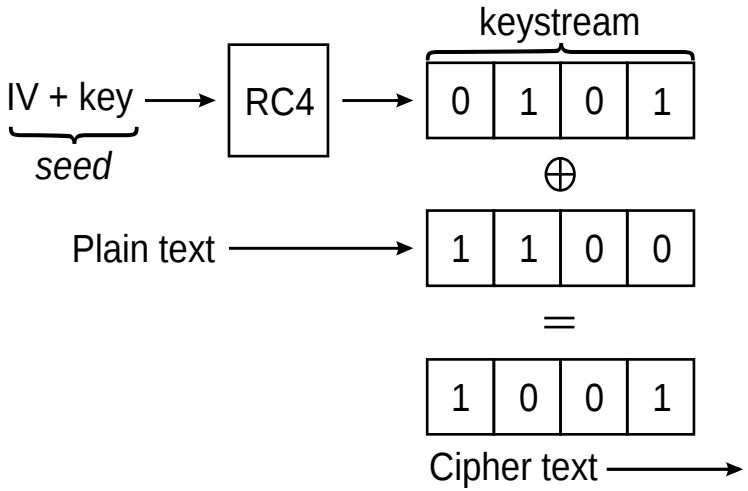
- Algorithmes de chiffrement par flux (Stream Ciphers),
- Algorithmes de chiffrement par bloc (Block Ciphers).

## > Les algorithmes cryptographiques symétriques par flux

- Message à chiffrer de **longueur quelconque**,
- Chiffré **bit par bit**, ou **octet par octet**. (dans certains cas en mot de 32bits)

## > Les algorithmes cryptographiques symétriques par flux

- Génération d'une valeur « pseudo aléatoire » appelée **Keystream** à partir d'une clef *(**seed**).*




- **Keystream** utilisé ensuite pour chiffrer les données

  → en général avec l'opération binaire XOR

> Principaux algorithmes symétrique par flux:

- RC4
- SEAL
- Utilisé également dans PKZIP

## > Les algorithmes cryptographiques symétriques par bloc

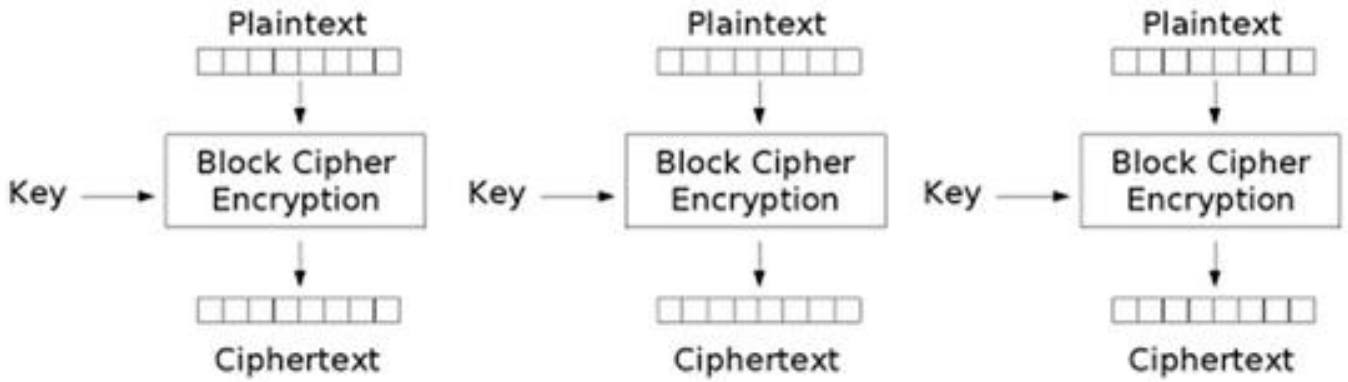
- Message à chiffrer de **longueur quelconque**,
- Le message est découpé en blocs de taille fixe.
- Le dernier bloc peut être comblé pour atteindre la taille du bloc. (**Padding**)
- ...

## > Les algorithmes cryptographiques symétriques par bloc (suite)

- ...
- Chaque bloc clair donnera **toujours** le même bloc chiffré,
- On peut traiter les blocs de différente façon : **modes d'opération**.

## > Le mode d'opération ECB (Electronic Codebook)

- Chiffrement (déchiffrement) :

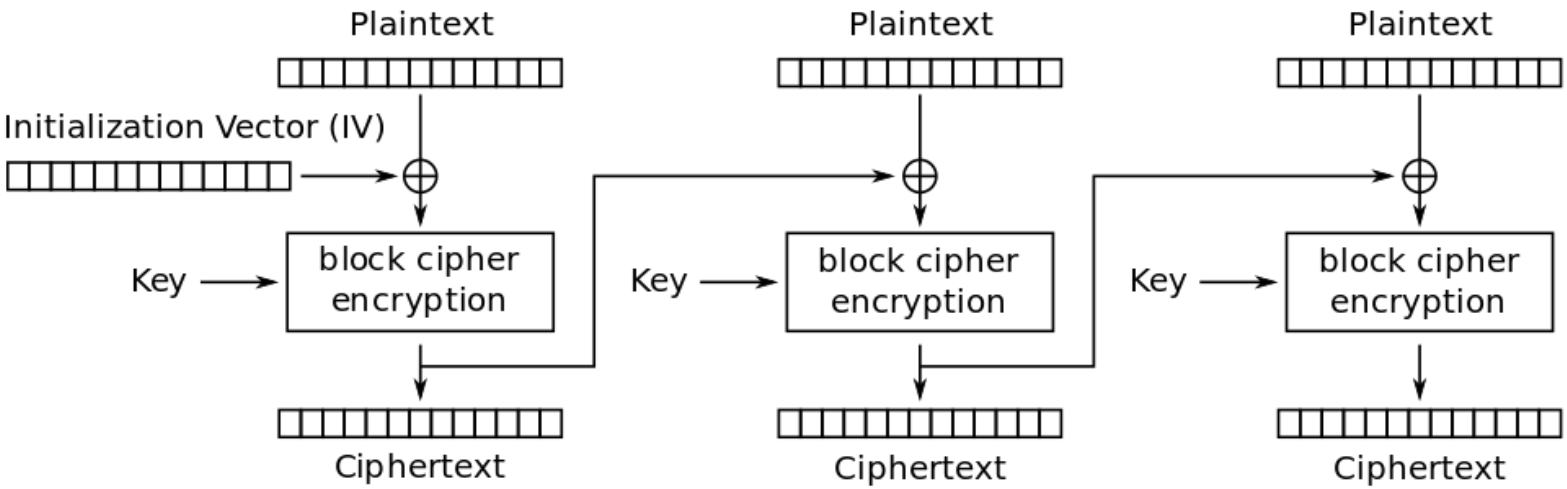



Electronic Codebook (ECB) mode encryption

DO NOT USE!

## > Le mode d'opération CBC (Cipher Block Chaining)

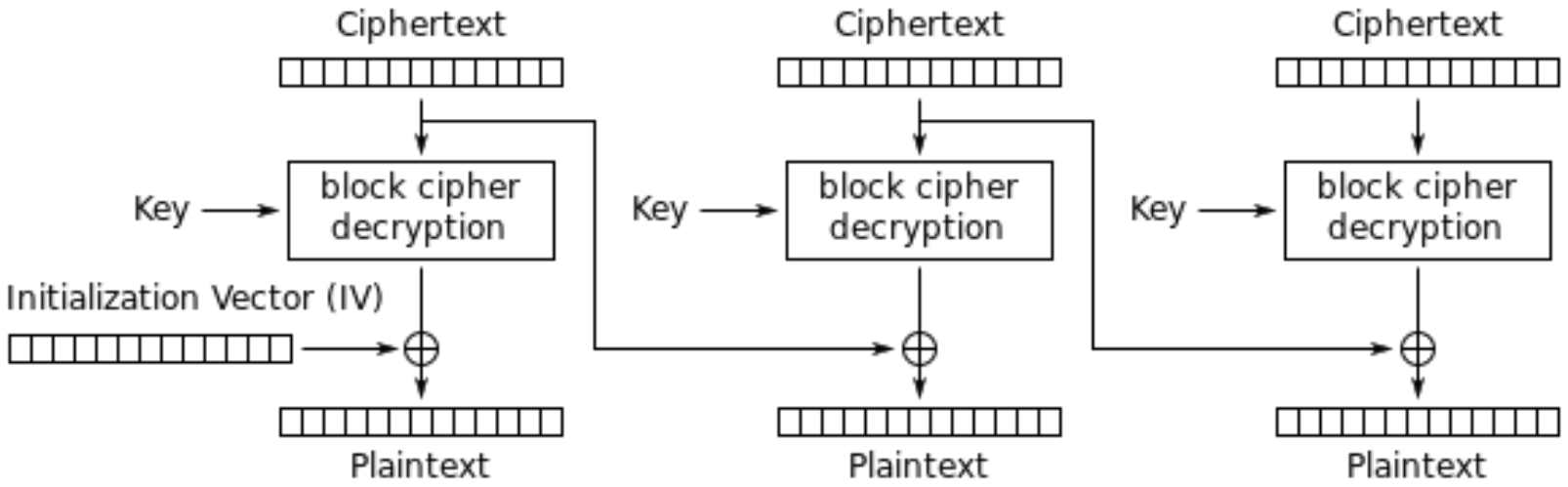
- Chiffrement :




Cipher Block Chaining (CBC) mode encryption

## > Le mode d'opération CBC (Cipher Block Chaining)

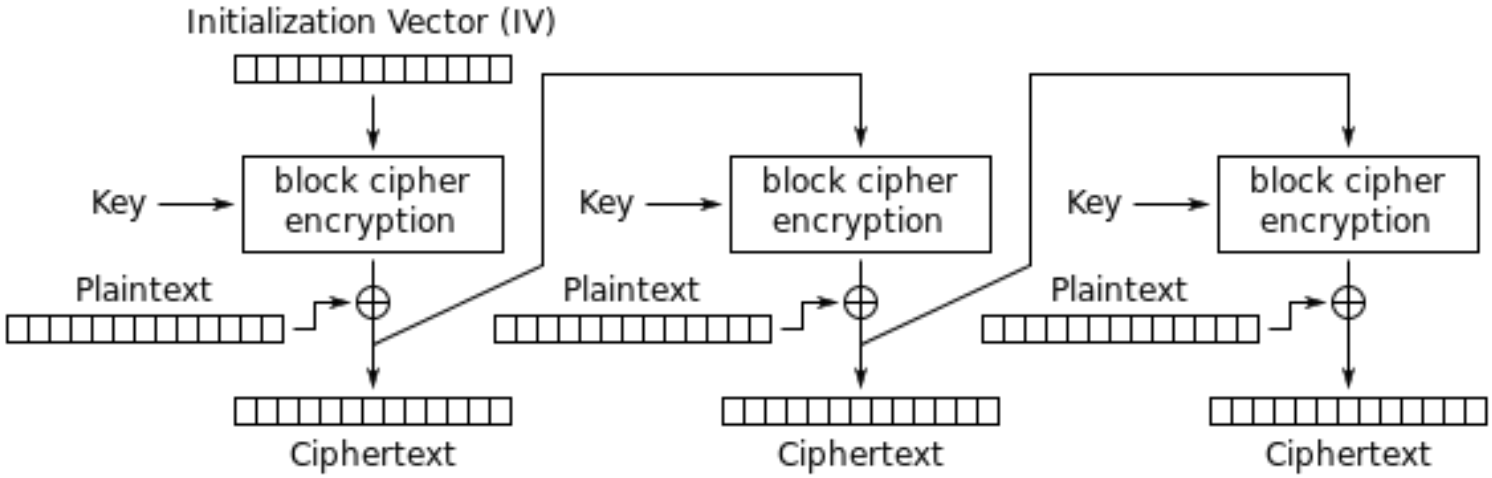
- Déchiffrement :




Cipher Block Chaining (CBC) mode decryption

## > Le mode d'opération CFB (Cipher Feedback)

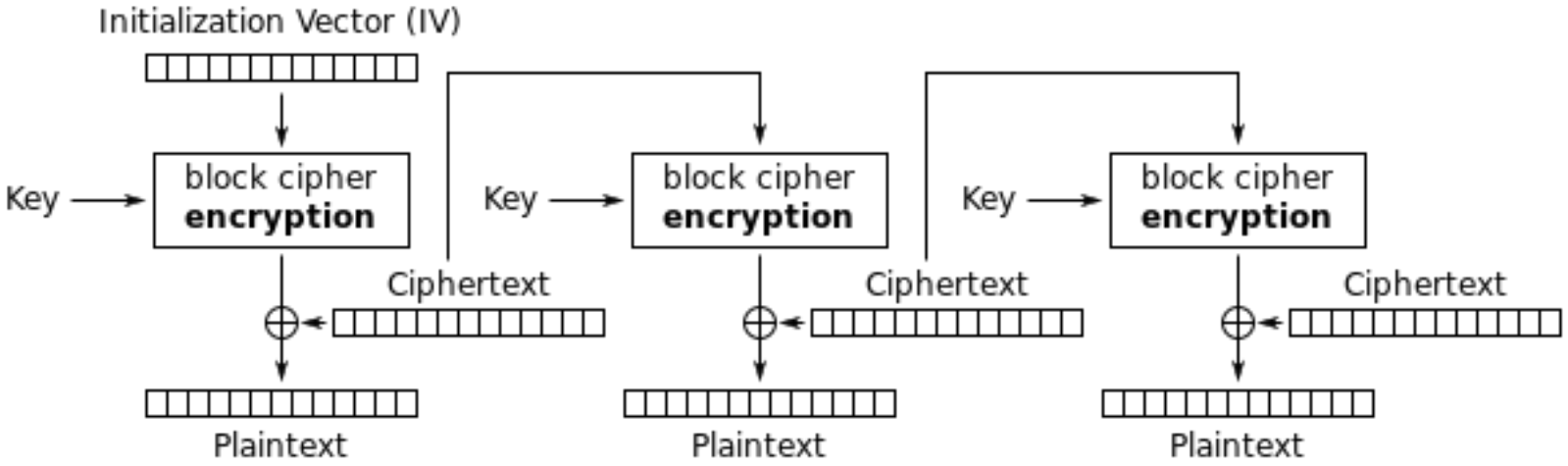
- Chiffrement :




Cipher Feedback (CFB) mode encryption

## > Le mode d'opération CFB (Cipher Feedback)

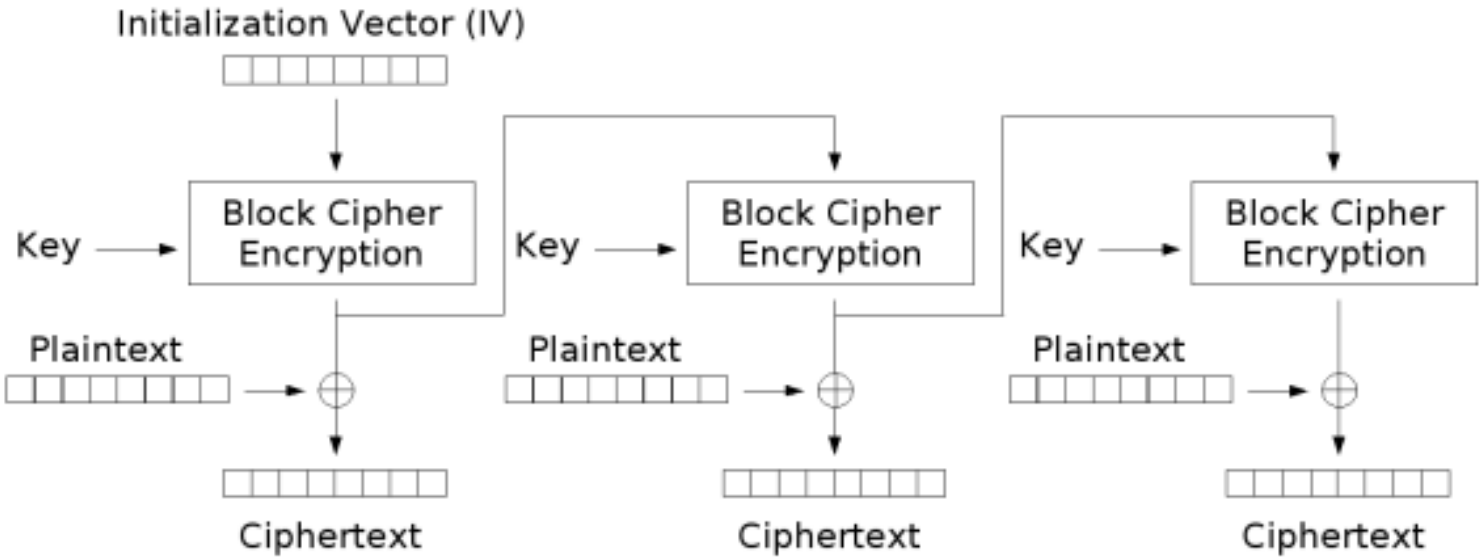
- Déchiffrement :




Cipher Feedback (CFB) mode decryption

## > Le mode d'opération OFB (Output Feedback)

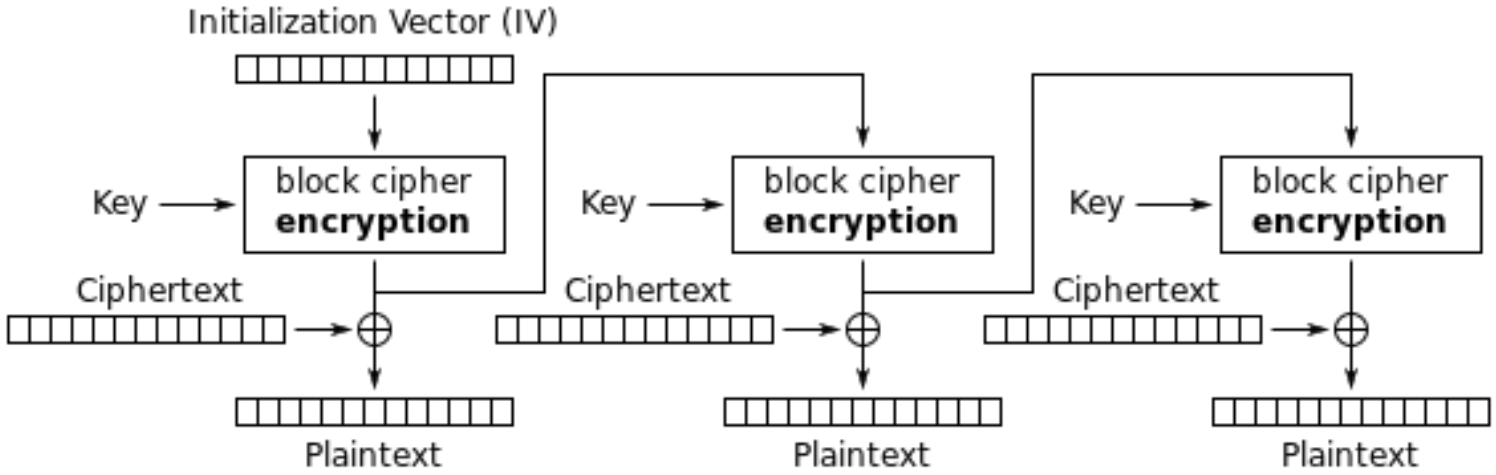
- Chiffrement :




Output Feedback (OFB) mode encryption

## > Le mode d'opération OFB (Output Feedback)

- Déchiffrement :




Output Feedback (OFB) mode decryption

## > A propos du Vecteur d'Initialisation (IV)

- Bloc de données aléatoire utilisé pour démarrer le chiffrement du premier bloc
- Ajoute une notion de hasard au chiffrement
- /!\ ne pas utiliser le même IV avec deux clefs différentes !
- Pas nécessaire de chiffrer l'IV, par contre :
  - Sa génération doit être aléatoire,
  - L'ensemble des IV doit être suffisamment grand !

> Principaux algorithmes symétriques par blocs :

- DES, 3DES (Data Encryption Standard) : **A bannir**
- AES (Advanced Encryption Standard)
- IDEA
- Blowfish
- SAFER
- …

**Il est recommandé d'utiliser AES-256-CBC**

> Garantir l'intégrité grâce à la cryptographie symétrique

- MAC (Message Authentication Code),
- HMAC (keyed-Hash Message Authentication Code)

## > MAC : principe de fonctionnement

- Génération d'une valeur appelée souvent **tag.**
- Le **tag** sert à authentifier le message.
- Il garantit son **intégrité** et son **authenticité** (authentification de l'expéditeur).

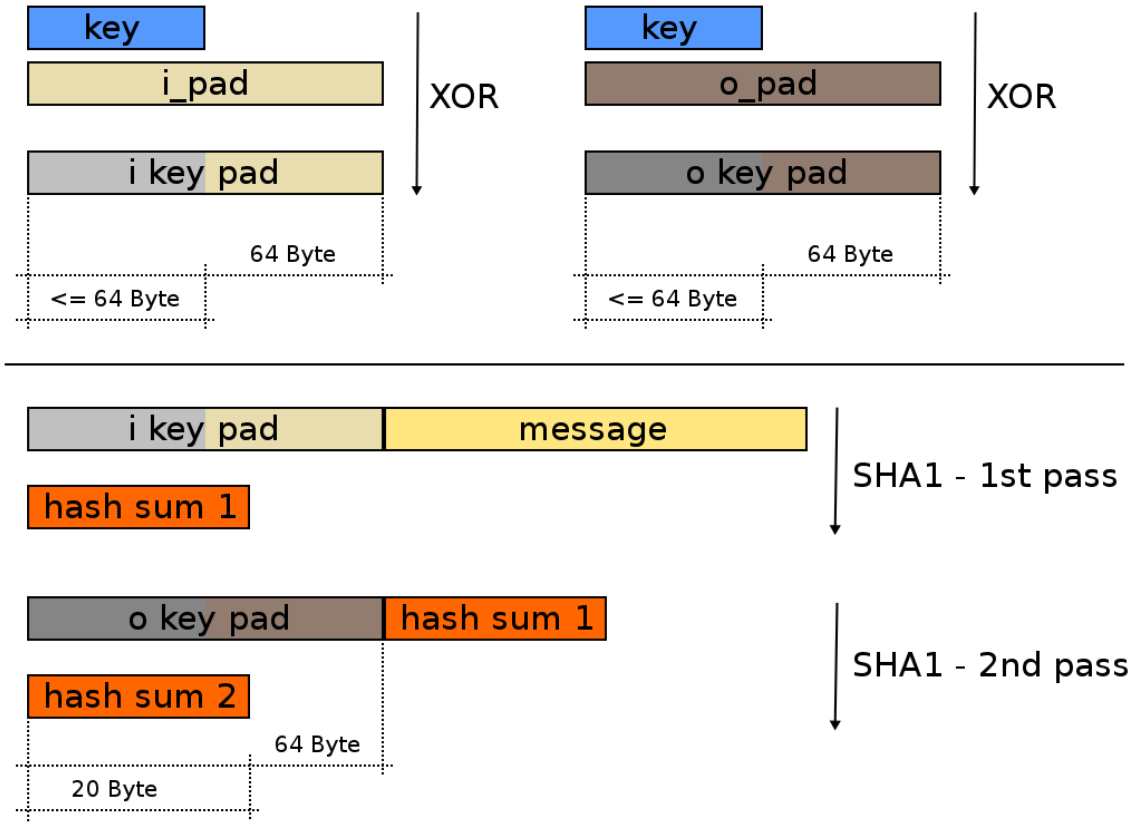
## > MAC : principe de fonctionnement (suite)

- Génération d'une clef secrète partagée.
- Signature du message avec la clef.
- Vérification de la signature à l'aide :
  - Du **message**,
  - Du **tag**, et
  - De la **clef**.

## > HMAC : principe de fonctionnement

- Fonctions MAC particulières,
- Génération d'une clef secrète dérivée ensuite pour générer deux secrets.
- Ensuite deux passes de fonction de hachage salées par ces deux secrets.
- Nom de l'algorithme :
  - HMAC-{Fonction_de_hachage}
  - Ex : HMAC-SHA1

## > HMAC : principe de fonctionnement (en image)




Algorithme de génération d'un HMAC SHA1 (source : wikipedia)

## > HMAC : génération d'un HMAC-SHA1 avec openssl

```
user1@Workstation1:~/Bureau$echo -n "Ceci est un message secret" | openssl dgst -sha1
-hmac "secretKey"
(stdin)= 86a3dc985a6bfa8739f4af6bf5f254d3226ffeaa
```

- `dgst` indique l'algorithme de hachage à utiliser lors la génération du HMAC.
- `-hmac` indique de créer un MAC « haché » en utilisant la clef `"secretKey"`.

OpenSSL
Cryptography and SSL/TLS Toolkit

> A quoi sert la cryptographie symétrique ?

- Authentification
- Confidentialité

> Possibilité de combiner avec MAC :

- Garantir en plus l'intégrité du message.

> Avantages ...

- Niveau de confidentialité élevé si le choix des algorithmes et de la taille des clefs est correctement pensé,
- Chiffrement et déchiffrement « relativement rapide »

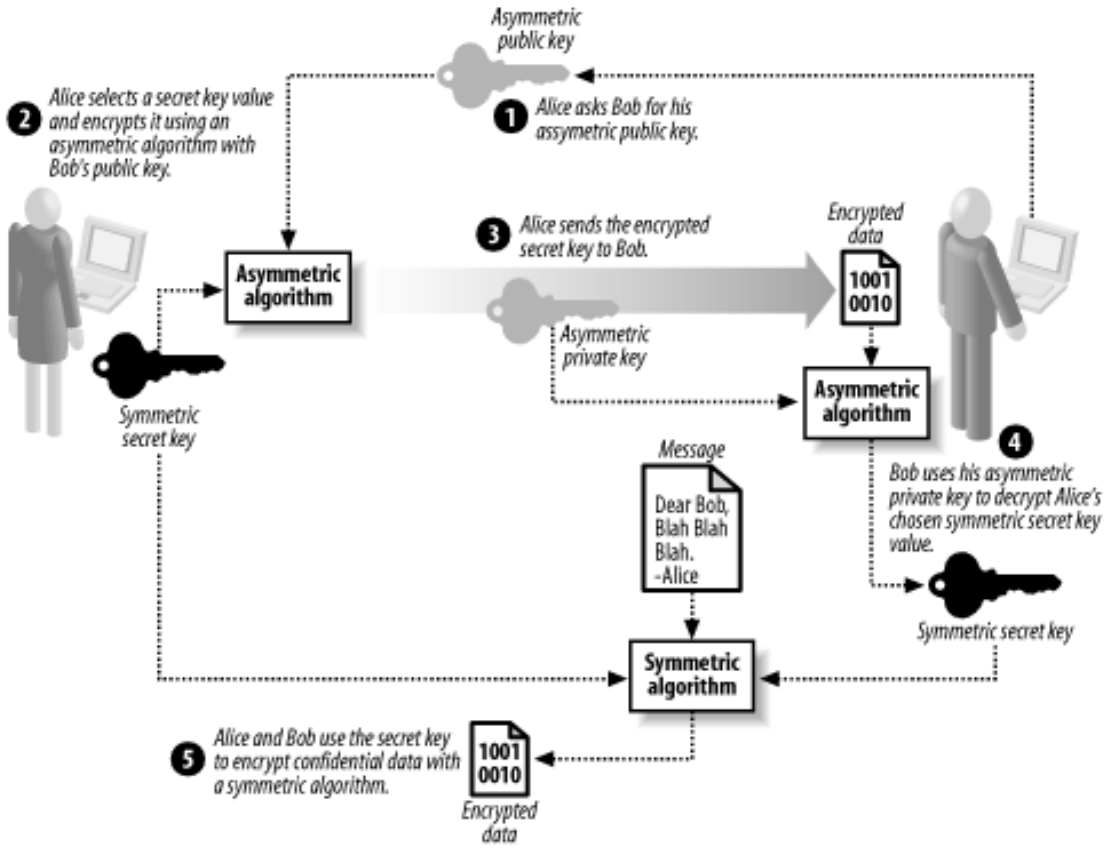
> et inconvénients :

- Le nombre de clefs augmente avec le nombre d'interlocuteur.

  `n(n - 1)/2 clefs` nécessaires où `n` est le nombre de participants.
- **Si l'effectif est trop élevé, cela devient rapidement ingérable**.
- L'**échange de la clef** est un point déterminant de la sécurité des échanges. Il est alors nécessaire d'utiliser un algorithme d'échange de clefs pour sécurisé ce point là. (par exemple Diffie-Hellman)

# LES PRINCIPAUX SYSTÈMES CRYPTOGRAPHIQUES
## La cryptographie symétrique

> L'échange de clef symétrique Diffie-Hellman (en image)




source : http://etutorials.org

# La cryptographie asymétrique
# (Ou à clef publique)

## > Principe de fonctionnement

- Chaque participant dispose d'une paire de clefs (biclef)
  - Une clef privée
  - Une clef publique
- L'une sert à chiffrer, l'autre à déchiffrer.
- Le même algorithme est utilisé pour le chiffrement et le déchiffrement.

## > Principe de fonctionnement (suite)

- Repose essentiellement sur des problèmes mathématiques complexes :
  - factorisation d'un nombre entier formé de grands facteurs premiers,
  - résolution d'un logarithme discret sur un corps fini,
  - résolution d'un logarithme discret sur une courbe elliptique.

# LES PRINCIPAUX SYSTÈMES CRYPTOGRAPHIQUES
## La cryptographie asymétrique (à clef publique)

> Deux applications :

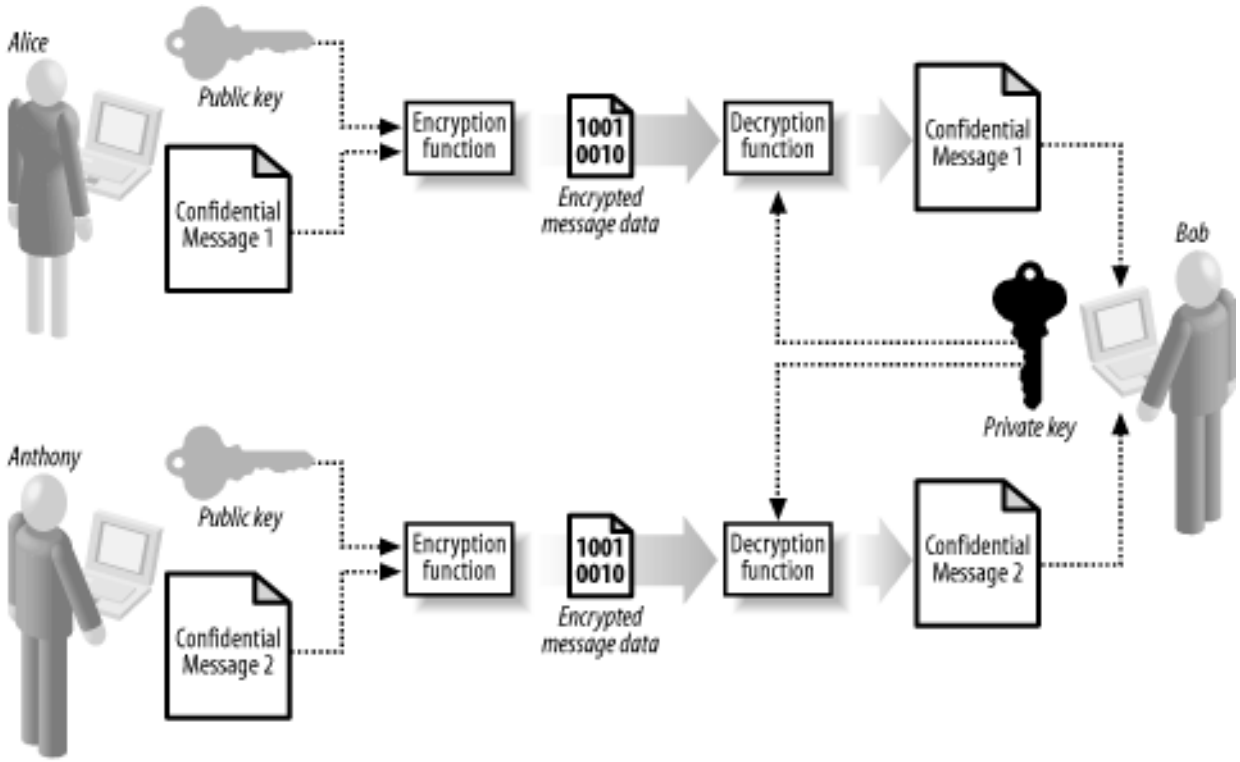
- Chiffrement,
- La signature électronique.

> Le chiffrement avec la cryptographie asymétrique :

- Alice chiffre son message avec **la clef publique de Bob,**
- Alice envoie le message chiffré à Bob.
- Bob déchiffre le message de Alice avec **sa propre clef privée**.

# LES PRINCIPAUX SYSTÈMES CRYPTOGRAPHIQUES
## La cryptographie asymétrique (à clef publique)

> Le chiffrement avec la cryptographie asymétrique :



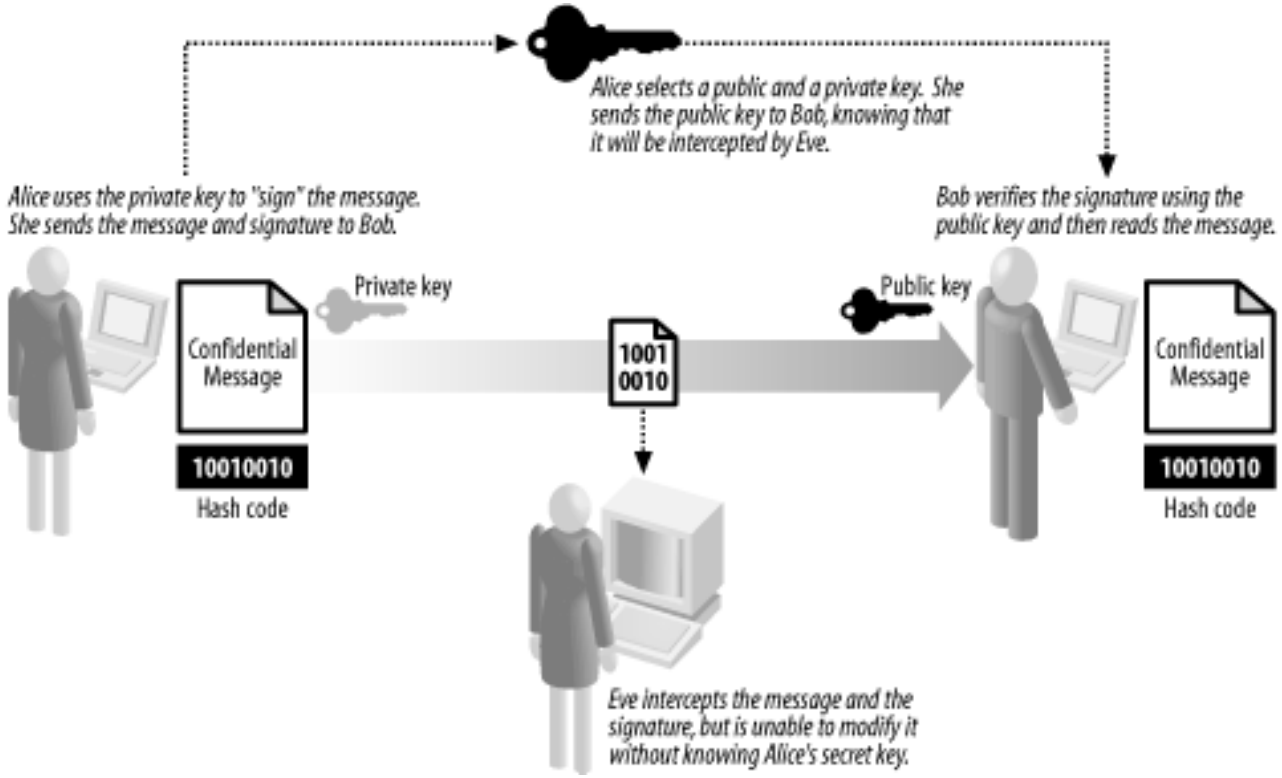

source : http://etutorials.org

> La signature électronique avec la cryptographie asymétrique :

- Alice chiffre son message avec **sa propre clef privée**,
- Alice envoie le message, ainsi que la signature à Bob.
- Bob déchiffre la signature avec **la clef publique de Alice**.

# LES PRINCIPAUX SYSTÈMES CRYPTOGRAPHIQUES
## La cryptographie asymétrique (à clef publique)

> La signature électronique avec la cryptographie asymétrique :




source : http://etutorials.org

# LES PRINCIPAUX SYSTÈMES CRYPTOGRAPHIQUES
## La cryptographie asymétrique (à clef publique)

> Principaux algorithmes asymétriques :

- RSA (Rivest Shamir Adleman),
- ElGamal
- Systèmes à courbes elliptiques,
- DSA (Digital Signature Algorithm) (uniquement pour la signature électronique),
- ...

**Il est recommandé d'utiliser RSA avec des clefs de 2048 bits.**

## > Un exemple d'algorithme asymétrique : RSA

- Fondé en 1977 par Ron Rivest, Adi Shamir et Leonard Adleman.
- Basé sur la complexité de factoriser des grands nombres. (Problématique calculatoire)

## > A propos de RSA, détermination de la clef publique :

- On choisi `n` produit de deux nombres premiers `p` et `q` (quoi doivent rester secrets), donc `n = p.q .`
- On choisi `e` tel que `e` soit premier avec `(p-1)` et `(q-1) .`
- Le nombre `n` est appelé **modulus** de la clef.
- Le nombre `e` est appelé **exposant publique** de la clef.
- Le couple `(n,e)` représente la **clef publique**.

> A propos de RSA, calcul de la clef privée :

- L'entier $d$ est alors calculé avec l'algorithme d'Euclide tel que $d = e^{-1} \bmod ((p-1)(q-1))$
- L'entier $d$ est la **clef privée**.

# LES PRINCIPAUX SYSTÈMES CRYPTOGRAPHIQUES
## La cryptographie asymétrique (à clef publique)

> Génération d'une clef privée RSA avec openssl :

```
user1@Workstation1:~/Bureau$openssl genrsa -out testRSA.priv 2048
```

> Export d'une clef publique correspondante avec openssl :

```
user1@Workstation1:~/Bureau$openssl rsa -in testRSA.priv -pubout -out testRSA.pub
```

# LES PRINCIPAUX SYSTÈMES CRYPTOGRAPHIQUES
## La cryptographie asymétrique (à clef publique)

> Composantes de la clef publique :

```
user1@Workstation1:~/Bureau$openssl rsa -in testRSA.pub -pubin -text
Public-Key: (2048 bit)
Modulus:
    00:cc:38:96:49:b9:92:bd:d3:d7:ee:88:0b:e6:7a:
    b8:1b:73:26:76:fe:a3:0d:57:06:aa:32:05:22:ff:
    c8:b7:a9:d5:8e:dc:00:d7:96:73:2f:ab:b8:dd:f1:
    bc:ca:56:67:94:69:64:b7:98:9b:48:fe:62:f0:9f:
    c0:d1:ec:86:b7:d4:9f:6f:a0:77:25:91:cd:76:f1:
    b1:cf:4f:71:7a:ae:7a:c5:20:70:24:af:31:74:58:
    97:4d:eb:98:93:e6:bc:54:ce:98:6f:60:bf:db:2b:
    b8:fd:bc:ed:c4:bb:3f:03:72:9f:b0:92:0c:d6:b4:
    27:01:ef:19:d0:c9:55:74:0c:ee:c0:a6:c8:00:26:
    2f:54:70:f2:18:24:12:76:2a:1e:b8:71:79:60:8c:
    8e:ab:46:c6:12:82:a8:5a:04:a9:5e:52:c2:f8:19:
    f6:d3:85:f7:ef:e7:95:b6:bb:f9:59:9f:23:30:aa:
    52:c1:d0:c9:fb:a8:7d:f3:25:8e:10:d6:37:86:08:
    c0:ab:0a:2c:8f:5e:6e:47:fc:b2:ba:e0:db:a9:a6:
    bf:ca:9a:da:e7:d5:43:06:12:b9:cc:f0:79:11:32:
    3d:51:09:d8:f1:61:2c:2d:0f:b9:03:31:8e:b1:ce:
    73:f3:b2:4b:10:51:ec:69:b5:fa:ec:39:0d:e6:d1:
    48:7f
Exponent: 65537 (0x10001)
writing RSA key
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAzDiWSbmSvdPX7ogL5nq4
G3Mmdv6jDVcGqjIFIv/It6nVjtwA15ZzL6u43fG8ylZnlGlkt5ibSP5i8J/A0eyG
t9Sfb6B3JZHNdvGxz09xeq56xSBwJK8xdFiXTeuYk+a8VM6Yb2C/2yu4/bztxLs/
A3KfsJIM1rQnAe8Z0MlVdAzuwKbIACYvVHDyGCQSdioeuHF5YIyOq0bGEoKoWgSp
XlLC+Bn204X37+eVtrv5WZ8jMKpSwdDJ+6h98yWOENY3hgjAqwosj15uR/yyuuDb
qaa/ypra59VDBhK5zPB5ETI9UQnY8WEsLQ+5AzGOsc5z87JLEFHsabX67DkN5tFI
fwIDAQAB
-----END PUBLIC KEY-----
```

## > A propos de RSA, chiffrement :

Soit :

- $M$ le message clair,
- $C$ le message chiffré,
- $e$ l'exposant publique,
- $n$ le modulus.

Alors :

$$C = M^e \bmod n$$

## > A propos de RSA, déchiffrement :

Soit :

- $M$ le message clair,
- $C$ le message chiffré,
- $d$ la clef privée,
- $n$ le modulus.

Alors :

$$M = C^d \bmod n$$

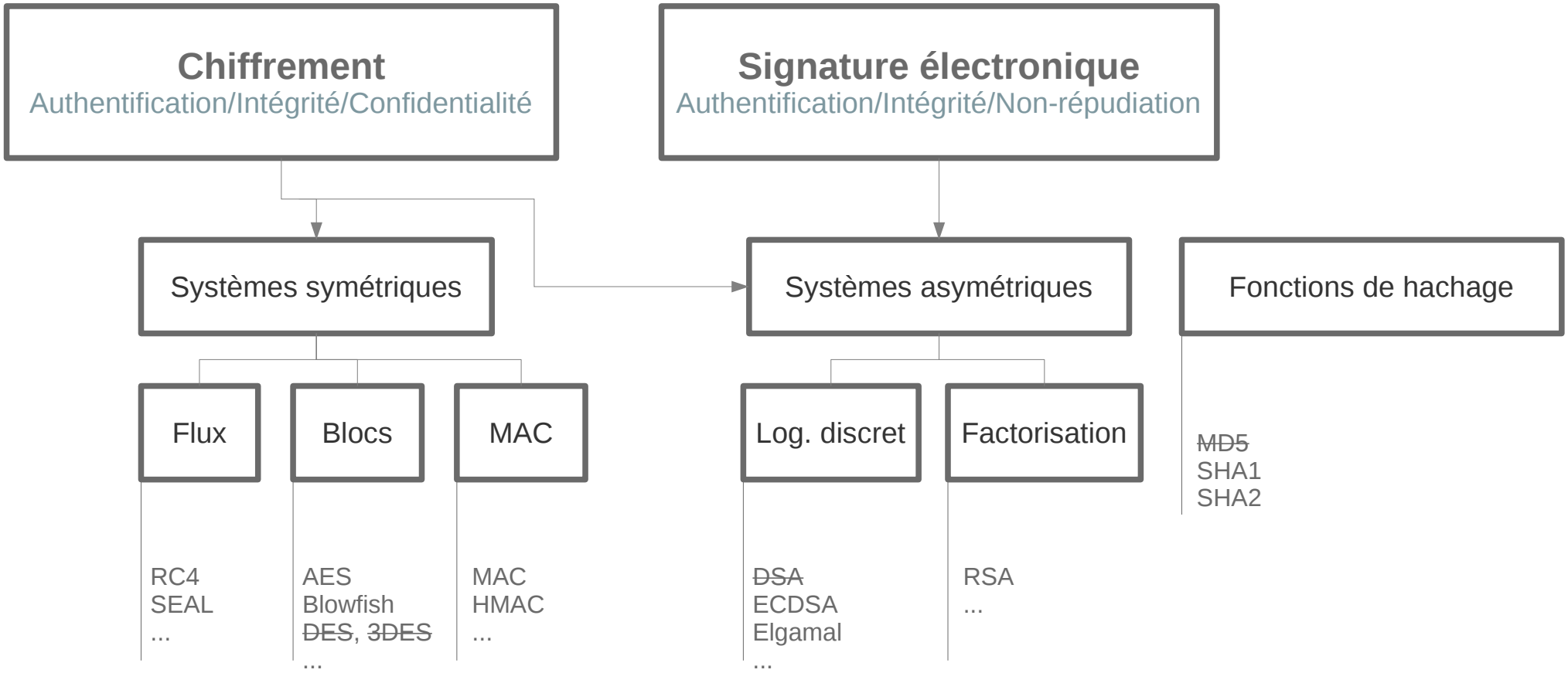
# LES PRINCIPAUX SYSTÈMES CRYPTOGRAPHIQUES
## La cryptographie asymétrique (à clef publique)

> A quoi sert la cryptographie asymétrique ?

- Authentification,
- Confidentialité,
- Intégrité,
- Non-répudiation.

# LES PRINCIPAUX SYSTÈMES CRYPTOGRAPHIQUES
En bref !

## > La cryptographie en bref ...



# LES PRINCIPAUX SYSTÈMES CRYPTOGRAPHIQUES

Merci pour votre attention !

> Des questions ?

# INTRODUCTION À LA CRYPTOGRAPHIE APPLIQUÉE

Thierry MEYER Consultants, est un cabinet de conseil, audit, et expertise technique spécialisé en sécurité des systèmes d'information depuis sa création en  2005.

contact@tm-consultants.fr

@Th1tux