# Modes Opératoires et Authentification Symétrique

Alexandre Duc

# Modes Opératoires

- Un algorithme de **chiffrement par flot** permet de chiffrer des données de **n'importe quelle longueur**.
- Un algorithme de **chiffrement par bloc** ne peut que chiffrer **64 ou 128 bits à la fois**, typiquement.

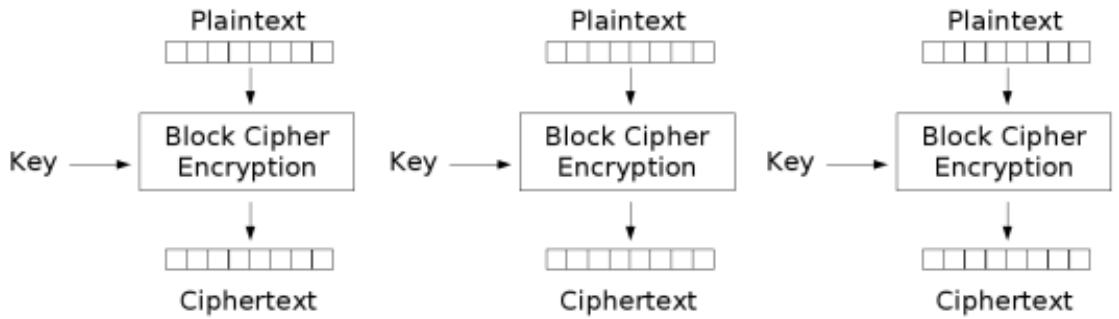
**Question**

Comment chiffrer plus de données avec un algorithme de chiffrement par bloc ?

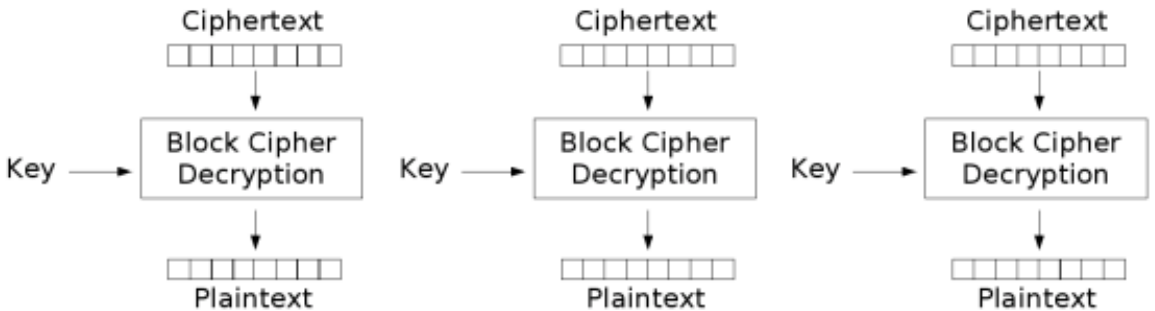
# Modes Opératoires

- Un **mode opératoire** est une façon d'utiliser un algorithme de chiffrement par bloc pour chiffrer des données plus grandes que la taille de son bloc.
- Modes opératoires classiques :
  - «Electronic Code Book» (ECB)
  - «Cipher Block Chaining» (CBC)
  - «Counter Mode» (CTR)
  - CFB, OFB, ...

# «Electronic Code Book»




Electronic Codebook (ECB) mode encryption




Electronic Codebook (ECB) mode decryption

Source des illustrations : http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation

# «Electronic Code Book»

## Sécurité

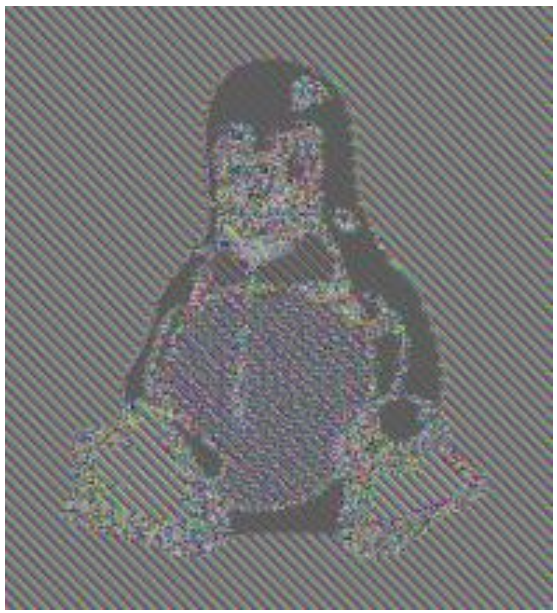
- Deux **blocs de textes clairs identiques** seront **chiffrés de manières identiques**. On a donc une perte potentielle de confidentialité.
- Un adversaire peut facilement introduire un ou plusieurs nouveaux blocs de texte chiffré de manière délibérée ou changer l'ordre des blocs durant la transmission. Le mode ECB résiste donc très mal aux attaques visant à modifier l'intégrité du texte chiffré.

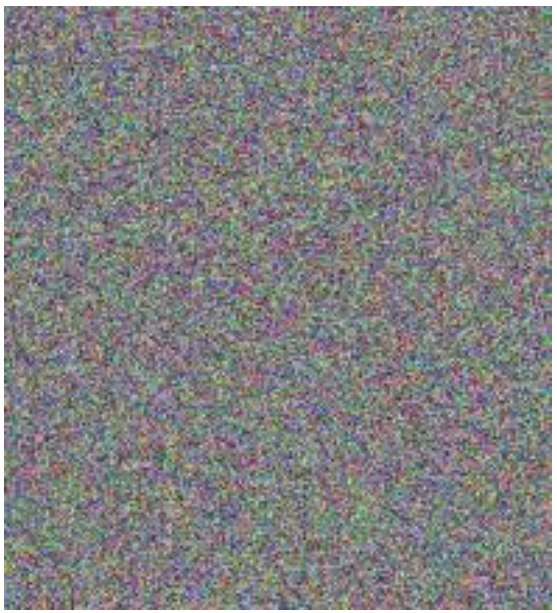
# «Electronic Code Book»

Perte de confidentialité avec ECB :

```
Chantemargue␣␣␣79'999.99
Cherpillod␣␣␣␣␣53'953.83
Guyot␣␣␣␣␣␣␣␣␣␣65'912.53
Germano␣␣␣␣␣␣␣␣35'900.99
Pellegrin␣␣␣␣␣␣53'123.45
Philipp␣␣␣␣␣␣␣␣49'049.71
Roger␣␣␣␣␣␣␣␣␣␣60'070.97
```

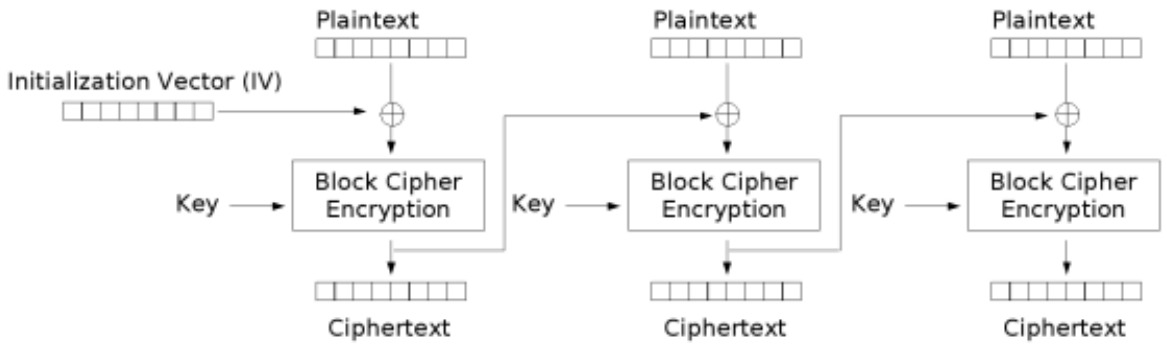
# «Electronic Code Book»





Source des illustrations : http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation

# «Cipher Block Chaining»




Cipher Block Chaining (CBC) mode encryption

## Question

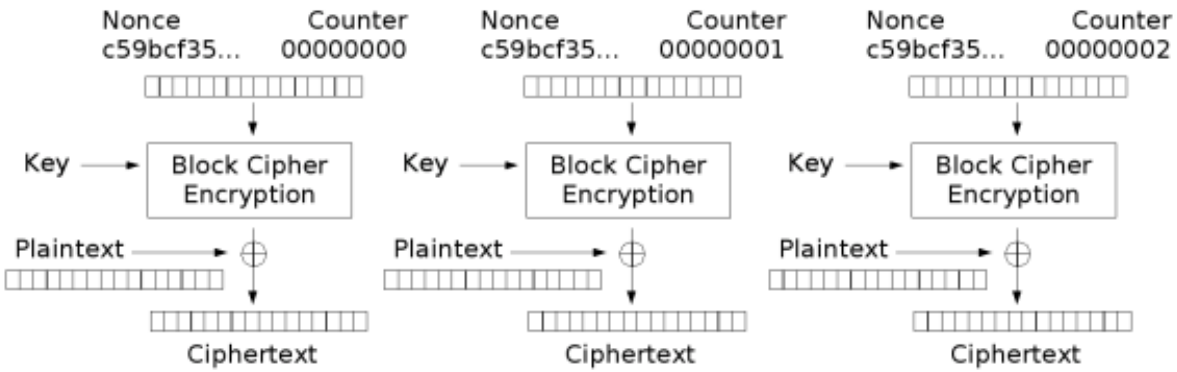
Comment déchiffrer ?

Source des illustrations : http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation

# «Cipher Block Chaining»

- Les données à chiffrer sont divisées en blocs de la taille du bloc de l'algorithme de chiffrement.
- Avant d'être chiffré, un bloc est combiné au moyen d'un XOR avec le texte chiffré précédent.
- Le premier bloc est combiné avec un **vecteur d'initialisation** (IV) **aléatoire**.
- L'IV peut être envoyé en clair.
- Chaque bloc de texte chiffré dépend de tous les précédents.
- Le chiffrement est séquentiel mais le déchiffrement peut être parallélisé.

# «Counter Mode»




Counter (CTR) mode encryption

## Question

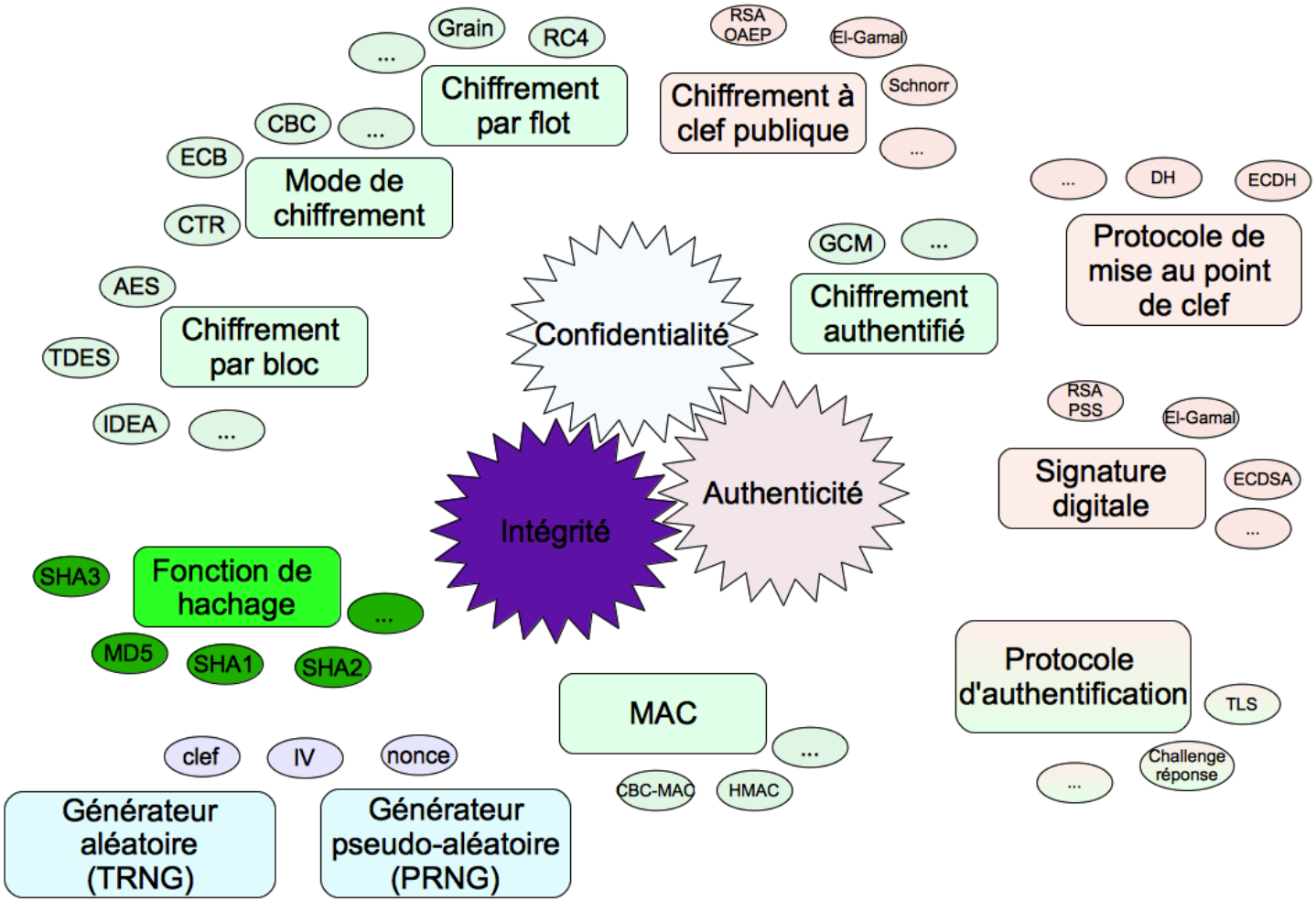
Comment déchiffrer ?

Source des illustrations : http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation

# «Counter Mode»

- Un compteur (à **usage unique**) est chiffré par l'algorithme de chiffrement par bloc, et est incrémenté pour chaque bloc.
- Le flux chiffré est XORé au texte clair.
- Il n'y a pas besoin de compléter les blocs de données si la taille des données n'est pas un multiple de la taille du bloc.
- La sécurité repose fortement sur la qualité du nonce/IV.
- L'IV/nonce peut être envoyé en clair.
- Mauvaise résistance à un adversaire attaquant l'intégrité d'un texte chiffré.

# Vue Synoptique



# Fonctions de Hachage

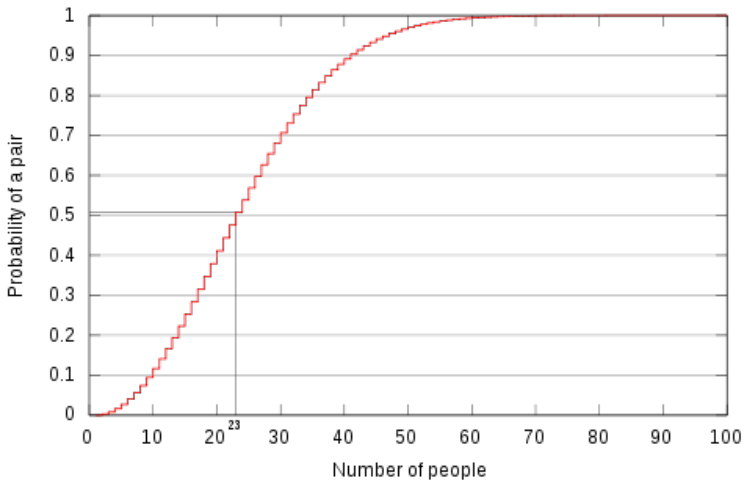
## Définition (Fonction de Hachage)

Une **fonction de hachage** cryptographiquement sûre est une fonction $H : \{0,1\}^* \longrightarrow \{0,1\}^\ell$ tel que :

- Le calcul de l'empreinte («digest») $h = H(m)$ d'un message $m$ est une opération qui peut se réaliser de manière **rapide**.
- Étant donnée une empreinte $h$, il est impossible en pratique de trouver un message $m$ tel que $h = H(m)$ (**résistance à la première préimage**).
- Étant donnée un message $m$ et son empreinte $h = H(m)$, il est impossible en pratique de trouver un message $m' \neq m$ tel que $h = H(m')$ (**résistance à la seconde préimage**).
- Il est impossible en pratique de trouver deux messages $m \neq m'$ tels que $H(m) = H(m')$ (**résistance aux collisions**).

# Paradoxe des Anniversaires

Quelle est la probabilité que deux personnes partagent la même date d'anniversaire dans un groupe de 23 personnes ?




Source de l'illustration : http://en.wikipedia.org/wiki/Birthday_problem

# Paradoxe des Anniversaires

- Plus précisément, Dans un espace de taille $d$, la probabilité d'avoir une collision avec $n$ tirage est approximativement de

$$1 - e^{-n^2/(2d)} .$$

- En d'autres termes, trouver avec une bonne probabilité une collision sur une empreinte de $\ell$ bits coûte au plus seulement $2^{\frac{\ell}{2}}$ évaluations de la fonction de hachage.

# Coût Générique des Différentes Attaques

Soit $H : \{0,1\}^* \longrightarrow \{0,1\}^\ell$ une fonction de hachage.

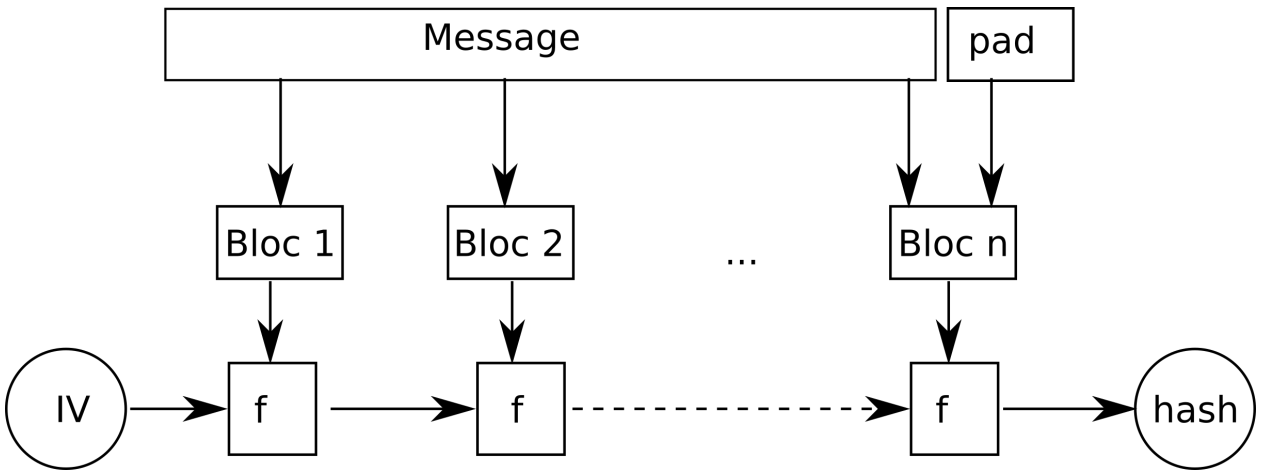
- Une attaque de type **première préimage** coûte $O(2^\ell)$ (recherche exhaustive)
- Une attaque de type **seconde préimage** coûte $O(2^\ell)$ (recherche exhaustive)
- Une attaque par **collisions** coûte $O(2^{\ell/2})$ (paradoxe des anniversaires)

**Question**

Comment effectue-t-on ces attaques ?

# Construction de Merkle-Damgård

La construction de Merkle-Damgård permet de transformer une **fonction de compression** en une fonction de hachage.




Attention au **padding** ! Typiquement : un 1 suivi de 0s puis de la longueur du message encodée sur 64 bits.

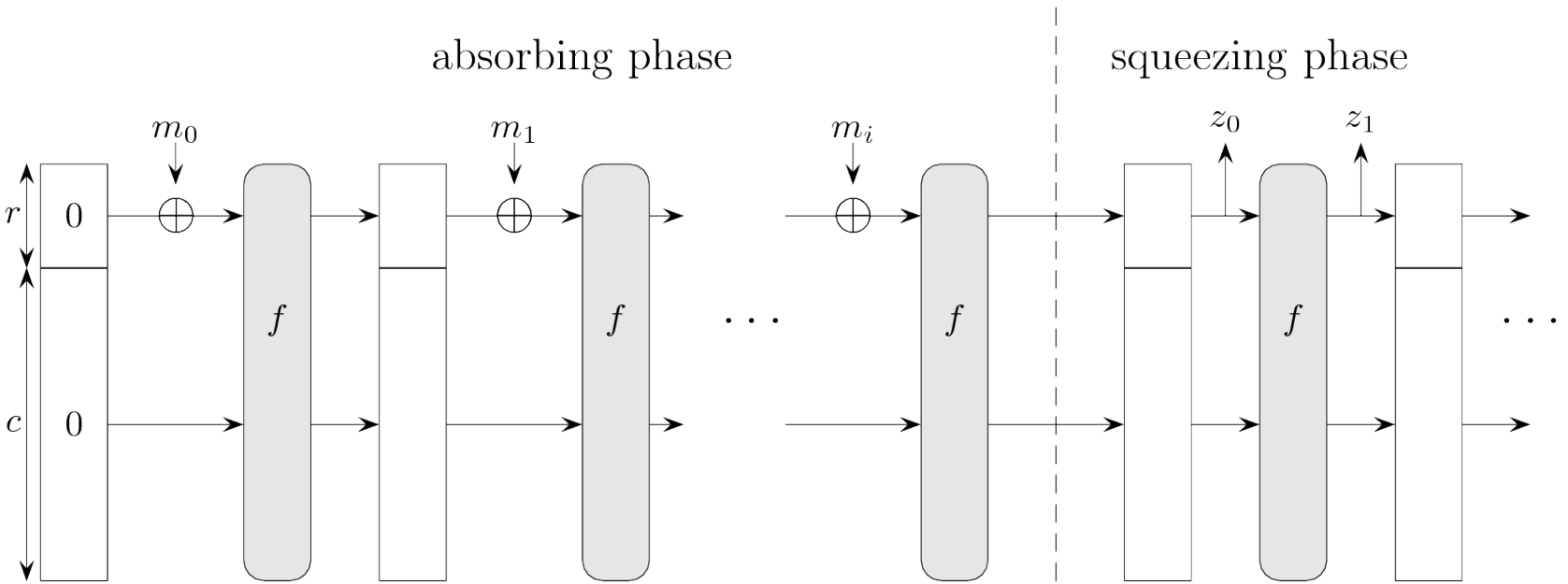
# Quelques Fonctions de Hachage

- MD5
  - Construction de Merkle-Damgård.
  - Empreintes de 128 bits
  - **Cassée** ! Il est possible de trouver des collisions en quelques secondes sur un PC standard.
- SHA-1
  - Construction de Merkle-Damgård.
  - Empreintes de 160 bits
  - **Cassée** ! Il est possible de trouver des collisions en quelques secondes sur un PC standard.

# Quelques Fonctions de Hachage (2)

- SHA-256, SHA-512 (aussi appelé SHA-2)
  - Construction de Merkle-Damgård
  - Empreintes de 224, 256, 384 et 512 bits
  - **Sûre** (pour l'instant...)
- SHA-3
  - Empreintes de 224, 256, 384 et 512 bits
  - Processus de sélection similaire à celui d'AES. 5 finalistes (BLAKE, Grøstl, JH, Keccak et Skein), contre 51 candidats acceptés dans la compétition. Keccak (Joan Daemen et al.) a gagné la compétition en octobre 2012 et est devenu le standard NIST FIPS 202 en août 2015.
  - Construction **éponge**.

# Construction Eponge



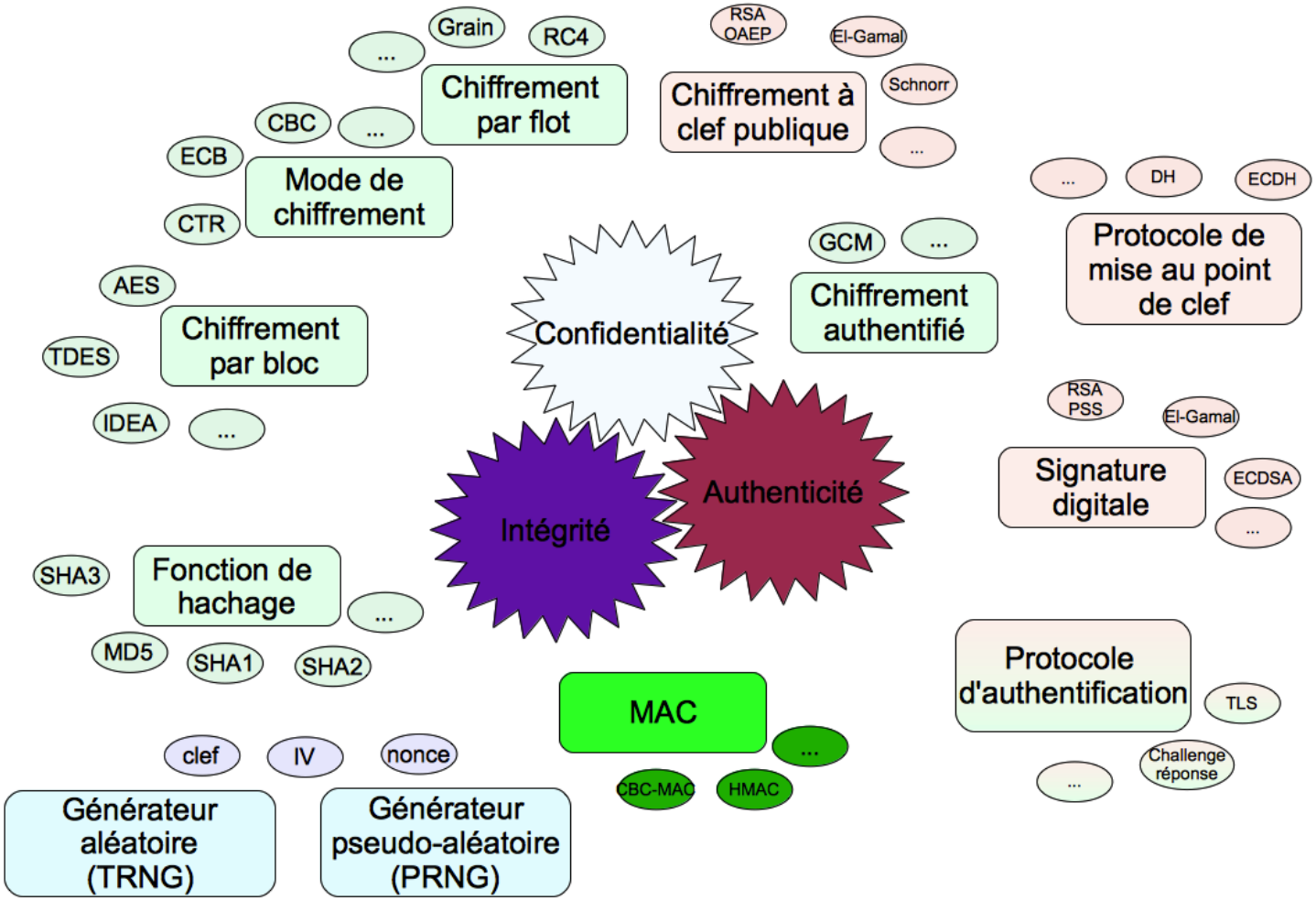
1. Modes Opératoires

2. Fonctions de Hachage

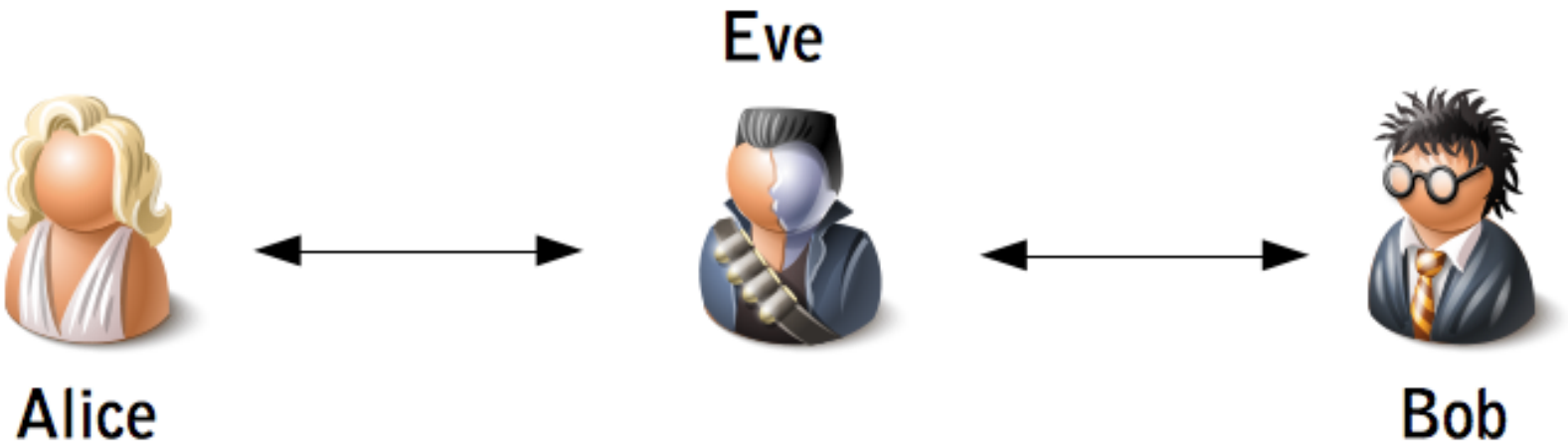
3. Authentification Symétrique
   - MACs
   - HMAC
   - CBC-MAC

4. Chiffrement Authentifié

# Vue Synoptique



# Rappel : Adversaires Actifs



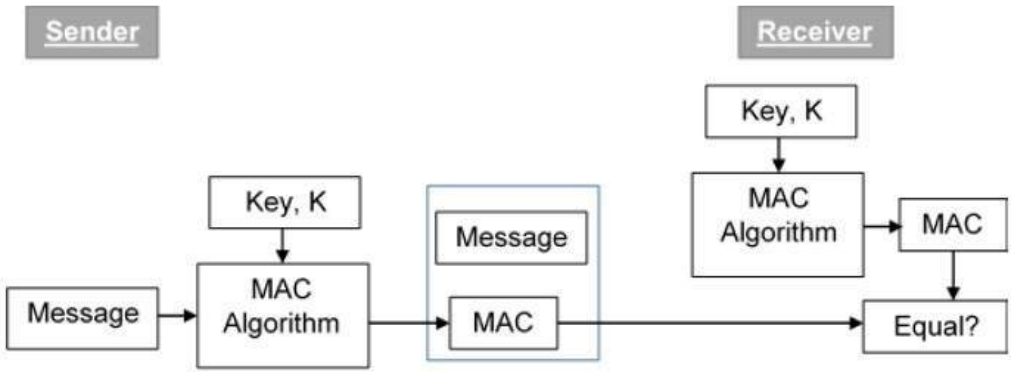
# Authenticité et Fonctions de Hachage

- La cryptographie symétrique peut garantir l'authenticité d'un canal de communication

**Question**

Pourquoi est-ce qu'une fonction de hachage isolée n'est pas un bon moyen de garantir l'authenticité d'un canal ? $(M, H(M))$

# «Messages Authentication Codes (MAC)»




- Alice et Bob peuvent utiliser un **MAC** pour vérifier **l'authenticité d'un message**, à condition qu'ils possèdent un secret en commun.
- **Attention** : Un MAC n'est pas utilisé pour protéger la confidentialité d'un message. Il est utilisé en parallèle d'un message afin de garantir que ce dernier n'ait pas été modifié.

# Un Bon MAC ?

**Question**

Que pensez-vous du MAC suivant :

$$\mathrm{SHA256}(X) \oplus K$$

# Fonctionnement d'un MAC

- Alice et Bob se mettent d'accord sur un secret partagé $K$.
- Alice envoie son message $X$ ainsi que la valeur de MAC $\tau = \mathrm{MAC}_K(X)$ à Bob.
- Bob calcule $\tau = \mathrm{MAC}_K(X')$ avec le message $X'$ reçu.
- Le message est accepté si et seulement si $\tau = \tau'$, où $\tau'$ est la valeur du MAC attaché au message.

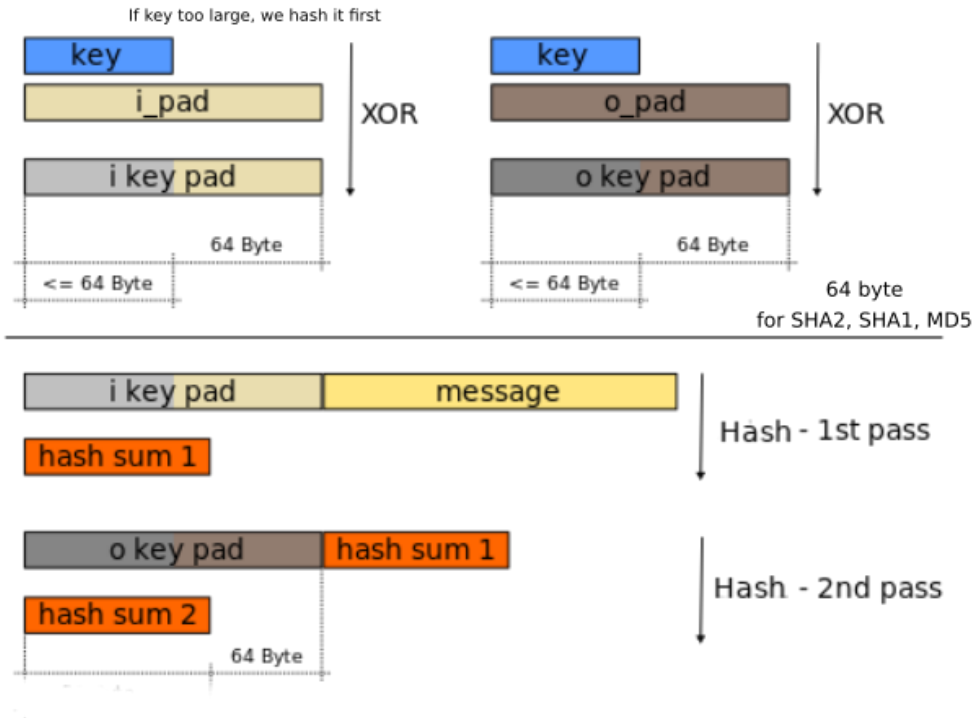
# Sécurité d'un MAC

- On dira qu'un MAC est **cryptographiquement sûr** s'il est impossible en pratique de forger une paire valide $(x', \mathrm{MAC}_K(x'))$ avec $x_i \neq x'$ à partir d'une ou de plusieurs paires $(x_i, \mathrm{MAC}_K(x_i))$.
- Le message $x'$ n'a **pas** besoin d'avoir du sens !

# HMAC

- «Hash-based Message Authentication Code» (HMAC)
  - Publié en 1996 par Bellare, Canetti et Krawczyk
  - Standardisé dans la RFC 2104, et dans le document NIST FIPS PUB 198
  - Construit un MAC à partir de n'importe quelle fonction de hachage cryptographiquement sûre
  - HMAC-MD5 et HMAC-SHA1 sont très largement utilisées en pratique (TLS, IPSec, ...)
  - Transition vers HMAC-SHA2.

# HMAC

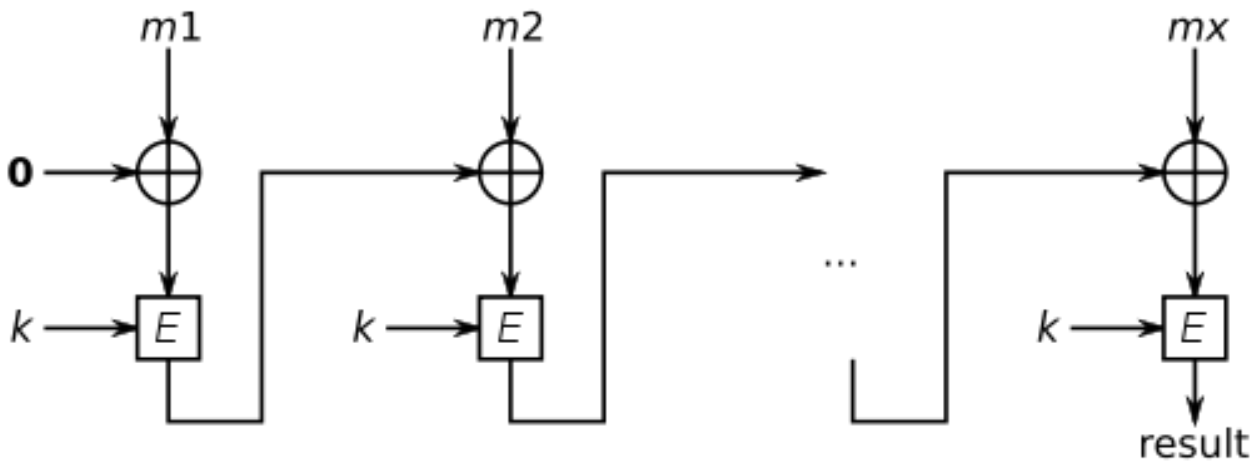



- Spécifications :
  - Constantes

$$\begin{aligned} \texttt{o\_pad} &= \texttt{0x5C5C5C}\ldots\texttt{5C} \\ \texttt{i\_pad} &= \texttt{0x363636}\ldots\texttt{36} \end{aligned}$$

# CBC-MAC

- CBC-MAC
  - Basé sur le mode de chiffrement symétrique CBC
  - Utilise un algorithme de chiffrement par bloc $E_K(.)$
  - Seulement sûr pour des messages de **taille fixe** !




Source de l'illustration : http://en.wikipedia.org/wiki/CBC-MAC

# Attaque contre CBC-MAC

**Question**

Trouver comment forger un MAC avec CBC-MAC en ayant à disposition un seul message dont le MAC est connu.

# CBC-MAC avec Messages de Taille Variable

- Pour des messages de taille variable, il faut soit
  - surchiffrer le résultat avec une seconde clef («encrypted CBC-MAC») ;
  - ou simplement ajouter la longueur du message comme **premier** bloc.
  - Encore mieux : **le standard "CMAC"** : on XOR une constante dépendant de la clef [1] après la dernière itération et on tronque le résultat.

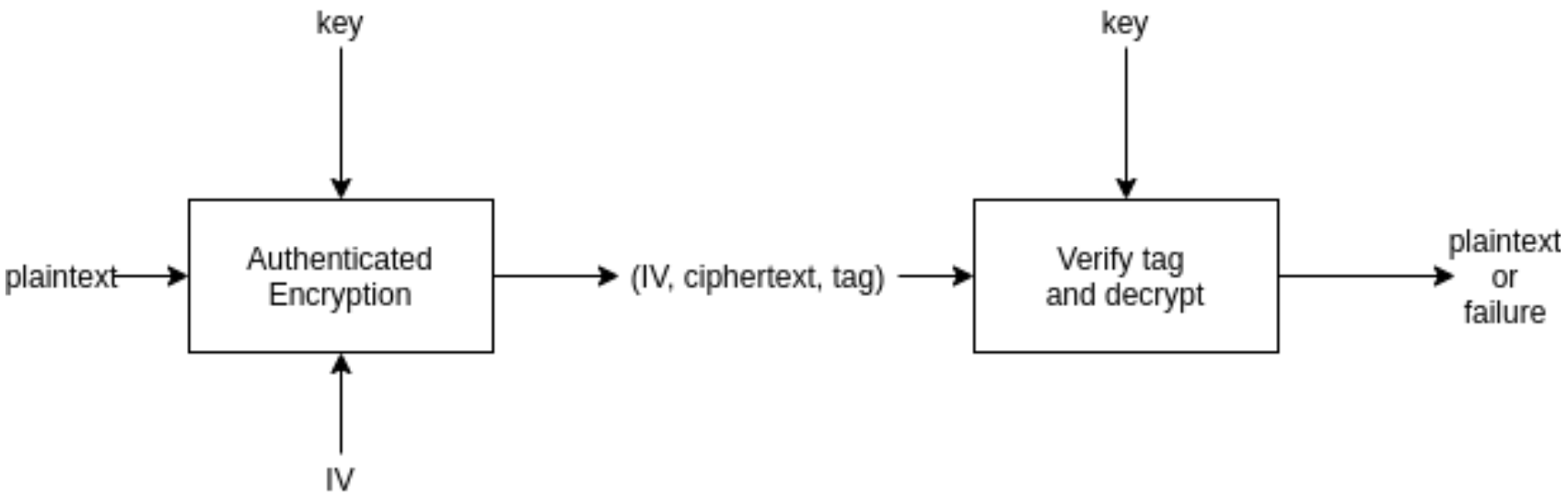
1. $E_K(0)$ shifté d'un bit vers la gauche XOR une constante si carry

# Chiffrement Authentifié

- Le **chiffrement authentifié** offre trois propriétés de sécurité simultanément : la **confidentialité**, **l'authenticité** et **l'intégrité**.
- Souvent obtenu en combinant un mode opératoire avec un MAC.
- Possibilité d'authentifier des données sans les chiffrer ("authenticated data" (AD)).
- Février 2019 : résultats concours CAESAR : **nouveaux** systèmes de chiffrement authentifié.

☞ Si possible, **préférer** le chiffrement authentifié au chiffrement normal.

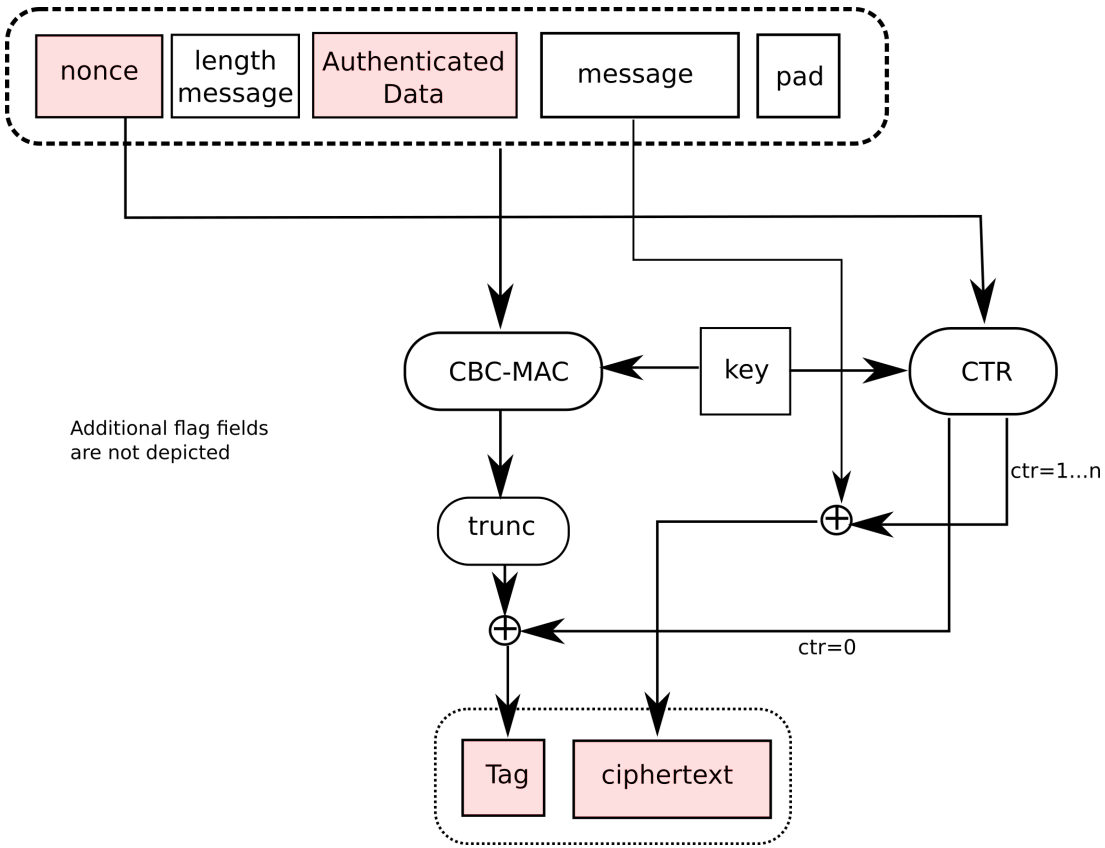
# Chiffrement Authentifié



# CCM

- CCM combine CBC-MAC et le mode opératoire CTR.
- CCM utilise un système de chiffrement par bloc de 128 bits.
- L'IV (nonce) ne doit **jamais** être répété.
- CCM est utilisé dans 802.11i, IPSec et TLS.
- Strictement **amélioré** dans le mode opératoire EAX.

# CCM


nonce
length message
Authenticated Data
message
pad
CBC-MAC
key
CTR
Additional flag fields
are not depicted
ctr=1...n
trunc
ctr=0
Tag
ciphertext

# CCM

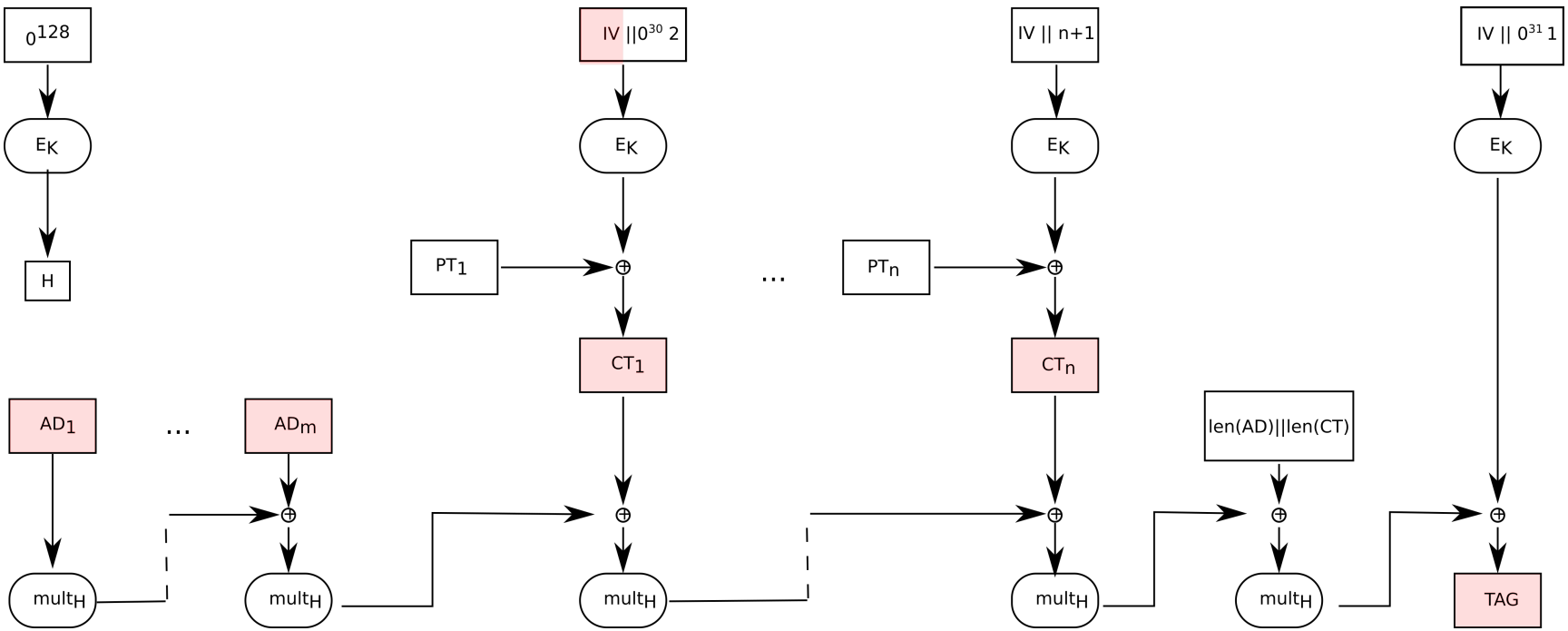
**Question**

Comment déchiffrer ?

# GCM

- GCM est basé sur un système de chiffrement par blocs qui prend des messages de 128 bits. Il mélange le mode opératoire CTR avec le **Galois message authentication code**.
- GCM utilise des multiplications dans $\mathrm{GF}(2^{128})$ construit comme $\mathbb{Z}_2[x]/(x^{128} + x^7 + x^2 + x + 1)$.
- L'IV (nonce) ne doit **jamais** être répété.
- GCM est utilisé dans IPSec, TLS et SSH.

# GCM


$0^{128}$
$E_K$
H
$AD_1$
...
$AD_m$
$mult_H$
$mult_H$
$IV \,||0^{30}\, 2$
$E_K$
$PT_1$
$CT_1$
$mult_H$
...
IV || n+1
$E_K$
$PT_n$
$CT_n$
$mult_H$
len(AD)||len(CT)
$mult_H$
IV || $0^{31}$ 1
$E_K$
TAG

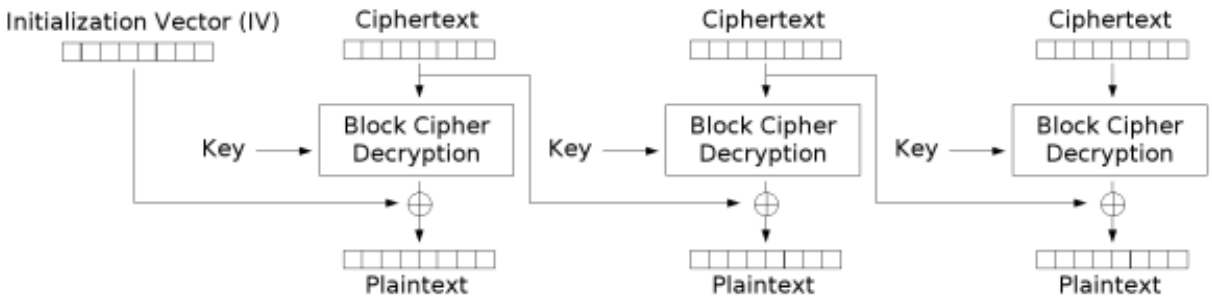
# GCM

**Question**

Comment déchiffrer ?

# Solutions

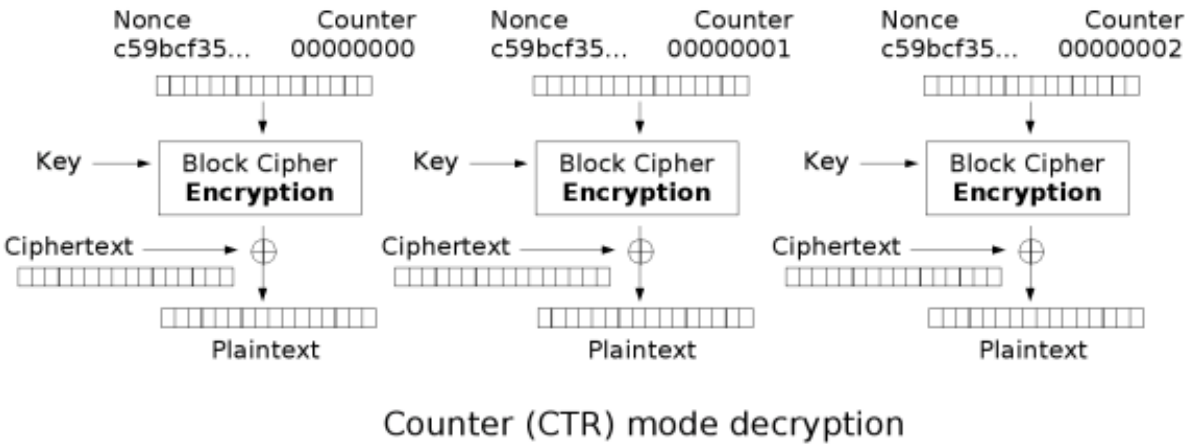
# «Cipher Block Chaining» : déchiffrement




Cipher Block Chaining (CBC) mode decryption

Source des illustrations : http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation

# «Counter Mode» : déchiffrement




Counter (CTR) mode decryption

Source des illustrations : http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation

# Coût Générique des différentes Attaques

## Solution

- Pour les attaques de première et seconde préimage, il suffit d'effectuer une recherche exhaustive (bruteforce) sur les préimages possibles.
- Pour les collisions, il faut faire des requêtes aléatoires et les stocker dans une hashtable (table de hashage). A chaque nouvel hash calculé, vérifier s'il se trouve déjà dans la table.

# Authenticité et Fonctions de Hashage

**Solution**

C'est catastrophique. N'importe qui peut authentifier un message car les algorithmes sont connus et ne dépendent d'aucune clef secrète.

# Un Bon MAC ?

**Solution**

C'est catastrophique. Il est possible, en connaissant un MAC de récupérer la clef secrète. Etant donné un message $X$ et un Mac $\tau$,

$$\mathrm{SHA256}(X) \oplus \tau = K.$$

# Attaque contre CBC-MAC

**Solution**

Nous avons à notre disposition un message $m$ et son MAC $\tau$. Nous pouvons calculer le MAC du message $m \| (m \oplus \tau)$ qui vaut $\tau$, où $\|$ est le symbole de concaténation.

# CCM

**Solution**

A l'aide de la clef et du nonce, nous pouvons tout d'abord déchiffrer le texte chiffré. Ensuite, nous pouvons recalculer le MAC et le comparer avec le MAC reçu. S'ils sont différent, nous rejetons le message.

# GCM

**Solution**

Dans GCM, nous allons tout d'abord vérifier l'intégrité du texte chiffré. Pour cela, nous recalculons tout d'abord $H$ suivi du TAG. Nous comparons le TAG calculé avec celui que nous avons reçu. S'il est correct, alors nous pouvons déchiffrer le message à l'aide de CTR.