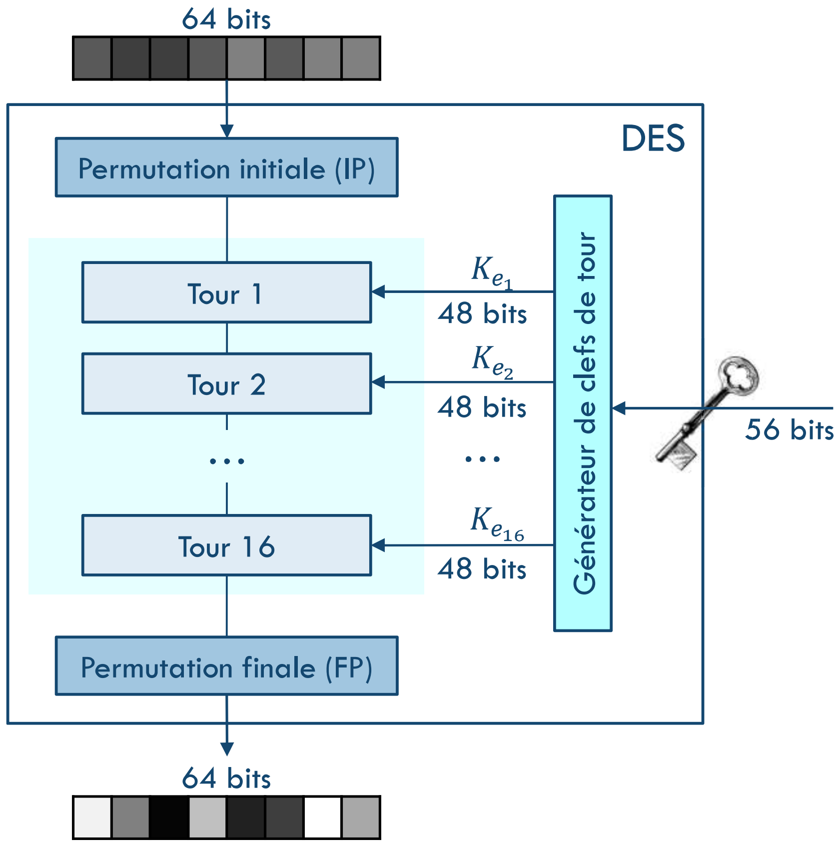
DATA ENCRYPTION STANDARD (DES)

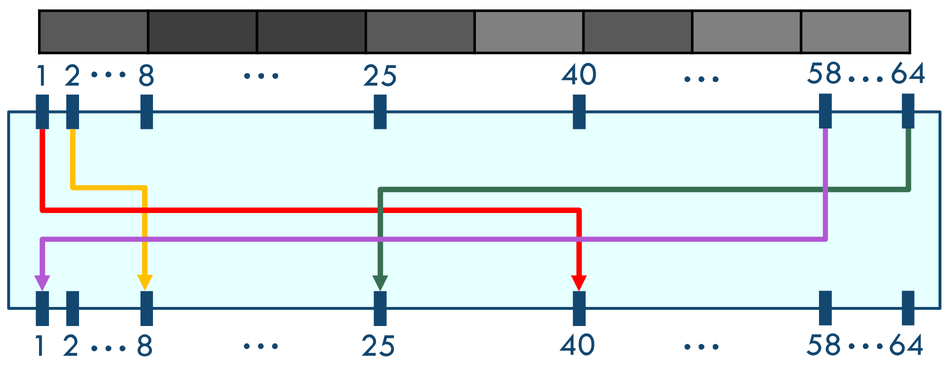
- Algorithme de chiffrement symétrique, **par blocs**
- Publié en **1975**, par **IBM**
- Aujourd'hui : considéré comme **non-sécurisé**
- Caractéristiques techniques :
  - Taille de la clef : **56 bits** (+ **8 bits** de parité)
  - Taille des blocs : **64 bits**
  - Nombre de tours : **16**



DATA ENCRYPTION STANDARD (DES)

- **Permutation initiale (IP)**

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

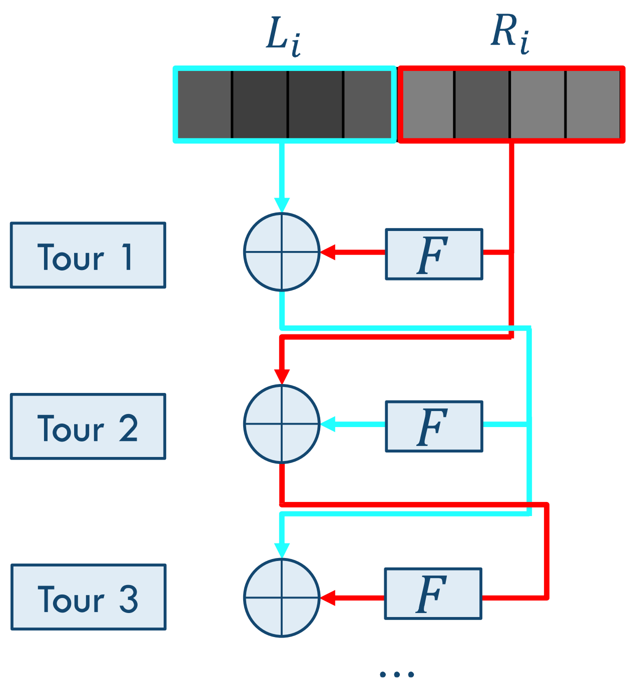


Q : Quelle est la nouvelle position du 8ème bit initialement ?

DATA ENCRYPTION STANDARD (DES)

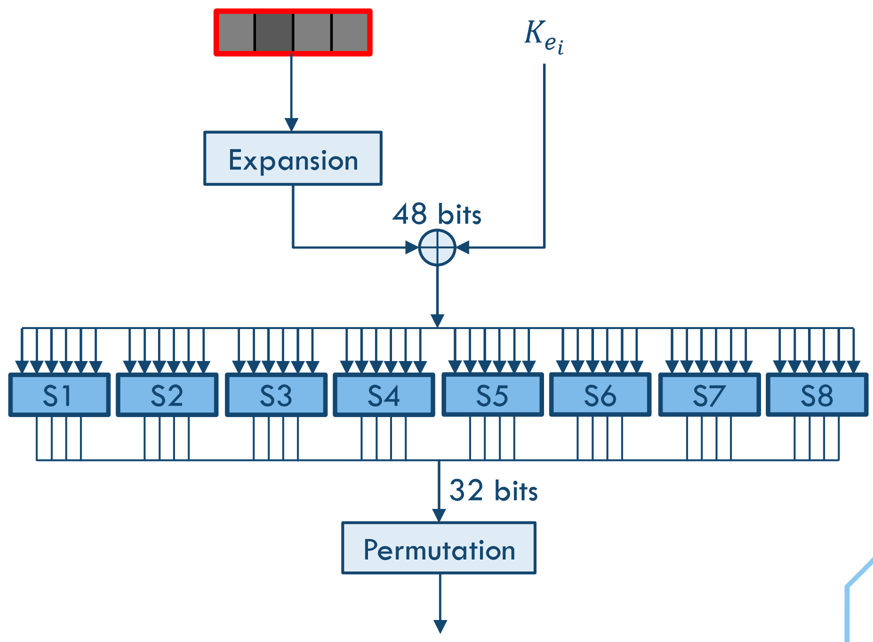
- **Structure générale des tours**
  - Découpage de chaque bloc en 2 sous-blocs
  - Application de la fonction F (de Feistel) à l'un des sous-blocs
  - Ou-exclusif avec l'autre sous-bloc

$$\begin{aligned} L_{i+1} &= R_i \\ R_{i+1} &= L_i \oplus F(R_i, K_{e_{i+1}}) \end{aligned}$$



DATA ENCRYPTION STANDARD (DES)

- **Fonction de Feistel (F)**
  - **Expansion** : 32 bits → 48 bits, soit 8x6 bits (duplication de la moitié des bits)
  - **Mélange avec la clef** : XOR avec la clef de tour
  - **Substitution** : Passage par les S-box, 6 bits → 4 bits (x8) (garantit la sécurité du DES, casse la linéarité)
  - **Permutation** : Passage par la P-box, 32 bits réarrangés



DATA ENCRYPTION STANDARD (DES)

- **Confusion** : Supprimer les relations entre le message en clair et le message chiffré
  - Outil : Boîtes de substitution (**S-box**)
- **Diffusion** : Propager l'information relative à chaque bit du message en clair dans le message chiffré
  - Outil : Boîtes de permutation (**P-box**)
- **Rappel** : Différence entre ces deux concepts

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 4 & 1 & 5 & 2 & 6 & 3 & 7 \end{pmatrix}$$

Permutation :
2164523 → 1422653

Substitution :
2164523 → 1432615

DATA ENCRYPTION STANDARD (DES)

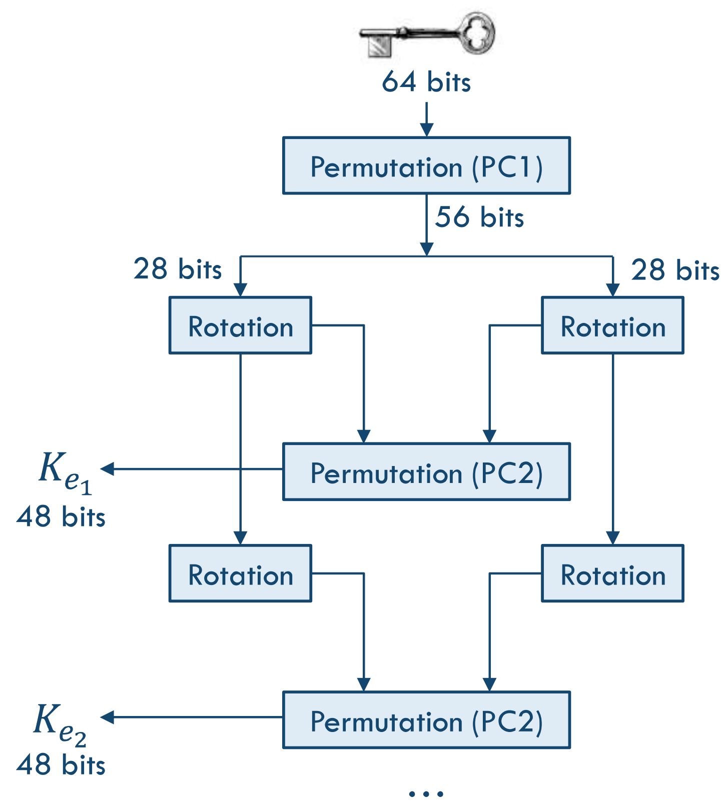
- **Boîte de substitution (S-box)**
  - Par exemple, avec en entrée : **011011**

S5		4 bits au centre de l'entrée															
		0000	**0001**	**0010**	**0011**	**0100**	**0101**	**0110**	**0111**	**1000**	**1001**	**1010**	**1011**	**1100**	**1101**	**1110**	**1111**
Bits externes	**00**	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	**1001**	1000	0110
	10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

Q : Qu'obtient-on en prenant 110101 en entrée ?
Q : A quelle(s) entrée(s) peut correspondre la sortie 0101 ?

DATA ENCRYPTION STANDARD (DES)

- **Génération des clefs de tour**
  - **Permutation (PC1) :** 64 bits → 56 bits, soit 2x28 bits (autres bits pour contrôle)
  - **Rotation :** Vers la gauche, d'un ou deux bits
  - **Permutation (PC2) :** 28 bits → 24 bits, soit 48 bits

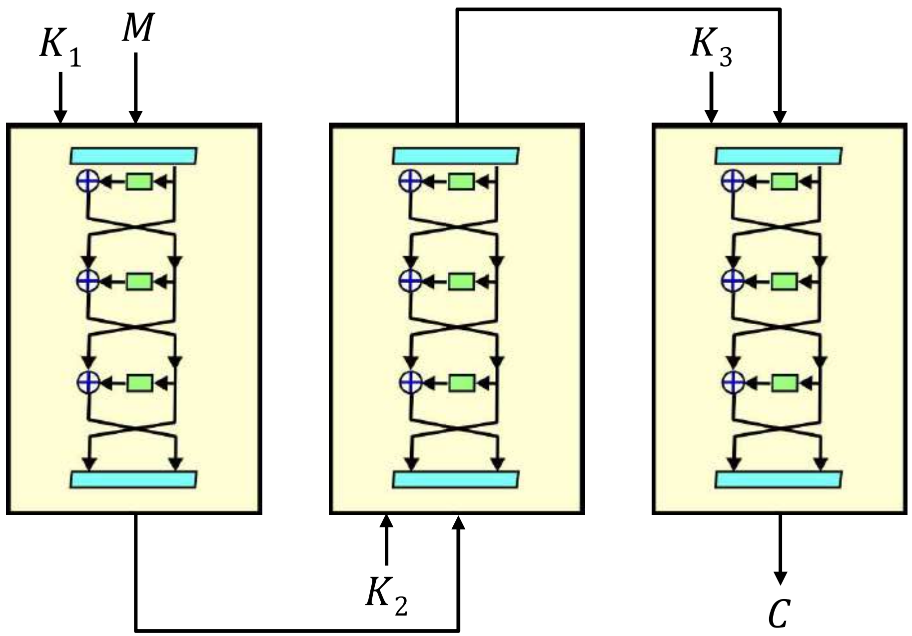


DATA ENCRYPTION STANDARD (DES)

- **Attaques**
  - **Attaque par force brute** possible
    - **Diffie-Hellman** en 1977 (US$ 20M), clef retrouvée en 1 jour → théorique
    - **Wiener** en 1993 (US$ 1M), clef retrouvée en 7h → théorique
    - **Electronic Frontier Foundation** en 1998 (US$ 250k), clef retrouvée en 2 jours → mise en pratique
    - **COPACOBANA** (Univ. Bochum & Kiel, en Allemagne) en 2006 (US$ 10k) → mise en pratique
  - **Cryptanalyse différentielle**
    - **Biham-Shamir** en 1980 (CPA –Attaque par 2^{47} clairs choisis)
  - **Cryptanalyse linéaire**
    - **Matsui** en 1994 (KPA –Attaque par 2^{43} clairs connus)
    - **Junod** en 2001 (KPA –Attaque par 2^{40} clairs connus)

TRIPLE DES (3DES)

- Publié en **1998**, dérivé de **DES**
- Plus grande taille des clefs pour éviter l'attaque par force brute : **168 bits** possible
- Idée : Utiliser 3 clefs de **56 bits** chacune



$$C = E_{K_3}(D_{K_2}(E_{K_1}(M)))$$

$$M = D_{K_1}(E_{K_2}(D_{K_3}(C)))$$

TRIPLE DES (3DES)

- Attaque « **Meet-in-the-middle** » sur 2DES (même principe pour 3DES)
  - Attaque **KPA** : M_1, M_2, C_1, C_2 connus tels que $C_1 = E_{K_2}(E_{K_1}(M_1))$ et $C_2 = E_{K_2}(E_{K_1}(M_2))$
  - Chiffrer une fois le message en clair revient à déchiffrer une fois le message chiffré
  - On a donc : $E_{K_1}(M) = D_{K_2}(C)$
  - Calcul et stockage des 2^{56} couples $(K, E_K(M_1))$
  - Pour chaque clef K', calcul de $D_{K'}(M_1)$ et recherche de correspondance
  - Si couple de clefs candidates $K_1 = K$ et $K_2 = K'$, vérification $C_2 = E_{K_2}(E_{K_1}(M_2))$
  - Nombre maximal d'essais : $\mathbf{2 \times 2^{56} = 2^{57}} \ll \mathbf{2^{112}}$

ADVANCED ENCRYPTION STANDARD (AES)

- Algorithme de chiffrement symétrique, **par blocs**
- Gagnant d’un concours lancé en 1997
- Standard depuis **2001** (NIST)
- Vrai nom : **Rijndael** → Créé par deux belges **Joan Daemen** et **Vincent Rijmen**

ADVANCED ENCRYPTION STANDARD (AES)

- **Pourquoi un nouveau standard ?**
  - DES est devenu attaquable par force brute
  - Développement de systèmes d'évaluation : analyse différentielle et linéaire
  - Possible d'avoir une méthode de chiffrement plus rapide en utilisant des instructions processeur

- **Pourquoi un concours public ?**
  - Rassembler la communauté travaillant sur la cryptographie
  - Encourager la recherche autour des systèmes sécurisés
  - Prévenir les « backdoors »
  - Accélérer l'acceptation et l'adoption d'un standard

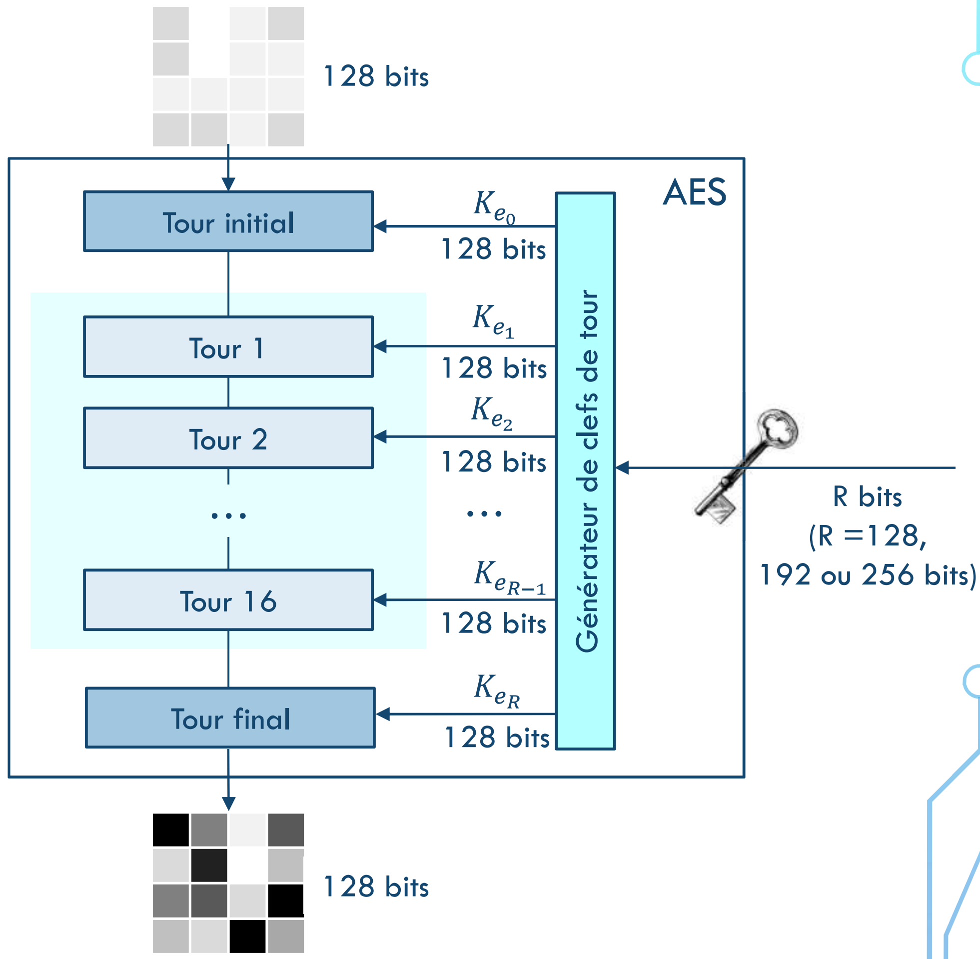
ADVANCED ENCRYPTION STANDARD (AES)

- **Description générale**
  - Nombre de tours : **10, 12** ou **14** (suivant la taille de la clef)
  - Chaque tour : **4** opérations
  - Taille des blocs du message : **128 bits** (4 colonnes de 4 octets)
  - Taille de la clef de chiffrement : **128, 192** ou **256 bits**

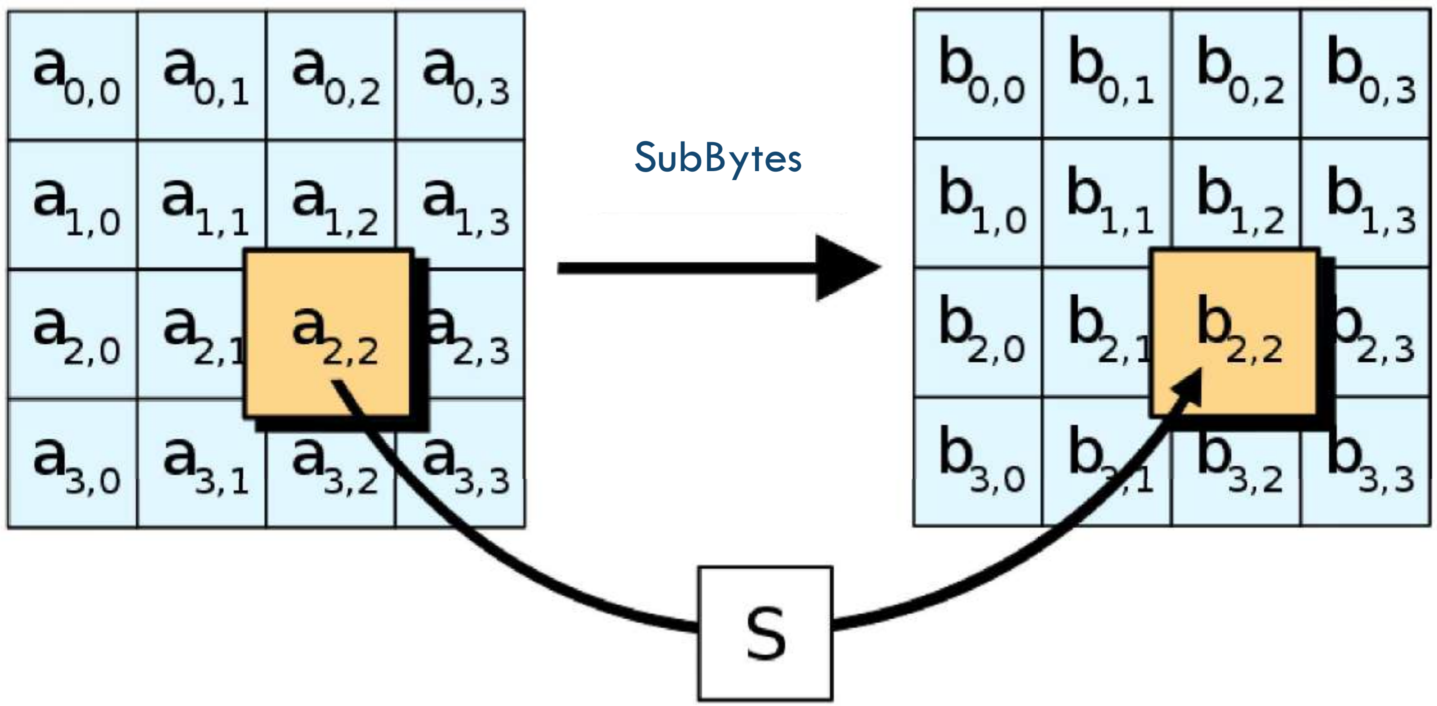
$p(0,0)$	$p(0,1)$	$p(0,2)$	$p(0,3)$
$p(1,0)$	$p(1,1)$	$p(1,2)$	$p(1,3)$
$p(2,0)$	$p(2,1)$	$p(2,2)$	$p(2,3)$
$p(3,0)$	$p(3,1)$	$p(3,2)$	$p(3,3)$

ADVANCED ENCRYPTION STANDARD (AES)

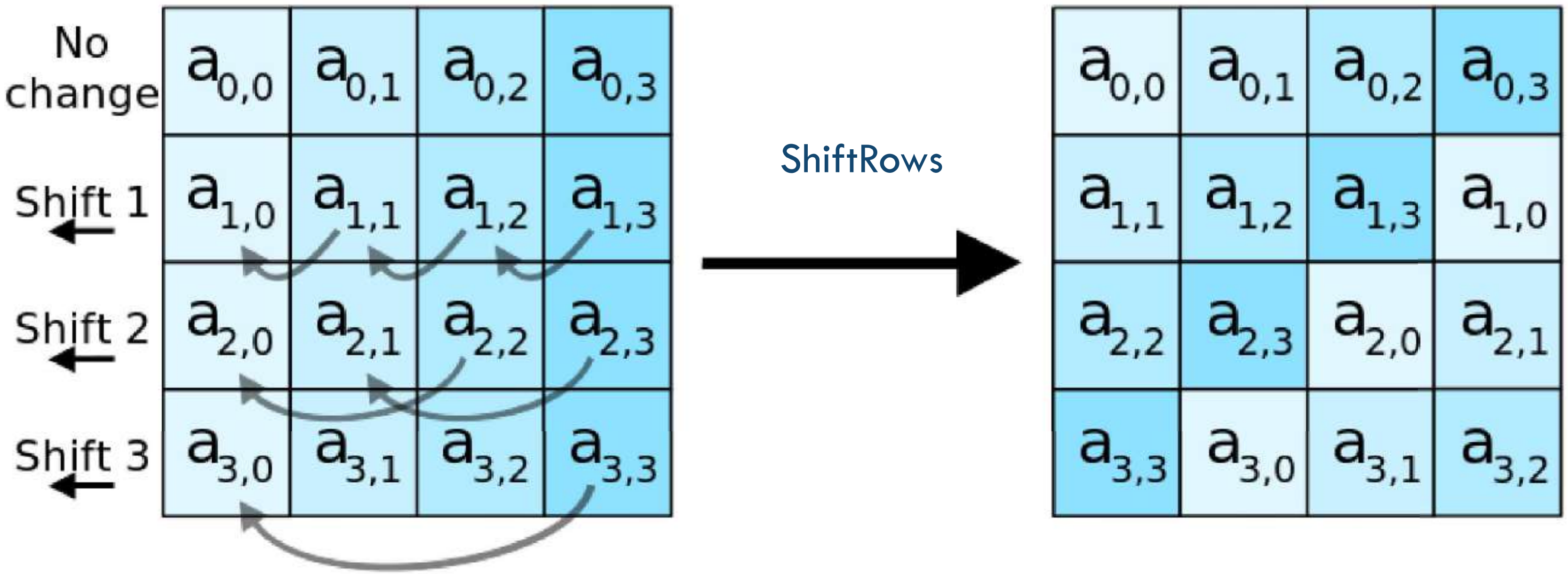
- **Description générale**
  1. KeyExpansion
  2. Tour initial
     1. AddRoundKey
  3. Pour chaque tour suivant
     1. SubBytes
     2. ShiftRows
     3. MixColumns
     4. AddRoundKey
  4. Tour final (pas de MixColumns)
     1. SubBytes
     2. ShifRows
     3. AddRoundKey



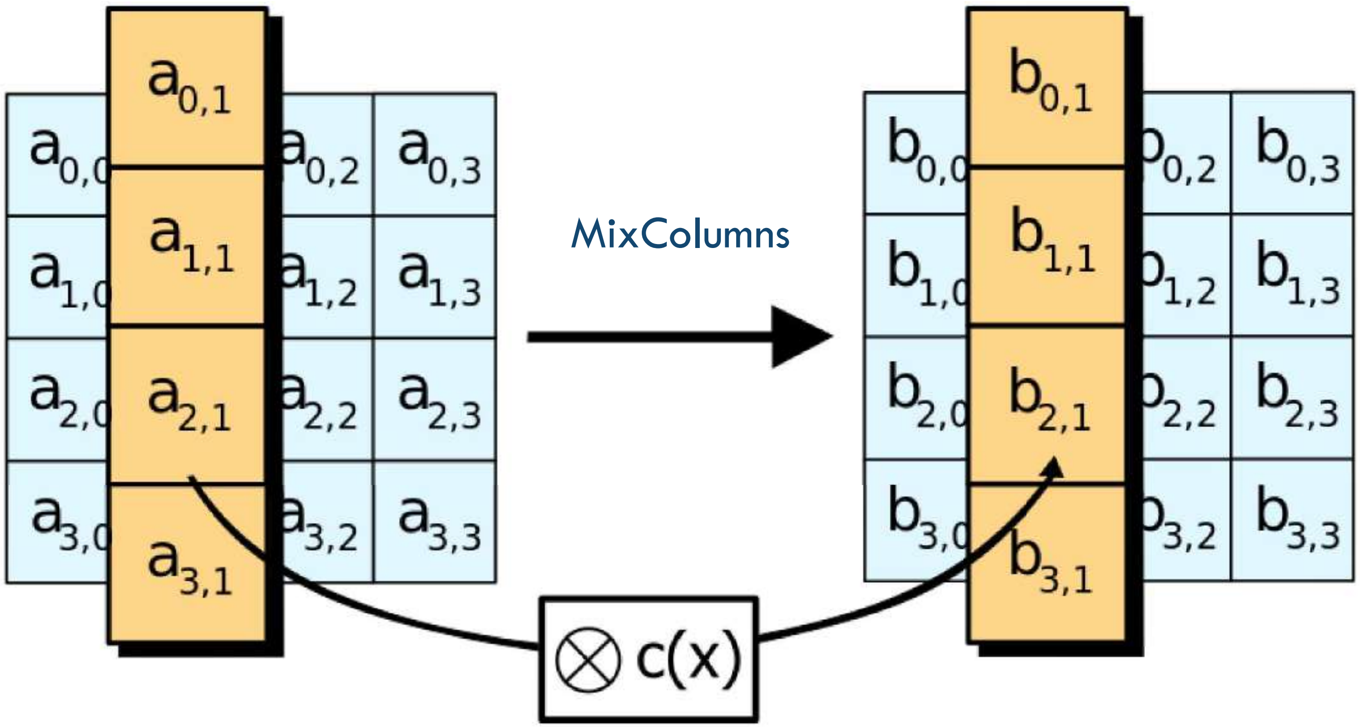
ADVANCED ENCRYPTION STANDARD (AES)



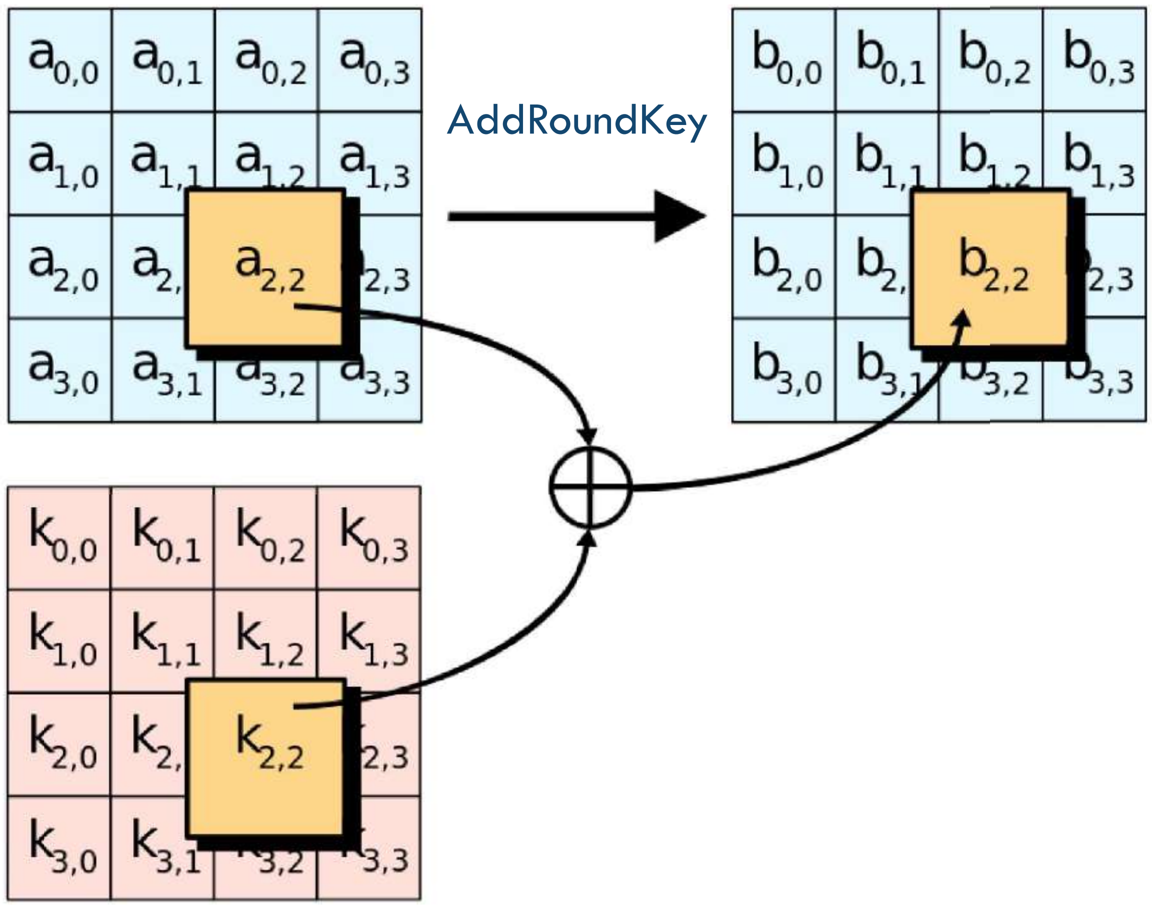
ADVANCED ENCRYPTION STANDARD (AES)



ADVANCED ENCRYPTION STANDARD (AES)

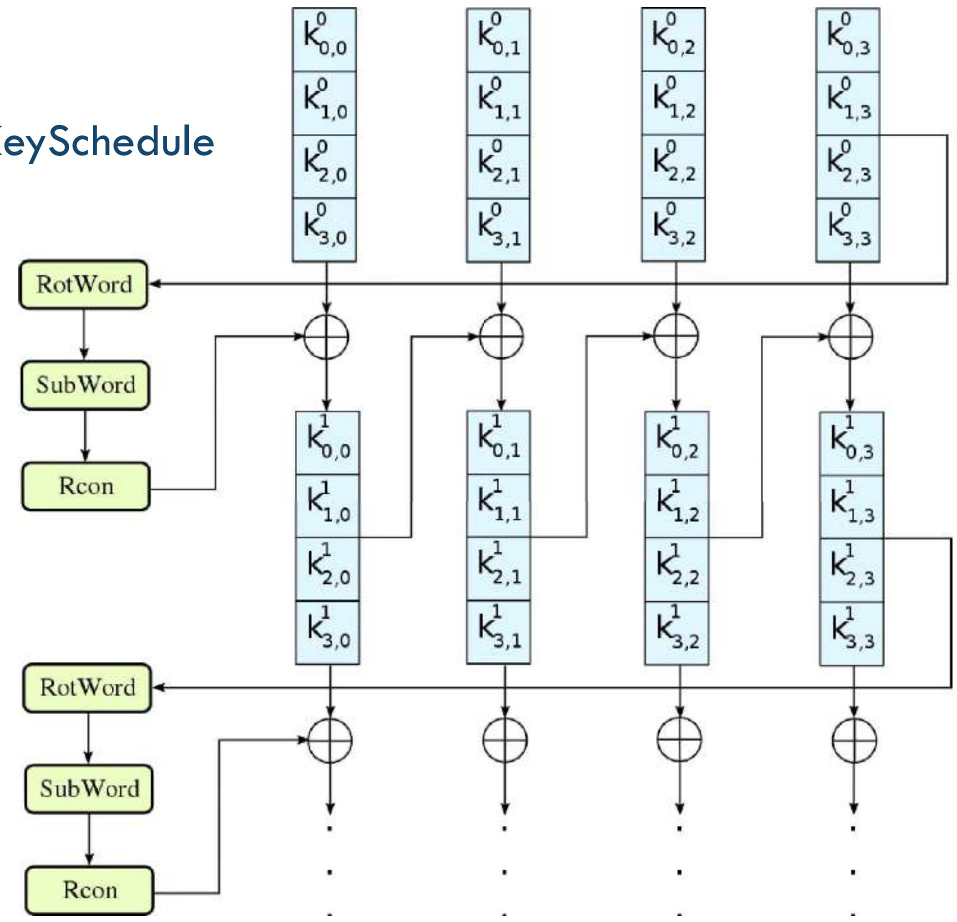


ADVANCED ENCRYPTION STANDARD (AES)



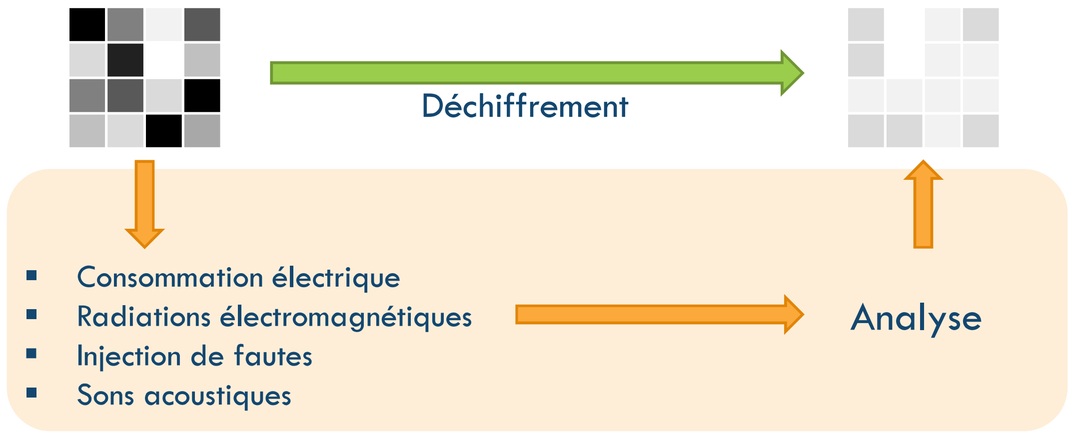
ADVANCED ENCRYPTION STANDARD (AES)

KeySchedule



ADVANCED ENCRYPTION STANDARD (AES)

- **Attaque par canal auxiliaire**
  - Recherche et exploitation des failles dans l'implémentation, logicielle ou matérielle
  - Ne remet pas en cause la robustesse théorique des méthodes et procédures de sécurité
  - Une sécurité « mathématique » ne garantit pas forcément une sécurité lors de l'utilisation en « pratique »



- Beaucoup d'attaques publiées **non faisables** en pratique (2013)
- **Considéré sûr à ce jour**

CRYPTOGRAPHIE ASYMÉTRIQUE

- Caractéristiques
  - Une **clef privée** $\boldsymbol{K_{priv}}$ et une **clef publique** $\boldsymbol{K_{pub}}$
  - Propriétés :
    - La connaissance de la clef publique K_{pub} ne permet pas de déduire la clef privée K_{priv}
    - $D_{K_{priv}}(E_{K_{pub}}(M)) = M$
  - Principe : **Fonction unidirectionnelle à trappe**
    - « Facile » à calculer dans un sens, « difficile » à inverser
    - Sauf si on connaît une information secrète (**la trappe**)
    - Algorithmes basés sur des opérations d’exponentiation en algèbre modulaire

CRYPTOGRAPHIE ASYMÉTRIQUE

- Caractéristiques
  - Génération des clefs :
    - A partir de grands nombres premiers $K_{pub} = f(K_{priv})$
    - Calcul de $K_{priv} = f^{-1}(K_{pub})$ **impossible**
    - Taille des clefs : **512 bits** ou **1024 bits**
    - Performances : **1000 fois plus lents** que les algorithmes symétriques !
    - Nombre de clefs : autant de paires que d'entités
    - Distribution des clefs : Facilitée car **pas d'échange** de clefs secrètes
      - Clef secrète conservée par les entités
      - Clef publique échangée

PROTOCOLE DE DIFFIE-HELLMAN (1976)

Alice

Bob

1) Alice et Bob choisissent un grand nombre premier p et d'un entier $1 \leq a < p$

2) Alice choisit secrètement x_A	2) Bob choisit secrètement x_B
3) Alice calcule $y_A = a^{x_A} \, (mod\, p)$	3) Bob calcule $y_B = a^{x_B} \, (mod\, p)$

4) Alice et Bob s'échangent les valeurs de y_A et y_B

5) Alice calcule $y_B{}^{x_A} = (a^{x_B})^{x_A} = a^{x_B x_A} \, (mod\, p) = K$	5) Bob calcule $y_A{}^{x_B} = (a^{x_A})^{x_B} = a^{x_A x_B} \, (mod\, p) = K$

Q : Sur quoi repose la sécurité de l'échange des clefs ?

RIVEST-SHAMIR-ADLEMAN (RSA)

- Créé en **1977** par Ron Rivest, Adi Shamir et Leonard Adleman
- Breveté par le MIT en **1983**
- Basé sur le problème de la **factorisation des grands nombres entiers**

RIVEST-SHAMIR-ADLEMAN (RSA)

- **Génération du couple de clefs** (K_{pub}, K_{priv})
  - Choisir p et q, deux nombres premiers distincts
  - Calculer leur produit $n = pq$ (**module de chiffrement**)
  - Calculer $\varphi(n) = (p-1)(q-1)$
  - Choisir un nombre e premier avec $\varphi(n)$ et strictement inférieur à ce nombre (**exposant de chiffrement**)
  - Calculer l'entier naturel d, inverse de e modulo $\varphi(n)$ (**exposant de déchiffrement**)
  - On a $\boldsymbol{K_{pub} = (e, n)}$ **et** $\boldsymbol{K_{priv} = d}$

Q : Quel algorithme utilise t-on pour calculer d, l'inverse de e modulo $\varphi(n)$?

RIVEST-SHAMIR-ADLEMAN (RSA)

- **Algorithme d'Euclide étendu**
  - Idée :
    - d est l'inverse de e modulo $\varphi(n)$, c'est-à-dire $ed \equiv 1\ (mod\ \varphi(n))$
    - D'après le théorème de Bachet-Bézout il existe deux entiers d et k tels que $ed + k\varphi(n) = 1$
  - Par exemple : $e = 23$ et $\varphi(n) = 120$

Q : Dérouler l'algorithme d'Euclide pour calculer le PGCD de $\boldsymbol{e}$ et $\boldsymbol{\varphi(n)}$.
Q : Utiliser les résultats obtenus pour trouver $\boldsymbol{d}$.

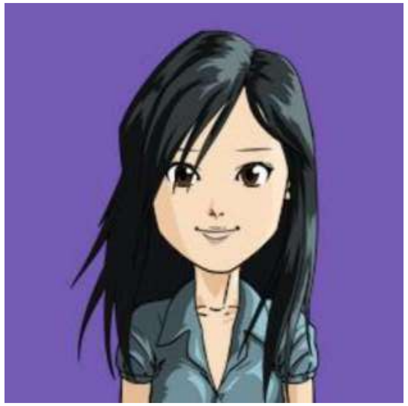
RIVEST-SHAMIR-ADLEMAN (RSA)

- **Algorithme d'Euclide étendu**
  - Idée :
    - d est l'inverse de e modulo $\varphi(n)$, c'est-à-dire $ed \equiv 1 \ (mod\ \varphi(n))$
    - D'après le théorème de Bachet-Bézout il existe deux entiers d et k tels que $ed + k\varphi(n) = 1$
  - Par exemple : $e = 23$ et $\varphi(n) = 120$
    - $120 = 1 \times 120 + 0 \times 23$
    - $23 = 0 \times 120 + 1 \times 23$
    - $5 = 120 - 5 \times 23 = 1 \times 120 - 5 \times 23$
    - $3 = 23 - 4 \times 5 = 1 \times 23 - 4(1 \times 120 - 5 \times 23) = -4 \times 120 + 21 \times 23$
    - $2 = 5 - 1 \times 3 = (1 \times 120 - 5 \times 23) - 1 \times (-4 \times 120 + 21 \times 23) = 5 \times 120 - 26 \times 23$
    - $1 = 3 - 1 \times 2 = (-4 \times 120 + 21 \times 23) - 1 \times (5 \times 120 - 26 \times 23) = -9 \times 120 + 47 \times 23$
    - Donc $d = 47$ et $k = -9$

RIVEST-SHAMIR-ADLEMAN (RSA)

- **L'algorithme de chiffrement/déchiffrement**

Alice

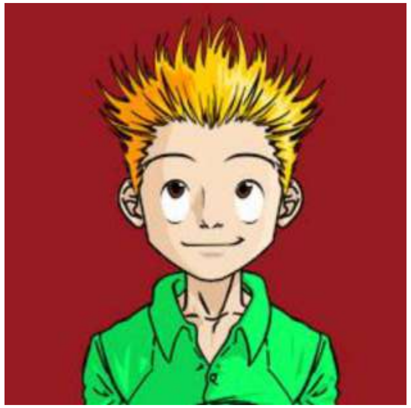


$K_{pub}(A) = (e_A, n_A)$

$K_{priv}(A) = d_A$

$C = M^{e_B} (mod\ n_B)$

Bob



$K_{pub}(B) = (e_B, n_B)$

$K_{priv}(B) = d_B$

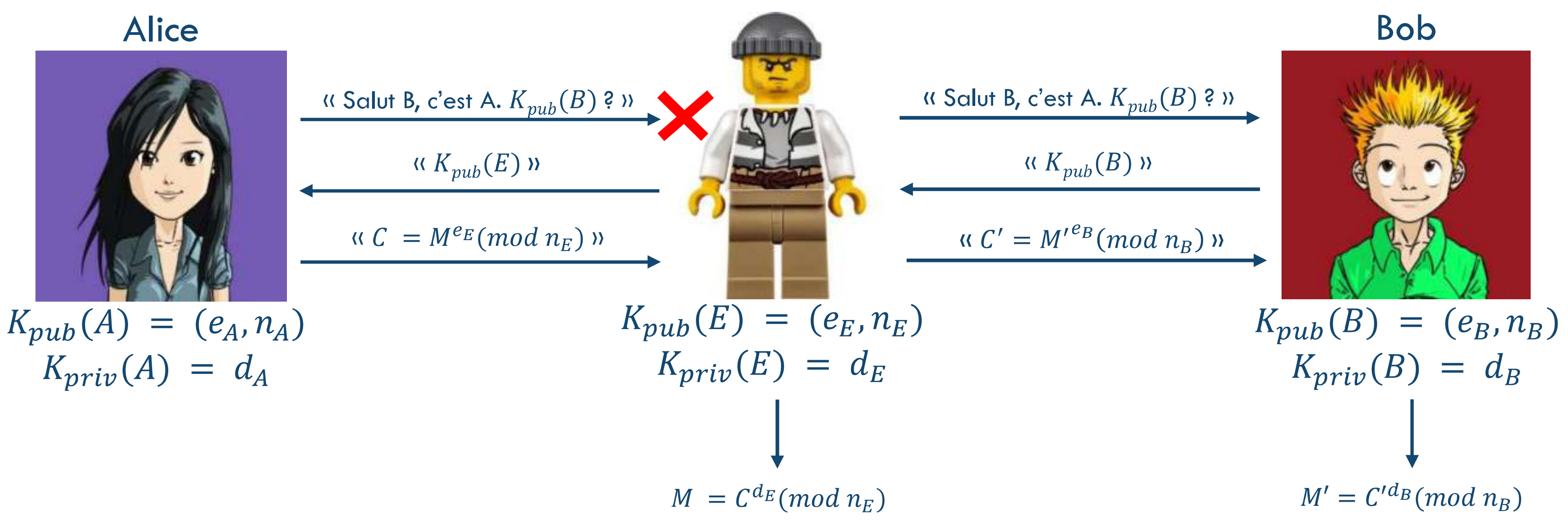
$M = C^{d_B}\ (mod\ n_B)$

$M = (M^{e_B})^{d_B} (mod\ n_B)$

Q : Comment calculer efficacement l'exponentiation modulaire ?

RIVEST-SHAMIR-ADLEMAN (RSA)

- **Attaque de « l'homme du milieu »**



Par exemple, M = « Viens me chercher à la gare » et M' = « Viens me chercher au stade »

Q : Comment remédier à ce problème ?

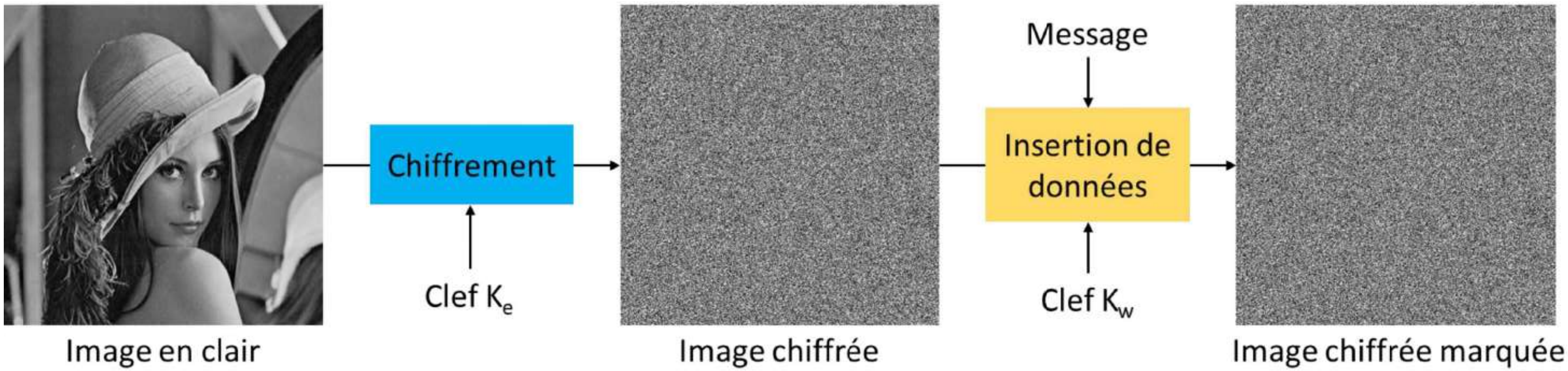
TAILLE DES CLEFS

n	2^n	Ordre de grandeur
32	2^{32}	Nombre d'humains sur Terre
46	2^{46}	Distance de la Terre au Soleil, en millimètres
46	2^{46}	Nombre d'opérations faites par un CPU mono-cœur (1GHz)
55	2^{55}	Nombre d'opérations faites par un CPU mono-cœur (1GHz), en un an
63	2^{63}	Nombre d'opérations faites par un serveur quadri-processeurs de 22 cœurs (2,2GHz), en un an
82	2^{82}	Masse de la Terre, en kilogrammes
89	2^{89}	Nombre d'opérations faites en 13,8 milliards d'années, à raison d'un milliard d'opérations par seconde
155	2^{155}	Nombre de molécules d'eau sur Terre
256	2^{256}	Nombre d'électrons dans l'univers

PLAN DU COURS

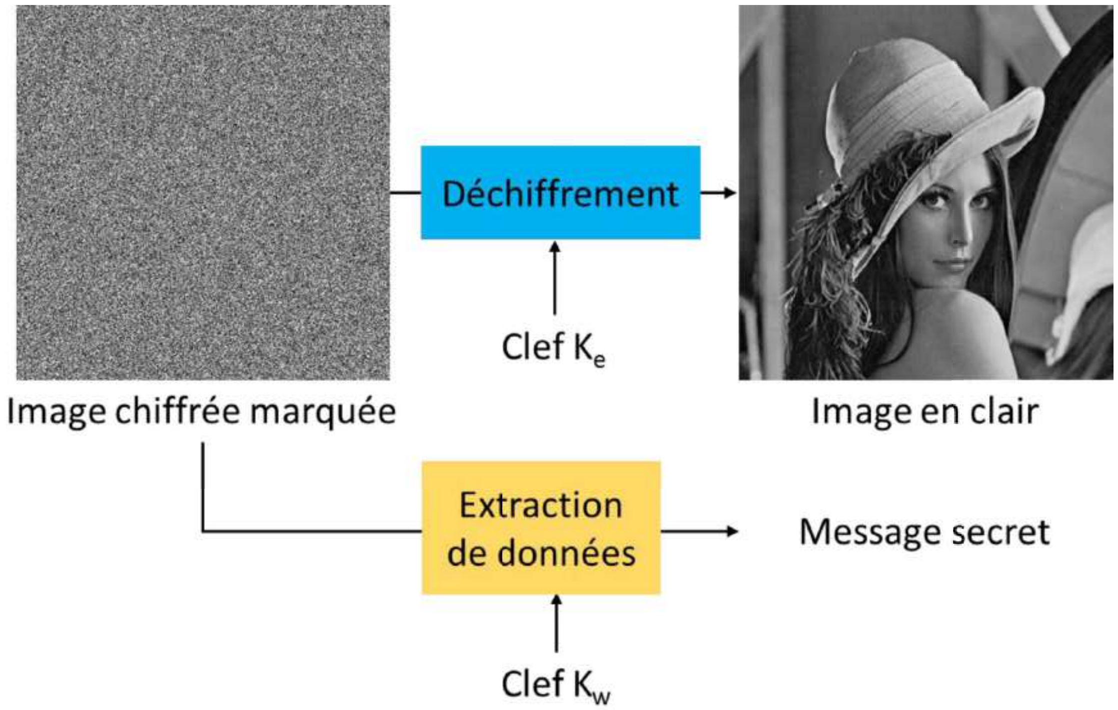
- Introduction à la cryptographie
  - Enjeux de sécurité
  - Rappels de mathématiques, terminologie et outils
- Méthodes de chiffrement classiques
  - Bref historique
  - Cryptographie moderne
- Conclusion
  - Ouverture : Analyse et traitement des images dans le domaine chiffré

INSERTION DE DONNÉES CACHÉES DANS LE DOMAINE CHIFFRÉ



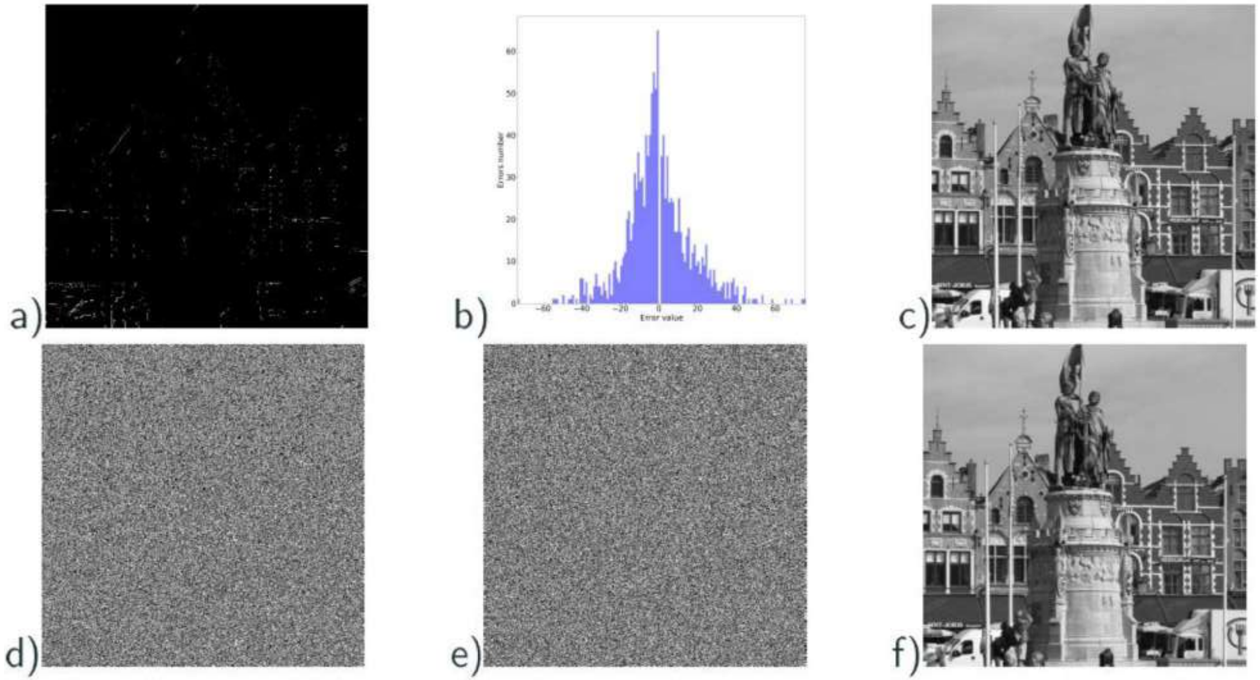
Chiffrement de l'image et insertion du message

INSERTION DE DONNÉES CACHÉES DANS LE DOMAINE CHIFFRÉ



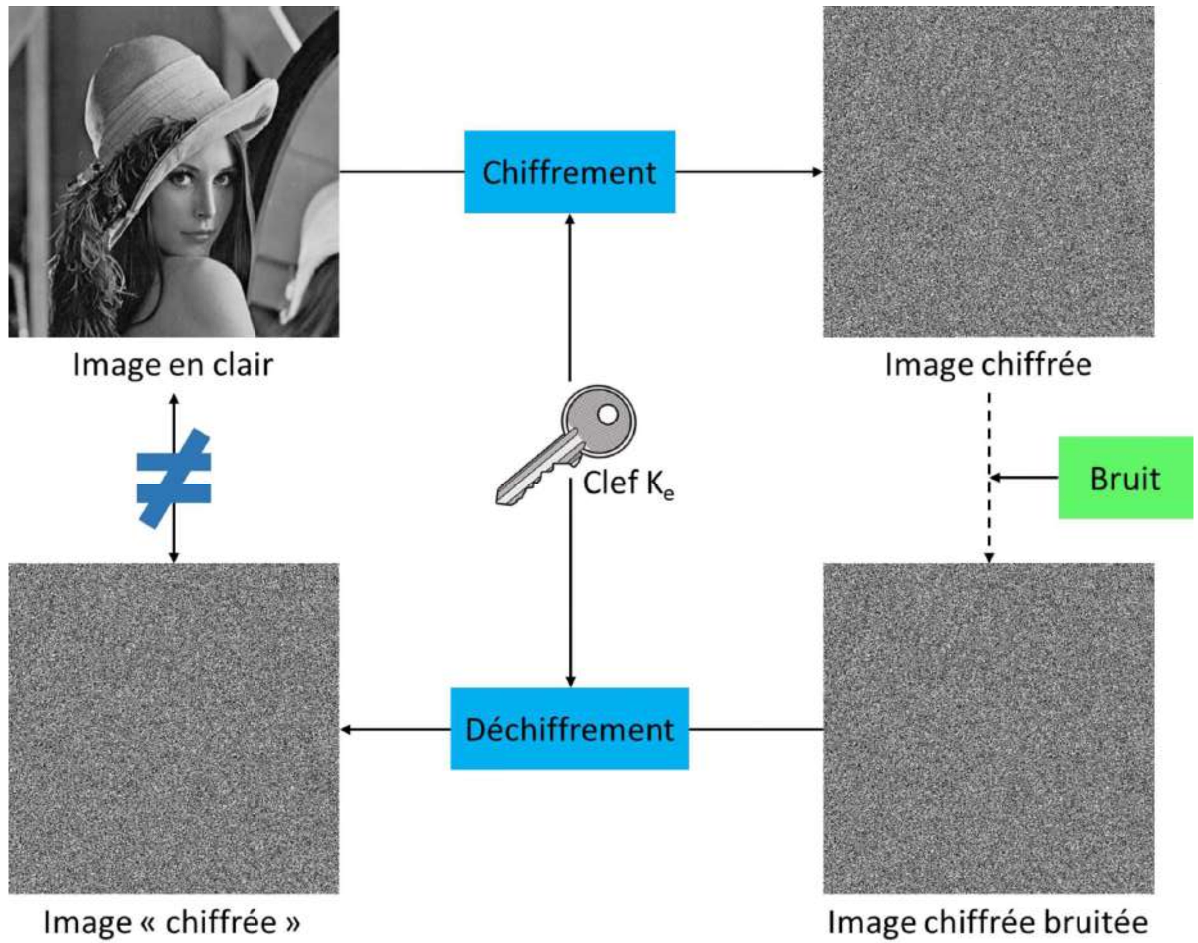
Déchiffrement de l'image et extraction du message

INSERTION DE DONNÉES CACHÉES DANS LE DOMAINE CHIFFRÉ



Exemple détaillé pour l'approche$_{1\,bpp}$: a) Emplacement des erreurs de prédiction, nombre d'erreurs $= 1242$ $(0,47\%)$, b) Histogramme des erreurs de prédiction, c) Image pré-traitée, PSNR $= 46,87$ dB, d) Image chiffrée, e) Image chiffrée marquée, capacité d'insertion $= 1$ bpp, f) Image reconstruite, PSNR $= 46,87$ dB, SSIM $= 0,99$.

CORRECTION D'IMAGES CHIFFRÉES BRUITÉES



CORRECTION D'IMAGES CHIFFRÉES BRUITÉES

Image originale (256 niveaux de gris)

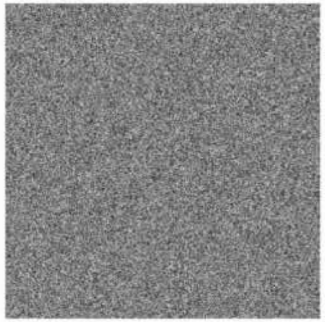
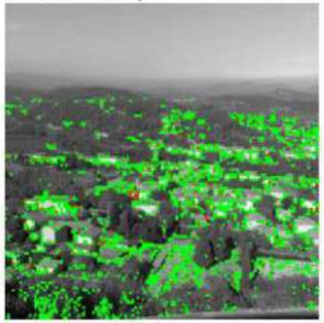


Image chiffrée bruitée (PSNR = $8,85\ dB$, BER = $2,6 \times 10^{-3}$)

Image reconstruite, sans correction (PSNR = $16,51\ dB$)

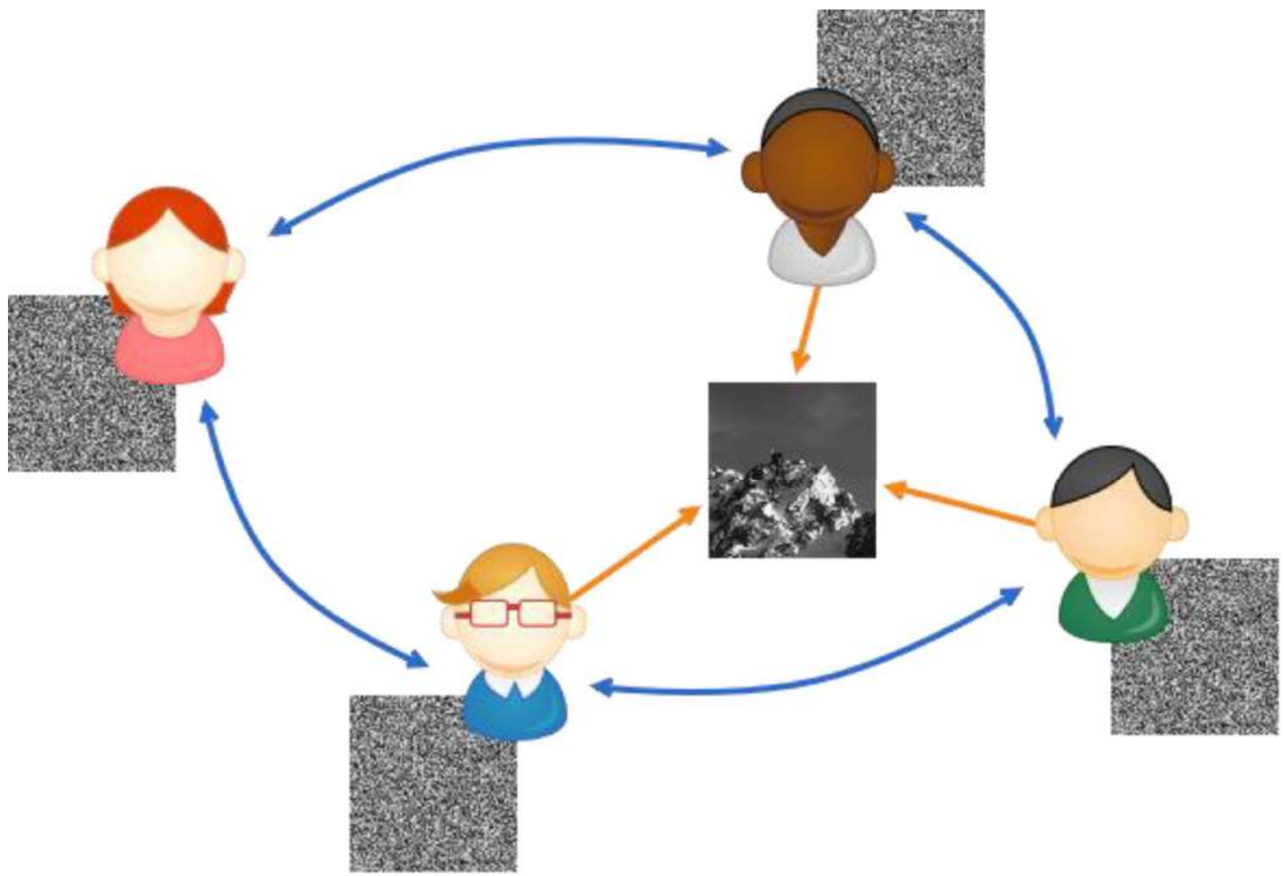
Correction de l'image en utilisant l'entropie des distances, sans quantification (256 niveaux de gris, PSNR = $16,37\ dB$)

Correction de l'image en utilisant l'entropie des distances, après quantification (8 niveaux de gris, PSNR = $29,73\ dB$)



Emplacement des blocs où il y a plus d'une configuration possible en clair

PARTAGE D'IMAGES SECRÈTES (VISUAL SECRET SHARING – VSS)



PARTAGE D'IMAGES SECRÈTES (VISUAL SECRET SHARING – VSS)

Image originale

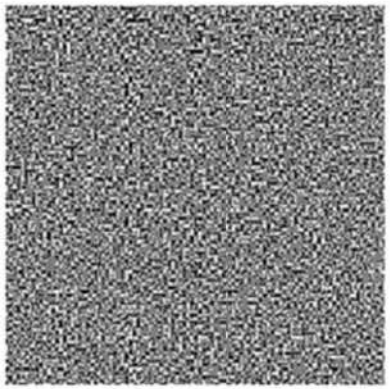


Image share 1

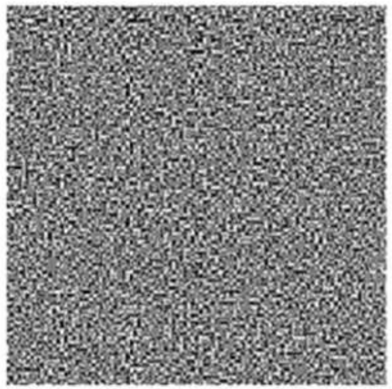
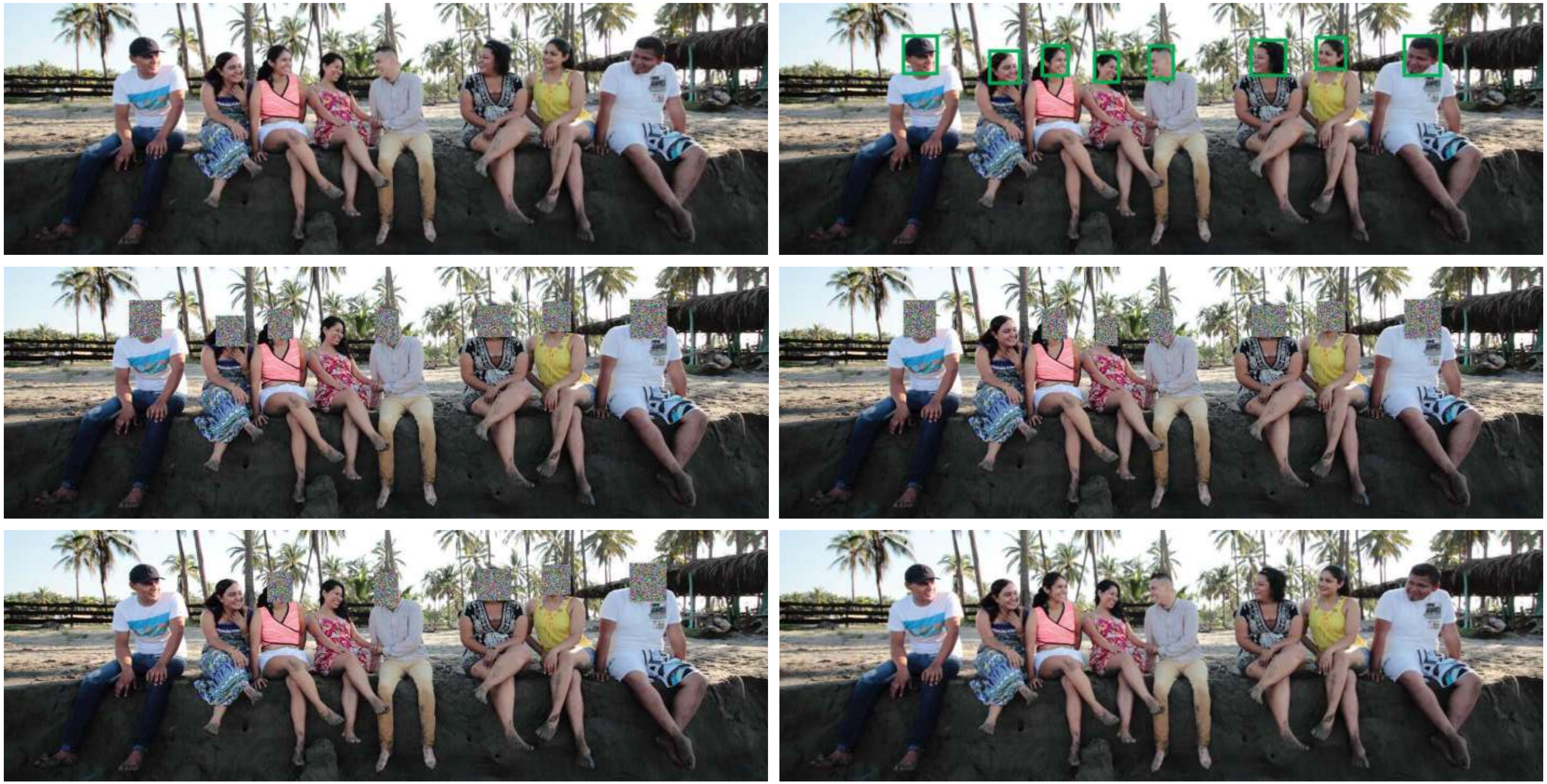


Image share 2

Image reconstruite

PARTAGE D'IMAGES SECRÈTES (VISUAL SECRET SHARING – VSS)



RECOMPRESSION D'IMAGES CRYPTO-COMPRESSÉES