Introduction à la cryptographie

École des Mines, 3e année

Documents

- Pour aller plus loin : transparents de cours de David Pointcheval.
  `web` : demander à Google

Plan

Le chiffrement autrefois

- Le chiffrement de César : décalage des lettres
- Disque de chiffrement (Léone Battista Alberti 1466)

Le chiffrement autrefois

- Le chiffrement de César : décalage des lettres
- Disque de chiffrement (Léone Battista Alberti 1466)

→ sujet à des analyses statistiques

Le chiffrement : période technique

Substitutions et permutations automatiques

Enigma

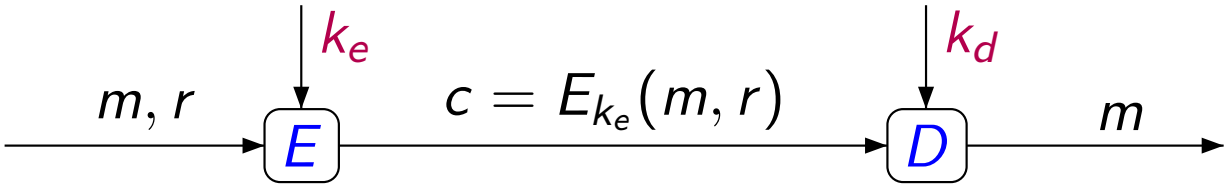
Avantages et inconvénients

- Niveau de sécurité qui dépend du nombre de rotors
- Pas de preuves de sécurité

Le chiffrement aujourd'hui

Trois algorithmes

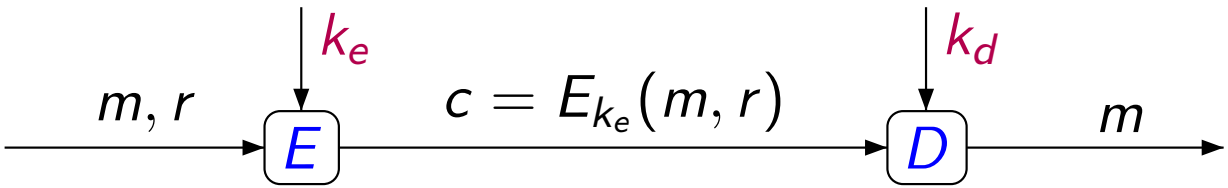
- G - génération de clef
- E - chiffrement
- D - déchiffrement



Le chiffrement aujourd'hui

Trois algorithmes

- G - génération de clef
- E - chiffrement
- D - déchiffrement



$k_e = k_d$ chiffrement symétrique

$k_e \neq k_d$ chiffrement asymétrique

Sûreté du chiffrement asymétrique

données publiques :

- $c = E_{k_e}(m, r)$ chiffré
- k_e clef de chiffrement

Un unique message m satisfait la relation (avec éventuellement plusieurs r possibles)
$\longrightarrow$ Une recherche exhaustive sur m et r permet de trouver m !

Sûreté du chiffrement asymétrique

données publiques :

- $c = E_{k_e}(m, r)$ chiffré
- k_e clef de chiffrement

Un unique message m satisfait la relation (avec éventuellement plusieurs r possibles)
$\longrightarrow$ Une recherche exhaustive sur m et r permet de trouver m !

$\Rightarrow$ Le secret inconditionnel est impossible, il faut se baser sur des hypothèses algorithmiques.

Plan

La cryptographie moderne sur des problèmes difficiles

Exemple : factorisation des nombres premiers.

- Étant donné un entier n
- trouver deux nombres premiers p, q tels que $n = p.q$.

La cryptographie moderne sur des problèmes difficiles

Exemple : factorisation des nombres premiers.

- Étant donné un entier n
- trouver deux nombres premiers p, q tels que $n = p.q$.

$\longrightarrow$ Calculs dans le groupe

$$(\mathbb{Z}/n\mathbb{Z})^*$$

avec $n = p.q$, p, q nombres premiers.

Factorisation d'entiers et RSA

$\longrightarrow$ Utilisation de problèmes algorithmiquement difficiles.

Factorisation :

- $p, q \mapsto n = p.q$ facile (quadratique)
- $n = p.q \mapsto p, q$ difficile

Factorisation d'entiers et RSA

$\longrightarrow$ Utilisation de problèmes algorithmiquement difficiles.

Factorisation :

- $p, q \mapsto n = p.q$ facile (quadratique)
- $n = p.q \mapsto p, q$ difficile

fonction RSA $n = pq$, p et q premiers.

e : exposant public

- $x \mapsto x^e \mod n$ facile (cubique)
- $y = x^e \mapsto x \mod n$ difficile
  $x = y^d$ où $d = e^{-1} \mod \phi(n)$

Problèmes algorithmiquement difficiles

On considère le groupe $G = (\mathbb{Z}/n\mathbb{Z})$, avec $n = p.q$, p, q nombres premiers et g un générateur du groupe.

- RSA : étant donnés g^a et a, trouver g.

Problèmes algorithmiquement difficiles

On considère le groupe $G = (\mathbb{Z}/n\mathbb{Z})$, avec $n = p.q$, p, q nombres premiers et g un générateur du groupe.

- RSA : étant donnés g^a et a, trouver g.
- Discrete logarithm (DL) : étant donnés g^a et g, trouver a.

Problèmes algorithmiquement difficiles

On considère le groupe $G = (\mathbb{Z}/n\mathbb{Z})$, avec $n = p.q$, p, q nombres premiers et g un générateur du groupe.

- RSA : étant donnés g^a et a, trouver g.
- Discrete logarithm (DL) : étant donnés g^a et g, trouver a.
- Computational Diffie-Hellman (CDH) : étant donnés g, g^a et g^b, trouver g^{ab}.

Problèmes algorithmiquement difficiles

On considère le groupe $G = (\mathbb{Z}/n\mathbb{Z})$, avec $n = p.q$, p, q nombres premiers et g un générateur du groupe.

- RSA : étant donnés g^a et a, trouver g.
- Discrete logarithm (DL) : étant donnés g^a et g, trouver a.
- Computational Diffie-Hellman (CDH) :
  étant donnés g, g^a et g^b, trouver g^{ab}.
- Decisional Diffie-Hellman (DDH) :
  étant donnés g, g^a, g^b et g^c, a-t-on $c = ab \mod |G|$?

$$DDH < CDH < DL$$

Estimations pour la factorisation d'entiers

Lenstra-Verheul 2000

Module (bits)	Opérations (en $\log_2$)
512	58
1024	80
2048	111
4096	149
8192	156

$\approx 2^{60}$ ans

$\rightarrow$ Bornes inférieures pour RSA aussi.

Plan

Modélisation de l'adversaire

On veut modéliser un attaquant :

- le plus intelligent possible
  $\longrightarrow$ il peut faire toutes les opérations qu'il souhaite
- qui dispose d'un temps limité.

Modélisation de l'adversaire

On veut modéliser un attaquant :

- le plus intelligent possible
  $\longrightarrow$ il peut faire toutes les opérations qu'il souhaite
- qui dispose d'un temps limité.
  - on ne souhaite pas considérer les attaques faisables en 2^{60} ans

Modélisation de l'adversaire

On veut modéliser un attaquant :

- le plus intelligent possible
  $\longrightarrow$ il peut faire toutes les opérations qu'il souhaite
- qui dispose d'un temps limité.
  - on ne souhaite pas considérer les attaques faisables en 2^{60} ans
  - Sinon, l'adversaire peut toujours énumérer toutes les clefs (temps exponentiel en $2^{taille(clefs)}$)

Modélisation de l'adversaire

On veut modéliser un attaquant :

- le plus intelligent possible
  $\longrightarrow$ il peut faire toutes les opérations qu'il souhaite
- qui dispose d'un temps limité.
  - on ne souhaite pas considérer les attaques faisables en 2^{60} ans
  - Sinon, l'adversaire peut toujours énumérer toutes les clefs (temps exponentiel en $2^{taille(clefs)}$)

Modèle : on considère n'importe quelle machine de Turing

- pour modéliser n'importe quel algorithme

Modélisation de l'adversaire

On veut modéliser un attaquant :

- le plus intelligent possible
  $\longrightarrow$ il peut faire toutes les opérations qu'il souhaite
- qui dispose d'un temps limité.
  - on ne souhaite pas considérer les attaques faisables en 2^{60} ans
  - Sinon, l'adversaire peut toujours énumérer toutes les clefs (temps exponentiel en $2^{taille(clefs)}$)

Modèle : on considère n'importe quelle machine de Turing

- pour modéliser n'importe quel algorithme
- probabiliste : l'adversaire peut générer des clefs, tirer au sort certaines étapes de son comportement

Modélisation de l'adversaire

On veut modéliser un attaquant :

- le plus intelligent possible
  $\longrightarrow$ il peut faire toutes les opérations qu'il souhaite
- qui dispose d'un temps limité.
  - on ne souhaite pas considérer les attaques faisables en 2^{60} ans
  - Sinon, l'adversaire peut toujours énumérer toutes les clefs (temps exponentiel en $2^{taille(clefs)}$)

Modèle : on considère n'importe quelle machine de Turing

- pour modéliser n'importe quel algorithme
- probabiliste : l'adversaire peut générer des clefs, tirer au sort certaines étapes de son comportement
- polynomiale en la taille des clefs : représente le temps raisonnable d'éxécution.

Principe des preuves de sécurité

Preuve par réduction

1. Hypothèse : Le problème algorithmique P est difficile = il n'y a pas d'algo polynomial ($P = RSA, DL, DDH, CDH$...)

Principe des preuves de sécurité

Preuve par réduction

1. Hypothèse : Le problème algorithmique P est difficile = il n'y a pas d'algo polynomial ($P = RSA, DL, DDH, CDH$...)
2. Réduction :
   - Si A un adversaire (polynomial) casse le schéma de chiffrement,
   - Alors A peut-être utilisé pour résoudre P en temps polynomial.

Principe des preuves de sécurité

Preuve par réduction

1. Hypothèse : Le problème algorithmique P est difficile = il n'y a pas d'algo polynomial ($P = RSA, DL, DDH, CDH$...)
2. Réduction :
   - Si A un adversaire (polynomial) casse le schéma de chiffrement,
   - Alors A peut-être utilisé pour résoudre P en temps polynomial.
3. Résultat de sécurité : il n'existe pas d'adversaire polynomial

Définitions

Comment définir la sécurité d'un algorithme de chiffrement ?

$\longrightarrow$ Plusieurs propositions.

One-Wayness (OW)

Adversaire $\mathcal{A}$: une machine de Turing probabiliste et polynomial (PPTM)

Notion de sécurité de base : One-Wayness (OW)

Sans la clef privée, il est impossible d'obtenir le texte en clair :

$$\Pr_{m,r}[c = E(m; r) \mid \mathcal{A}(c) = m]$$

est négligeable.

One-Wayness (OW)

Adversaire $\mathcal{A}$: une machine de Turing probabiliste et polynomial (PPTM)

Notion de sécurité de base : One-Wayness (OW)
Sans la clef privée, il est impossible d'obtenir le texte en clair :

$$\Pr_{m,r}[c = E(m; r) \mid \mathcal{A}(c) = m]$$

est négligeable.

Négligeabilité : f est négligeable si pour tout polynôme p, il existe η_0 t.q. pour tout $\eta \geq \eta_0$

$$f(\eta) \leq 1/p(\eta)$$

Ce n'est pas assez

- Cela n'empêche pas de connaître la moitié du texte en clair
- On peut avoir une connaissance partielle du message :
  - Sujet : XXXX
  - Ma réponse est : XXXX

Ce n'est pas assez

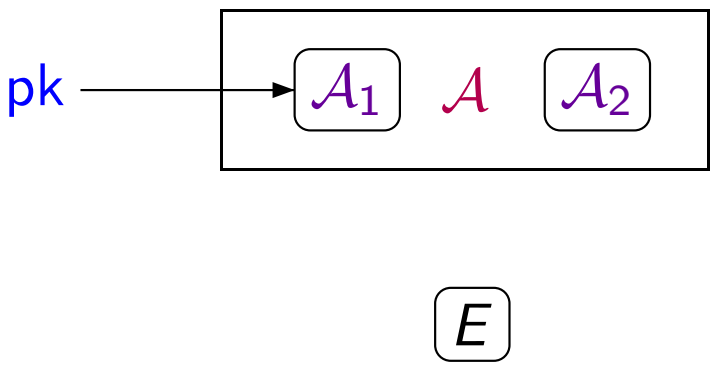
- Cela n'empêche pas de connaître la moitié du texte en clair
- On peut avoir une connaissance partielle du message :
  - Sujet : XXXX
  - Ma réponse est : XXXX

→ Introduction d'une notion d'indistinguabilité :
l'adversaire ne doit pas pouvoir deviner ne serait-ce qu'un bit du message.

Indistinguabilité (IND)

Jeu Adversaire : $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$

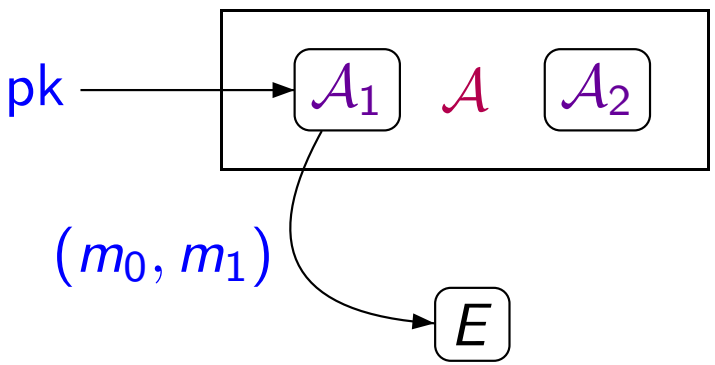
1. On donne à l'adversaire $\mathcal{A}_1$ la clef publique pk.



Indistinguabilité (IND)

Jeu Adversaire : $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$

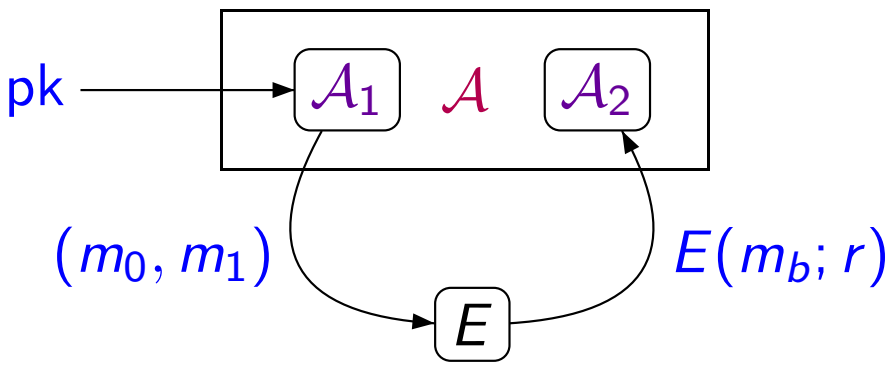
1. On donne à l'adversaire $\mathcal{A}_1$ la clef publique pk.
2. L'adversaire $\mathcal{A}_1$ choisit deux messages m_0, m_1.



Indistinguabilité (IND)

Jeu Adversaire : $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$

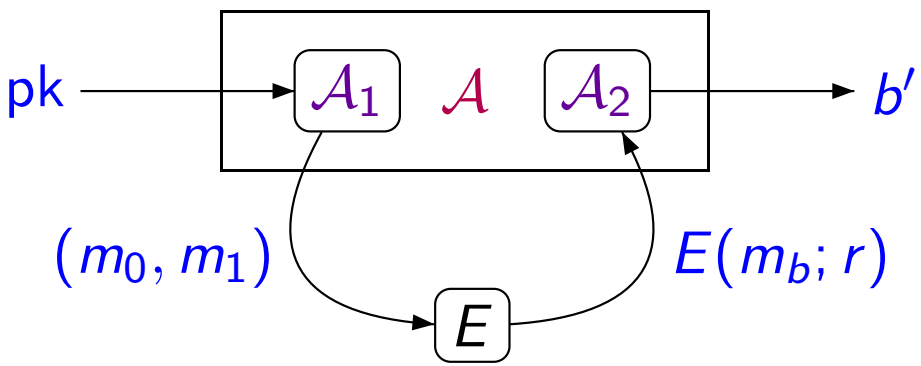
1. On donne à l'adversaire $\mathcal{A}_1$ la clef publique pk.
2. L'adversaire $\mathcal{A}_1$ choisit deux messages m_0, m_1.
3. un bit $b = 0, 1$ est choisi au hasard et on donne $c = E(m_b; r)$ à l'adversaire.



Indistinguabilité (IND)

Jeu Adversaire : $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$

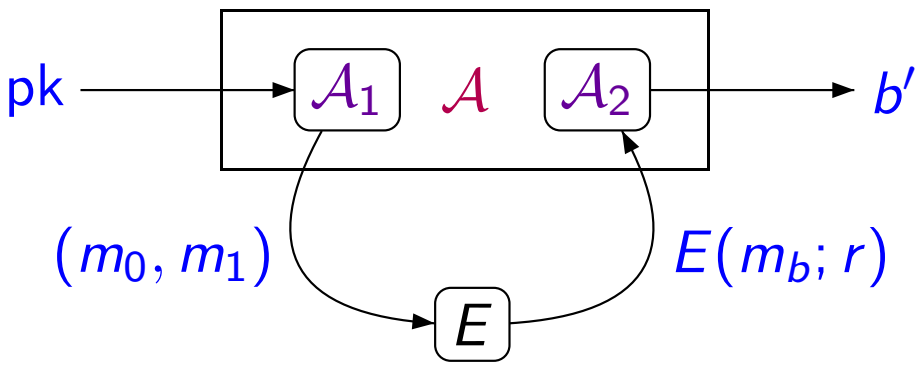
1. On donne à l'adversaire $\mathcal{A}_1$ la clef publique pk.
2. L'adversaire $\mathcal{A}_1$ choisit deux messages m_0, m_1.
3. un bit $b = 0, 1$ est choisi au hasard et on donne $c = E(m_b; r)$ à l'adversaire.
4. L'adversaire $\mathcal{A}_2$ répond par b'.



Indistinguabilité (IND)

Jeu Adversaire : $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$

1. On donne à l'adversaire $\mathcal{A}_1$ la clef publique pk.
2. L'adversaire $\mathcal{A}_1$ choisit deux messages m_0, m_1.
3. un bit $b = 0, 1$ est choisi au hasard et on donne $c = E(m_b; r)$ à l'adversaire.
4. L'adversaire $\mathcal{A}_2$ répond par b'.



La probabilité $\Pr[b = b'] - \frac{1}{2}$ doit être négligeable.

Encore plus fort !

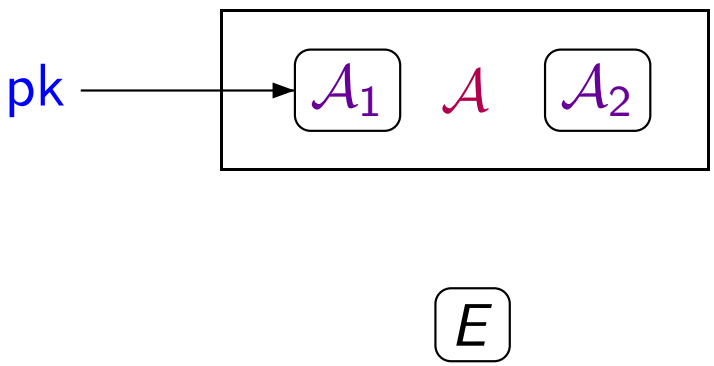
Non Malléabilité (NM)

Étant donné un chiffré $E(m; r)$, l'adversaire ne doit pas être capable de créer un chiffré $E(m'; r')$ tel que les messages m et m' aient un lien.

Non Malléabilité (NM)

Game Adversaire : $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$

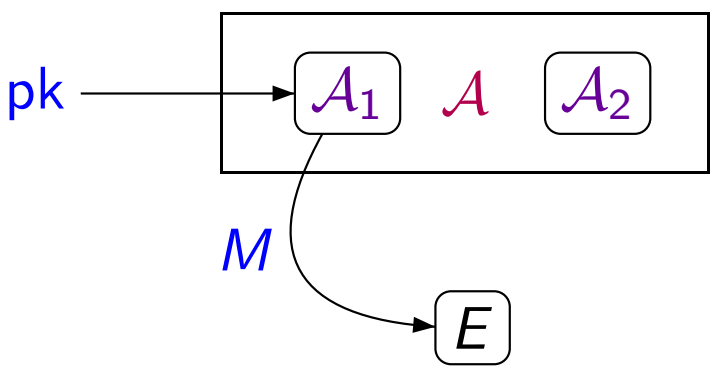
1. On donne la clef publique pk à l'adversaire $\mathcal{A}_1$



Non Malléabilité (NM)

Game Adversaire : $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$

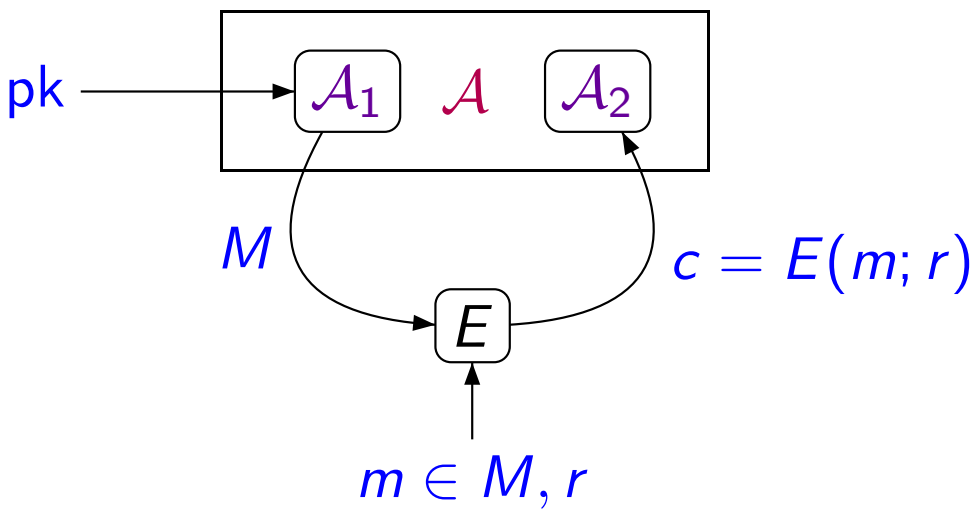
1. On donne la clef publique pk à l'adversaire $\mathcal{A}_1$
2. L'adversaire $\mathcal{A}_1$ choisit un ensemble de messages M.



Non Malléabilité (NM)

Game Adversaire : $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$

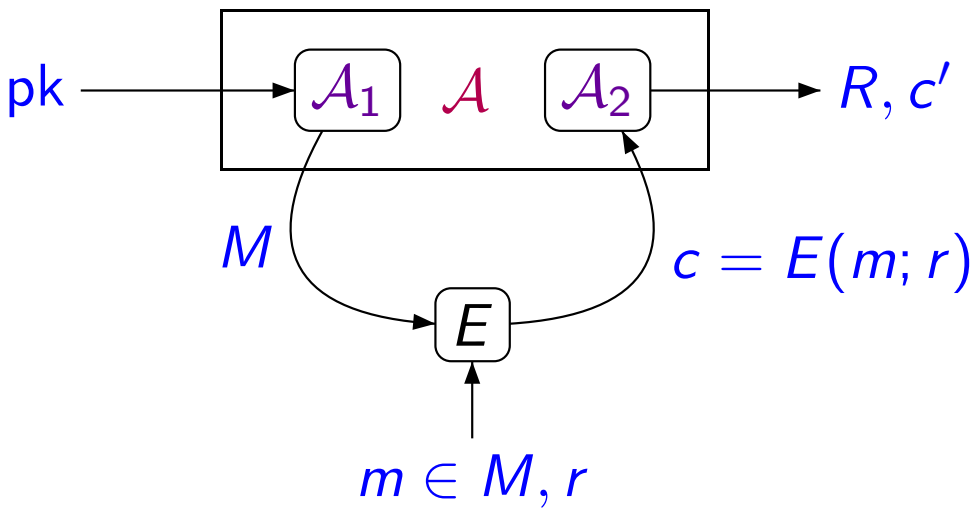
1. On donne la clef publique pk à l'adversaire $\mathcal{A}_1$
2. L'adversaire $\mathcal{A}_1$ choisit un ensemble de messages M.
3. Deux messages m et m^* sont choisis au hasard dans M et on donne $c = E(m; r)$ à l'adversaire.



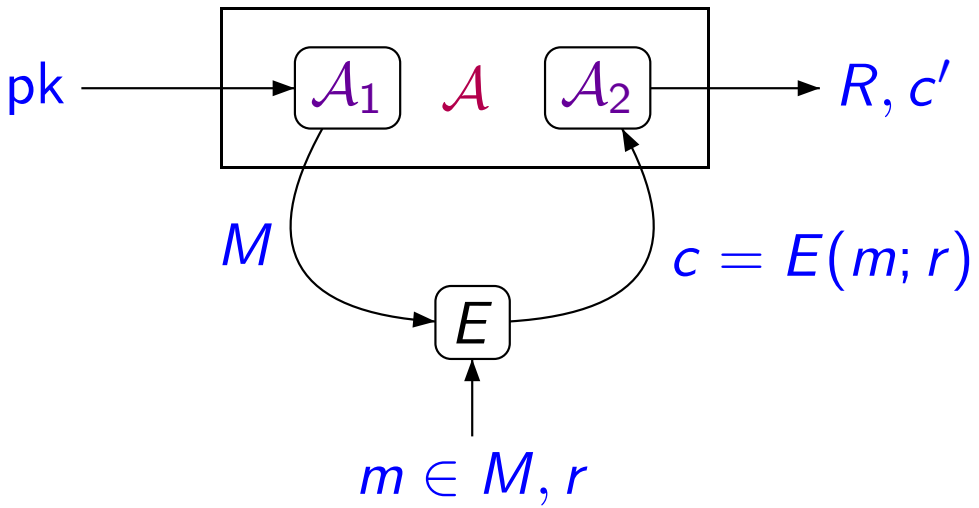
Non Malléabilité (NM)

Game Adversaire : $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$

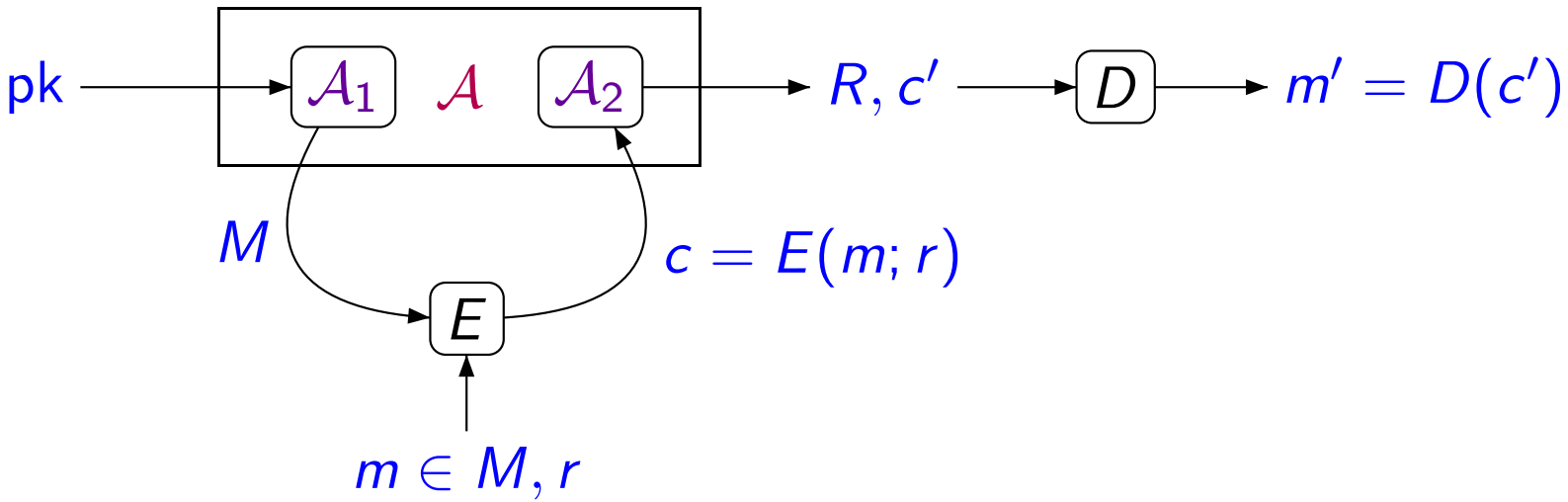
1. On donne la clef publique pk à l'adversaire $\mathcal{A}_1$
2. L'adversaire $\mathcal{A}_1$ choisit un ensemble de messages M.
3. Deux messages m et m^* sont choisis au hasard dans M et on donne $c = E(m; r)$ à l'adversaire.
4. L'adversaire $\mathcal{A}_2$ renvoie une relation binaire R et un chiffré c'.



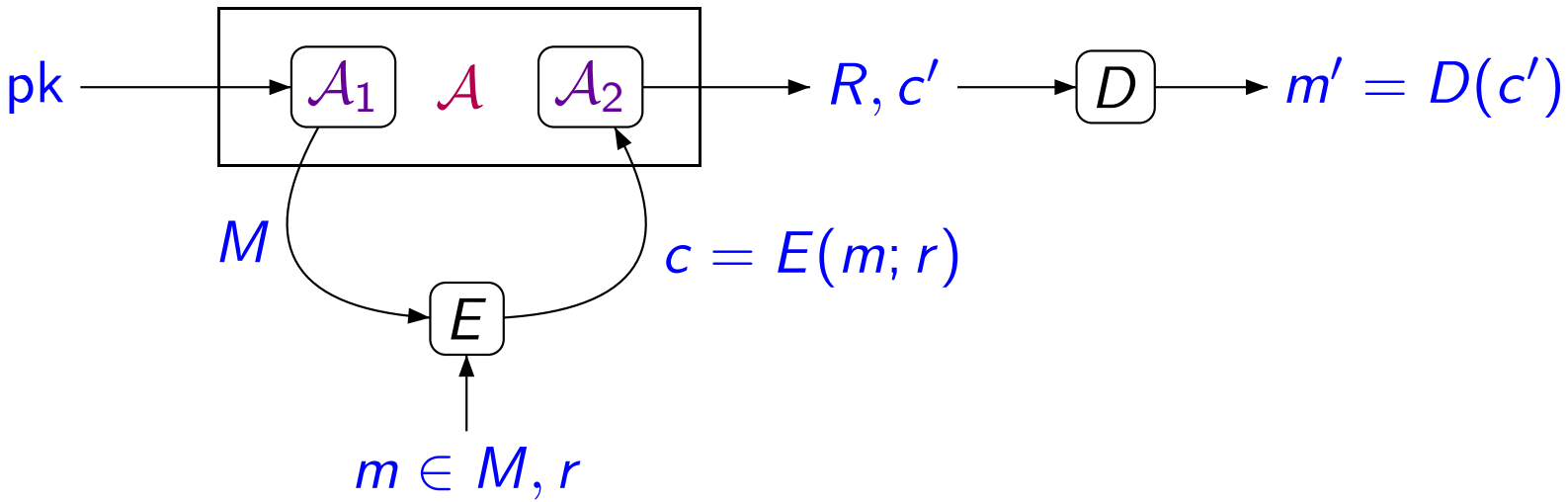
Non Malléabilité (NM)



Non Malléabilité (NM)



Non Malléabilité (NM)



La probabilité que $\Pr[R(m, m')] - \Pr[R(m, m^*)]$ doit être négligeable.

Relations

Non Malléabilité

$\Downarrow$

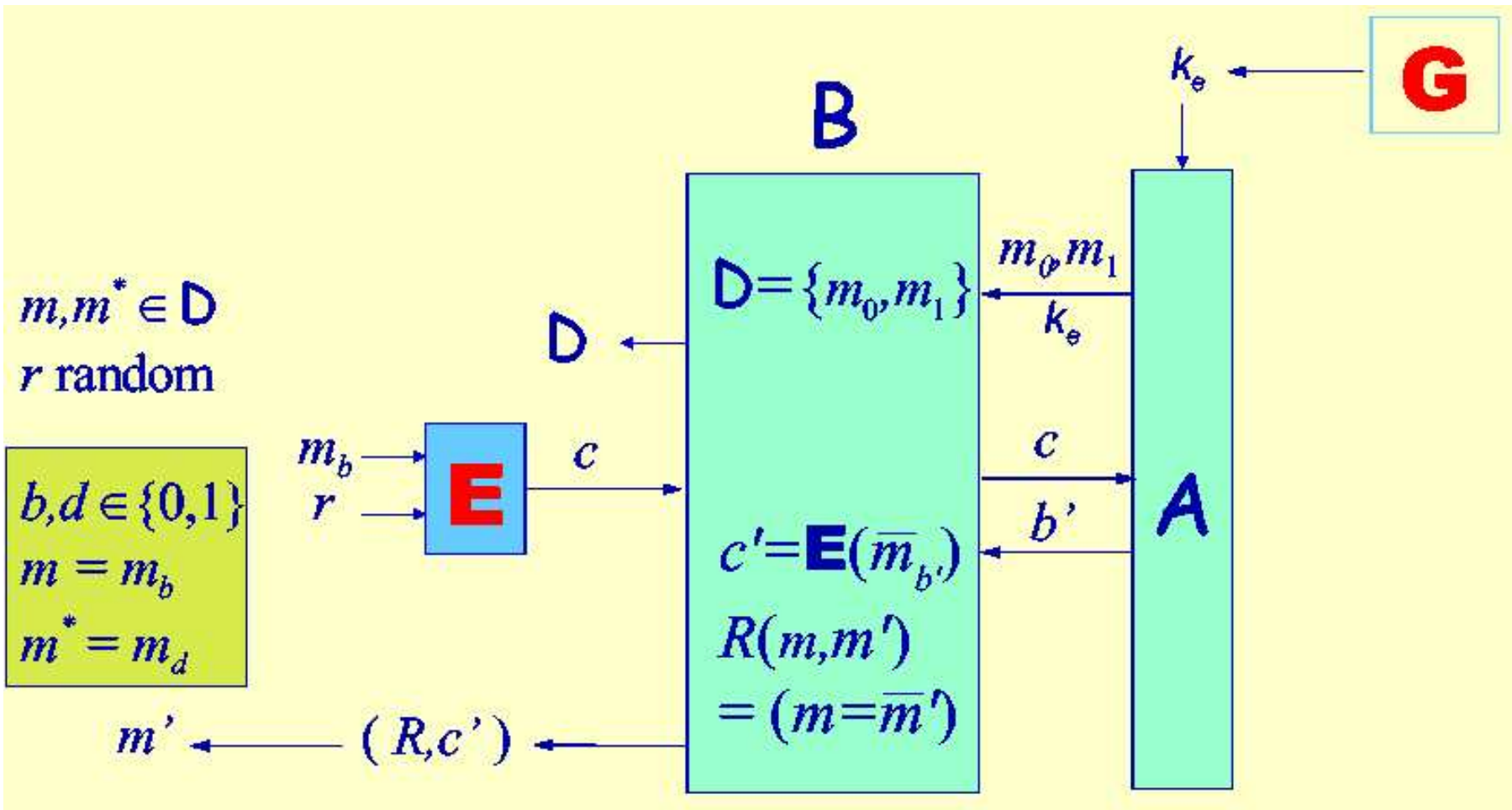
Indistinguabilité

$\Downarrow$

One-Wayness

IND ⇒ NM - Construction

transparent emprunté à David Pointcheval



IND ⇒ NM - Preuve (1)

transparent emprunté à David Pointcheval

$c' = \mathbf{E}(m_{b'})$ $R(m,m') = (m = \overline{m}')$

- Under the assumptions
  - $m_0 \neq m_1$ and $b \neq d$
  - otherwise: advantage = 0
- If $b' = b$: $m' = \overline{m}_b$

  $R(m,m') = (m_b = \overline{m}') = 1$ $R(m^*,m') = (m_d = \overline{m}') = (b = d)$
- If $b' \neq b$: $m' = \overline{m}_{1-b}$

  $R(m,m') = (m_b = \overline{m}') = 0$ $R(m^*,m') = (m_d = \overline{m}') = (b \neq d)$

IND $\Rightarrow$ NM - Preuve (2)

transparent emprunté à David Pointcheval

$$\mathrm{Adv}^{nm}(\mathsf{B}|b \neq d)$$

$$= \Pr[R(m,m') \mid b \neq d] - \Pr[R(m^*,m') \mid b \neq d]$$

$$= 1 \times \Pr[b' = b \mid b \neq d] + 0 \times \Pr[b' \neq b \mid b \neq d]$$

$$- \tfrac{1}{2} \times \Pr[b' = b \mid b \neq d] - \tfrac{1}{2} \times \Pr[b' \neq b \mid b \neq d]$$

$$= \Pr[b' = b \mid b \neq d] - \tfrac{1}{2} = \mathrm{Adv}^{ind}(\mathsf{A} \mid b \neq d)/2$$

$$\mathrm{Adv}^{nm}(\mathsf{B} \mid b = d) = 0$$

$$\mathrm{Adv}^{ind}(\mathsf{A} \mid m_0 = m_1) = 0$$

$$\mathrm{Adv}^{nm}(\mathsf{B}) = \mathrm{Adv}^{ind}(\mathsf{A})/4$$

Ajouter encore plus de sécurité

L'adversaire a accès à des oracles :

$\longrightarrow$ chiffrement de tous les messages de son choix
$\longrightarrow$ déchiffrement de tous les messages de son choix

Trois niveaux classiques de sécurité :

- Chosen-Plaintext Attacks (CPA)

Ajouter encore plus de sécurité

L'adversaire a accès à des oracles :

$\longrightarrow$ chiffrement de tous les messages de son choix
$\longrightarrow$ déchiffrement de tous les messages de son choix

Trois niveaux classiques de sécurité :

- Chosen-Plaintext Attacks (CPA)
- Non adaptive Chosen-Ciphertext Attacks (CCA1)
  $\longrightarrow$ accès à l'oracle seulement avant le challenge.

Ajouter encore plus de sécurité

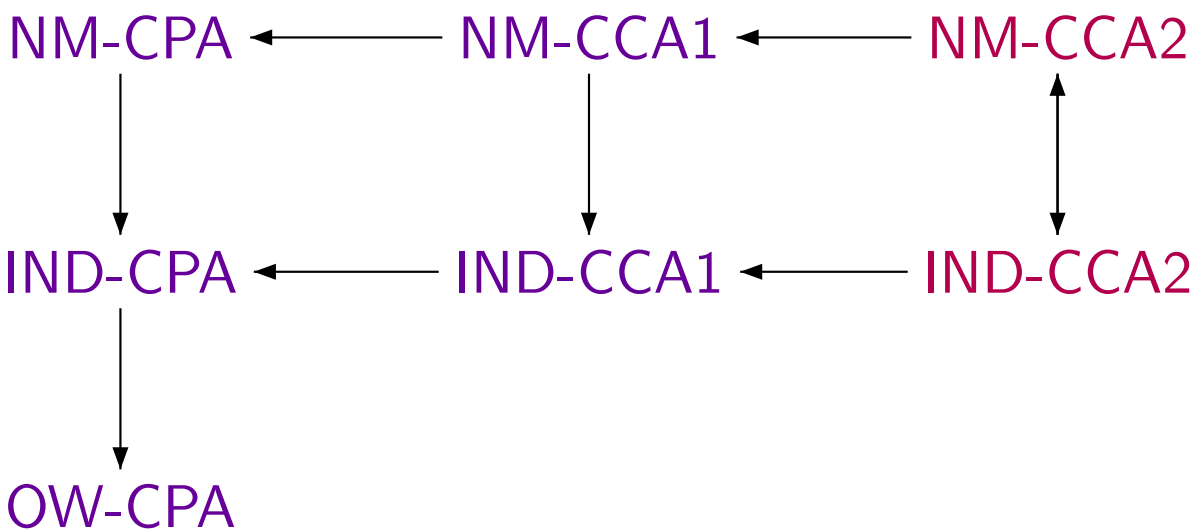
L'adversaire a accès à des oracles :

$\longrightarrow$ chiffrement de tous les messages de son choix
$\longrightarrow$ déchiffrement de tous les messages de son choix

Trois niveaux classiques de sécurité :

- Chosen-Plaintext Attacks (CPA)
- Non adaptive Chosen-Ciphertext Attacks (CCA1)
  $\longrightarrow$ accès à l'oracle seulement avant le challenge.
- Adaptive Chosen-Ciphertext Attacks (CCA2)
  $\longrightarrow$ accès illimité à l'oracle (sauf pour le challenge)

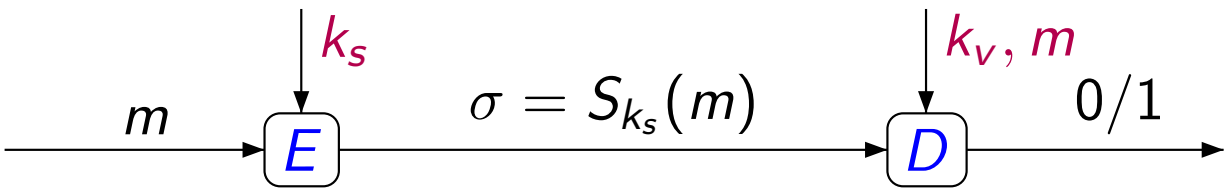
Relations



Signatures

Trois algorithmes

- G - génération de clef
- S - signature
- V - vérification



Sécurité des signatures

Propriété de base : Existentially unforgeable

Definition

Sans la clef privée, il est impossible de créer un couple de (m, σ) valide.

$$\Pr[V(m,\sigma) = 1 \mid \mathcal{A}(k_v) = (m,\sigma)]$$

Plan

Exemple : RSA

public	privé
$n = pq$	$d = e^{-1} \mod \phi(n)$
e (clef publique)	(clef privée)

Chiffrement RSA

- $E(m) = m^e \mod n$
- $D(c) = c^d \mod n$

OW-CPA = problème RSA par définition !

Exemple : RSA

public	privé
$n = pq$	$d = e^{-1} \mod \phi(n)$
e (clef publique)	(clef privée)

Chiffrement RSA

- $E(m) = m^e \mod n$
- $D(c) = c^d \mod n$

OW-CPA = problème RSA par définition !

Mais sujet à des attaques de type chiffré choisi adaptable (Daniel Bleichenbacher, 1998).

El Gamal

Basé sur le problème du logarithme discret dans le groupe $G = (\mathbb{Z}/n\mathbb{Z})^*$, avec $n = p.q$, p, q nombres premiers et g un générateur du groupe.

On choisit d tel que $1 \leq d \leq n-2$. Soit $b = g^d \mod n$.

Clé publique : $k = (n, g, b)$
Clé privée : d

El Gamal

Basé sur le problème du logarithme discret dans le groupe $G = (\mathbb{Z}/n\mathbb{Z})^*$, avec $n = p.q$, p, q nombres premiers et g un générateur du groupe.

On choisit d tel que $1 \leq d \leq n-2$. Soit $b = g^d \mod n$.

Clé publique : $k = (n, g, b)$
Clé privée : d

Chiffrement

$E_k(M) = C = (c_1, c_2)$ avec $c_1 = g^s \mod n$ et $c_2 = M.b^s \mod n$

s nombre aléatoire généré par celui qui chiffre le message.

El Gamal

Basé sur le problème du logarithme discret dans le groupe $G = (\mathbb{Z}/n\mathbb{Z})^*$, avec $n = p.q$, p, q nombres premiers et g un générateur du groupe.

On choisit d tel que $1 \leq d \leq n-2$. Soit $b = g^d \mod n$.

Clé publique : $k = (n, g, b)$
Clé privée : d

Chiffrement

$E_k(M) = C = (c_1, c_2)$ avec $c_1 = g^s \mod n$ et $c_2 = M.b^s \mod n$

s nombre aléatoire généré par celui qui chiffre le message.

Déchiffrement

$D_d(C) = c_2/(c_1)^d = (M.b^s)/(g^s)^d = M.g^{(d.s)}/g^{(d.s)} = M \mod n$

Sécurité de El Gamal (OW-CPA)

Theorem (OW-CPA)

El Gamal est OW-CPA sous l'hypothèse CDH difficile.

Preuve : En exercice

Sécurité de El Gamal (IND-CPA)

Clé publique : $k = (n, g, b)$

Clé privée : d

$E_k(M) = C = (c_1, c_2)$ avec $c_1 = g^s \mod n$ et $c_2 = M.b^s \mod n$

Theorem (IND-CPA)

El Gamal est IND-CPA sous l'hypothèse DDH difficile.

Sécurité de El Gamal (IND-CPA)

Clé publique : $k = (n, g, b)$
Clé privée : d

$E_k(M) = C = (c_1, c_2)$ avec $c_1 = g^s \mod n$ et $c_2 = M.b^s \mod n$

Theorem (IND-CPA)

El Gamal est IND-CPA sous l'hypothèse DDH difficile.

Preuve : (intuition) Soit $\mathcal{A}$ un adversaire qui casse El Gamal pour IND-CPA. On construit $\mathcal{B}$ qui casse DDH.

- $\mathcal{B}$ reçoit (g^a, g^b, g^c). Il doit dire si $g^{ab} = g^c$.
- $\mathcal{B}$ envoie la clef publique g^a à $\mathcal{A}$
- $\mathcal{A}$ calcule (m_0, m_1)
- $\mathcal{B}$ choisit $b \in \{0, 1\}$ au hasard et envoie $c = (B, Cm_b)$.
- $\mathcal{A}$ envoit b'
- $\mathcal{B}$ répond par $(b = b')$.

Plan

RSA-OAEP

Format de chiffrement RSA normalisé

OAEP (Optimal Asymmetric Encryption Padding)
schéma de remplissage

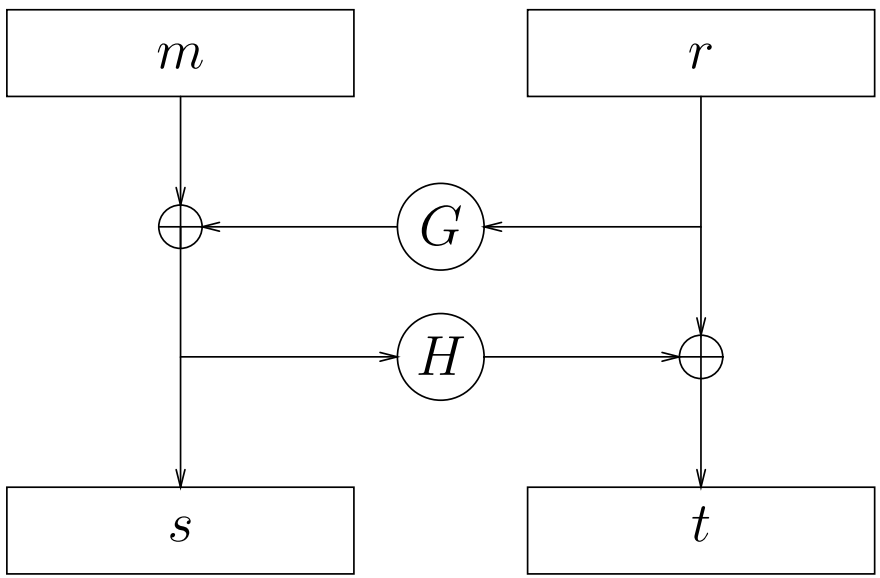
- Introduit en 1994 par Mihir Bellare et Phil Rogaway.
- Il nécessite une source d'aléa,
- ainsi que deux fonctions de hachage.

Reconnu par de nombreuses normes :
RSA-PKCS, SET, IETF, IEEE, ISO

Description de RSA-OAEP

Soient G et H deux fonctions de hachage.
Soit r un nombre aléatoire.

$$\mathrm{OAEP}(M) = [(M \oplus G(r))||(r \oplus H(M \oplus G(r)))]$$

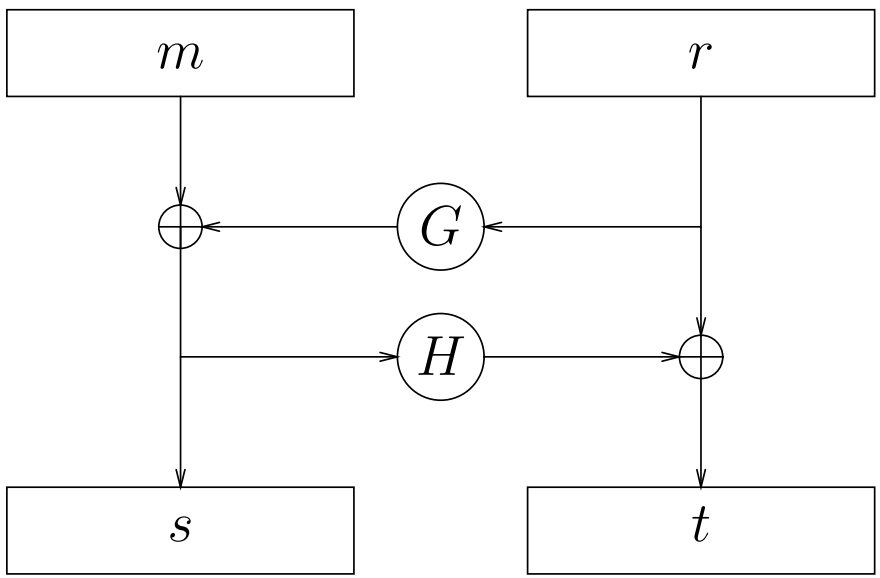


Description de RSA-OAEP

Soient G et H deux fonctions de hachage.
Soit r un nombre aléatoire.

$$\mathrm{OAEP}(M) = [(M \oplus G(r))||(r \oplus H(M \oplus G(r)))]$$

$$\mathrm{RSA-OAEP}_{n,e}(M) = \mathrm{OAEP}(M)^e \mod n$$



Sécurité de RSA-OAEP

Theorem

Le schéma de chiffrement RSA-OAEP est IND-CCA2.

Pour chiffrer de longs messages

On combine chiffrement asymétrique et symétrique :

$$\mathrm{enca}(m, \mathrm{pub}(a)) = [k]_{\mathrm{pub}(a)}, \{m\}_k$$

- k est une clef symétrique fraîche
- $[\_]_\_$ est un algorithme de chiffrement asymétrique (RSA-AOEP) par exemple
- $\{\_\}_\_$ est un algorithme de chiffrement symétrique

Plan

Chiffrement parfait

- Utilisation d'un masque aléatoire qui est la clé.
- Ou exclusif bit-à-bit

$$\begin{aligned} 0 \oplus 0 &= 0 \\ 0 \oplus 1 &= 1 \\ 1 \oplus 0 &= 1 \\ 1 \oplus 1 &= 0 \end{aligned}$$

Chiffrement : $\{m\}_k = m \oplus k$

Chiffrement parfait

- Utilisation d'un masque aléatoire qui est la clé.
- Ou exclusif bit-à-bit

$$\begin{aligned} 0 \oplus 0 &= 0 \\ 0 \oplus 1 &= 1 \\ 1 \oplus 0 &= 1 \\ 1 \oplus 1 &= 0 \end{aligned}$$

Chiffrement : $\{m\}_k = m \oplus k$

Exemple :

- $M = 0110101011010100$ (message en clair)
- $K = 0101011011100110$ (la clé K secrète)
- Chiffrement : $C = M \oplus K = 0011110000110010$
- Déchiffrement : $M = C \oplus K = 0110101011010100$

Avantages et Inconvénients

Avantages et Inconvénients

Avantages

- Parfaitement sûr
- Très rapide

Inconvénients

- La clef ne doit pas être réutilisée
- Il faut un bon générateur d'aléa pour la clef

Avantages et Inconvénients

Avantages

- Parfaitement sûr
- Très rapide

Inconvénients

- La clef ne doit pas être réutilisée
- Il faut un bon générateur d'aléa pour la clef

Applications essentiellement militaires (téléphone rouge, ...)

DES (Data Encryption Standard) - 1

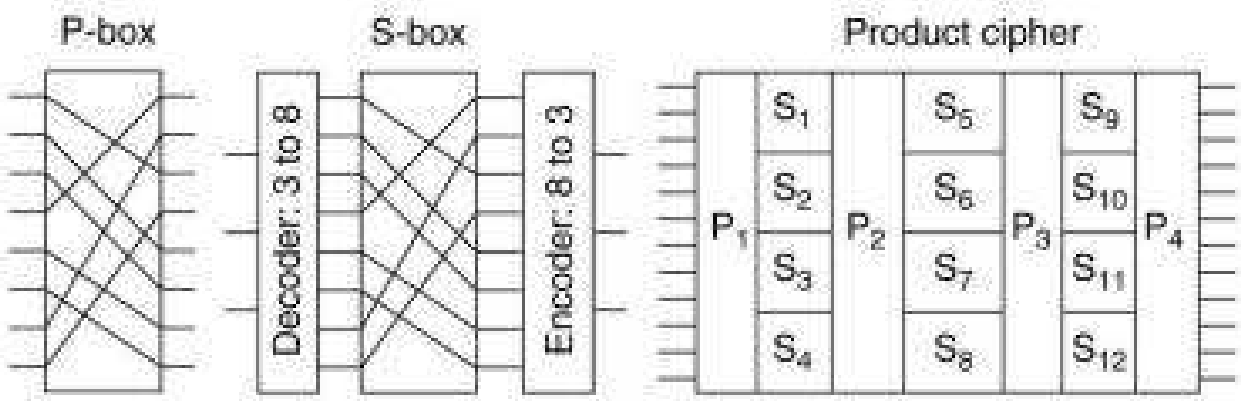
- Inventé par IBM au début des années 70 (128 bits)
- Modifié par la NSA avant sa déclassification (56 bits)

DES (Data Encryption Standard) - 1

- Inventé par IBM au début des années 70 (128 bits)
- Modifié par la NSA avant sa déclassification (56 bits)

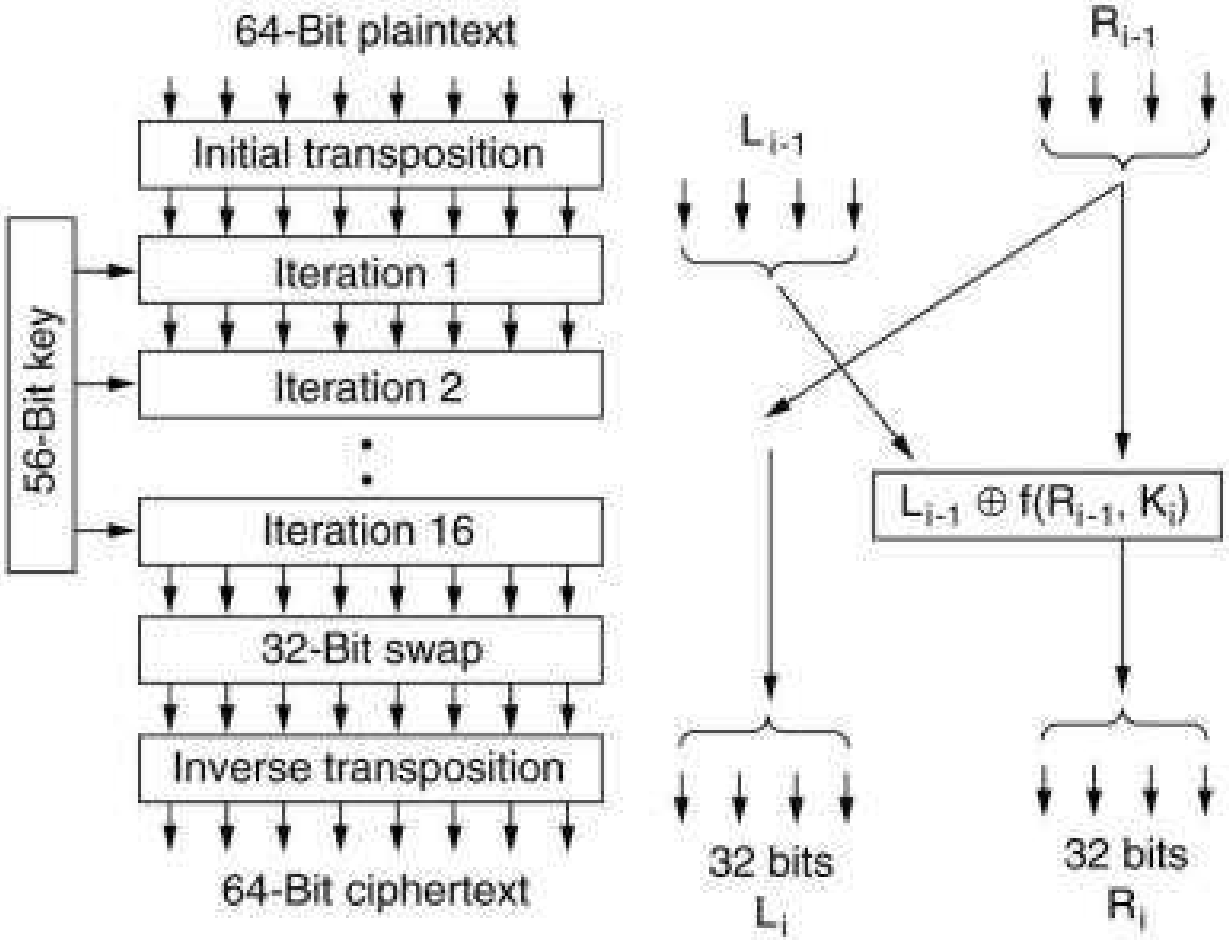
Comment ça marche ?

- substitutions *S*-Box
- permutations *P*-Box



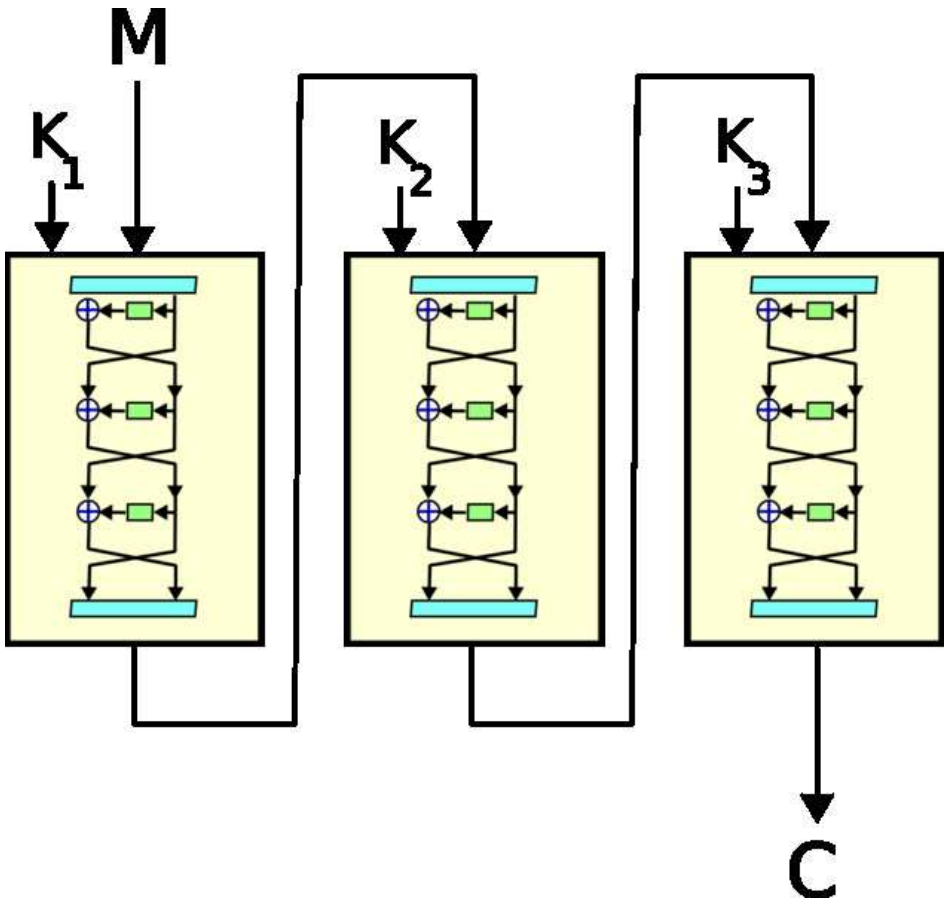
→ Contrairement à la cryptographie asymétrique, pas de sécurité prouvée.

DES (Data Encryption Standard) - 2



Triple DES

Passage à des clefs de 112 bits



AES (Advanced Encryption Standard)

- Résultat d'une compétition pour le nouveau standard de chiffrement américain
- Gagné par Rijndael avec 2001
- Chiffrement à 128 ou 256 bits
- Basé sur la théorie de Galois

Pour chiffrer de longs messages

- On sait chiffrer des petits messages (blocs)
  $\longrightarrow$ Chiffrement par bloc
- Comment assembler les blocs ?

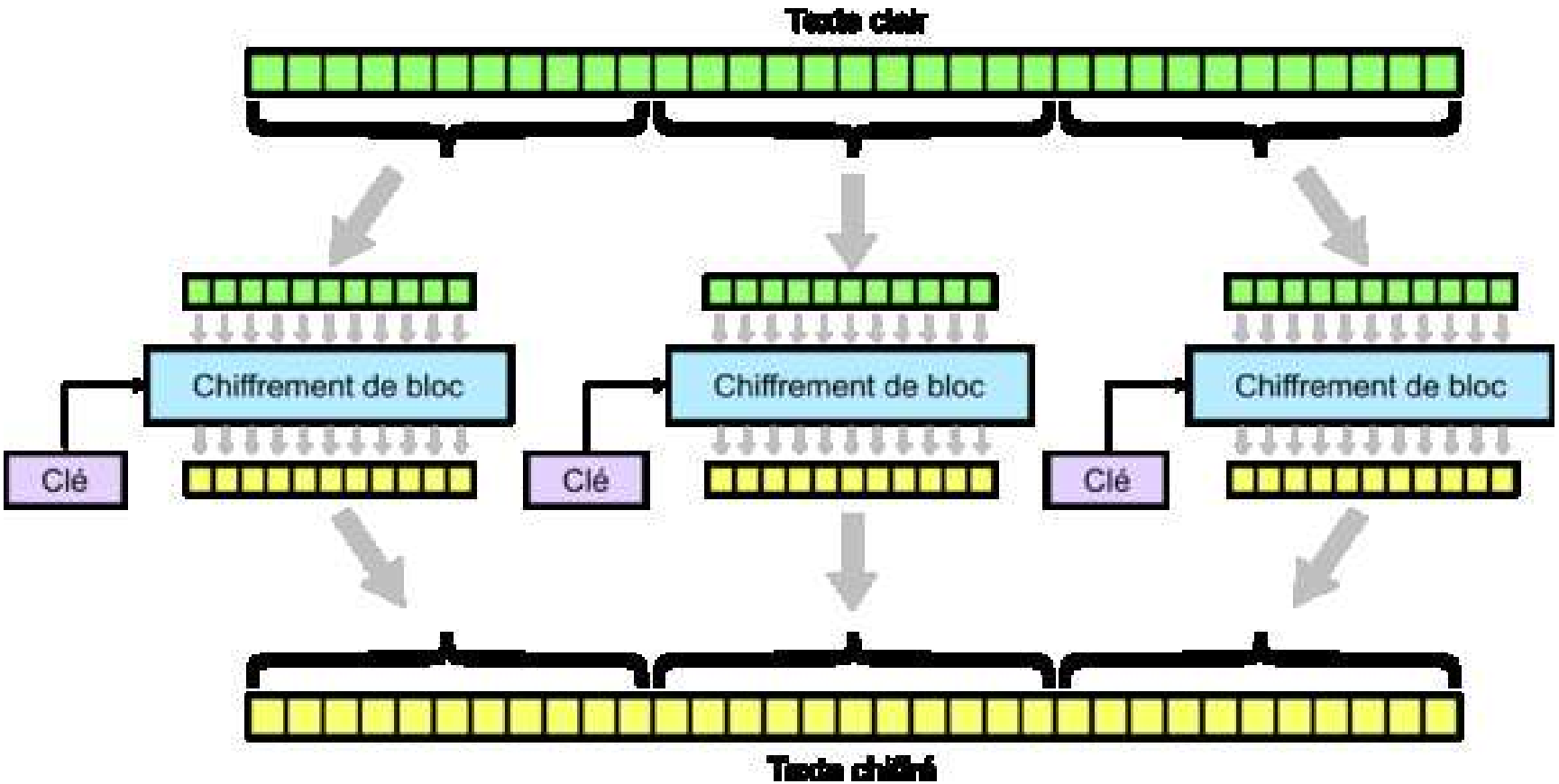
Pour chiffrer de longs messages

- On sait chiffrer des petits messages (blocs)
  $\longrightarrow$ Chiffrement par bloc
- Comment assembler les blocs ?

Deux modes classiques

- Electronic codebook (ECB)
- Cipher Block Chaining (CBC)

Electronic codebook (ECB)



Inconvénients

?

Inconvénients

- Malléable :

  À partir de $\{m1 \cdot m_2 \cdots m_p\}_k = \{m_1\}_k \cdot \{m_2\}_k \cdots \{m_p\}_k$,
  on peut calculer $\{m2 \cdot m_1 \cdots m_p\}_k$

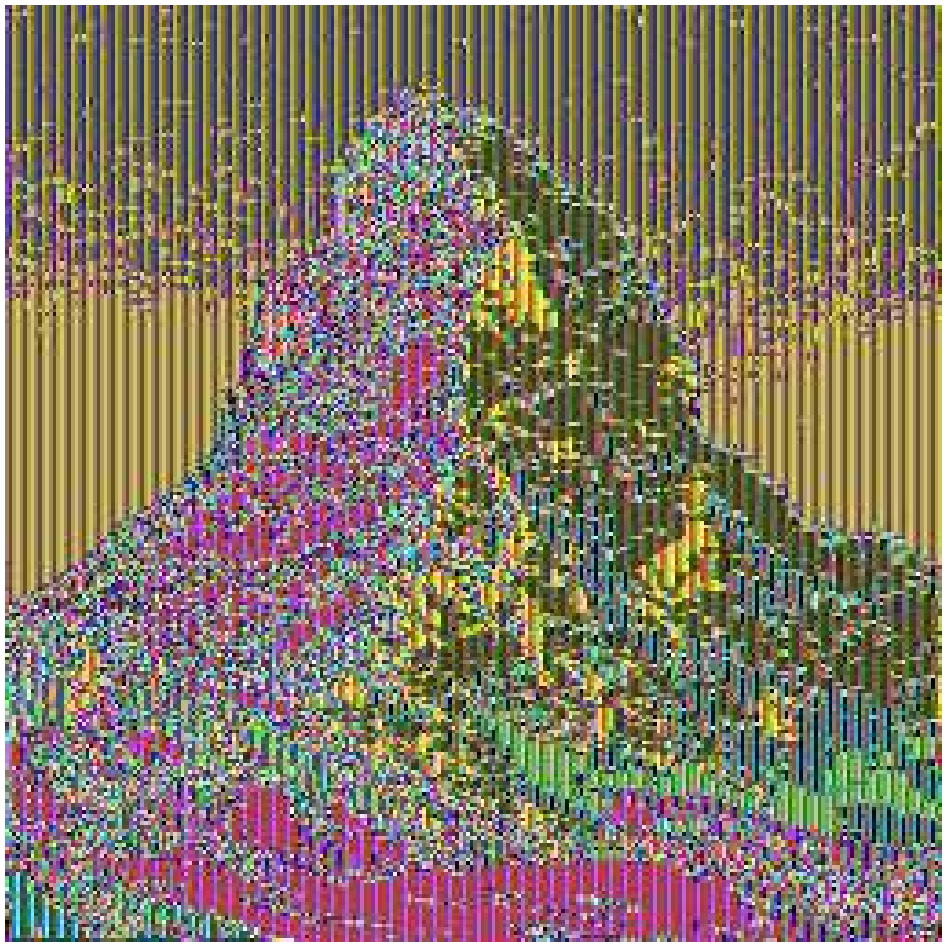
- On peut tester l'égalité de certains blocs :
  $\{m_1\}_k = \{m_2\}_k \Rightarrow m_1 = m_2$

Exemple en images

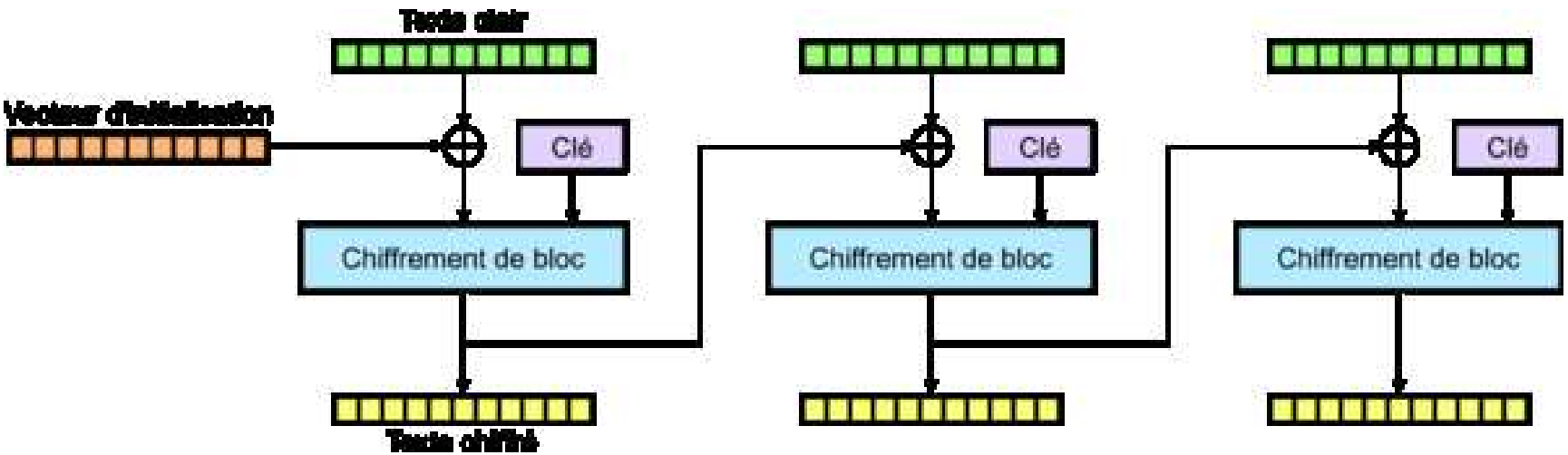
Le Cervin

Exemple en images

Le Cervin chiffré avec ECB



Cipher Block Chaining (CBC)



Exemple en images

Le Cervin chiffré avec CBC