Introduction à la sécurité
TD 1

Ninon Devis: ninon.devis@ircam.fr

Master 2021

1.1 Taille des clefs

1.1 Combien de déchiffrement DES cette machine est-elle capable d'effectuer par seconde ?

- DES: (Data Encryption Standard) système de chiffrement symétrique par bloc de 64 bits. Approuvé en 1978 par le NBS (National Bureau of Standards)

- 16 cartes-support avec chacune 8 FPGAs capable d'exécuter 4 “coeurs” de chiffrement DES en parallèle. Cadencé à 140 MHz avec un chiffrement DES par cycle d'horloge.
- Rappel 1 Mhz = 10^6 Hz

→ 16 x 8 x 4 x 140 x 10^6 = 71 680 000 000 DES/sec

1.1 Taille des clefs

1.2 En combien de temps une COPACOBANA peut-elle casser le DES (qui a des clefs de 56 bits) ?

→ $2^{56} / 71{,}680.10^9 \approx$ 1 005 268 sec soit entre 11 et 12 jours.

1.3 En combien de temps une COPACOBANA peut-elle casser SkipJack (qui a des clefs de 80 bits), en admettant qu'un chiffrement skipjack prend le même temps qu'un chiffrement DES ?

→ $2^{80} / 71{,}680.10^9 \approx$ 0.5 millions d'années.

1.1 Taille des clefs

1.4 Combien y a-t-il de mots de passe de 8 caractères, en autorisant minuscules, majuscules, chiffres et quelques signes de ponctuation ? Quelle est la taille de clef secrète correspondante ? (ceci est parfois nommé l'entropie du mot de passe).

→ 26 MAJ + 26 min + 10 Chiffres + 2 ponctuations = 64 caractères.

→ 64^8 possibilités, soit $(2^6)^8 = 2^{48}$ combinaisons.

→ Un mot de passe « compliqué » a donc la même « force » qu'une clef de 48 bits!

1.5 Un coeur d'un CPU contemporain, est capable de faire 8 millions de chiffrements AES par seconde. Combien de coeurs faut-il pour casser un tel mot de passe en une semaine ?

Un coeur: $7 \times 24 \times 3600 \times 8 \cdot 10^6 \approx 4.8 \cdot 10^{12}$ essais par semaine.

Il faut faire 2^{48} essais en tout => $2^{48} / 4.8 \cdot 10^{12} \approx$ une soixantaine de coeurs.

1.1 Taille des clefs

1.6 Bremermann a démontré qu'un système matériel auto-suffisant ne peut pas réaliser plus de $1.36 \cdot 10^{50}$ opérations par seconde et par kilo de matière (cette valeur est environ c^2/h, où c est la vitesse de la lumière et h est la constante de Planck). Sachant que la planète terre pèse $5.972 \cdot 10^{24}$ kg, donner une borne inférieure sur le temps nécessaire pour casser une clef secrète de 512 bits par force brute.

➔ La terre peut faire au maximum $5.972 \cdot 10^{24} \times 1.36 \cdot 10^{50} \approx 8 \cdot 10^{74}$ opérations par seconde.

➔ Donc, pour faire les $2^{512} \approx 1.34 \cdot 10^{154}$ opérations qui s'imposent, il faut au moins $\approx 5 \cdot 10^{71}$ années. (Rappel : a priori,le soleil disparaît dans 10^9 années.)

➔ NB. Il faudrait 10^{-36} sec pour casser une clef 128 et 2 min pour une clef 256.

1.1 Taille des clefs

1.9 Considérons l'algorithme suivant =>
Quelle est sa probabilité de succès,
en fonction de n et k ?

```
procedure ManyTrials(C, n, k)
    K ← Random bit string of size n
    P ← D(K, C)
    repeat
        if P ≠ ⊥ then
            Return (P, K)
        end if
    until k trials have failed
    Return ⊥
end procedure
```

- Probabilité de succès pour un essai ?

➔ 2^n clefs possibles donc 2^{-n} probabilité de succès

- On fait k essais, quelle probabilité de succès totale ? (Inégalité de Boole)

➔ Avec k essais, $P_{\text{succès}} \leqslant k2^{-n}$ Si k est faible devant 2^n, alors cette majoration est précise.

1.1 Taille des clefs

Comment obtenir un résultat plus précis ?

➔ regarder la probabilité que ManyTrials ne *trouve pas* la solution.

- Avec quelle probabilité chaque essai échoue ?

➔ $1 - 2^{-n}$ Chacun des k essai est indépendant des k autres, donc la probabilité que les k essais ratent est $(1 - 2^{-n})^k$

- Quelle est la probabilité que l'attaque réussisse après k essais ?

➔ $1 - (1 - 2^{-n})^k$.

- Cette formule est valable *quelles que soient* les valeurs de k et n.

1.1 Taille des clefs

1.10 Qu'est-ce qui est le plus probable : « ManyTrial casse une clef de 128 bits avec un million d'essais » ou « Je joue 4 fois à l'euro-millions et je gagne les 4 fois »

- D'après l'énoncé de 1.8: 139 838 160 combinaisons possibles pour l'Euromillions.

➔ La probabilité de gagner 4 fois de suite est de $(139\,838\,160)^4 \approx 3.85 \cdot 10^{32}$

- Quelle est la probabilité de 'deviner' une clef de 128 bits en un essai ?

➔ 1 chance sur 2^{128}

- Qu'est ce qui est plus grand entre 10^{32} et 2^{128} ?

➔ On passe au logarithme:

$\ln 10^{32} = 32 \times \ln 10 \approx 32 \times 2.3 \approx 73.6$

$\ln 2^{128} = 128 \times \ln 2 \approx 128 \times 0.7 \approx 89.6$

2^{128} est environ $e^{89.6-73.6} \approx 8\,886\,110$ fois plus grand que 10^{32}

=> environ dix millions d'essais pour dépasser la prob de 4 gains consécutifs

Exercice ‘plus dur’

1. Modes opératoires et propriétés de sécurité

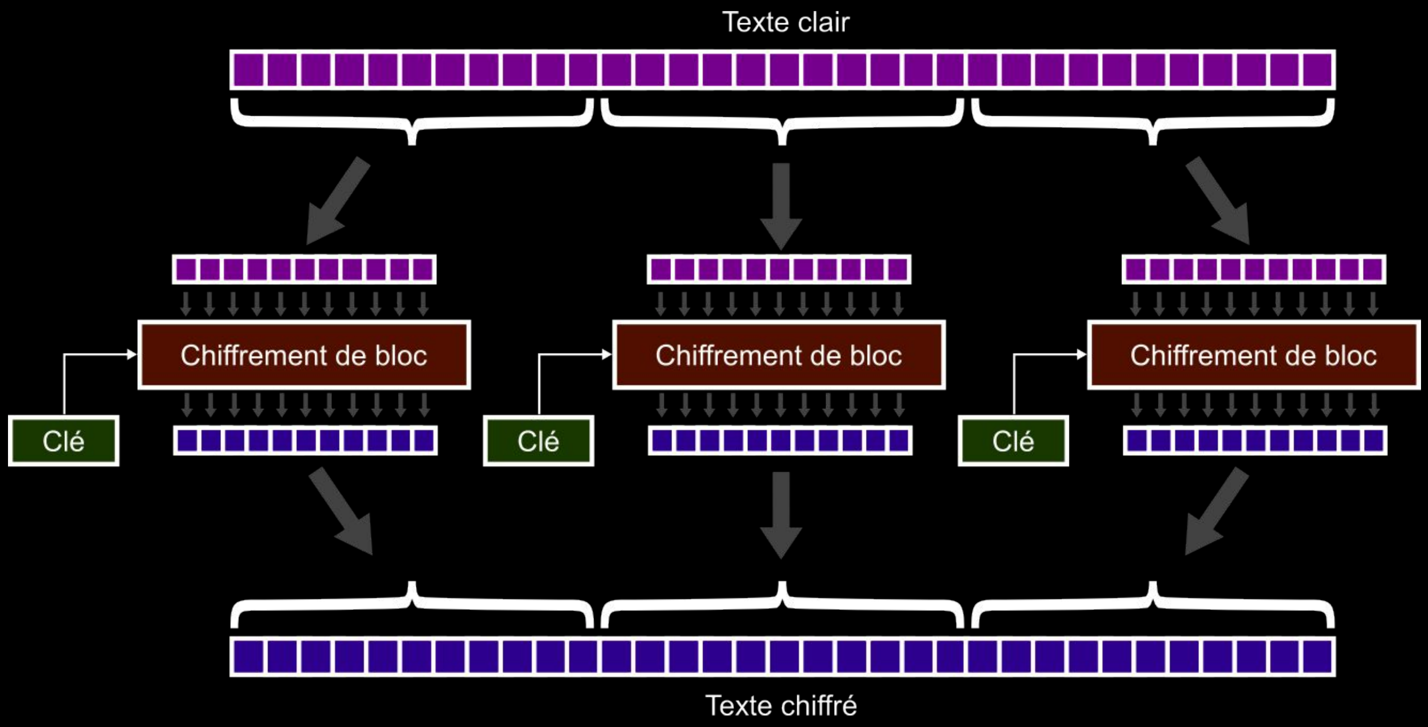
- Considérons un système de chiffrement par bloc E qui chiffre des blocs de n bits (i.e. $M = \{0, 1\}^n$).

a. Montrer que le mode opératoire ECB n’assure pas la sécurité sémantique.

- Qu’est ce que la sécurité sémantique ?
  - Soient deux messages M_0 et M_1 choisis par l’adversaire.
  - pour $b \in \{0, 1\}$, soit le chiffré C du message M_b
    - *La sécurité sémantique est assurée si l’adversaire ne peut pas obtenir la valeur du bit b avec une probabilité significativement meilleure que ½.*
  - Sans sécurité sémantique: fuite d’information, mais ne signifie pas qu’un attaquant peut déchiffrer tous les messages interceptés!

Exercice ‘plus dur’

- Qu’est ce que le mode opératoire ECB (Electronic Codebook) ?



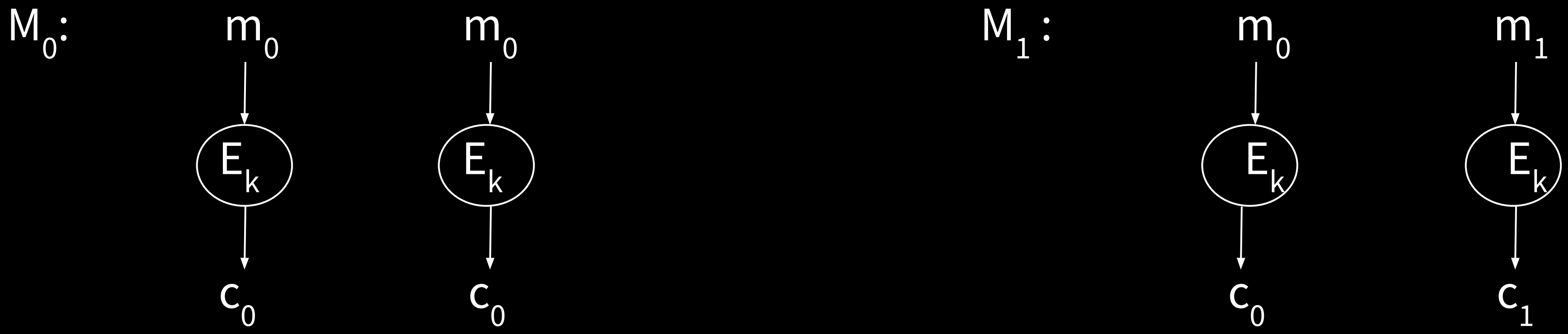
Exercice ‘plus dur’

a. Montrer que le mode opératoire ECB n’assure pas la sécurité sémantique.

Soit $M_0 = m_0 m_0$ (le même message clair M_0 se répète)

$M_1 = m_0 m_1$ (M_1 ne se répète pas)

Faisons passer ces deux messages dans ECB:



$C = c_0 c_0$ si C est le chiffré de M_0, il se répète.

$C = c_0 c_1$ ici, C ne se répète pas.

Exercice ‘plus dur’

a. Supposons que E est utilisé en mode compteur CTR. Montrer que si le nombre de blocs de suite chiffrante est suffisamment grand, alors il est facile de distinguer la suite chiffrante d’une suite aléatoire. Donner la longueur de la suite chiffrante pour que le distingueur ait un avantage supérieur à 1/2 si le chiffrement par bloc E opère sur des blocs de 64 bits (comme le DES).

- Quel est l’un des moyens de s’assurer qu’un chiffrement est efficace ? A quoi doit-il être semblable ?

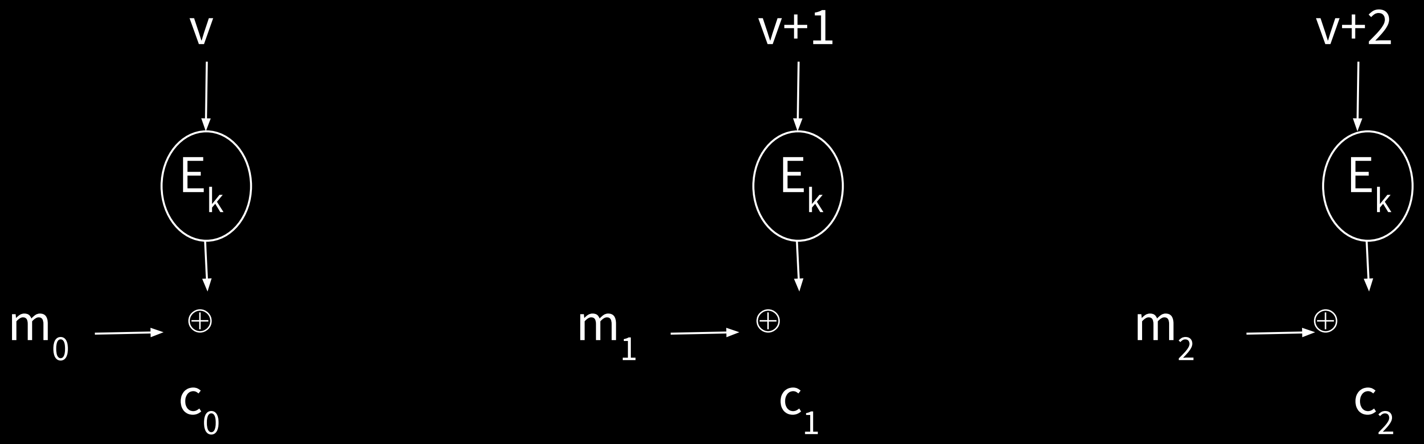
➔ Une suite aléatoire est totalement indéchiffrable => Plus le chiffrement ressemble à une suite aléatoire, plus il sera dur à attaquer.

Exercice ‘plus dur’

- Étudions cette question sur un système de chiffrement par bloc en mode compteur (CTR)

➔ Soit v un vecteur d’initialisation (aussi appelé Nonce) concaténé avec un compteur qui augmente avant chaque utilisation du bloc de chiffrement:

v est secret:



Exercice ‘plus dur’

- Soient S_A une suite aléatoire et S_{CTR} un chiffrement par CTR. Peut-on les différencier ? On s’aide du paradoxe des anniversaires:
  - Combien doit-on réunir de personnes dans une pièce pour avoir plus d’une chance sur deux que deux personnes soient nées le même jour ? ≈ 30 !
  - Démonstration: quelle probabilités d’avoir des anniversaires *différents*?
    - pour 2 personnes: $\overline{p_2} = \frac{364}{365} = 1 - \frac{1}{365}$
    - pour 3 personnes: $\overline{p_3} = (1 - \frac{1}{365})(1 - \frac{2}{365})$
    - pour k personnes: $\overline{p_k} = (1 - \frac{1}{365})(1 - \frac{2}{365})\cdots(1 - \frac{k-1}{365})$
  - D’où: $p_k = 1 - \overline{p_k}$

Exercice ‘plus dur’

- Généralisation: $P_k = 1 - \frac{365!}{(365-k)! \times 365^k}$

- Pour E un ensemble fini: (avec #E = cardinal de E = nombre d’éléments de E)

$$P_n = 1 - \frac{\#E!}{(\#E-n)! \times \#E^n}$$

- Alors, à partir du moment ou n est plus grand que: $\sqrt{2ln(2)\#E}$

  on a plus d’une chance sur deux de tirer deux fois le même élément.

Exercice ‘plus dur’

- Notre suite S_A contient-elle des répétitions ?
  - #E = ensemble des blocs de n bits = 2^n
  - Pour que S_A contienne une répétition avec un proba > ½ il faut donc qu’elle contienne plus de: $(2ln(2)2^n)^{\frac{1}{2}} \simeq 2^{\frac{n}{2}}$

- La suite S_{CTR} contient-elle des répétitions ?
  - E_k est une fonction de chiffrement: c’est une permutation. On a donc: $E_k(a) = E_k(b) \Leftrightarrow a = b$
  - Tant que S_{CTR} contient moins que 2^n blocs il n’y a aucune répétition.

Exercice ‘plus dur’

- Donc pour différencier S_A et S_{CTR}, il suffit de voir s’il y a ou non des répétitions lorsque les suites contiennent au moins $2^{n/2}$ blocs mais moins de 2^n blocs.

- Conclusion: On peut distinguer un chiffrement en mode compteur d’une suite aléatoire à condition que les messages à chiffrer soient très longs. C’est une condition forte !