Introduction à la sécurité
TD3

Ninon Devis: ninon.devis@ircam.fr

Master 2021

Exercice 1 - Chiffrement avec blanchiment

Nous considérons un chiffrement par blocs E qui utilise des clefs de k bits pour chiffrer des blocs de n bits et une variante (dite avec blanchiment) qui utilise une clef de k + n bits de la forme $K = (K_1, K_2) \in \{0, 1\}^k \times \{0, 1\}^n$ et qui chiffre un bloc m de n bits sous la forme $c = E_{K1}(m) \oplus K_2$. Montrer qu'il existe une attaque à deux clairs connus contre cette variante de E qui demande 2^{k+1} évaluations de la fonction E. (i.e. que cette variante ne ralentit la recherche exhaustive que d'un facteur 2).

→ On s'intéresse à la complexité des recherches exhaustives.

- Pour un chiffrement par bloc E avec des clefs de k bits pour chiffrer des blocs de n bits: Complexité de la Brute force = 2^k
- Pour augmenter cette complexité, on applique un blanchiment: $c = E_{K_1}(m)$

$\longrightarrow c = E_{K_1}(m) \oplus K_2$ Blanchiment

Exercice 1 - Chiffrement avec blanchiment

- Avec blanchiment, quel est l'espace de clefs ? k_1 , K_2

$k_1 \in \{0,1\}^k \quad K_2 \in \{0,1\}^n \quad K_3 \in \{0,1\}^{n+k}$

→ La complexité est donc de:

2^{k+n}

- On suppose que l'on accès à un couple de clairs/chiffrés suivants: (m_1, c_1) et (m_2, c_2) avec:
  - $c_1 = \mathcal{E}_{k_1}(m_1) \oplus K_2$
  - $c_2 = \mathcal{E}_{k_1}(m_2) \oplus K_2$
- Comment faire diminuer la complexité du calcul ?

$c_1 \oplus c_2 = \mathcal{E}_{k_1}(m_1) \oplus K_2 \oplus \mathcal{E}_{k_1}(m_2) \oplus K_2 = \mathcal{E}_{k_1}(m_1) \oplus \mathcal{E}_{k_1}(m_2)$

→ Le XOR fait disparaitre le blanchiment.

Exercice 1 - Chiffrement avec blanchiment

- Nous avons donc obtenu: $c_1 \oplus c_2 = e_{K1}(m_1) \oplus e_{K1}(m_2)$

➔ Il nous faut calculer toutes les combinaisons de K_1 sur

2 messages

➔ La complexité est donc:

$$2 \times 2^k = 2^{k+1}$$

Point Culture

- Le système de chiffrement par bloc DES est résistant à la plupart des attaques connues, néanmoins sa faiblesse réside dans la longueur de ses clefs...de 56 bits.
- La recherche exhaustive est donc possible, c'est pourquoi des alternatives ont été proposées pour augmenter la complexité de la force brute:
    - le blanchiment...qui n'est finalement pas si efficace: $c = DES_{K1}(m) \oplus K_2$
    - le pré-blanchiment $c = DES_{K1}(m \oplus K_2)$ qui n'est pas plus sûre
    - En 1984, Rivest propose la variante: $c = DES_{K1}(m \oplus K_2) \oplus K_3$

- L'une des idées pour accroître la sécurité serait de *surchiffrer* le clair avec un chiffrement double (et donc deux clefs aléatoires et indépendantes):
    - On obtient le chiffré par: $c = e_{K2}(e_{K1}(m))$
    - Et donc le clair par: $m = D_{K1}(D_{K2}(c))$

Exercice 2 - Double Chiffrement

1. Montrer que le double chiffrement n'apporte pas toujours un gain de sécurité par rapport au chiffrement simple.

- Pouvez-vous citer un exemple simple de chiffrement pour lequel le chiffrement double revient à un chiffrement simple ?

➔ Chif. de César : décaler de 3 lettres puis de 5 lettres = décaler de 8 lettres.

- Deuxième exemple: le xor avec un masque.

$\mathcal{E}_{k_1} : m \to m \oplus k_1$

$\mathcal{E}_{k_2} : m \to m \oplus k_2$

$$\mathcal{E}_{k_1} \circ \mathcal{E}_{k_2} = \mathcal{E}_{k_1}(\mathcal{E}_{k_2}(m)) = m \oplus \underbrace{k_1 \oplus k_2}_{k_3}$$

complexité inchangée !

➔ Complexité inchangée!

Exercice 2 - Double Chiffrement

2. Exprimer la taille de l'espace des clefs du chiffrement double en fonction de la taille k des clefs du système de chiffrement sous-jacent. Donner l'accroissement de la complexité d'une recherche exhaustive de la clef.

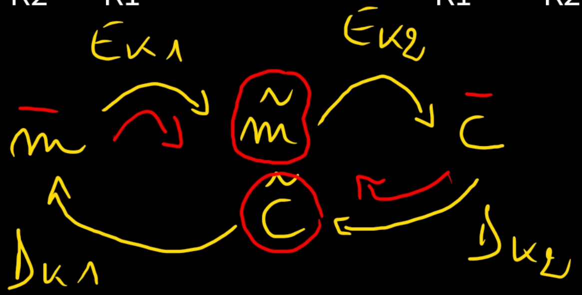
- Soient K_1 et K_2 les deux clefs. Avec: $K_1, K_2 \in \{0,1\}^k$
- Dans le cas ou le chiffrement double n'est pas équivalent à un chiffrement simple, la recherche exhaustive s'élève théoriquement à :

$$2^{2k}$$

Exercice 2 - Double Chiffrement

3. Montrer que le double chiffrement est vulnérable à une attaque à clairs connus si l'attaquant dispose de quelques couples clair/chiffré et calcule $E_k(m)$ et $D_k(c)$ pour toutes les clefs k en mémorisant les résultats obtenus une table.

- Soit un couple clair/chiffré (m, c) obtenu par chiffrement double. comment obtenir les clefs utilisées grâce à une table de hachage ?
    - On a $c = E_{K2}(E_{K1}(m))$ et $m = D_{K1}(D_{K2}(c))$ que l'on peut schématiser par:



- Comment procéder le plus efficacement ?

Exercice 2 - Double Chiffrement

- Calculer le chiffré $\tilde{c} = \mathcal{E}_{k_1}(m)$ pour toutes les clefs possibles.
    - Stocker les valeurs de $(\tilde{c}, k_1)$ obtenues dans une table de hachage. $\times$
- Calculer le déchiffré $\tilde{m} = \mathcal{D}_{k_2}(m)$ pour toutes les clefs possibles.
    - Chercher si $\tilde{m}$ apparaît comme un $\tilde{c}$ dans la table de hachage!
- Pour chaque $\tilde{c} = \tilde{m}$ retourner alors les (k_1, k_2) associées !

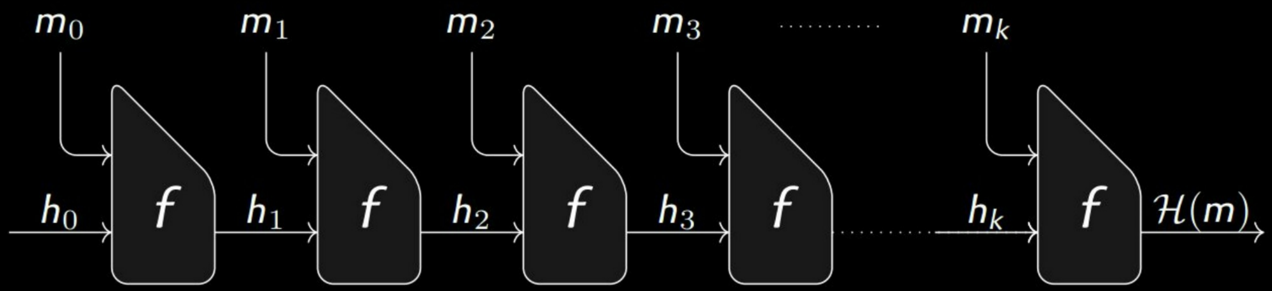
Analyser la complexité en temps et en mémoire de cette attaque.

- Combien de chiffrements calculé pour c̃ ? 2^k
- Combien pour m̃ ? 2^k
- Complexité en temps: combien d'évaluation de l'algorithme ? $2 \times 2^k = 2^{k+1}$
- Complexité en espace: combien d'espace pour la table de hachage ? $k \times 2^k$

Exercice 3 - Multicollisions pour les fonctions de hachage itérées

Nous considérons une fonction de hachage $H : \{0, 1\}^r \to \{0, 1\}^n$ construite à partir d'une fonction de compression $f : \{0, 1\}^n \times \{0, 1\}^l \to \{0, 1\}^n$ par la méthode de Merkle-Damgard (avec $l > 2n$).

Rappels:



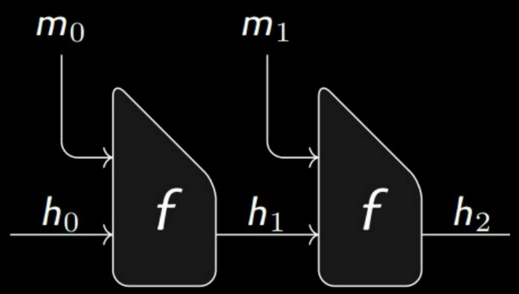
Construction de Merkle-Damgard: itération d'une fonction de compression en découpant le message en blocs et en appliquant cette fonction successivement à chaque bloc concaténé au résultat de la compression du bloc précédent.

Exercice 3 - Multicollisions pour les fonctions de hachage itérées

Soit $H_c : \{0, 1\}^{c \cdot l} \rightarrow \{0, 1\}^n$ une fonction construite à partir de f par la méthode de Merkle-Damgard mais sans ajouter de bourrage et utilisée uniquement pour les messages de longueur fixe égale à un multiple de l.

1. En cherchant deux collisions bien choisis pour la fonction de compression, montrer comment obtenir une 4-multicollision pour H_2.

- Représentons H_2:
  - On traite d'abord $f(h_0, m_0) = h_1$
  - puis ensuite $f(h_1, m_1) = h_2$



- En choisissant astucieusement une collision par étape on peut exhiber une 4-multicollision

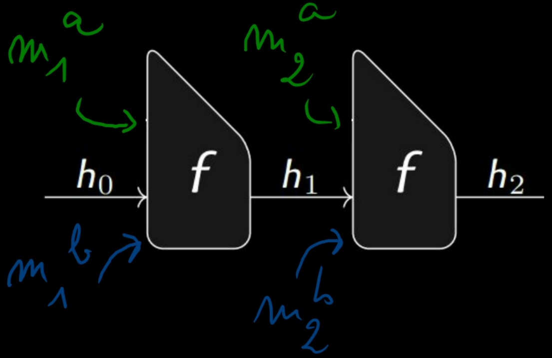
Exercice 3 - Multicollisions pour les fonctions de hachage itérées

- Pour la première étape, on choisit deux blocs de clairs différents:
  - La première collision est donc de la forme:

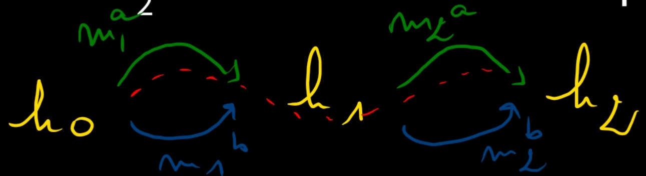
$$f(h_0, m_1^a) = f(h_0, m_1^b) = h_1$$

- De la même manière pour la seconde étape:
  - on choisit m_2^a, m_2^b
  - la collision est donc de la forme:

$$f(h_1, m_2^a) = f(h_2, m_2^b) = h_2$$



- On a donc pour H_2 une 4-multicollision puisque:



$$H_2(m_1^b, m_2^b) = H_2(m_1^a, m_2^a) = H_2(m_1^a, m_2^b) = H_2(m_1^b, m_2^a)$$

- Il est possible “d’emprunter tous les chemins” pour arriver à h_2:

Exercice 3 - Multicollisions pour les fonctions de hachage itérées

2. Expliquer comment transformer cette 4-multicollision pour H_2 en une 4-multicollision pour H. $M = m_1 m_2 \widehat{m_3} \rightarrow M^* = m_1 m_2 m_3 000 \rightarrow M^f = m_1 m_2 m_3 00 \| 3$

- Les messages de la multicollision sont de la même longueurs, il suffit donc d'ajouter le padding et un bloc contenant la longueur du message pour obtenir directement une multicollision pour H.

3. Généraliser en montrant qu'on peut obtenir une 2^t -multicollision pour H pour le coût de t collisions sur f.

$$\begin{tikzcd} h_0 \arrow[r, bend left, "m_1^a"] \arrow[r, bend right, "m_1^b"'] & h_1 \arrow[r, bend left, "m_2^a"] \arrow[r, bend right, "m_2^b"'] & h_2 \arrow[r, bend left, "m_3^a"] \arrow[r, bend right, "m_3^b"'] & h_3 & \cdots \arrow[r, bend left, "m_t^a"] \arrow[r, bend right, "m_t^b"'] & h_t \end{tikzcd}$$

On recherche une collision de la forme: $f(h_0, m_1^a) = f(h_0, m_1^b) = h_1$

puis: $f(h_{i-1}, m_i^a) = f(h_{i-1}, m_i^b) = h_i$ $\quad i \in \{2, \dots, t\}$.

avec: $m_i^a \neq m_i^b$

Exercice 3 - Multicollisions pour les fonctions de hachage itérées

3. Généraliser en montrant qu'on peut obtenir une 2^t -multicollision pour H pour le coût de t collisions sur f.

- Combien de ~~couples~~ obtient-on au total ? 2^t

  t couples !

- Si f se comporte comme une fonction aléatoire, quel est le coût pour obtenir une collision ?

  $2^{t/2}$

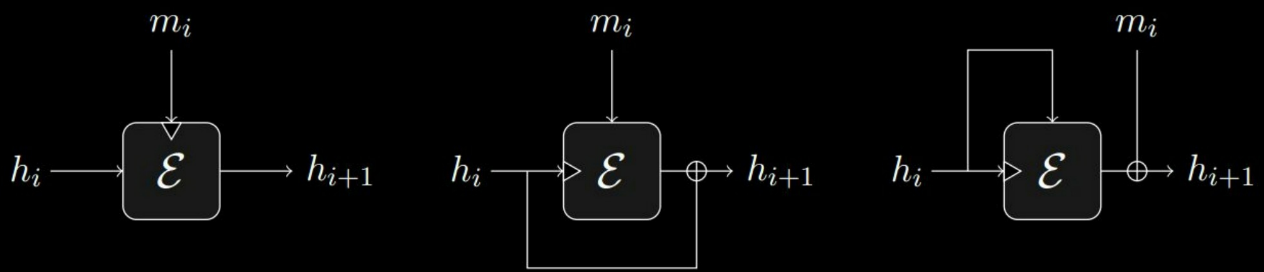
- Quel est finalement le coût pour obtenir une 2^t-multicollision ?

  $t \times 2^{t/2}$

Exercice 4 - Chiffrement par bloc et fonction de compression

Soit $E : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ un système de chiffrement par blocs qui utilise des clefs de n bits pour chiffrer des messages de n bits. Montrer que les trois fonctions de compression f_1, f_2 et f_3 ne sont pas résistantes à la pré-image.

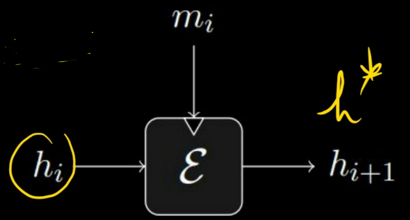
1. $f_1 : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}^n$, $f_1(h, m) = E_m(h)$
2. $f_2 : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}^n$, $f_2(h, m) = E_h(m) \oplus h$
3. $f_3 : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}^n$, $f_3(h, m) = E_h(h) \oplus m$



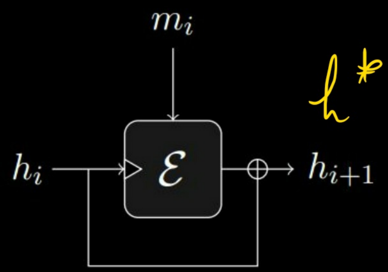
Rappel: Résistance à la pré-image: étant donnée une empreinte *h*, il doit être calculatoirement difficile de trouver un message *m* tel que: *H(m) = h*

Exercice 4 - Chiffrement par bloc et fonction de compression

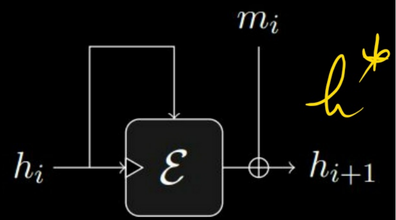
1. $f_1 : \{0, 1\}^n \times \{0, 1\}^n \to \{0, 1\}^n$, $f_1(h, m) = E_m(h) = h^*$ $\quad H(m) = h^*$
2. $f_2 : \{0, 1\}^n \times \{0, 1\}^n \to \{0, 1\}^n$, $f_2(h, m) = E_h(m) \oplus h = h^*$
3. $f_3 : \{0, 1\}^n \times \{0, 1\}^n \to \{0, 1\}^n$, $f_3(h, m) = E_h(h) \oplus m = h^*$



1. $f_1(h, m) = E_m(h)$: étant donné h* $\{0, 1\}^n$, pour tout message m choisit, il est possible de calculer: $h = E_m^{-1}(h^*)$
   alors: m $\quad f_1(h, m) = E_m(E_m^{-1}(h^*)) = h^*$



2. $h^* = f_2(h, m) = E_h(\cancel{h}) \oplus h$: étant donné h* $\{0, 1\}^n$, pour tout h choisit, il est possible de calculer: $m = E_h^{-1}(h^* \oplus h)$
   alors: $f_2(h, m) = E_h(\underbrace{E_h^{-1}(h^* \oplus h)}) \oplus h = h^* \oplus h \oplus h = h^*$



3. $h^* = f_3(h, m) = E_h(h) \oplus m$: étant donné h* $\{0, 1\}^n$, pour tout h choisit, il est possible de calculer: $m = E_h(h) \oplus h^*$
   alors: $f_3(h, m) = E_h(h) \oplus E_H(h) \oplus h^* = h^*$