Introduction à la sécurité
TD4

Ninon Devis: ninon.devis@ircam.fr

Master 2021

Retour sur les exercices précédents

$$\begin{tikzcd}
m \arrow[rr, bend left, "E_{K1}"] & & \tilde{m} \arrow[rr, bend left, "E_{K2}"] & & c \arrow[dl, bend left, "d_{K2}"] \\
& & & \tilde{c} \arrow[ulll, bend left, "d_{K1}"] &
\end{tikzcd}$$

- On a $c = E_{K2}(E_{K1}(m))$ et $m = D_{K1}(D_{K2}(c))$.
- Stocker les valeurs de $(K_1, \tilde{c})$ obtenues dans une table de hachage. Chercher si $\tilde{m}$ apparaît comme un $\tilde{c}$ dans la table de hachage: retourner les K_1 et K_2 . $\in \{0,1\}^{\ell}$

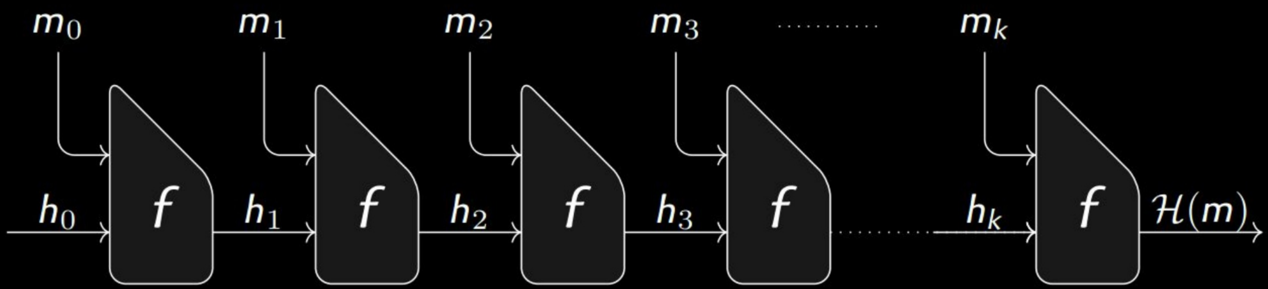
Analyser la complexité en temps et en mémoire de cette attaque.

- Combien de chiffrements calculé pour $\tilde{c}$ et $\tilde{m}$? 2^{ℓ}
- Complexité en temps: combien d'évaluation de l'algorithme ? $2 \times 2^{\ell} = 2^{\ell+1}$
- Complexité en espace: combien d'espace pour la table de hachage ?
    - On stocke 2^{ℓ} possibilités de la clef K_2 de longueur ℓ
    - On stocke donc 2^{ℓ} messages correspondants $\tilde{c}$ de longueur m
    - La table aura donc une complexité en espace de: $O(2^{\ell} \times \ell + 2^{\ell} \times m)$

$\rightarrow O(\ell \times 2^{\ell})$

Exercice 1 - Construction de Merkle Damgard

Rappels:



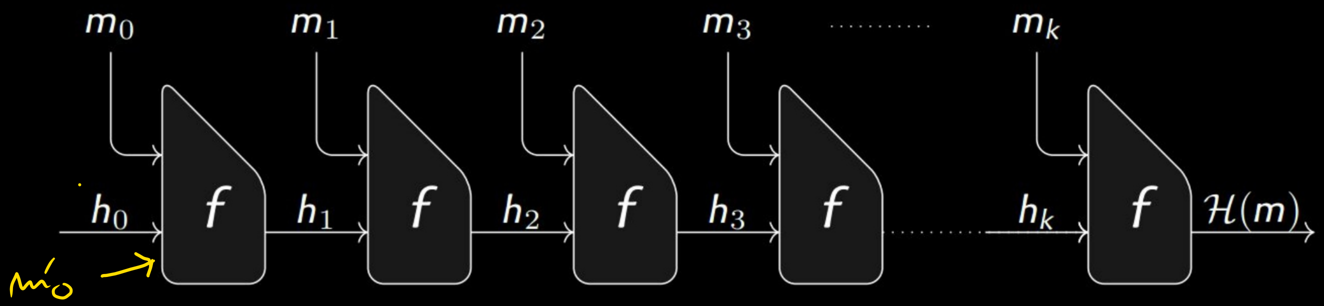
Le padding R est défini sur $\{0, 1\}^*$ et vérifie $|R(m)| \equiv 0 \bmod l$, $\forall\, m \in \{0, 1\}^*$

➔ transforme le message à hacher en un message dont la longueur est un multiple de l.

La valeur de H construite à partir de f et R, est définie par $H(m) = f(h_k, m_k)$ où:

- la valeur R(m) en (k + 1) blocs de l bits $R(m) = (m_0, \ldots, m_k) \in (\{0, 1\}^l)^k$
- $h_i = f(h_{i-1}, m_{i-1})$ pour tout $i \in \{1, \ldots, k\}$

Exercice 1 - Construction de Merkle Damgard



1. Supposons que les messages dont la longueur n'est pas un multiple de la longueur du bloc l sont complétés par une chaîne de zéros jusqu'à ce que la longueur soit un multiple de l (i.e. en posant i = |m| mod l, $R(m) = m \| 0^{l-i}$). Montrer que la fonction itérée obtenue à partir de f et R n'est pas résistante aux collisions.

- Il existe un exemple simple pour lequel deux messages différents donnent le même haché pour ce type de padding: $m = 1$ $\quad m' = 10$

→ $\tilde{m} = 10\ldots0$ non résistant aux collisions.

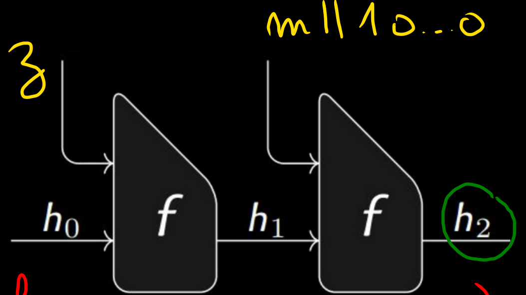
Exercice 1 - Construction de Merkle Damgard

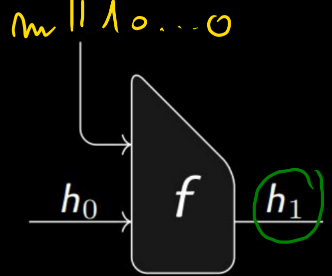
2. Le processus de bourrage est défini par: $R(m) = m||10^{l-i-1}$ avec $i = |m| \bmod l$. Montrer que si l'on dispose d'un bloc de message z tel que $f(h_0, z) = h_0$ alors il est possible de trouver des collisions pour la fonction itérée obtenue à partir de f et R.

- Soient $M_1 = z \,||\, m$ et $M_2 = m$. On fait passer ces deux messages dans R puis H:

$R(M_1) = z||m||10\ldots0$

$R(M_2) = m||10\ldots0$





or $h_1 = h_0$ puisque $f(h_0, z) = h_0$

On a bien $h_2 = h_1$ les deux empreintes sont égales pour des entrées différentes.

Exercice 1 - Construction de Merkle Damgard

$H = f(f(f \ldots))$

3. Supposons enfin qu'un dernier bloc contenant la longueur binaire du message est concaténé au procédé de bourrage de la question précédente (i.e. en notant τ_m un encodage binaire de la longueur $|m|$ de m, nous avons $R(m) = (m||10^{l-i-1}||\tau_m)$ avec $i = |m| \bmod l$). Montrer que la fonction itérée obtenue à partir de f et R est résistante aux collisions si f est résistante aux collisions.

$\left[a \neq b \Rightarrow f(a) \neq f(b) \right]$
$\left[f(a) = f(b) \Rightarrow a = b \right.$

- On cherche à prouver que si f est résistante aux collisions alors H est résistante aux collisions. ∀ m et m' tels que m ≠ m', $f(h, m) \neq f(h', m')$ ssi $H(m) \neq H(m')$ $\Rightarrow$
- Équivalent à : $H(m) = H(m')$ ssi $f(h, m) = f(h', m')$
- Considérons deux messages m et m' tels que $H(m) = H(m')$.

  On va traiter le cas ou m et m' sont de même longueurs, puis le cas où m et m' sont de longueurs différentes.

Exercice 1 - Construction de Merkle Damgard

- On cherche à prouver que si f est résistante aux collisions alors H est résistante aux collisions. H(m) = H(m’) => f(h, m) = f(h’, m’)
- Considérons deux messages m et m’ tels que H(m) = H(m’).
    - Si m et m’ ne sont pas de même longueur binaire : $\tau_m \neq \tau_{m'}$
        - On a, avec k et k’ le nombre de blocs de l bits de R(m) et R(m’):

$$f(h_k, \tau_m) = H(m) = H(m') = f(h_{k'}, \tau_{m'})$$

➔ On obtient bien une collision explicite soit: $f(h_k, \tau_m) = f(h_{k'}, \tau_{m'})$

- Si m et m’ de même longueur binaire: $\tau_m = \tau_{m'}$ et $k = k'$ // $f(h_k, \tau_m) = f(h_{k'}, \tau_{m'})$

Pour pouvoir prouver que la collision de H provient de la collision de f et non des égalités précédentes, il faut “remonter” jusqu’à avoir une collision liée à f.

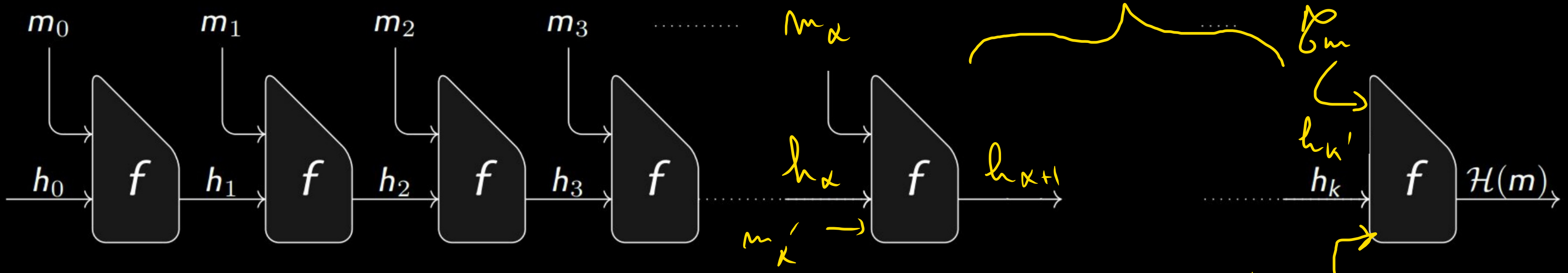
Exercice 1 - Construction de Merkle Damgard

- Comme $\tau_m = \tau_{m'}$, on cherche l'endroit où les deux messages sont ≠ .

En notant α le plus petit entier tel que $m_\alpha \neq m'_\alpha$

- Soit on a: $f(h_\alpha, m_\alpha) = f(h'_\alpha, m'_\alpha)$ Alors on a une collision explicite pour f.
- Soit pas, dans ce cas on cherche β le plus petit entier dans {α, …, k} tel que:

$h_{\beta+1} = h'_{\beta+1}$ (β existe car $h_{k+1} = H_m = H(m') = h'_{k+1}$)



On a alors: $h_\beta \neq h'_\beta$ et $h_{\beta+1} = f(h_\beta, m_\beta) = f(h'_\beta, m_{\beta'}) = h'_{\beta+1}$

On a bien une collision explicite pour la fonction de compression f.

Exercices Cryptographie Asymétrique

2.2. Quel niveau de sécurité (= taille de clef secrète équivalente) offrent les clefs de 2048 bits recommandées aujourd'hui ? $N = a \times b$

- On cherche à calculer la complexité de factorisation:
  - Selon la NFS (Number Field Sieve): $O\left(\exp\left((1{,}92 + o(1))(\ln N)^{1/3}(\ln\ln(N))^{2/3}\right)\right)$
  - On peut grossièrement calculer la complexité avec la formule donnée à la question 1: $T(N) = \exp\left(\alpha\left(N \ln^2 N\right)^{1/3}\right)$ où $\alpha = 1{,}70$.
  - Pour une clef de taille b: Brute force $\rightarrow$ temps $= 2^b$
    A l'inverse : si on a le temps $2^b = T$, pour retrouver b : $\log_2(T)$
  - Pour trouver la taille (en bits) équivalente: on applique donc $\log_2$
  - Ici N = 2048 d'où: $\log_2 T(2048) = (\log_2 e) \times \alpha\left(2048 \ln^2 2048\right)^{1/3}$

$\log_a x^y = \log_a(x) \times y$

$= 120{,}6 \rightarrow$ taille de clef équivalente

Exercices Cryptographie Asymétrique

2.3. Quelle taille de clef RSA faut-il choisir pour s'assurer un niveau de sécurité équivalent à des clefs secrètes de 128 bits ?

- En faisant le raisonnement inverse:
  - on part de la taille de clef équivalente pour revenir à la taille des clefs RSA on tombe sur un problème de type: $\log_2(T(N)) = a$

$2^a = e^{\alpha (N \ln^2 N)^{1/3}}$ $\ln(2^a) = \alpha (N \ln^2 N)^{1/3} \rightarrow A = N \ln^2 N$

- On évalue plutôt le résultat avec la méthode précédente.
- Pour une clef de 2358 on trouve un niveau de sécurité de 128,01 bits.
- On doit donc choisir une clef de 4096

Exercices Cryptographie Asymétrique

2.4. Le meilleur code publiquement disponible de factorisation, CADO-NFS, a besoin de 90 jours pour factoriser un nombre de 512 bits sur un coeur à 2Ghz. On peut estimer qu'un serveur qui contient 36 coeurs comparables coûte 4000 euros, et consomme 400W. Quel budget est nécessaire pour casser RSA-1024 en 3 mois ?

- On calcule le ratio des complexité pour 512 et 1024:

$\alpha = 1,70$

$$T(1024)/T(512) = \exp\left(\alpha\sqrt[3]{1024 \ln^2 1024} - \alpha\sqrt[3]{512 \ln^2 512}\right)$$
$$= 10^7$$

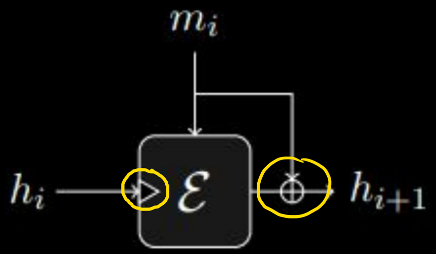
- Il faut donc autant de coeurs pour tenir la deadline de 3 mois ce qui correspond à: $10^7/36 = 277\ 778$ serveurs.
- Soit un budget d'environ: 1 milliard d'€

Exercices Cryptographie Asymétrique

2.5. Quelle puissance électrique est nécessaire ? (un serveur qui contient 36 coeurs comparables coûte 4000 euros, et consomme 400W)

$277\,778 \times 400 = 100\,MW$ tout pile \o/

Exercice 3 : Sécurité de la construction de Matyas-Meyer-Oseas avec le DES



Montrer que la fonction de compression f n'est pas résistante aux collisions lorsque E = DES dans la construction de Matyas-Meyer-Oseas.

La fonction de compression $f : \{0, 1\}^n \times \{0, 1\}^l \rightarrow \{0, 1\}^n$ est définie par $f(h, m) = E_h(m) \oplus m$ où E est un système de chiffrement par blocs de n bits.

Le DES possède une propriété forte: la complémentation. Ainsi:

$\forall h \in \{0,1\}^{56}$ et $\forall m \in \{0,1\}^{64}$

$$\boxed{DES_{\bar h}(\bar m) = \overline{DES_h(m)}}$$

$$f(\bar h, \bar m) = \mathcal{E}_{\bar h}(\bar m) \oplus \bar m = \overline{\mathcal{E}_h(m)} \oplus \bar m$$

$$= \mathcal{E}_h(m) \oplus m = f(h, m)$$

$$\bar A \oplus \bar B = A \oplus B$$

f n'est donc pas résistante aux collisions !

Exercice 4 : Attaque en collision contre fonctions de hachage concaténées

Soient H_1 et H_2 deux fonctions de hachage de même domaine qui produisent des empreintes de n bits et considérons la fonction de hachage H définie pour tout message m par $H(m) = H_1(m)||H_2(m)$ (H produit des empreintes de 2n bits)

1. Montrer que H est résistante aux collisions dès que l'une des fonctions H_1 ou H_2 est résistante aux collisions.

On suppose que l'on a deux messages m_1 et m_2 avec $m_1 \neq m_2$ tels que $H(m_1) = H(m_2)$

Alors: $H(m_1) = \underline{H_1(m_1)} \,||\, \underline{H_2(m_1)} = H(m_2) = \underline{H_1(m_2)} \,||\, \underline{H_2(m_2)}$

Une collision de H fournit donc une collision pour H_1 et une pour H_2. Si l'une des deux fonctions est résistante aux collisions alors la fonction H concaténée est résistante aussi.

Exercice 4 : Attaque en collision contre fonctions de hachage concaténées

$H(m_1^0, m_2^0) = H(m_1^1, m_2^1) = H(m_1^0, m_2^1) = H(m_1^1, m_2^0)$

$\rightarrow$ 4 – multicollisions –

2. En supposant que H_1 est une fonction de hachage itérée vulnérable à l'attaque des multi collisions de l'exercice précédent, proposer un algorithme pour construire une collision pour la fonction H en environ $2^{n/2}$ (n/2) évaluations de la fonction de hachage H_1 et $2^{n/2}$ évaluations de la H_2.

- D'après l'exercice du TD3 (multi collisions pour les fonctions de hachages itérées): Le coût total pour obtenir une 2^t-multicollision est de l'ordre de:

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

- Selon le même principe, on peut construire une $\rightarrow 2^{n/2}$ – multicollision pour la fonction H_1 -

➔ A combien d'évaluations de H_1 cela correspond ?

$2^{n/2} \times \frac{n}{2}$ évaluations –

Exercice 4 : Attaque en collision contre fonctions de hachage concaténées

- Selon le même principe, on construit une $2^{n/2}$-collision pour la fonction H_1
- A combien d'évaluations de H_1 cela correspond ? $2^{n/2}$ $(n/2)$

➔ Combien de couples sont obtenus ? $n/2$

➔ A combien de messages en collision cela correspond ? $2^{n/2}$

➔ Il suffit alors d'évaluer la fonction H_2 en le nombre de messages obtenus:

Par le paradoxe des anniversaires :

➔ En répétant cette attaque on obtient une collision pour H_2

$\Rightarrow$ On obtient donc une collision pour H