Cryptanalyse

Cours 10 - Cryptanalyse et Fonctions de Hachage

Maxime Bombar

Mardi 12 Novembre

Organisation

Pensez à rendre vos challenges ! Deadline Mardi 17 Décembre 08h00.

Examen final de cryptanalyse : Mardi 17 Décembre 14h30

Est-ce que vous souhaitez une autre séance de révision sans machine ?

Introduction

Fonction de Hachage Cryptographique

$H : \{0,1\}^\star \to \{0,1\}^n$, *e.g.*, $n = 256$.

- **Objectif :** Produire une empreinte d'une source (*e.g.*, un fichier).
- **Sécurité :** L'empreinte ne fournit pas d'information sur la source.

Si la longueur de l'entrée est **fixée**, H est simplement une **fonction de compression**.

Sécurité 1 : Résistance à la Pré-Image

On se **fixe** $H : \{0,1\}^\star \to \{0,1\}^n$

Résistance à la pré-image (aPre)

Pour $\mathbf{h} \leftarrow \{0,1\}^n$, il doit être difficile de retrouver $\mathbf{M}$ tel que $\mathbf{h} = H(\mathbf{M})$ avec bonne probabilité (sur le choix de $\mathbf{h}$).

Authentification par mot de passe

On peut stocker (User, h) et vérifier que $H(\mathrm{password}) = h$.

Si la base de données fuite, un attaquant n'a pas accès aux mots de passe.

Sécurité 2 : Résistance à la Seconde Pré-Image

On se **fixe** $H : \{0,1\}^\star \to \{0,1\}^n$

Résistance à la seconde pré-image (aSec)

Soit $m > n$. Pour $\mathbf{M}_1 \leftarrow \{0,1\}^m$ il doit être difficile de retrouver $\mathbf{M}_2 \neq \mathbf{M}_1$ tel que $H(\mathbf{M}_2) = H(\mathbf{M}_1)$ (avec bonne probabilité sur le choix de $\mathbf{M}_1$).

- Signature de type Hash-and-Sign : On signe $H(\mathrm{message})$.
- Intégrité de fichiers.

Impossible de forger une signature.

Sécurité 3 : Résistance aux Collisions

On se **fixe** $H : \{0,1\}^\star \to \{0,1\}^n$

Résistance aux collisions (Informel)

Il doit être difficile de trouver deux messages $\mathbf{M}_1 \neq \mathbf{M}_2$ avec $H(\mathbf{M}_1) = H(\mathbf{M}_2)$.

Collisions à Prefixes Choisis

Une collision ne produit en général pas de messages ayant du sens.

Préfixes-Choisis

On se fixe $\mathbf{P}_1, \mathbf{P}_2 \in \{0,1\}^m \times \{0,1\}^\ell$ et on cherche une collision de la forme

$$H(\mathbf{P}_1||\mathbf{M}_1) = H(\mathbf{P}_2||\mathbf{M}_2).$$

Attaques dans la vie réelle

- Forge de certificats, ou clés GPG/SSH malveillants.
- Virus Flame (2012) avec le même hash MD5 qu'une mise à jour Windows [a].

a. `https://msrc.microsoft.com/blog/2012/06/flame-malware-collision-attack-explained/`

Quelques Exemples Célèbres

- **MD5** :
  - Hash de 128 bits, inventée en 1992 par Rivest (RFC 1321).
  - Collisions trouvées **à la main** (Wang, Yu, 2005).
  - Forge de certificats (Stevens *et. al.*, 2009)
- **SHA-0** :
  - Hash de 160 bits, publiée en 1993 par la NSA.
  - Collisions théoriques par Joux et Chabaud (1998).
- **SHA-1** :
  - Hash de 160 bits, extension de SHA-0.
  - Collisions théoriques en 2005 (Wang *et. al.*).
  - Premières collisions en 2017 (Stevens *et. al.*, 2009)
  - Collisions à prefixes choisis (Leurent, Peyrin, 2020).
  - Encore très utilisée de nos jours...

Les Standards Actuels

- **SHA-2**
  - Publiée par la NSA en 2001.
  - Famille de fonctions de hachage de 224, 256, 384, 512 bits.
- **SHA-3**
  - Issue d'une compétition du NIST : KECCAK.
  - Nouveau design (Construction éponges).
  - Famille de fonctions de hachage de 224, 256, 384, 512 bits, publiée en 2015.

Fun Facts

Collisions MD5 sur PS3 (2007)

Marc Stevens, Arjen Lenstra, Benne de Weger.

Exemple récent de collisions

Collision MD5 de textes ASCII (Marc Stevens, Mars 2024)

MD5(TEXTCOLLBYfGiJUETHQ4h**A**cKSMd5zYpgqf1YRDhkmxHkhPWptrkoyz28wnI9V0aHeAuaKnak)
=
MD5(TEXTCOLLBYfGiJUETHQ4h**E**cKSMd5zYpgqf1YRDhkmxHkhPWptrkoyz28wnI9V0aHeAuaKnak)

Sur l'Existence de Collisions

Principe des tiroirs

L'espace des message est beaucoup plus grand que l'espace des hash :

$$\{(M, M') \mid M \neq M' \text{ et } H(M) = H(M')\}$$

existe, est bien défini et est non vide.

Algorithme non uniforme

Pas de clé $\Rightarrow$ Il existe un algorithme qui renvoie en **temps constant*** une collision pour **n'importe quelle** fonction de hachage.

En pratique, on ne sait évidemment pas construire cet algorithme.

Attaque des Anniversaires

Algorithme Générique Recherche de Collisions

Données : $m \in \mathbb{N}$ avec $m > n$
Résultat : $x \neq x'$ tels que $H(x) = H(x')$
$\Delta \leftarrow \emptyset$
tant que Vrai faire
  $x \leftarrow \{0,1\}^m$;
  $y \leftarrow H(x)$;
  si Il existe $x' \neq x$ **tel que** $(x, y) \in \Delta$
   alors
    retourner (x, x')
  fin
  $\Delta \leftarrow \Delta \cup \{(x, y)\}$.
fin

On se fixe $H : \{0,1\}^\star \to \{0,1\}^n$.

- Quelle structure de donnée pour Δ ?
- L'algorithme termine-t-il ?
- Analyse ?

Analyse de l'algorithme

Recherche par force brute nécessiterait 2^n essais pour énumérer tous les y_i, ou $n \cdot 2^n$ en gardant l'approche probabiliste (collectionneur de coupons).

« Paradoxe » des Anniversaires

Le nombre moyen d'étapes de l'algorithme est $\approx \sqrt{\pi/2} \cdot 2^{n/2} \approx 1.25 \cdot 2^{n/2}$.

Modélisation

On se fixe $H : \{0,1\}^\star \to \{0,1\}^n$.

On se donne $y_1 = H(x_1), \ldots, y_\ell = H(x_\ell), \ldots$ Quelle est la valeur moyenne de ℓ avant de trouver une collision ?

Le problème est mal défini, dépend de H en général...

Modélisation

On se fixe $H : \{0,1\}^\star \to \{0,1\}^n$.

On se donne $y_1 = H(x_1), \ldots, y_\ell = H(x_\ell), \ldots$ Quelle est la valeur moyenne de ℓ avant de trouver une collision ?

Le problème est mal défini, dépend de H en général...

Heuristique de travail

Les y_i sont indépendants et identiquement distribués dans $\{0,1\}^n$.

Analyse

Soit X le nombre de y_i pour qu'une collision soit trouvée. On a

$$\mathbb{E}(X) = \sum_{\ell \geqslant 0} \mathbb{P}(X > \ell).$$

$$\begin{aligned}
\mathbb{E}(X) &= \sum_{k=1}^{\infty} k \cdot \mathbb{P}(X = k) = \sum_{k=1}^{\infty} \sum_{\ell=0}^{k-1} \mathbb{P}(X = k) = \sum_{k=1}^{\infty} \sum_{\ell=0}^{\infty} \mathbb{P}(X = k) \mathbb{1}_{\ell < k} \\
&= \sum_{\ell=0}^{\infty} \sum_{k=1}^{\infty} \mathbb{P}(X = k) \mathbb{1}_{k > \ell} \quad \text{(Tous les termes sont positifs.)} \\
&= \sum_{\ell=0}^{\infty} \sum_{k>\ell}^{\infty} \mathbb{P}(X = k) = \sum_{\ell=0}^{\infty} \mathbb{P}(X > \ell).
\end{aligned}$$

Analyse (Suite)

On veut estimer $\mathbb{P}(X > \ell)$ pour $\ell \geqslant 0$. Soit $N \stackrel{\text{def}}{=} 2^n$.

$$\begin{aligned}
\mathbb{P}(X > \ell) &= \frac{|\{\ell - \text{uplets de } \{0,1\}^n \text{ sans collision}\}|}{N^\ell} \\
&= \frac{N \times (N-1) \times \cdots \times (N-\ell+1)}{N^\ell} \\
&= \prod_{k=1}^{\ell-1} \left(1 - \frac{k}{N}\right) \approx \prod_{k=1}^{\ell-1} e^{-k/N} \\
&= e^{-\frac{\ell(\ell-1)}{2N}}
\end{aligned}$$

Analyse (Fin)

Le nombre moyen de tests à effectuer avant d'obtenir une collision est

$$\mathbb{E}(X) \approx \sqrt{\pi/2} \cdot 2^{n/2}.$$

Preuve :

$$\mathbb{E}(X) = \sum_{\ell>0} \mathbb{P}(X > \ell) \approx \sum_{\ell>0} e^{-\ell^2 \cdot 2^{-n-1}} \approx \int_0^{\infty} e^{-x^2 \cdot 2^{-n-1}} dx = \sqrt{\pi/2} \cdot 2^{n/2}.$$

En résumé

Une attaque sur les collisions d'une fonction de hachage est réussie si elle nécessite moins de $2^{n/2}$ évaluations de la fonction de hachage.

Remarque : Compromis Temps-Mémoire

L'algorithme générique présenté ici nécessite de stocker $2^{n/2}$ couples $(x, H(x))$ de $m + n > 2n$ bits. En réalité, on peut faire beaucoup mieux, tout en gardant un nombre d'évaluation de H de l'ordre de $2^{n/2}$.

Pour les exploiter en pratique, on veut obtenir des collisions sur des messages précis (par exemple préfixes choisis). Nécessite des méthodes avancées (*e.g.*, **Cryptanalyse différentielle** poussée).

Quelques ordres de Grandeurs

Le réseau bitcoin calcule 2^{60} hash SHA-2 par seconde, ou encore 2^{84} par an.

Une longueur de hash de 160 bits (comme SHA-1) est beaucoup trop court pour résister aux collisions, même avec des méthodes peu sophistiquées.

https ://www.blockchain.com/explorer/charts/hash-rate

La Construction de Merkle-Damgård

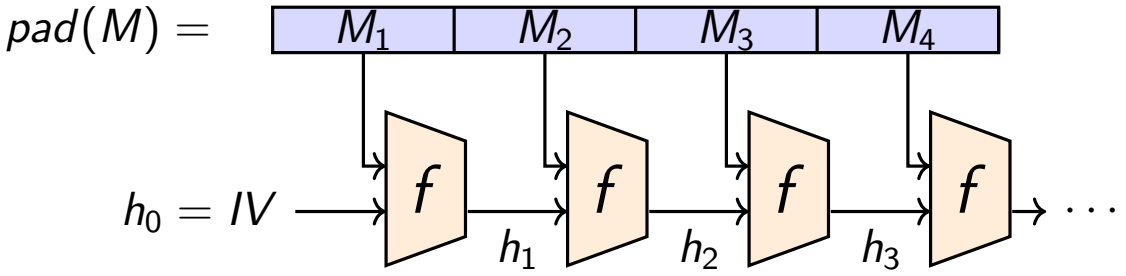
Construire des Fonctions de Hachages Cryptographiques ?

Une fonction de hachage à entrée de taille fixée $m > n$ est une fonction de compression.

$$f : \{0,1\}^n \times \{0,1\}^\ell \to \{0,1\}^n.$$

Idée : Chaîner les compressions ?

Merkle-Damgård



- Message $M = (M_1||M_2||\cdots||M_{k-1}||M_k)$ découpé en $k-1$ blocks de taille ℓ, où un padding est éventuellement appliqué sur le bloc $k-1$.
- Le bloc M_k encode le nombre de bits de M, écrit sur n bits.
- $h_1 = f(IV, M_1)$ $\quad h_{i+1} = f(h_i, M_{i+1})$ $\quad$ Hash final est $f(h_{\ell-1}, M_\ell)$.

Collisions et Prefixes Identiques

Soit $H : \{0,1\}^* \to \{0,1\}^n$ de type Merkle-Damgård avec une fonction de compression f.

Exercice

On suppose que l'on sait trouver des collisions pour n'importe quelle IV. Montrer qu'il est alors possible de trouver deux messages $\mathbf{M}$ et $\mathbf{M}'$ à préfixe identique, *i.e.*, de la forme $\mathbf{M} = (\mathbf{P}||\mathbf{N})$ et $\mathbf{M}' = (\mathbf{P}||\mathbf{N}')$, tels que $H(\mathbf{M}) = H(\mathbf{M}')$.

Length-Extension Attack

Soit $H : \{0,1\}^* \to \{0,1\}^n$ de type Merkle-Damgård avec une fonction de compression f.

Exercice

Soit $h = H(\mathbf{M})$ pour un certain message $\mathbf{M}$. Montrer qu'il est possible de calculer $H(\mathbf{M}||\mathbf{S})$ pour **n'importe quel suffixe** $\mathbf{S}$, même si $\mathbf{M}$ est inconnu.

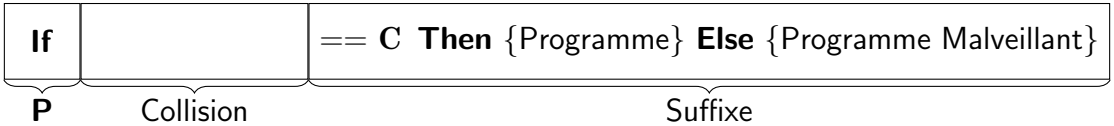
Application : Construction de Programmes Malveillants

Soit $H : \{0,1\}^* \to \{0,1\}^n$ de type Merkle-Damgård.

Collisions à contexte choisi

Pour n'importe quel préfixe $\mathbf{P}$ et suffixe $\mathbf{S}$, on cherche à calculer deux « blocs de collisions » $\mathbf{C} \neq \mathbf{C}'$ tels que $H(\mathbf{P}||\mathbf{C}||\mathbf{S}) = H(\mathbf{P}||\mathbf{C}'||\mathbf{S})$.

- On choisit **If** comme préfixe et on calcule une collision $H(\mathbf{P}||\mathbf{C}) = H(\mathbf{P}||\mathbf{C}')$ avec $\mathbf{C}, \mathbf{C}'$ incluant déjà le padding.
- On étend avec le même suffixe qui décrit les deux programmes



Compression et Résistance aux Collisions

Soit $H : \{0,1\}^* \to \{0,1\}^n$ de type Merkle-Damgård avec une fonction de compression f.

Exercice

Montrer que de toute collision pour H, on peut en déduire une collision pour f.

Compression et Chiffrement par Blocs

Idée : Construire une fonction de compression résistante aux collisions à l'aide d'un chiffrement par blocs

$$\mathrm{Enc} : \{0,1\}^n \times \{0,1\}^\kappa \to \{0,1\}^n$$

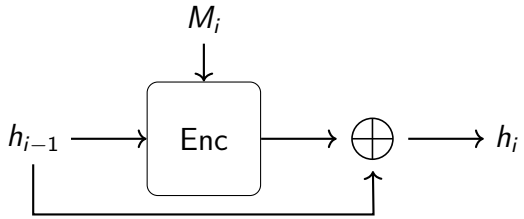
avec clés de κ bits et blocs de n bits.

Black, Rogaway, Shrimpton, 2002

Il existe 12 façons de construire une fonction de compression résistante aux collisions à partir d'un chiffrement par blocs :
`https://iacr.org/archive/crypto2002/24420321/24420321.pdf`

Transformée de Davies-Meyer

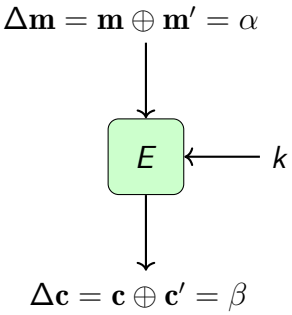
La construction la plus courante (Exemple MD5, SHA-1, SHA-2).



$$f(h_{i-1}, M_i) = Enc_{M_i}(h_{i-1}) \oplus h_{i-1}.$$

Cryptanalyse Avancée

Rappel : Cryptanalyse Différentielle



Objectif : Étudier comment une différence α entre deux messages se propage au travers du cryptosystème.

Question : Que dire d'une différentielle $\alpha \to 0$ avec $\alpha \neq 0$ dans le cas d'un chiffrement ?

Cas des Fonction de Hachage

Différences et Collisions

Une paire de messages $\mathbf{M}, \mathbf{M}' \stackrel{\text{def}}{=} \mathbf{M} + \Delta\mathbf{M}$ de différence ($\Delta\mathbf{M} \neq 0 \rightarrow 0$) fournit exactement une collision pour la fonction de hachage.

- Les fonctions de hachage sont en général **plus complexes** que les chiffrements par blocs habituels.
- Exemple : SHA-1 correspond à une généralisation à 5 branches d'un schéma de Feistel, itéré 80 fois.

Near-Collision Attacks

Les collisions sur une construction MD sont en général obtenues à partir de « presque-collisions » sur la fonction de compression

$$f(h, m) \stackrel{\text{def}}{=} E_m(h) \oplus h.$$

Presque-Collisions

Pour une différence choisie d et deux valeurs de chaînage h_{i-1} et h'_{i-1}, on cherche deux messages m_i et m'_i tels que

$$\Delta E_m(h_{i-1}) \stackrel{\text{def}}{=} E_{m'_i}(h'_{i-1}) \oplus E_{m_i}(h_{i-1}) = d.$$

i.e.,

$$\Delta h_i \stackrel{\text{def}}{=} f(h'_{i-1}, m'_i) \oplus f(h_{i-1}, m_i) = \Delta h_{i-1} \oplus d.$$

Application : Collision à Deux Blocs

- Étant donné un préfixe $P = (m_1||\ldots||m_p)$, on calcule h_p la valeur de chaînage après les p blocs.

Application : Collision à Deux Blocs

- Étant donné un préfixe $P = (m_1||\dots||m_p)$, on calcule h_p la valeur de chaînage après les p blocs.
- Via une attaque à « presque-collisions » avec chaînages (h_p, h_p)pour un certain d on cherche $m_{p+1}, m'_{p+1}, h_{p+1}, h'_{p+1}$ tels que

$$\Delta h_{p+1} = f(h_p, m'_{p+1}) \oplus f(h_p, m_{p+1}) = \Delta h_p \oplus d = d.$$

Application : Collision à Deux Blocs

- Étant donné un préfixe $P = (m_1 || \dots || m_p)$, on calcule h_p la valeur de chaînage après les p blocs.
- Via une attaque à « presque-collisions » avec chaînages (h_p, h_p)pour un certain d on cherche $m_{p+1}, m'_{p+1}, h_{p+1}, h'_{p+1}$ tels que

$$\Delta h_{p+1} = f(h_p, m'_{p+1}) \oplus f(h_p, m_{p+1}) = \Delta h_p \oplus d = d.$$

- On réexécute une « presque-collision » sur (h_{p+1}, h'_{p+1}) à différence d : on trouve m_{p+2}, m'_{p+2} et h_{p+2}, h'_{p+2} tels que

$$\Delta h_{p+2} = \Delta h_{p+1} \oplus d = 0.$$

Application : Collision à Deux Blocs

- Étant donné un préfixe $P = (m_1||\dots||m_p)$, on calcule h_p la valeur de chaînage après les p blocs.
- Via une attaque à « presque-collisions » avec chaînages (h_p, h_p)pour un certain d on cherche $m_{p+1}, m'_{p+1}, h_{p+1}, h'_{p+1}$ tels que

$$\Delta h_{p+1} = f(h_p, m'_{p+1}) \oplus f(h_p, m_{p+1}) = \Delta h_p \oplus d = d.$$

- On réexécute une « presque-collision » sur (h_{p+1}, h'_{p+1}) à différence d : on trouve m_{p+2}, m'_{p+2} et h_{p+2}, h'_{p+2} tels que

$$\Delta h_{p+2} = \Delta h_{p+1} \oplus d = 0.$$

- Alors $(P||m_{p+1}||m_{p+2})$ et $(P||m'_{p+1}||m'_{p+2})$ est une collision.

Cryptanalyse Différentielle de MD5

Quelques Références

- Première Collision : Wang et Yu (2005) `https://iacr.org/archive/eurocrypt2005/34940019/34940019.pdf`.
- Clarification de l'attaque : `https://eprint.iacr.org/2004/264.pdf`

Deux types de Différentielles

Pour deux mots de 32 bits X et $X^\star$:

- $\delta X \stackrel{\text{def}}{=} X^\star - X \mod 2^{32}$
- $\Delta X \stackrel{\text{def}}{=} X^\star \oplus X$

Outils Automatiques

Projet HashClash `https://github.com/cr-marcstevens/hashclash`.

Attaque Pratique : Blast Radius (Août 2024)

- Attaque Man-In-The-Middle sur le protocole radius.
- Calcule en **temps réel** des collisions MD5 à **prefixes choisis** via de la **cryptanalyse différentielle**.
- Autorise **n'importe qui** à se connecter sur un réseau de type Eduroam.

Il faut authentifier les requêtes ! À l'avenir, Radius over TLS.

Plus d'information : `https://www.blastradius.fail/`