TP4 - Fonctions de hachage

01 Octobre 2021

L'objectif de ces exercices est de vous familiariser avec les fonctions de hachage cryptographiques, et notamment avec leurs faiblesses.

> Vous êtes libre de choisir le langage de programmation de votre choix.

Exercice 1 : Recherche de collision avec l'algorithme de Floyd

L'algorithme de Floyd pour la recherche de cycle permet de trouver des collisions pour les fonctions de hachage si celle-ci engendre une suite récurrente (le même algorithme est utilisé pour la factorisation et le calcul du logarithme discret en utilisant la méthode ρ de Pollard).

1. Implémentez l'algorithme de détection de cycle de Floyd.
2. À l'aide de cet algorithme, trouvez une collision pour une entrée quelconque de la fonction de hachage `sha256_32` (SHA-256 tronquée à 32 bits pour avoir un attaque en un temps raisonnable) fournie dans le fichier sage.

Exercice 2 : SHA-256

En suivant les instructions données par le document du NIST [1], implémentez la fonction de hachage SHA-256. Vous êtes libre de choisir le langage de programmation que vous souhaitez (C recommandé pour faciliter les opération binaires).

La primitive doit pouvoir être initialisée (`SHA256_init`), l'état interne doit pouvoir être mis à jour (`SHA256_update`) autant de fois que nécessaire, et la finalisation du traitement (`SHA256_finalize`) inclut l'ajout du padding et la dernière mise à jour de l'état interne.

Pensez à gérer les cas particuliers (message vide par exemple).

Des vecteurs de test permettant de vérifier votre implémentation sont disponibles sur le site du NIST [2].

Exercice 3 : *Length-extension attack*

Les fonctions SHA1 et SHA2 (ainsi que MD5 et MD4 d'ailleurs) sont basées sur le schéma de Merkle-Damgård. Ce type de construction est vulnérable aux *length extension attacks* lorsque la sortie du traitement ne reçoit pas de modification finale. En effet, dans ce cas, la sortie correspond à l'état interne final de la fonction de hachage. Ainsi, en initialisant l'état de départ avec le condensat reçu, il est possible de "continuer" le hachage du message, et d'ajouter des données à la fin.

Soient K une clé sécrète de 128 bits, et M le message "Ce message a été écrit par le le responsable de TP.". Le condensat suivant a été généré de la manière suivante :

$$\texttt{SHA256}(K \mid\mid M) = \texttt{0xbc460fc2585f850bf417faba1718e27a887fb16d40ba6ce231d6f2198b95b949}$$

1. `https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf`
2. `https://csrc.nist.gov/CSRC/media/Projects/Cryptographic-Standards-and-Guidelines/documents/examples/SHA256.pdf`

Implémentez l'attaque pour générer le condensat de $K||M'$, avec $M' = "M \ || \text{ some data}"$. Le résultat attendu est la paire $(M',\ H(K \ || \ M'))$.
N'oubliez pas de prendre en compte le padding.