

INFO002 : cryptologie
TPD 0 : quelques exemples

Pierre Hyvernat
Laboratoire de mathématiques de l'université de Savoie
bâtiment Chablais, bureau 17, poste : 94 22
email : Pierre.Hyvernat@univ-smb.fr
www : <http://www.lama.univ-smb.fr/~hyvernat/>

Le plus simple pour répondre à ces questions est d'utiliser l'interface en ligne de commande de OpenSSL. (`$ man openssl` et `$ man openssl-<CMD>` pour le manuel...)

Toutes les questions (sauf la question 2.3) peuvent être résolues avec une combinaison de la commande `openssl`, des redirections “|” et les commandes suivantes :

- `xxd -p` pour transformer du binaire en représentation hexadécimale,
- `xxd -p -r` pour transformer une représentation hexadécimale en binaire,
- `printf` pour afficher des chaînes (Attention si vous utilisez la commande `echo` car elle ajoute un caractère “saut de ligne” !)

Exercice 1 : Chiffrement symétrique

Question 1. Chiffrez le message (ASCII)

`encipher`

avec la clé (hexadécimale)

01 23 45 67 89 ab cd ef

avec DES (sans remplissage). Le résultat sera donné en hexadécimal.

Question 2. Chiffrez le même message avec la clé

00 22 44 66 88 aa cc ee

puis la clé

01 23 45 67 89 ab cd ef

Que constatez-vous ? Expliquez.

Question 3. Déchiffrez le message (hexadécimal)

23 54 c0 e4 ed 15 f4 84 33 47 2d fe 61 06 f4 bb

avec la clé (hexadécimale)

f0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01

avec AES-128 (sans remplissage).

Question 4. Le message suivant (hexadécimal)

ee c3 ab 94 83 0c 2a e4 9f d0 25 31 b3 f5 11 b9
47 6c 16 9d a6 e9 d7 dd 71 62 59 4f fb 09 aa c4
9e 05 22 f4 6b 27 73 be 32 44 a7 d8 ed b8 3f cc

a été chiffré avec AES-256 en mode CBC avec la clé

f0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0f

L'IV utilisé se trouve en tête du message.

Déchiffrez le message.

Exercice 2 : fonctions de hachage

Question 1. L'empreinte MD5 de la chaîne (ASCII)

Vive les fonctions de hachage...

est d9d19cb4f238e26ee4eba628f609620b.

Vérifiez cette empreinte et calculez les empreintes SHA-1, SHA-256, SHA-512 de cette même chaîne.

Question 2.

- Comparez les empreintes de deux fichiers ayant un unique bit de différence au milieu. Que constatez-vous ?
- Comparez les empreintes du même fichiers avec deux extensions différentes (.jpg et .jpeg). Que constatez vous ?

Question 3. On considère la fonction SHA-256- n : les n premiers bits de la fonction SHA-256.

Cherchez une préimage pour les empreintes suivantes (hexadécimal) :

- de (SHA-256-8)
- dead (SHA-256-16)
- deadbe (SHA-256-24)
- deadbeef (SHA-256-32)

Estimez le temps de calcul d'une préimage pour SHA-256-64 et SHA-256-128.

Remarque : pour cette question, vous devrez écrire un programme pour faire la recherche. Un langage compilé est de loin préférable !

Pour information : mon code (en C, compilé avec l'option -O3 et la bibliothèque openssl pour calculer SHA-256) trouve une préimage pour dead en moins de 1 seconde après une centaine de milliers de calculs d'empreintes aléatoires.

Exercice 3 : MAC

Question 1. Calculez le MAC du message (ASCII)

Et quid des fonctions de tronçonnage ?

avec HMAC-sha256 et la clé (ASCII)

ZYXWVU

Comment un destinataire peut-il vérifier le MAC ?

Exercice 4 : Clé publique

Question 1. Générez une clé RSA 2048 bits. Quelle est l'exposant publique ?

Question 2. Générez la clé publique correspondante.

Question 3. Visualisez le contenu de la clé privée et de ses composants.

Question 4. Visualisez le contenu de la clé publique et vérifiez qu'elle ne contient pas l'exposants privé ou les nombres premiers utilisés.

Question 5. Modifiez légèrement le contenu de votre clé privée. Vérifiez la validité de la nouvelle clé.

Question 6. Chiffrez (avec une clé publique) et déchiffrez (avec une clé privée) un message de votre choix.

Question 7. Signez (avec votre clé privée) l'empreinte d'un document de votre choix, et vérifiez la signature du fichier disponible sur ma page web, avec la clé publique correspondante...