Groupe de TD : DIA 6

Membres : Sara TIEBERGUE, Henri SERANO et Eloi SEIDLITZ

Rendu de 5 pages, 2 pages pour répondre aux exercices et 3 pour les annexes.

Projet Applied Cryptography

Exercice 1

Partie 1

Pour cet exercice, nous avons choisi si la chaine de caractère : "Serano\_Henri\_\_Sara\_Thibierge\_\_Eloi\_Seidlitz\_DIA6".

Le nonce que nous avons trouvé est : 408284928

Nous avons obtenu le hash suivant : 9f12fa3ea9dd4cdb820f093c22f67e89f866701f637d49163f50c88940000000

Obtenir cet hash contenant 7 zéros nous a pris 16 minutes et 50.8 secondes.

Nous avons fait une fonction utilisable dans la partie 2. Il est possible de retrouver le code en annexe 1.

Partie 2

Ici nous avons utilisé la même chaine de caractère que pour la première partie en ajoutant les caractères 97 (a) à 126 (~) de la table ascii. Par exemple la première chaine de caractère est "Serano\_Henri\_\_Sara\_Thibierge\_\_Eloi\_Seidlitz\_DIA6a" et 0.29 secondes ont suffis à trouver un hash avec 5 zéros, cependant il a fallu 1 minute 18.49 secondes pour en trouver un avec 6 zéros.

Nous avons donc répété le processus 30 fois où n zéros est 5 et n+1 zéros est 6.   
Le temps moyen pour trouver n zéros (5) était de 1.25 secondes

Le temps moyen pour trouver n+1 zéros (6) était de 30.85 secondes

Il a donc fallu en moyenne 24.70 fois plus de temps pour trouver 6 zéros que 5.

Il est intéressant de noter qu’il est possible que la première chaine de caractère contenant n zéros à la fin en contienne en réalité n+1. C’était le cas pour la chaine de caractère : "Serano\_Henri\_\_Sara\_Thibierge\_\_Eloi\_Seidlitz\_DIA6e".

Pour plus de détaille allez sur l’annexe 2.

Exercice 2

Pour obtenir nos password et iv, nous les avons générés aléatoirement pour plus de sécurité.

Nous avons donc obtenu notre entier N que vous pourrez retrouver en déchiffrant C.

Vous trouverez notre fichier encrypté sous le nom ES\_HS\_ST.enc que nous avons généré grâce à la commande openssl suivante (‘password’ et ‘iv’ ne vous sont pas donné):

openssl enc -aes-256-ctr -pbkdf2 -in text\_to\_encrypt.txt -out ES\_HS\_ST.enc -pass pass:password -iv iv

Nous avons ensuite généré notre clé privée ‘b’ aléatoirement puis notre clé public B, 3089461795188824779451885, nous permettant d’obtenir notre chiffrement C de N : 5855849996723033070255305.  
  
Nous avons testé nos processus pour vérifier qu’ils fonctionnent.  
Pour plus d’information n’hésitez pas à consulter l’annexe 3.

Exercice 3

Partie 1

Pour cette partie, nous avons fait la démonstration de la signature ElGamal. Nous avons pris soin de ne pas utiliser de librairies ou packages qui faisaient tout à notre place.  
Je vous invite à aller consulter l’annexe 4 pour comprendre comment nous avons procédé. Pour vous aider le code a été documenté.

Partie 2

Pour cette seconde partie, nous avons fait un exemple d’utilisation du RSA dans openssl.

Comme pour la première partie, je vous invite à aller consulter l’annexe 5 pour comprendre comment nous avons procédé. Pour vous aider le code a été documenté.

Annexe 0 (Importations des librairies utilisées)

A black and grey background

Description automatically generated with medium confidence

Annexe 1

A screenshot of a computer program

Description automatically generated

Annexe 2

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

Annexe 3

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Annexe 4

A screenshot of a computer

Description automatically generated

A black rectangular object with a black border

Description automatically generated

Annexe 5

A screenshot of a computer

Description automatically generated

A black background with yellow text

Description automatically generated