CRY - Laboratoire 2

Exploitation

20 novembre 2020

Étudiants
Doran KAYOUMI

Professeur
Alexandre DUC
Assistant
Olivier KOPP

HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD
www.heig-vd.ch

Table des matières

1 CBC-MAC

1.1 Attaque

En regardant la construction de CBC-MAC, j'ai remarqué que j'avais le contrôle sur la valeur qui est transmise au premier bloc de chiffrement. Donc, je peux modifier à ma guise le premier bloc du message et l'IV.

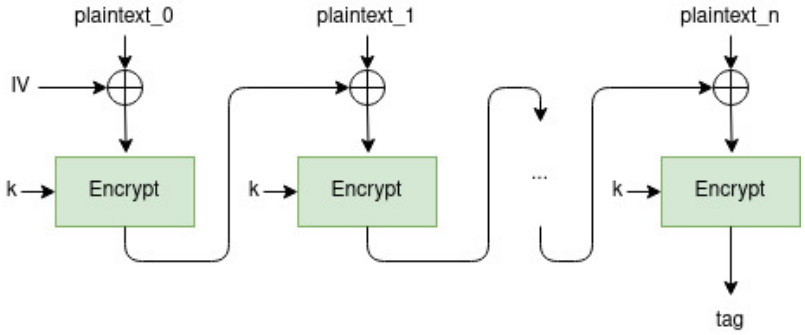

FIGURE 1 – Structure chiffrement CBC

Après avoir modifié le premier bloc du message (e.g. changer un oui par un non), j'ai calculé un nouvel IV pour que la valeur transmise au premier bloc de chiffrement soit identique à celle obtenue avec les valeurs originales. En faisant ça, le tag obtenu sera identique à celui du message original.

Pour trouver ce nouvel IV, j'ai simplement fait un XOR entre la valeur qu'il faut transmettre au premier bloc de chiffrement et le premier bloc de mon message forgé.

$$og\_fblock \oplus og\_iv \oplus forged\_fblock = forged\_iv$$ [1]

Et voilà ! Il suffit d'envoyer ce nouvel IV avec le message forgé.

1.2 Amélioration possible

La faille du système provient de l'utilisation d'un IV aléatoire. Donc, pour améliorer la construction, il suffit d'utiliser un IV **constant**. Par exemple 0.

1. Calcul compliqué pour trouver un nouvel IV

2 Speck

2.1 Récupérer le mot de passe sans l'IV et avec la clé

Le message que la radio transmet suit la structure suivante, **password0000** diffusé en boucle.

FIGURE 2 – Structure du message transmis par la radio

Lors du déchiffrement, chaque bloc peut-être déchiffré indépendamment des autres et que seul le premier bloc doit être déchiffré avec l'`IV`

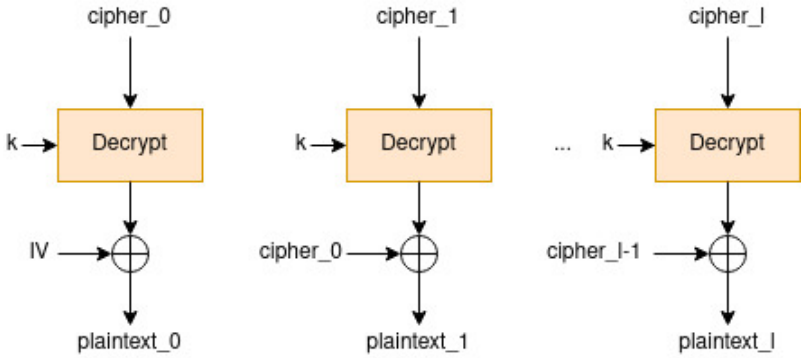

FIGURE 3 – Structure du déchiffrement utilisant CBC

Donc, pour obtenir le mot de passe, il suffit de déchiffrer un bloc (autre que le premier vu que l'on ne possède pas l'IV) qui contient le mot de passe. Par exemple le troisième bloc.

2.2 Attaque

Pour cette attaque, je me suis basé sur le `paradoxe des anniversaires`. L'idée est de simplement trouver une collision sur la valeur qui est donnée à la fonction de chiffrement, c'est-à-dire :

$$plain \oplus previous\_cipher$$

Donc, s'il existe une collision pour deux textes clair différents, on peut poser :

$$plain_4 \oplus cipher_3 = plain_{11} \oplus cipher_{10}.$$

En suivant la structure du message, on peut déterminer que $plain_{11}$ correspond aux zéros, ce qui veut dire qu'on peut simplement l'ignorer. Et donc le mot de passe (i.e. $plain_4$) peut être déterminé en faisant $cipher_3 \oplus cipher_{10}$ (i.e. un xor entre les ciphers précédents les ciphers en collision).

Le mot de passe que la radio a transmis est **UPss**

3 Algorithme de chiffrement par blocs

3.1 Attaque

Pour cette attaque, j'ai pu me baser sur l'`Attaque des Chiffrements Affines`, car je possédais $n+1$ paires de texte clair connu et que le chiffrement est décrit par l'opération $y = Kx+k$.

La première chose que j'ai faite, c'est de créer deux matrices en utilisant les n premières paires de texte connu. La première matrice (Y) ayant pour colonne les textes chiffrés soustrait au texte chiffré de la $n+1$éme paire. Et la deuxième (X) ayant pour colonne les textes claire. [2].

Ensuite, j'ai utilisé ces deux matrices afin de retrouver la première clé K en faisant $K = YX^{-1}$. Une fois la première clé trouvée, je suis reparti de l'opération de chiffrement et j'ai pu retrouver la deuxième clé k en faisant $k = y - Kx$.

Une fois les deux clé dans ma possession, j'ai pu déchiffrer le challenge et j'ai obtenu : **NeverGonnaEspied**.

2. Je n'ai pas eu besoin de soustraire chaque texte clair avec le texte clair de la $n+1$éme paire car car ce dernier est un zéro