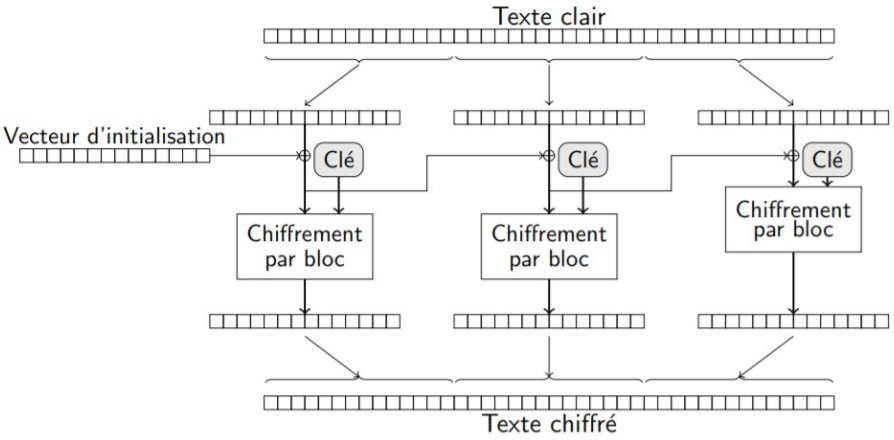
TD n°2

Exercice I. Modes opératoires et propriétés de sécurité

c. Montrer que le mode opératoire CBC n'assure pas la sécurité sémantique pour des messages suffisamment longs.

- Considérer 2 blocs égaux de chiffré (a quelle condition y a-t-il une forte probabilité de blocs égaux?), faire passer ces blocs dans CBC.

*Exercice II. Mode opératoire CBC**



1. Quel est l'avantage de CBC ? Retrouve t-on cette propriété pour CBC* ? Et pour le déchiffrement CBC* ?

2. Montrer que ce mode n'assure pas la sécurité sémantique.

- Sélectionner un message dont le chiffré est reconnaissable après passage dans CBC*

Exercice III. Attaque CBC avec le padding RFC2040

Même en utilisant un mode opératoire pour étendre l'espace des textes clairs, un système de chiffrement par bloc qui opère sur des blocs de n bits ne peut chiffrer que des messages dont la longueur est un multiple de n. Il est donc nécessaire de définir un processus de bourrage ou remplissage (padding, en anglais) qui transforme un message de longueur arbitraire en un message dont la longueur est un multiple de n. Dans la norme RFC2040, le processus de bourrage suivant à été proposé lorsque les messages à chiffrer sont constitués d'octets et que n est un multiple de 8. Étant donné un message M de longueur 8l, nous effectuons la division euclidienne de 8l par n et obtenons $8l = \alpha n + 8i$ de sorte qu'il manque i octets au dernier bloc de message. En notant $m_1 m_2 \ldots m_{b-i}$ les octets du dernier bloc de message (avec b = n/8 la taille d'un bloc en octets), le processus de bourrage consiste à concaténer i fois l'octet 0i à la fin du dernier bloc. Ainsi pour un algorithme de chiffrement qui opère sur des blocs de 128 bits, soit 16 octets, le bloc de clair m1 . . . m12 sera transformé en m1 . . . m12||04040404.

Considérons un attaquant qui a intercepté un chiffré C = (C1, C2, . . . , Cn) produit par un système de chiffrement à blocs en mode CBC avec le processus de bourrage RFC2040. On suppose aussi v connu.

1. Montrer que si l'attaquant dispose d'un oracle qui détermine si le message clair associé à un chiffré arbitraire est bien formé pour l'encodage RFC2040, alors il peut déterminer l'encodage effectivement utilisé pour le chiffré C.

- Que vérifie l'oracle ? Quand retourne t-il une erreur ?
- Si on modifie le chiffré C, comment se propage l'erreur introduite ?
- Comment utiliser cette faille pour déterminer l'encodage du message ?

2. Modifier l'attaque pour qu'il détermine le dernier octet du dernier bloc de clair.

- Exprimer la valeur du message Mi en fonction de Ci, faire de même pour un message modifié. Quelle égalité déduit-on ?

Soit M_i le dernier bloc de clair de la forme: M = m1...m11 || 05 05 05 05 05

- Que cherche t-on à connaître ?
- Pour cela, on modifie le chiffré afin d'obtenir un message avec un padding d'un bit de plus. Quel sera le clair associé ?
- Grâce à l'égalité précédente, trouver le chiffré associé.
- Combien de requêtes doit-on a faire à l'oracle au maximum pour déterminer l'encodage ?

3. En itérant ce processus, montrer que l'attaquant peut obtenir ainsi le message clair en intégralité.