

TD4 — Codes d'authentification de messages

Ex1. CBC-MAC Dans cet exercice on s'intéresse au contrôle d'intégrité par codes CBC-MAC.

- (a) Dessiner le schéma de calcul du code CBC-MAC d'un message $M = M_1 \cdots M_n$ avec une clé K et une fonction de chiffrement F .
- (b) L'utilisation d'un IV fixe est fondamental. Montrer que si l'IV est variable et communiqué avec le message chiffré alors un adversaire peut forger des codes CBC-MAC valides.
- (c) Lors du chiffrement de messages en mode CBC, l'utilisation d'IV non prédictibles est fondamental. Expliquer cette apparente contradiction avec la question précédente.
- (d) On dit de CBC-MAC qu'il n'est pas sûr pour le contrôle d'intégrité des messages de taille variable. Expliquer comment forger un code CBC-MAC valide pour un nouveau message à partir de deux messages accompagnés de leurs codes CBC-MAC.
- (e) Pétronille propose d'utiliser l'algorithme suivant pour transmettre de manière confidentielle, avec contrôle d'intégrité, un message M avec une clé secrète K . Un adversaire peut-il forger des messages recevables par le destinataire?

```
def envoyer(M,K):  
    iv = Choisir un IV aléatoire  
    C = Chiffrer M avec AES-256 en mode CBC avec clé K et IV iv  
    I = Calculer le CBC-MAC AES-256 de M avec clé K  
    send(iv,C,I)  
  
def recevoir(K):  
    R = receive()  
    (iv,C,I) = R  
    M = Déchiffrer C avec AES-256 en mode CBC avec clé K et IV iv  
    J = Calculer le CBC-MAC AES-256 de M avec clé K  
    si I == J alors  
        retourner M  
    sinon  
        lever une exception
```

Ex2. HMAC Dans cet exercice on s'intéresse au contrôle d'intégrité par codes HMAC.

- (a) Écrire la formule de calcul du code HMAC d'un message M avec une clé K et une fonction de hachage H .
- (b) TERENCE propose d'utiliser l'algorithme suivant pour effectuer le contrôle d'intégrité. Qu'en pensez-vous? Quelle différence avec HMAC-MD5?

```
def micmac(M,K):  
    M' = concatener(K,M)  
    retourner MD5(M')
```