Université Marie and Louis Pasteur, ISIFC, Cryptographie. TP sur le chiffrement Symétrique et AES-GCM.

Jean-François Couchot

September, 18th 2026

1 Chiffrement et déchiffrement

Ce premier exercice illustre le mode AES-GCM. Vous devez utiliser le langage Python et une bibliothèque de chiffrement comme `pycryptodome`.

Dans une sous-section de votre notebook, écrire un premier script dans différentes cellules qui successivement

1. Génère une clé de 128 bits.
2. Sauvegarde cette clé dans un fichier `secret.key`.
3. Chiffre une phrase comme "Le mot de passe est 1234" en utilisant l'algorithme `AES` en mode `GCM`.
4. Récupère le `nonce`, l'en-tête, le texte chiffre et le `tag`.
5. Encode ces quatre éléments en base64.
6. Sauvegarde l'ensemble dans un fichier `message.json`.

Dans une seconde sous-section de votre notebook, écrire un second script dans différentes cellules qui successivement

1. Lit le fichier `secret.key` pour en extraire la clé
2. Charge le fichier `message.json` et en extrait les éléments
3. Déchiffre le message et l'affiche.

2 Vérification de l'intégrité d'un message

Le mode GCM permet de vérifier l'intégrité des données.

1. Ouvrir le fichier `message.json` avec un éditeur de texte et modifier une seule lettre dans la valeur du texte chiffre. Sauvegarder alors le fichier modifie sous le nom `message.json`.
2. Exécutez a nouveau le script de déchiffrement de la section précédente
3. Le programme doit générer une erreur. Expliquer pourquoi cette vérification est essentielle pour les communications sur un réseau.

3 Le danger du `nonce` réutilisé

Le mode GCM utilise un compteur interne. Nous allons démontrer qu'il ne faut jamais utiliser un même `nonce` deux fois avec la même clé.

1. Dans une seconde section de votre notebook, recopier tout le code de la première section.
2. Modifier la première sous-section de chiffrement comme suit.
    1. Fixer la valeur du `nonce` manuellement dans le code.
    2. Chiffrer un premier message M_1 et stocker le résultat chiffre C_1.
    3. Chiffrer un deuxième message M_2 plus long que M_1 avec le même nonce et la même clé et stockez le résultat chiffre C_2.
3. Modifier la seconde sous-section de déchiffrement comme suit.
    1. Réaliser l'opération logique `XOR` entre C_1 et C_2.
    2. Réaliser ensuite l'opération logique `XOR` entre M_1 et M_2.
    3. Afficher les deux résultats et comparer ces deux valeurs.
    4. Faire une démonstration que si la clé et le nonce sont identiques alors $C_1 COR C_2 = M_1 COR M_2$.
    5. Faire un COR entre $C_1 COR C_2$ et M_2 qui sont connus. Montrer qu'on retrouve le message original. En faire la démonstration théorique.
    6. Donner une conclusion à cet exercice.