TP 2

Objectif

- Génération et utilisation des Clés secrètes
- Génération et utilisation des Clés publiques
- Chiffrement de données via les commandes de la bibliothèque openSSL
- Signature de données via les commandes de la bibliothèque openSSL

Outils

- Kali Linux
- Bibliothèque openSSL

Exercice 1 : Génération et utilisation des clés secrètes (clés symétriques)

- Génération d'une clé secrète pour AES :
  - **openssl rand -out cle_secrete.key 32**
  - Pourquoi est-il important d'utiliser une clé de 32 octets pour AES-256 ?
- Chiffrement d'un fichier avec AES en utilisant une clé secrète :
  - **openssl enc -aes-256-cbc -pbkdf2 -in fichier.txt -out fichier_chiffre_aes.enc -pass file:cle_secrete.key**
  - Expliquez chaque partie de la commande
  - Quelle est la différence entre utiliser un mot de passe (-k) et une clé secrète (-pass file) ?
- Déchiffrement du fichier :
  - **openssl enc -aes-256-cbc –d -pbkdf2 -in fichier_chiffre_aes.enc -out fichier_dechiffre_aes.txt -pass file:cle_secrete.key**
  - Expliquez chaque partie de la commande
  - Pourquoi est-il nécessaire d'utiliser la même clé secrète pour le déchiffrement ?

Exercice 2 : Génération et utilisation des clés publiques (clés asymétriques)

- Génération de la clé ECC (Elliptic Curve Cryptography) :
  - **openssl ecparam -genkey -name secp256k1 -out cle_privee_ecc.pem**
  - Expliquez chaque partie de la commande
- Création d'une clé publique à partir de la clé privée ECC
  - **openssl ec -in cle_privee_ecc.pem -pubout -out cle_publique_ecc.pem**
  - Expliquez chaque partie de la commande
- Quelle est la différence entre une clé publique ECC et une clé publique RSA ?

Exercice 3 : Signature de données avec ECC

- Signature d'un fichier avec une clé privée ECC :
  - **openssl dgst -sha256 -sign cle_privee_ecc.pem -out signature_ecc.bin fichier.txt**
  - Expliquez chaque partie de la commande
  - Pourquoi est-il préférable de signer un haché plutôt que le fichier original ?
- Vérification de la signature avec une clé publique ECC :
  - **openssl dgst -sha256 -verify cle_publique_ecc.pem -signature signature_ecc.bin fichier.txt**
  - Expliquez chaque partie de la commande
  - Essayer d'effectuer la vérification avec un autre fichier (Ex : file.txt). Notez vos remarques.
  - Que se passe-t-il si la signature ne correspond pas au fichier original ?

Exercice 4 : Chiffrement hybride

- **Chiffrement hybride :**
  - Générer une clé symétrique aléatoire
  - Chiffrer un fichier (**file.txt**) avec la clé symétrique (AES)
  - Chiffrer la clé symétrique avec une clé publique RSA
- **Déchiffrement hybride :**
  - Déchiffrer la clé symétrique avec la clé privée RSA
  - Déchiffrer le fichier avec la clé symétrique déchiffrée
- **Questions**
  - Pourquoi utilise-t-on un chiffrement hybride plutôt qu'un chiffrement asymétrique seul ?
  - Quels sont les avantages de cette approche en termes de performance et de sécurité ?