المدرسة الوطنية للذكاء الاصطناعي والرقمنة - بركان
ÉCOLE NATIONALE DE L'INTELLIGENCE ARTIFICIELLE ET DU DIGITAL - BERKANE

TP – Cryptographie avec OpenSSL

AES – Diffie-Hellman – RSA – ECDSA

Module : Cryptographie

Réalisé par:

- *AIT Mbarek Karim*
- *Belfaida barae*
- *El Hami Yassine*
- *Idchaoudi Youssef*
- *Tahri Mohamed*

Encadré par:

- *Pr. FARTITCHOU Mohamed*

Outils : Linux, OpenSSL

Installation

Sous Linux (Debian/Ubuntu) :

```
sudo apt update
sudo apt install openssl
```

Objectif général : Comprendre et manipuler les algorithmes cryptographiques modernes à l'aide d'OpenSSL, en analysant leurs propriétés de sécurité et leurs usages réels.

Objectifs pédagogiques

À la fin de ce TP, l'étudiant sera capable de :

- Utiliser OpenSSL pour le chiffrement symétrique et asymétrique
- Comprendre la différence entre chiffrement, échange de clés et signature
- Implémenter un schéma de chiffrement hybride
- Manipuler Diffie-Hellman et ECDSA
- Analyser la sécurité des choix cryptographiques

Exercice 1 — Chiffrement symétrique AES (niveau avancé)

1. Créez un fichier texte `message.txt` contenant au moins deux paragraphes.
2. Chiffrez ce fichier avec **AES-256-CBC** en utilisant OpenSSL :

```
openssl enc -aes-256-cbc -salt -in message.txt -out message.enc
```

3. Vérifiez que le fichier chiffré n’est plus lisible avec `cat` ou `strings`.
4. Déchiffrez le fichier et vérifiez l’intégrité du contenu :

```
openssl enc -d -aes-256-cbc -in message.enc -out message.dec
```

5. Affichez les paramètres cryptographiques utilisés (IV, sel) avec l’option `-p`.
6. Questions :
    - Quel est le rôle du **sel (salt)** ?
    - Pourquoi AES est-il qualifié d’algorithme **symétrique** ?
    - Que se passe-t-il si l’IV est réutilisé ?

Exercice 2 — Échange de clé Diffie-Hellman avec OpenSSL

1. Générez des paramètres Diffie-Hellman de 2048 bits :

```
openssl dhparam -out dhparam.pem 2048
```

2. Analysez le fichier généré :

```
openssl dhparam -in dhparam.pem -text -noout
```

3. Identifiez les valeurs **p** et **g**.
4. Expliquez le rôle de Diffie-Hellman dans un protocole sécurisé.
5. Questions :
    - Diffie-Hellman chiffre-t-il des données ?
    - Quel problème de sécurité résout-il ?
    - Pourquoi DH est souvent combiné avec AES ?
    -

Exercice 3 — Cryptographie asymétrique RSA

1. Générez une clé privée RSA de 2048 bits :

```
openssl genrsa -out rsa_private.pem 2048
```

2. Affichez les paramètres de la clé :

```
openssl rsa -in rsa_private.pem -text -noout
```

3. Générez la clé publique associée :

```
openssl rsa -in rsa_private.pem -pubout -out rsa_public.pem
```

4. Créez un fichier aléatoire de 32 octets :

```
openssl rand 32 -out secret.key
```

5. Chiffrez ce fichier avec la clé publique RSA :

```
openssl rsautl -encrypt -pubin -inkey rsa_public.pem -in secret.key -out secret.enc
```

6. Déchiffrez-le avec la clé privée.
7. Questions :
    - Pourquoi ne chiffre-t-on pas directement de gros fichiers avec RSA ?
    - Quelle clé est utilisée pour le chiffrement ? pour le déchiffrement ?
    - Quel est le lien avec le chiffrement hybride ?

Exercice 4 — Chiffrement hybride RSA + AES

1. Utilisez la clé `secret.key` (Exercice 3) comme mot de passe pour chiffrer un fichier avec AES :

```
openssl enc -aes-256-cbc -in message.txt -out message_hybrid.enc -pass file:secret.key
```

2. Déchiffrez le fichier en utilisant la même clé.
3. Expliquez pourquoi cette méthode est plus efficace et plus sûre que :
    - AES seul
    - RSA seul
4. Question :
    - Quelle partie du système assure la **confidentialité** ?
    - Quelle partie assure l'**échange sécurisé de clé** ?

Exercice 5 — Signature numérique avec ECDSA

1. Générez une clé privée ECDSA basée sur la courbe `prime256v1` :

```
openssl ecparam -name prime256v1 -genkey -noout -out ecdsa_private.pem
```

2. Générez la clé publique associée :

```
openssl ec -in ecdsa_private.pem -pubout -out ecdsa_public.pem
```

3. Signez le fichier `message.txt` :

```
openssl dgst -sha256 -sign ecdsa_private.pem -out signature.bin message.txt
```

4. Vérifiez la signature :

```
openssl dgst -sha256 -verify ecdsa_public.pem -signature signature.bin message.txt
```

5. Modifiez un caractère dans `message.txt` et vérifiez à nouveau la signature.
6. Questions :
    - Que garantit une signature numérique ?
    - Pourquoi ECDSA est-il préféré à RSA dans les systèmes modernes ?
    - Quelle est la différence entre **signature** et **chiffrement** ?

Exercice 6 — Analyse et réflexion (obligatoire)

Répondez de manière argumentée :

1. Comparez AES, RSA, Diffie-Hellman et ECDSA (rôle, usage, performance).
2. Pourquoi dit-on que la cryptographie moderne repose sur des **schémas hybrides** ?
3. Quels seraient les risques d'utiliser :
    - une clé RSA de 1024 bits ?
    - un mot de passe faible avec AES ?
4. Citez un exemple réel d'utilisation pour chaque algorithme étudié.