# Travaux Pratiques

# Cryptographie Symétrique

## Objectifs

1. Comprendre le principe du chiffrement symétrique.
2. Mettre en œuvre les algorithmes DES et AES dans CrypTool 2.
3. Tester différents modes de chiffrement (ECB, CBC, CFB, OFB).
4. Visualiser la propagation des erreurs et la diffusion.
5. Comprendre la sensibilité aux clés et au texte clair.

## Partie-1 :Cryptool 2

### Exercice 1 — Prise en main de CrypTool 2

**Objectif :**

Découvrir l'environnement et exécuter un premier chiffrement symétrique.

**Étapes :**

1. Ouvrir CrypTool 2.
2. Menu **"File → New → Workspace"**.
3. Ajouter :
   - un bloc **"Text Input"**,
   - un bloc **"Symmetric Encryption (DES)"**,
   - un bloc **"Text Output"**.
4. Connecter les blocs :
   `Text Input → DES → Text Output`
5. Entrer le message :

   ```
   MESSAGE : CRYPTOGRAPHIE
   CLÉ : 133457799BBCDFF1
   ```

6. Lancer le chiffrement et observer la sortie (texte chiffré en hexadécimal).

**Question :**
Que remarquez-vous si vous changez une seule lettre du texte clair ?

## Exercice 2 — Étude de l'algorithme DES

**Objectif :**

Observer le fonctionnement interne de DES.

**Étapes :**

1. Ajouter le bloc **"Visualization of DES"**.
2. Entrer :
   - **Texte clair** : `0123456789ABCDEF`
   - **Clé** : `133457799BBCDFF1`
3. Lancer la visualisation.
4. Analyser :
   - la permutation initiale (IP),
   - les 16 rondes de Feistel,
   - la permutation finale (FP),
   - les sous-clés générées à chaque ronde.

**Question :**
Quelle est la taille de chaque sous-clé ?
Combien de bits changent dans la sortie si un seul bit d'entrée change ?

## Exercice 3 — Étude de l'AES

**Objectif :**

Comprendre les transformations internes de l'AES.

**Étapes :**

1. Ouvrir un nouveau workspace.
2. Ajouter :
   - **"Text Input"**,
   - **"AES Encryption"**,
   - **"Visualization of AES"**,
   - **"Text Output"**.
3. Clé : `2B7E151628AED2A6ABF7158809CF4F3C`
   Texte clair : `3243F6A8885A308D313198A2E0370734`
4. Observer :
   - SubBytes, ShiftRows, MixColumns, AddRoundKey.
   - Génération des clés de ronde.

**Question :**
Combien de tours sont appliqués pour AES-128 ?
Que se passe-t-il si l'on change un bit de la clé ?

## Exercice 4 — Étude des modes de chiffrement

**Objectif :**

Comparer les modes ECB, CBC, CFB et OFB.

**Étapes :**

1. Créer une chaîne :
   - **"File Input"** → **"Image"** (ex. une photo en noir et blanc)
   - **"AES Encryption"**
   - **"File Output"**
2. Tester successivement :
   - ECB,
   - CBC,
   - CFB,
   - OFB.

**Astuce :**
Utilisez la même clé et le même IV (pour CBC, CFB, OFB).

**Questions :**

- Quelle différence visuelle observez-vous entre l'image chiffrée en **ECB** et les autres modes ?
- Pourquoi le mode **ECB** n'est-il pas recommandé ?

## Exercice 5 — Sensibilité au changement de clé et au texte clair

**Objectif :**

Illustrer l'effet avalanche.

**Étapes :**

1. Prenez le même texte clair : `"HELLO WORLD"`.
2. Chiffrez-le deux fois :
   - avec la clé `AAAAAAAAAAAAAAAA`,
   - avec la clé `AAAAAAAAAAAAAAA9`.
3. Comparez les résultats en hexadécimal.

**Question :**
Combien de bits changent ?
Pourquoi ce phénomène est-il important pour la sécurité ?

# Partie-2 : OpenSSL

## Exercice-6 — Génération d'une clé et chiffrement basique

**But :**

Comprendre le chiffrement symétrique avec OpenSSL.

**Étapes :**

1. Crée un fichier texte :

```
echo "La cryptographie protège l'information." > message.txt
```

2. Chiffre le message avec **AES-128-CBC** :

```
openssl enc -aes-128-cbc -in message.txt -out message.enc -k
secret123
```

3. Visualise le fichier chiffré :

```
hexdump -C message.enc | head
```

4. Déchiffre le fichier :

```
openssl enc -aes-128-cbc -d -in message.enc -out
message_dechiffre.txt -k secret123
```

5. Compare :

```
diff message.txt message_dechiffre.txt
```

**Questions :**

- Quelle différence observes-tu entre le fichier clair et le fichier chiffré ?
- Que se passe-t-il si tu changes un seul caractère de la clé ?

## Exercice 7 — Étude de différents algorithmes

**But :**

Comparer les algorithmes symétriques pris en charge par OpenSSL.

**Étapes :**

1. Liste tous les algorithmes :

```
openssl enc -ciphers
```

2. Chiffre avec différents algorithmes :

```
openssl enc -aes-256-cbc -in message.txt -out aes256.enc -k key123
openssl enc -des-ede3-cbc -in message.txt -out des3.enc -k key123
openssl enc -bf-cbc -in message.txt -out blowfish.enc -k key123
```

3. Compare les tailles des fichiers chiffrés :

```
ls -l *.enc
```

**Questions :**

- Quelle est la différence de taille entre les sorties ?
- Quel algorithme est le plus rapide ? (tu peux utiliser `time`)

## Exercice 8 — Étude des modes de chiffrement

**But :**

Visualiser la différence entre ECB et CBC.

**Étapes :**

1. Télécharge une petite image :

```
Wget
https://upload.wikimedia.org/wikipedia/commons/7/77/Delete_key1.jpg -
O image.jpg
```

2. Chiffre avec le mode **ECB** :

```
openssl enc -aes-128-ecb -in image.jpg -out image_ecb.jpg -k password
```

3. Chiffre avec le mode **CBC** :

```
openssl enc -aes-128-cbc -in image.jpg -out image_cbc.jpg -k password
-iv 00000000000000000000000000000000
```

4. Ouvre les deux images :

```
eog image_ecb.jpg &
eog image_cbc.jpg &
```

**Questions :**

- Quelle image laisse apparaître la structure originale ?
- Pourquoi le mode ECB n’est-il pas sûr pour les images ?

## Exercice 9 — Sensibilité à la clé et au texte clair

**But :**

Observer l'effet avalanche.

**Étapes :**

1. Crée deux fichiers similaires :

```
echo "HELLO WORLD" > m1.txt
echo "HELLO WORLE" > m2.txt
```

2. Chiffre-les avec la même clé :

```
openssl enc -aes-128-cbc -in m1.txt -out c1.enc -k key123
openssl enc -aes-128-cbc -in m2.txt -out c2.enc -k key123
```

3. Compare les fichiers chiffrés :

```
diff c1.enc c2.enc
```

**Question :**

- Pourquoi une petite différence dans le message produit-elle un résultat complètement différent ?

**BON TRAVAIL**