# TP2 - Chiffrement asymétrique, Openssl et SSH

## Chiffrement par OpenSSL document :

1. **Générer la clé privée RSA** :

openssl genpkey -algorithm RSA -out private_key.pem -aes256

Cette commande génère une clé privée RSA protégée par une phrase de passe en utilisant le chiffrement AES-256. Si vous ne souhaitez pas utiliser de chiffrement, vous pouvez simplement omettre `-aes256`.

2. **Générer la clé publique RSA** :

openssl rsa -pubout -in private_key.pem -out public_key.pem

Cette commande extrait la clé publique de la clé privée générée précédemment et la stocke dans un fichier séparé.

### Utiliser la clé RSA générée pour chiffrer un document texte :

openssl rsautl -encrypt -inkey public_key.pem -pubin -in document.txt -out document.enc

Cette commande utilise la clé publique (`public_key.pem`) pour chiffrer le fichier `document.txt` et enregistre le résultat chiffré dans `document.enc`.

### Pour déchiffrer le document chiffré, utilisez la commande suivante :

openssl rsautl -decrypt -inkey private_key.pem -in document.enc -out document_decrypted.txt

Cette commande utilise la clé privée (`private_key.pem`) pour déchiffrer le fichier `document.enc` et enregistre le résultat déchiffré dans `document_decrypted.txt`.

### Chiffrer l'image avec la clé publique RSA :

Chiffré :
openssl rsautl -encrypt -inkey public_key.pem -pubin -in image.png -out image_rsa.enc

Déchiffré :
openssl rsautl -decrypt -inkey private_key.pem -in image_rsa.enc -out image_rsa_decrypted.png

### Chiffrement de l'image avec AES-CBC :

Générer une clé et un vecteur d'initialisation (IV) pour AES
openssl rand -base64 32 > aes_key.bin

openssl rand -base64 16 > aes_iv.bin

**Chiffrer l'image avec AES-CBC :**

openssl enc -aes-256-cbc -in image.png -out image_aes.enc -K $(xxd -p -c 256 aes_key.bin) -iv $(xxd -p -c 128 aes_iv.bin)

**Déchiffrer l'image avec AES-CBC :**

openssl enc -d -aes-256-cbc -in image_aes.enc -out image_aes_decrypted.png -K $(xxd -p -c 256 aes_key.bin) -iv $(xxd -p -c 128 aes_iv.bin)

**Remarques et observations :**

1. **RSA** :
    - **Limitation de la taille** : RSA est principalement utilisé pour chiffrer de petites quantités de données (comme des clés de chiffrement) plutôt que des fichiers volumineux tels que des images. Cela est dû à la taille fixe de l'entrée qu'un cryptosystème RSA peut chiffrer.
    - **Performance** : Le chiffrement RSA est beaucoup plus lent que les algorithmes symétriques comme AES, en particulier pour des fichiers volumineux.
2. **AES-CBC** :
    - **Adapté pour les grands fichiers** : AES (Advanced Encryption Standard) est bien adapté pour le chiffrement de grands fichiers tels que des images. Il est rapide et efficace pour le chiffrement de volumes importants de données.
    - **Utilisation des clés** : AES utilise des clés symétriques, ce qui signifie que la même clé est utilisée pour le chiffrement et le déchiffrement. La gestion des clés doit donc être sécurisée, souvent en les chiffrant avec une clé publique RSA avant de les transmettre.