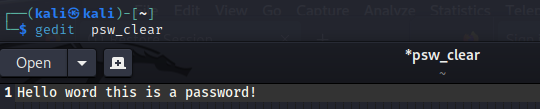
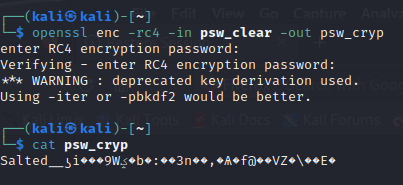
CHIFFREMENT SYMETRIQUE avec L’algorithme RC4

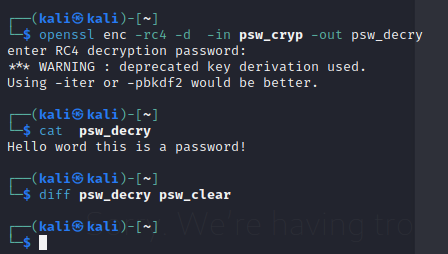
1.chiffré



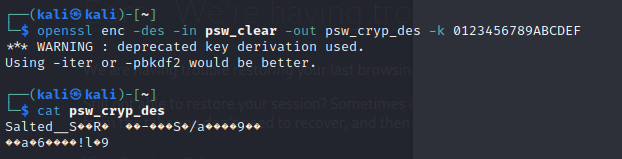


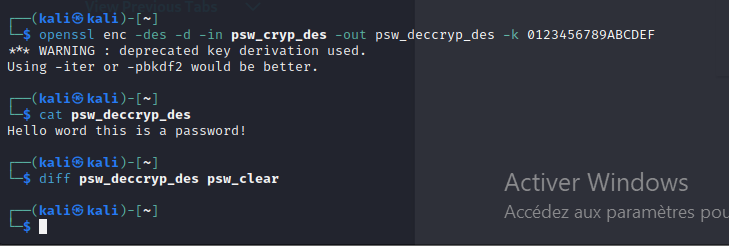
Password : enc

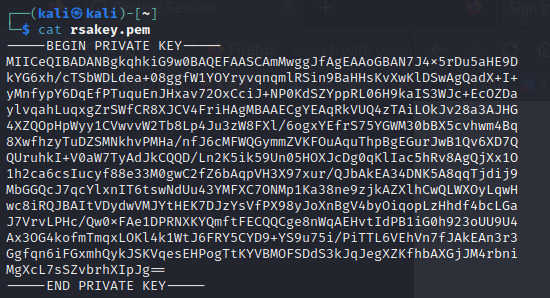
2. déchiffrer et comparer



CHIFFREMENT SYMETRIQUE avec L’algorithme DES

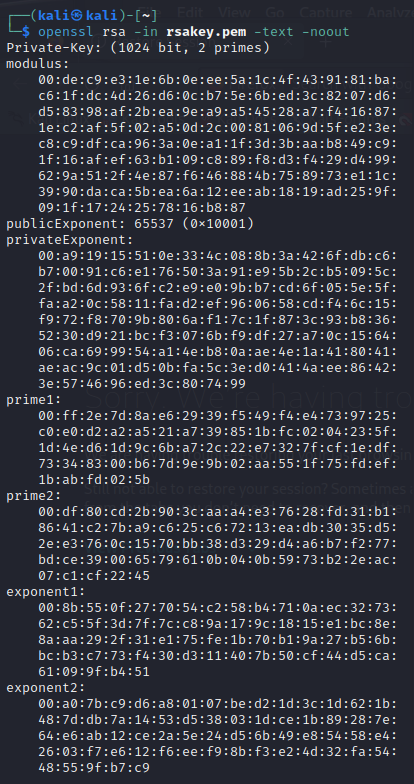




2. CHIFFREMENT ASYMETRIQUE

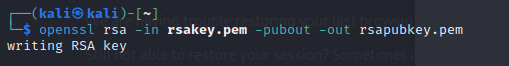
## Générer une paire de clés de taille 1024 stocker-la dans le fichier rsakey.pem

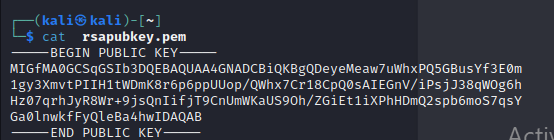
## visualiser les clés en format complet



Extraire la clé publique de la clé privée et sauvegarder le résultat dans le fichier

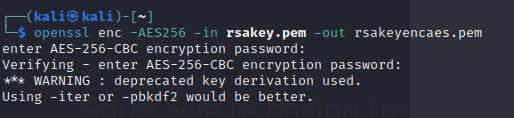
rsapubkey.pem

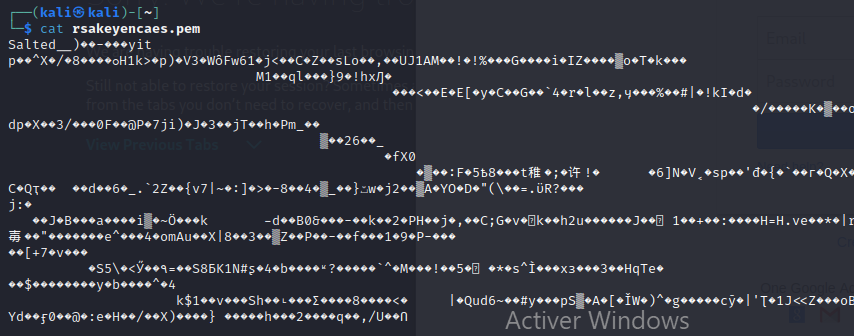




Chiffrement de la clé RSA par l’algorithme

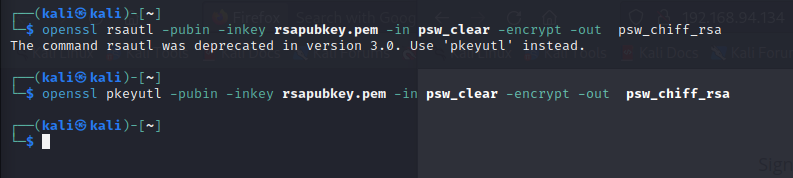
**commande qui permet de chiffrer le fichier rsakey.pem et produit ainsi un fichier rsakeyencaes.pem.**



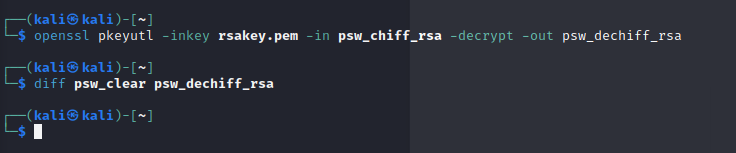


**Chiffrement/déchiffrement de données avec RSA**

**Chiffrer psw\_clear avec la clé publique rsapubkey.pem**



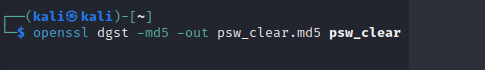
Déchiffrer et comparer

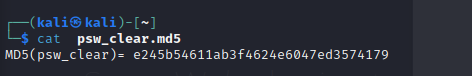


3. SIGNATURE NUMERIQUE

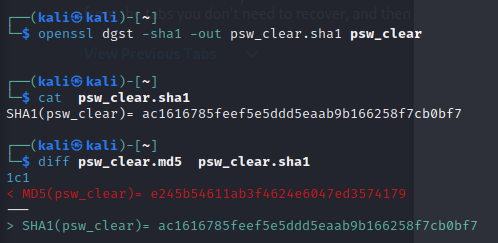
Génération d’une empreinte d’un fichier

\*MD5





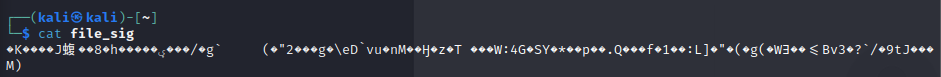
\*SH1



On remarque que MD5 produit une empreinte plus courte (32 caractères) par rapport à SHA-1 (40 caractères)

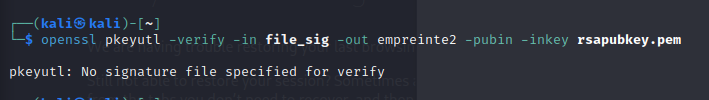
Signature d’un fichier





**Quelle est la clé que vous devez utiliser pour signer le fichier fichier\_nom\_eleve.sha1?**

**Vous devez utiliser la clé privée (rsakey.pem) pour la signature.**



**Quelle est la clé que vous devez utiliser pour vérifier la signature du fichier fichier\_sig?**

**Vous devez utiliser la clé publique (rsapubkey.pem) pour la vérification.**

CERTIFICAT NUMERIQUE

Génération de la clé privée