Opérations pratiques

Sécurité Informatique (Licence 3)

Abdelmalik Bachir

Génération des clés privée et publique (openssl)

`$openssl genrsa -out private_key.pem 2048`

- génère une clé privée qui va être sauvegardée dans le fichier `private_key.pem`
- l'algorithm utilisé est RSA
- la taille de la clé est 2048 bits

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

- extraire la clé publique `public_key.pem` à partir de la clé privée `private_key.pem`

Chiffrement d’un texte avec une clé publique

`$openssl rsautl -in message.txt -out message-rsa.crypt -inkey public_key.pem -pubin -encrypt`

- le fichier en clair `message.txt` contenait la phrase “Je suis un secret”
- le fichier crypté `message-rsa.crypt` avec la clé publique `public_key.pem` contient : “«ñ<94>Îòµb$;
Ç^AÜ<83>^\<91>ñ×¸^W<83>X<93>'ÕÁDÃ<90>~FeÔvI<83>P^Q<91>±I<95>æ<8c>^]<85>®<84>ÀÜDã'µÞQý#s£ç&ugÑø¤ìõ<9e>¨<8b>$=
¤å<99>§$Ü<80><9b>&<88><85>Â<96>Æ<8a><%6<8a>8@·Wû<95>^Oü<95>{¬Y<93>£Þ¾ðåþø<92>+(<94>^@çúÔyq<9f><98> ¨<9e>yX}
·|<84>w2ødª^XZ^^øÜ6*Õ,<85>¨^Kp^U<80>WïSÌ<8a><8e>²Kg+^KÖÓSÃsmÌ^Rm^Q^\>Ù"yÁ¨^QÍ^_R½^[H©+g<88>^K^?A5^T<8a^MÓ¿<87>A<
82>{<99>í^PSTM<90>eË©{;éºl«<86>Åh+·{s^K1ù2aÑ$<81>Ì^]X<84>t¸#þº:÷ý`ÍX”
-

Déchiffrement du message avec la clé privée

```
$openssl rsautl -decrypt -in message-rsa.crypt -out
message-rsa.txt -inkey private_key.pem
```

- le résultat du déchiffrement est mis dans le fichier `message-rsa.txt`
- l'algorithme de déchiffrement utilisé est RSA et la clé privée de décryptage est **`private_key.pem`**

Chiffrement d’un texte avec une clé partagée

```
$openssl enc -aes256 -in message.txt -out message-aes.crypt -pass
pass:myPassword123
```

- le fichier en clair `message.txt` contenait la phrase “Je suis un secret”
- le fichier crypté `message-aes.crypt` avec la clé partagée `myPassword123` contient : “`Salted__<89>ÔW<80>d×lÂ/ÔCÌWIl)ÕFU#ø°6Ä²^E<90>JªC<9c>Q¼^K|Vê2<9d>`”

Déchiffrement du message avec la clé partagée

```
$openssl enc -in message-aes.crypt -out message-aes.txt -pass
pass:myPassword123 -d -aes256
```

- le résultat du déchiffrement est mis dans le fichier `message-aes.txt`
- l'algorithme de déchiffrement utilisé est AES256

Calcul du condensat (hachage) d’un message

```
$openssl dgst -sha256 -binary -out message-aes.crypt.dgst
message-aes.crypt
```

- le résultat du hachage du fichier `message-aes.crypt` est mis dans le fichier `message-aes.crypt.dgst`
- l’algorithme de hachage utilisé est le SHA256

Signature du condensat

```
$openssl rsautl -in message-aes.crypt.dgst -out
message-aes.crypt.dgst.sign -sign -inkey private_key.pem
```

- le résultat de la signature du fichier `message-aes.crypt.dgst` est mis dans le fichier `message-aes.crypt.dgst.sign`
- la signature est effectuée avec la clé privée `private_key.pem`

Vérification de la signature

```
$openssl rsautl -in message-aes.crypt.dgst.sign -out
message-aes.crypt.dgst1 -pubin -inkey public_key.pem
```

- pour vérifier la signature, on applique la clé publique `public_key.pem` au condensat signée `message-aes.crypt.dgst.sign`, le résultat de l'opération est mis dans `message-aes.crypt.dgst1`
- le message message-aes.crypt est bel et bien signé si `message-aes.crypt.dgst1` et `message-aes.crypt.dgst` sont égaux.

Références bibliographiques

Yacine Challal, *Ingénierie des protocoles et logiciels sécurisés*
http://y_challal.esi.dz/index.php/2016/04/30/ingenierie-des-protocoles-et-logiciels-securises-ipls/