Editor: Yuxin SHI
Date: 03/03/2018

TD1 Cryptographie

EXO9

a. P1 va utiliser la clé publique de P2 pour garantir la confidentiation du message. Puis P1 va utiliser sa clé privée et signer le message, ainsi garantir son authenticité.
b. P2 va utiliser la clé publique de P1 pour vérifier c'est bien P1 qui l'envoyer quand il recevoit le message, pour lire le message, il va utiliser sa clé privée pour décrypter le message.

EXO10

Dans cet exo, j'ai utilisé un raspberry pour jouer le rôle de P2.
Pour encrypter le message(P1):

```
teasyu@devdog:~/Documents/ESIPE/Security$ openssl rsautl -encrypt -pubin -inkey
rsa_pub_p2.pem -in secret -out secret.en
```

Pour signer le fichier puis envoyer au P2:

```
teasyu@devdog:~/Documents/ESIPE/Security$ openssl pkeyutl -sign -in secret -inke
y rsa_private_P1.pem -out secret.sig
teasyu@devdog:~/Documents/ESIPE/Security$ scp secret.en pi@192.168.1.80:/home/pi
/Documents/P2/
pi@192.168.1.80's password:
secret.en                                     100%  128    28.0KB/s   00:00
```

Du côté p2:
Pour vérifier la signature:

```
pi@raspberrypi:~/Documents/P2 $ openssl pkeyutl -verify -in secret -sigfile secr
et.sig -pubin -inkey rsa_pub_P1.pem
Signature Verified Successfully
```

Pour décrypter le message:

```
pi@raspberrypi:~/Documents/P2 $ ls
rsa_private_p2.pem  rsa_pub_P1.pem  rsa_pub_p2.pem
pi@raspberrypi:~/Documents/P2 $ scp rsa_pub_p2.pem teasyu@192.168.1.35:/home/tea
syu/Documents/ESIPE/Security/
teasyu@192.168.1.35's password:
rsa_pub_p2.pem                                100%  272     0.3KB/s   00:00
pi@raspberrypi:~/Documents/P2 $ ls
rsa_private_p2.pem  rsa_pub_P1.pem  rsa_pub_p2.pem  secret.en
pi@raspberrypi:~/Documents/P2 $ openssl rsautl -decrypt -inkey rsa_private_p2.pe
m -in secret.en -out secret
pi@raspberrypi:~/Documents/P2 $ cat secret
un message
From P1 to P2
```