ITC8250 - Cyber Security Technologies I
Group Work 2

214322IV - Maëlie LEBARON
214366IV - Jordan Béziaud
214307IV - Vincent Rossignol
214319IV - Marine Hervet
214310IV - Victor Thévin

November 9, 2021

1 Briefly explain the components of an asymmetric encryption scheme and the required properties.

An asymmetric encryption scheme uses 2 keys : *the public key and the private key.*

It can be use in two different ways :

- **To send an encrypted message to someone** : the public key is used to encrypt a message, so that only the intended receiver with the associated private key, can decrypt it.
- **To authenticate a sender**: Using the private key to encrypt a message means that only the message can be decrypted only using the public key, which therefore authenticate the sender

The main property is that the private has to remains private, because if it is shared someone could usurp the sender identity, and we can not anymore authenticate the sender for example. The key should also be cryptographically secure.

2 Suppose that Bob wishes to encrypt a message for Alice. He therefore needs Alice's public key which he either receives from Alice upon request or which he downloads from Alice's web page. If Mallet controls the communication channel used by Bob to receive Alice's public key, Bob has no way to verify whether Alice's public key is authentic, i.e., if it originates from her.

2.1 Describe schematically how such a man-in-the-middle attack works. What messages are exchanged between Alice, Bob, and the adversary.

2.2 How could this kind of attack be prevented? What must Alice and Bob be aware of if they want to use public key cryptography as described in the scenario?

To prevent man-in-the-middle attacks, users should:

- use https websites
- use a VPN (Virtual Private Network) and separate the networks with VLAN (Virtual Local Area Network)

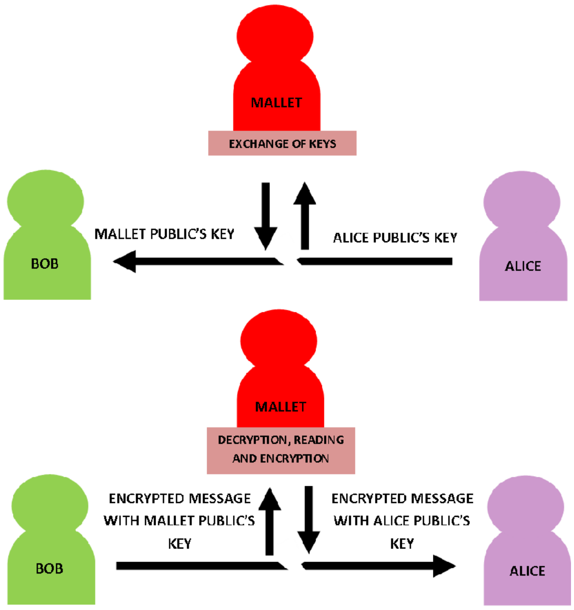

MALLET
EXCHANGE OF KEYS
BOB
MALLET PUBLIC'S KEY
ALICE PUBLIC'S KEY
ALICE
MALLET
DECRYPTION, READING AND ENCRYPTION
BOB
ENCRYPTED MESSAGE WITH MALLET PUBLIC'S KEY
ENCRYPTED MESSAGE WITH ALICE PUBLIC'S KEY
ALICE

Figure 1: Man is the middle attack

- avoid public networks
- use strong password for wifi and router login credentials
- use IDS (Intrusion Detection System) technologies
- use two-factor authentication

Using public key cryptography, Alice and Bob must be aware that they are sending their public key into a potentially insecure channel and someone could impersonate them.

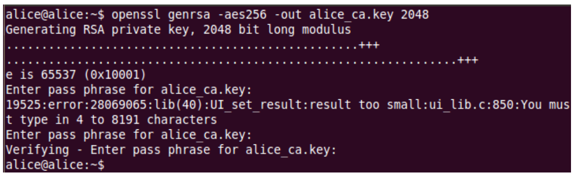
3 Suppose Alice publishes her public key on her web page. How can Bob determine its authenticity?

Bob can determine the authenticity of the public key using a certificate, which will certify that the public key is, in fact, linked to the private key of Alice. He should verify that the certificate has been authorized by a trusted authority.

4 Setup a certificate authority on alice's machine (show in report what commands you used)

We use the `openssl` tool to generate a RSA private key with the RSA-based private key generation utility `openssl-genrsa` with the following parameterss : - usage of the AES algorithm with length 256 bits - usage of a 2048 bit long modulus for improved security

Thus, by running the following command with the options specified above : `openssl genrsa -aes256 -out alice` 2048, we generated the private key named "alice_ca.key".



```
alice@alice:~$ openssl genrsa -aes256 -out alice_ca.key 2048
Generating RSA private key, 2048 bit long modulus
.................................................+++
.................................................................+++
e is 65537 (0x10001)
Enter pass phrase for alice_ca.key:
19525:error:28069065:lib(40):UI_set_result:result too small:ui_lib.c:850:You mus
t type in 4 to 8191 characters
Enter pass phrase for alice_ca.key:
Verifying - Enter pass phrase for alice_ca.key:
alice@alice:~$
```

Figure 2: Creation of alice's private key to use in the certificate

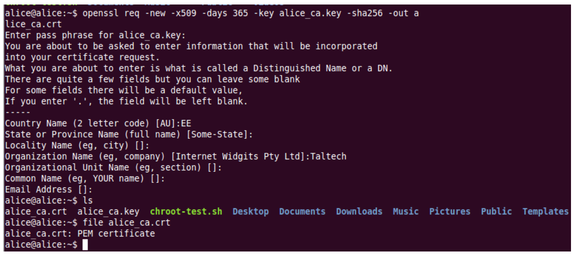
Afterward, by using the certificate generating utility of the `openssl` package called `openssl req`, we can generate a new certificate with the -new option.

The certificate will have the following parameters :

- expire in 365 days (`-days 365`)
- usage of SHA256 (`-sha256`) to create the hash
- usage of the X.509 standards (`-x509`) to declare the format of the public key certificate

Lastly, we will base the certificate on a private key that we generated on the previous step.

After running `openssl req -new -x509 -days 365 -key alice_ca.key -sha256 -out alice\_ca.crt`, we generated the certificate named "alice_ca.crt" !



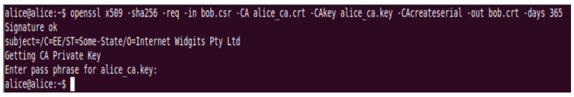
```
alice@alice:~$ openssl req -new -x509 -days 365 -key alice_ca.key -sha256 -out a
lice_ca.crt
Enter pass phrase for alice_ca.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:EE
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Taltech
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:
Email Address []:
alice@alice:~$ ls
alice_ca.crt  alice_ca.key  chroot-test.sh  Desktop  Documents  Downloads  Music  Pictures  Public  Templates
alice@alice:~$ file alice_ca.crt
alice_ca.crt: PEM certificate
alice@alice:~$ 
```

Figure 3: Generation of the certificate based on Alice's private key

5 Create a new key pair and use the CA to sign the public key. Explain the commands you use.

Let's generate a second private key, one of Bob's key for example by running `openssl genrsa -out bob.key 2048`

Then, we will generate a certificate signing request upon the private key with the same hash algorithm as before : sha256 with the following command: `openssl req -sha256 -new -key bob.key -out bob.csr`



```
alice@alice:~$ openssl x509 -sha256 -req -in bob.csr -CA alice_ca.crt -CAkey alice_ca.key -CAcreateserial -out bob.crt -days 365
Signature ok
subject=/C=EE/ST=Some-State/O=Internet Widgits Pty Ltd
Getting CA Private Key
Enter pass phrase for alice_ca.key:
alice@alice:~$ 
```

Figure 4: Generating Bob's certificate signing request based on his private key

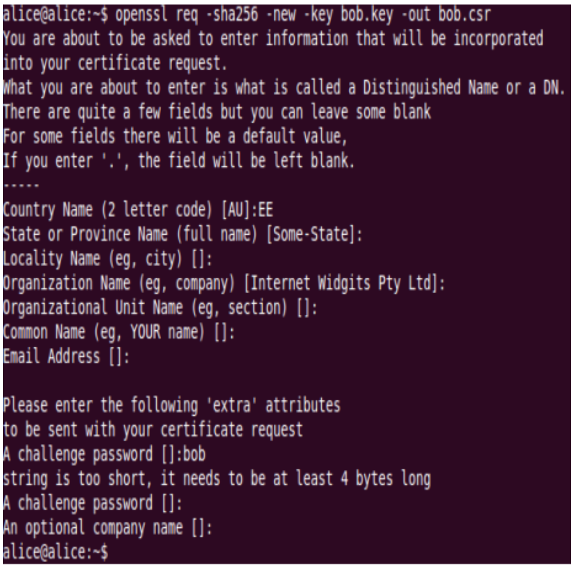
Now that we have the file to sign the certificate with, we can proceed to the signature using the openssl req utility with the following parameters :

- which certificate to sign with the `-in` option
- which certificate authority to use with the `-CA` option, we will use the one previously created : alice_ca.crt
- which certificate authority key, same as above by using the `-CAkey` option with the alice's private key : alice_ca.key
- the creation of a unique serial number to identify the certificate with the `-CAcreateserial option`
- the expiration time of 365 days

Finally, running the following command allows bob to sign the certificate using alice's certification authority : `openssl x509 -sha256 -req -in bob.csr -CA alice\_ca.crt -CAkey alice\_ca.key -CAcreat`

6 Briefly characterize the following building blocks of a public key infrastructure (PKI): public key, private key and certificate.

- **The public key** is meant to be shared to anyone
- **The private key** is only detain by one and only one person, I is private and should never be shared.



```
alice@alice:~$ openssl req -sha256 -new -key bob.key -out bob.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:EE
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:bob
string is too short, it needs to be at least 4 bytes long
A challenge password []:
An optional company name []:
alice@alice:~$
```

Figure 5: Signing Bob's certificate with Alice's certificate authority

- **The certificate** is a digital document that bind the public key with its owner (the one who possess the private key). It allow us to link the public key and the private key together. The certificate allows us to ensure that the public key we used to decrypt a message is linked to the private key of the person we want to authenticate.

7 Suppose that Alice wants to authenticate Bob using certificate-based authentication. Suppose further that Alice holds the CA's certificate and that Bob has a certificate for his public key issued by the CA. Explain the necessary steps of the authentication process.

Alice should first verify that Bob's public key is valid using the certificate of Bob. Then, she should verify the certificate using the CA's certificate, to ensure that the authority that delivered Bob's certificate can be trusted.

The command to use is the following :

```
openssl verify -CAfile ca\_certificate.crt bob\_certificate.crt
```