Public key cryptography in OpenSSL
HW6 - CNS Sapienza

Lombardo Andrea 1893440

12/12/2019

1 Introduction

The main goal of this abstract is to get familiar with :

- generating keys
- generating certificates
- converting certificates
- digital signing

Using command line from bash I manage to generating a pairs of keys(public and private) RSA and DSA through genpkeys options of the openssl tool. Then I need to sign and verify a file and also in that case I use the OpenSSL command line tool.

2 Design

Generate an RSA keypair with a 2048 bit private key:

```
openssl genpkey -algorithm RSA -out private_rsa_key.pem -pkeyopt rsa_keygen_bits:2048
```

instead for the DSA keypair with a 2048 bit first I should create a parameter for building the private key:

```
openssl genpkey -genparam -algorithm DSA -out parameter_dsa_key.pem -pkeyopt dsa_paramgen_bits:2048
```

Extracting the public key from an RSA keypair and put it in a new file, public_rsa_key.pem

```
openssl rsa -pubout -in private_rsa_key.pem -out public_rsa_key.pem
```

instead for DSA first I should define the private_key:

```
openssl genpkey -paramfile parameter_dsa_key.pem -out private_dsa_key.pem
```

after this

make sure to prevent other users from reading your keys allowing the rights after defining both private_keys:

```
chmod go-r private_rsa_key.pem && chmod go-r paramater_dsa_key.pem
```

I evaluate the follwing command:

```
openssl dsa -pubout -in private_dsa_key.pem -out public_dsa_key.pem
```

Now I should create a self-certification: **openssl req -x509 -new -key private_rsa_key.pem -out certification_rsa.pem** and then according to the following script I can check if it's ok: **openssl x509 -in certification_rsa.pem -keyform PEM -text -noout**

Now we can see that modifying the word from rsa to dsa we can build and check DSA certification.

Now for the digital signature I take a file called ”data.txt” and insert inside a content choice randomic and according the keys built before I should use the following scripts:
openssl dgst -sign private_rsa_key.pem -keyform PEM -sha256 -out data.txt.sign -binary data.txt

instead for verify the signature I should use the following scripts:
openssl dgst -verify public_rsa_key.pem -keyform PEM -sha256 -signature data.txt.sign -binary data.txt .
Now first I sign the file into a new file called data.txt.sign and then I check if the bash return ”Verified OK” that measn that everything is OK. How is it describe in the article in the following link:
”https://www.go4expert.com/articles/digital-certificate-formats-filename-t24831/” is possible to modify the format of the X.509 certificates: where the most common formats are

- **em - (Privacy Enhanced Mail) - PEM**
- **cer, .crt, .der,**
- **p7b, .p7c - PKCS#7 - PKCS #7,p12 - PKCS#12 ,**
- **pfx - PFX (Personal Information Exchange)**

1) convert an x509 certificate from cer to PEM format: openssl x509 –in cert.cer –out cert.pem

2) Convert PEM Format Certificate to PFX Format Certificate: openssl x509 pkcs12 -export -out certificate.pfx -inkey rsa_key.key -in certificate.pem

References

[1] https://en.wikibooks.org/wiki/Cryptography/Generate_a_keypair_using_OpenSSL

[2] https://gist.github.com/tsaarni/14f31312315b46f06e0f1ecc37146bf3

[3] https://stackoverflow.com/questions/10782826/digital-signature-for-a-file-using-o

[4] https://rietta.com/blog/openssl-generating-rsa-key-from-command/

[5] https://developers.yubico.com/PIV/Guides/Generating_keys_using_OpenSSL.html

[6] https://security.stackexchange.com/questions/5096/rsa-vs-dsa-for-ssh-authenticati

[7] https://www.ssh.com/ssh/keygen/

[8] https://security.stackexchange.com/questions/161526/why-does-generating-a-self-si

[9] https://www.zimuel.it/blog/sign-and-verify-a-file-using-openssl

[10] https://www.openssl.org/docs/manmaster/man1/genpkey.html

[11] https://stackoverflow.com/questions/21297139/how-do-you-sign-a-certificate-signi

[12] https://ingegneria.online/questions/146306/openssl-genera-diversi-tipi-di-certif