Public key cryptography in OpenSSl
HW6 - CNS Sapienza

Edoardo Puglisi 1649359

12/12/2019

Contents

1 Overview

The main purpose of this paper is to define the OpenSSL workflow for generating key-pairs and certificates, converting certificates and digital signing documents in particular using RSA and DSA.

2 Generating keys

In OpenSSL secret keys can be generated with *genpkey* command. The use of it change according on the algorithm used. For RSA just set as parameters algorithm and output file:

```
openssl genpkey -algorithm RSA -out pkeyRSA.pem
```

Other key generation option can be set such as the number of bits of generated key (deafult 2048) or the public key exponent value.

```
—–BEGIN PRIVATE KEY—–
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBA...
—–END PRIVATE KEY—–
```

For DSA algorithm the procedure is a bit diffent. First we must define the set of parameters for the key generator:

```
openssl genpkey -genparam -algorithm DSA -out dsap.pem
```

then create the key from the previously generated parameters set.

```
openssl genpkey -paramfile dsap.pem -out pkeyDSA.pem
```

```
—–BEGIN PRIVATE KEY—–
MIIBSgIBADCCASsGByqGSM44BAEwggEeAoGBAM...
—–END PRIVATE KEY—–
```

Given this new fileS we can extract the public keys:

```
openssl rsa -pubout -in pkeyRSA.pem -out pubRSA.pem
openssl dsa -pubout -in pkeyDSA.pem -out pubDSA.pem
```

```
—–BEGIN PUBLIC KEY—–
MIIBtzCCASsGByqGSM44BAEwggEeAoGBAM1kgabPEgZe0Ijj....
—–END PUBLIC KEY—–
```

3 Generating and verifying X.509 certificate

To generate a selfsigned certificate (X.509) we run the command:

```
openssl req -new -x509 -sha256 -days 365 -key pkeyRSA.
   pem -out certificate.crt
```

It takes as argument the validity and the private key we generated previously. You will be asked to insert certificate attributes such as location, organization name and common name.

```
Country Name (2 letter code) []:IT
State or Province Name (full name) []:Italy
Locality Name (eg, city) []:Roma
Organization Name (eg, company) []:Sapienza
Organizational Unit Name (eg, section) []:CNS
Common Name (eg, fully qualified host name) []:Diag
Email Address []:
```

To verify the self-signed certificate:

```
openssl verify -CAfile certificate.crt certificate.crt
______________
certificate.crt: OK
```

4 Signing and verifying a document

To sign a document use command *dgst* as following:

```
openssl dgst -sign pkeyRSA.pem -out signature input.
   txt
```

This command compute the sha256 of the input file and sign it with the private key. To verify the document given the public key instead:

```
openssl dgst -verify pubRSA.pem -signature signature
   input.txt
______________
Verified OK
```

5 Converting certificate formats

The same command used to create a selfsigned X.509 certificate can be used to convert certificates in different formats.

```
openssl x509 -in certificate.crt -out converted-
   certificate.der -outform DER
```

In this example the previously created certificated is converted in DER format.