M1 Informatique, dec. 4, 2017

TD 4: openssl

1 General overview

`openSSL` is a toolbox implementing the TLS protocol which offers:

(1) a `C` library allowing to write secure client/server applications
(2) a CLI (`openssl`) offering:

- to create RSA or DSA keys, certificates
- the computation of fingerprints (MD5, SHA,...)
- enciphering and deciphering (DES, AES, IDEA, RC4, BLOWFISH,...)
- client server testing for SSL/TLS
- mail signature and enciphering (S/MIME)

General overview of the functionalities offered by `man openssl`; general syntax of the commands:

```
openssl < command > < options >
```

You may also have a look at the following url:

http://www.madboa.com/geek/openssl/#digest-verify

2 Secret keys

A `man enc` gives you all the secret key ciphers offered by `openssl`. Shortly, to encipher the file `test` by blowfish in CBC mode with a password and if the output is the file `test.crypt`, type

```
openssl enc -bf-cbc -in test -out test.crypt
```

and for the deciphering:

```
openssl enc -bf-cbc -d -in test.crypt -out test.plain
```

To provide evidence that the files are identical, you may use `diff`.

(1) the file `ToDecrypt.crypt` was enciphered with idea in OFB mode with a password
   (a) The `md5` fingerprint of the password is: `30c6677b833454ad2df762d3c98d2409`. Decode it (use https://www.hashkiller.co.uk/md5-decrypter.aspx)
   (b) Decipher (`idea-ofb`) the file `ToDecrypt.crypt` available in the archive.
   (c) What happens when you use a bad password when deciphering?
   (d) Compare the sizes of the original file and of its enciphered version. Explain the difference.

3 Public keys

3.1 Key generation

We can generate a pair of RSA keys by typing:

```
openssl genrsa -out <file> <size>
```

where `file` represents the file containing the key and `size` the size in bits of the modulus. The output is a file in `pem` (Privacy Enhanced Mail) format.

(1) build a pair of RSA keys of size 1024 bits named `Student.pem`

The command `rsa` provides a readable output of the `pem` keypair.

```
openssl rsa -in <fichier> -text -noout
```

(1) Read the key that you've just generated;
(2) What is the effect of the `-noout` option?
(3) What is the effect of the `-pubout` option?
(4) Export the public contents of your key in a file `StudentPK.pem`.

3.2 Data enciphering

The command `rsautl` offers to encipher/decipher the data (`-encrypt`/`-decrypt`):

```
openssl rsautl -encrypt -in <file_in> -inkey <key> -out <file_out>
```

- `file_in`: file to encipher (1024 bits key, filesize < 116 octets)
- `key`: RSA keyfile. If it only contains the public key, add the option `-pubin`
- `file_out`: ciphertext

(1) Encipher the file `ToDecrypt.plain` with Blowfish in CBC mode. Encipher the password given to the symmetrical cipher with the recipient's public key (`BMPK.pem`). Add the enciphered password under the name `DigitalEnveloppe.rsa` and the enciphered file `ReEncrypted.crypt`.
(2) Provide the commands:
    - For recovering the password from the digital enveloppe;
    - To decrypt the file `ReEncrypted.crypt`.

3.3 Data signature

Only small documents can be signed. In order to sign large data, we first compute a fingerprint of this document by using the subcommand `dgst`.

```
openssl dgst <hash>  -out <fingerprint> <file_in>
```

where `hash` is a hash function to be chosen among `-sha256` `-sha1` or `-ripemd160`. Signing a document requires the computation of its digest and to sign it, which is done by using:

```
openssl dgst <hash> -sign Key.pem -out <signed_file> clear_file
```

Verification is done by

```
openssl dgst <hash> -verify PKey.pem -signature <signed_file> clear_file
```

4 Certificates

We will generate certificates. Then, we will see how to use them for signing or enciphering email.

4.1 Signing request creation

With your public key `Student.pem`, you can build a signing request to obtain a certificate. When creating the request file, you'll be asked to enter information X509 compliant. The request can be formulated by:

```
openssl req -new -key Student.pem -out myCSR.pem
```

The request file can also be viewed by:

```
openssl req -in myCSR.pem -text -noout
```

(1) Explain the different elements which are contained in this request. Does it contain the private key?

4.2 Certificate signing request

Once the certificate signing request has been created, you'll need to ask a certification authority to deliver the signed certificate after checking your identity.

4.2.1 The certifying authority M1I

In this lab, you'll play the role of the CA. To do this, you'll need the CA certificate and its corresponding keypair. You'll use the CA M1I which is contained in the archive as well as its keypair.
An openssl certificate is a `pem` file of the form:

```
-----BEGIN CERTIFICATE-----
-----END CERTIFICATE-----
```

To show a certificate's contents, use:

```
openssl x509 -in ACert.cer -text -noout
```

(1) After retrieving the CA certificate and its keypair, search for its expiration date and the keys lengths.

4.2.2 Certificate creation

To create and sign a certificate from a certificate signing request like `myCSR.pem`, the CA runs the `x509` command.

```
openssl x509 -days 10 -CA M1I.pem -CAkey M1I.pem -in myCSR.pem -req -out myCert.cer
```

(1) Create a certificate for your key (when signing, the `X509` command asks the CA for its password, if necessary).
(2) Control the certificate contents with the appropriate options of the `X509` command.

4.2.3 Certificates verification

One can verify the certificate validity with the command `verify`. To use it, it is necessary to have the CA certificate who emitted the certificate.

```
openssl verify -CAfile M1I.pem myCert.pem
```

Please make a directory on your computer with the requested files + answers.

Files :
- DigitalEnveloppe.rsa
- ReEncrypted.crypt
- Answers.txt
- Answers.sign

Keys :
- StudentPK.pem
- StudentCERT.cer

The file Answers.txt has to be copied/pasted from below in a text editor and please answer in the file

----------FILE ANSWERS ----------

NAME:
Surname:

Q1: Explain why you could recover the password from its md5 fingerprint

Q2: Why does the size of the original file differ from the size of the encrypted file.

Q3: Give the command for me to retrieve the password used to re-encipher the file ToDecrypt from the digital enveloppe assuming my public key is BMPK.pem

Q4: Provide the command I have to type to decrypt the file ReEncrypted.crypt provided I'll type the password I retrieved in Q3.

Q5: Give the command that verifies your certificate

Q6: Type the command to retrieve your public key from you certificate and the command to check their equality

Q7: Sign This file AFTER you answered all the questions and provide the command(s) I have to type to verify your signature of this file.

----------FILE ANSWERS ENDS ----------

When completed, send the zipped contents of the directory by mail with the name : NAME-Surname.zip (replace NAME by your Name and Surname by your Surname), before december 5 noon.