

FIT3031 Lab Exercise 02: OpenSSL utility – Symmetric/Asymmetric Encryption

The second Lab exercise is focused on running the Hands-on Exercise using OpenSSL utility. The exercise is a follow-up to the first lab exercise. Please follow Lab-01 for the steps required to start the Knoppix operating system under VMware workstation. Open the terminal window and login as **root**.

I. Symmetric encryption.

1. Using OpenSSL utility we can **encrypt** and **decrypt** files with strong ciphers function. For symmetric encryption, using openssl; List the command-lines for encrypting and decrypting a plaintext file using the Blowfish, Triple DES, and CAST5 ciphers:

First login as root by typing the command “su – root”

Next create a sample text file by running the following:

vi plaintext.txt

press “i” or “insert” to enter plain text in the file.

Then press “Esc” key “:wq” to save the file.

Run the following command to encrypt:

\$ openssl enc -e -bf -in plaintext.txt -out plaintext.blowfish

enter bf-cbc encryption password:

Run the following command to decrypt:

\$ openssl enc -d -bf -in plaintext.blowfish -out plaintext-decrypt.txt

enter bf-cbc decryption password:

Compare the files by generating the hash function of each files:

md5sum plaintext.txt plaintext.blowfish plaintext-decrypt.txt

Note: For other cipher function replace -bf for -des3 and -cast5-cbc respectively

II. Base64 Encode/Decode

If the file needs to be transported as an e-mail attachment, we use base64 to encode and decode the message. Sometimes this is referred to as "ASCII armor". List the commands using openssl to encode and decode a sample jpg file. Please download and save a sample jpg file from internet and provide the list of commands to encode and decode.

1. Using OpenSSL Utility and using the built-in Base64 algorithm **encode** and **decode** the sample jpg file.

Download a sample jpg file from internet and save it as sample.jpg file. Next encode and decode the file using the following commands

\$ openssl enc -e -base64 -in sample.jpg -out sample.base64

\$ openssl enc -d -base64 -in sample.base64 -out sample-decode.jpg

FIT3031 Lab Exercise 02: OpenSSL utility – Symmetric/Asymmetric Encryption

Base64 encoding is a standard method for converting 8-bit binary information into a limited subset of ASCII characters for safe transport through e-mail systems, and other systems that are not 8-bit safe. With OpenSSL, it is very easy to encode and decode Base64 data:

2. Using OpenSSL and base64 algorithm encode and decode any text file.

```
$openssl enc -base64 -in myfile.txt -out myfile.b64
```

```
$openssl enc -d -base64 -in myfile.b64 -out myfile.decrypt.txt
```

III. Cryptographic Hashing Functions

Hash functions can be generated using openssl utility. This will check to see if the file has not been tampered with? One simple way to do this is to generate a cryptographic hashing function. This will give you a fixed-length string (called a message digest) given an input file of any length. SHA-1 SHA-256 and RIPE-MD160 are considered current; MD-5 is considered outdated.

1. List the command line for creating a message digest for a simple plaintext file using the cryptographic hashing function such as SHA-1, RIPE-MD160 and MD5?

```
$ openssl dgst -sha1 -c plaintext.txt
```

```
$ openssl dgst -ripemd160 -c plaintext.txt
```

```
$ openssl dgst -md5 -c plaintext.txt
```

You can also see that the md5sum utility that is shipped with most GNU/Linux/debian distributions returns the same value as the openssl md5 message digest:

2. List the command line for creating a message digest for a simple text file using the built-in md5sum hashing function command?

```
$ md5sum plaintext.txt
```

The OpenSSL dgst (message digest/hashing) command also has numerous options for signing digests, verifying signatures, etc.

FIT3031 Lab Exercise 02: OpenSSL utility – Symmetric/Asymmetric Encryption

3. List the command-lines for encrypting and decrypting the message digest created in the previous step.

```
$ openssl enc -des3 -e -in MD5.txt -out MD5.enc  
$ openssl enc -des3 -d -in MD5.enc -out MD5.decrypt.txt
```

4. List the command-line for comparing the decrypted message digest file with the original message digest file.

```
$ openssl dgst -md5 MD5.txt MD5.decrypt.txt
```

IV. Asymmetric encryption.

1. For Asymmetric encryption, using openssl List the command-lines for encrypting and decrypting a plaintext file.

Hint: Now, for Asymmetric encryption you must first generate your private key and extract the public key.

```
$ openssl genrsa -out private.key 1024  
$ openssl rsa -in private.key -pubout > public.key
```

To encrypt:

```
$ openssl rsautl -encrypt -pubin -inkey public.key -in plaintext.txt -out encrypted.txt
```

To decrypt:

```
$ openssl rsautl -decrypt -inkey private.key -in encrypted.txt -out plaintext.txt
```