Asymmetric Key Operations with RSA and OpenSSL

Note:

The following commands should be executed in the OpenSSL window. You must open an OpenSSL session, which has the command prompt like the following:

OpenSSL> *[then type all commands here]*

RSA Private Key Generation

Link to Video:

https://drive.google.com/file/d/1tBALzGylT8THdvcFlMbNngWWr4A9z6No/view?usp=sharing

Generate 2048-bit RSA private/public key pairs using the following command. The private key is stored in **privateKey.pem**.

```
genrsa -out privateKey.pem 2048
```

You should get the following outputs in the console:

Generating RSA private key, 2048 bit long modulus .....+++
.....+++
e is 65537 (0x010001)

A file named **privateKey.pem** is generated, which is the private key.

If you want to see the content of the private key file, use the following command

```
rsa -in privateKey.pem -text -noout
```

You will see parameters such as modulus, public and private exponent, the prime numbers…. all the parameters required for RSA algorithm to work.

RSA Public Key Generation

In your OpenSSL session, type the following two commands, one-by-one, to generate the public key:

```
rsa -in privateKey.pem -pubout -out publicKey.pem
```

```
rsa -in publicKey.pem -pubin -text -noout
```

A file called publicKey.pem is generated, which is the public-key.

See next page for more information about encryption and decryption using RSA and OpenSSL…

OpenSSL command to encrypt a file using an RSA public key

Link to Video:

https://drive.google.com/file/d/1cO8oSQNCWwptT34SqYIwubr9yjRRJT7w/view?usp=sharing

Assumptions:

- You have access to an RSA public key file named publicKey.pem (see the key generation step above)
- The commands used in this section should be executed in an OpenSSL window. You must open an OpenSSL session, which has the command prompt like the following:

  `OpenSSL>` *[then type all commands here]*

- You have a plain-text file, named **`textFile.txt`**, with your name and student ID in that file. The may look like the followings:

`textFile.txt`

Student ID: S1234567 *Name: ABCDEF*

Encrypt the plain-text file **`textFile.txt`** with the public key publicKey.pem using the following command:

```
rsautl -in textFile.txt -out ciphertext.txt -pubin -inkey publicKey.pem –encrypt
```

An encrypted file named ciphertext.txt is generated, which contains something similar to the followings:

```
a$s(˜UQWgŸõ®FÖSº;]H‡øFµˇdπ¬ëGx◆xÎãèägéL=€Î4•eÓΩ–
Të">P)´Ã≥qÖŸÎ¨Üvê,Økj.åÊ´ßR/
,2O∂3îYBOi(qmÊUä˜–«ˇ¬Ú/VUÈi…Tiñ∫3,à…ƒ°ˇ≠s Ò•
a@!flãú7iùQM[êl‰ƒf÷wÛÛ«ÂôgÄ¶˝7§b
üL5Ù°Åê◊;ê^∑≥ãŸ5ÖœóVåK¨Ø¯‡H>k‹≠–
¬VP·Q¨µ∫zj˛áπÂZ•‡öÍ,\IüKÇQn~LÁ˙Ïˇ·∞æ/Yà:∗Beƒ'≈=!&
```

OpenSSL command to decrypt a ciphertext file using an RSA private key

Link to Video:

https://drive.google.com/file/d/1BnI1eRLi1qBuuFvvFm5-do9fnBrUu0Jf/view?usp=sharing

Assumptions:

- You have access to an RSA private key file named privateKey.pem (see the key generation step above).
- The commands used in this section should be executed in an OpenSSL window. You must open an OpenSSL session, which has the command prompt like the following:

  `OpenSSL>` *[then type all commands here]*

- You have a cipher-text file, named ciphertext.txt (see the encryption step above).

Use the following command to decrypt the ciphertext.txt and store the output in a file named decryptedText.txt:

rsautl -in ciphertext.txt -out decryptedText.txt -inkey privateKey.pem -decrypt
If the decryption is successful, the content of decryptedText.txt must be exactly the same as the content of the textFile.txt (see the encryption step above).

`decryptedText.txt`

Student ID: S1234567 *Name: ABCDEF*