1 Thông tin cá nhân

- Họ và tên: Trịnh Lê Nguyên Vũ
- MSSV: 20120630

2 Báo cáo tìm hiểu về khóa RSA của OpenSSL

- Thông tin về mã nguồn:
  - Ngôn ngữ lập trình: Python
  - Thư viện cần cài đặt: `cryptography`
    - `pip install cryptography`
- Video demo: https://youtu.be/44TChSIQBaQ

(keysize numbit = 512)

priv.pem (Khóa Bí Mật): RSA Private Key thông thường được lưu dưới dạng base64.

priv.pem

```
-----BEGIN PRIVATE KEY-----
MIIBVQIBADANBgkqhkiG9w0BAQEFAASCAT8wggE7AgEAAkEAxXL7M/7ZIBTEI/gJ
yxXTTxPYM3y5Z3Ev59i3vAYlKS1HRWZ/xnPrP0TnZEQh9ijBZjfqwotDSnYY5pe3
kwGYxwIDAQABAkAKLiJ7jNE/ETMHs7s1iFuLwTcteGnKxpA0wPzAKgTyeXbtWCo/
Vs6ecttKlejWSCaQPJmc3s1KuXq5Ih+NpfrZAiEA8LdMPX1t3IbPM8rqP7j6nvXx
t/wQm38tkWyzUGsNZ6sCIQDR/Ge3+zY+z2Ewkqs4Xa7dsWrNhYUFxmBI2arFkqmH
VQIgRh/f4fnsS1Yqfpgabb3hpPVZGZQg8mu2Rqs4AbFuEkkCIQCy04VwxVtY+jsG
pqFyCzZUTjdqd37lgB+XaudoApKlUQIhAIkWmbnHUXX/FN9SONSWK0P0T5RTG+U+
QxG+VmTaM3jg
-----END PRIVATE KEY-----
```


```
-----BEGIN PRIVATE KEY-----
MIIBVQIBADANBgkqhkiG9w0BAQEFAASCAT8wggE7AgEAAkEAxXL7M/7ZIBTEI/gJ
yxXTTxPYM3y5Z3Ev59i3vAYlKS1HRWZ/xnPrP0TnZEQh9ijBZjfqwotDSnYY5pe3
kwGYxwIDAQABAkAKLiJ7jNE/ETMHs7s1iFuLwTcteGnKxpA0wPzAKgTyeXbtWCo/
Vs6ecttKlejWSCaQPJmc3s1KuXq5Ih+NpfrZAiEA8LdMPX1t3IbPM8rqP7j6nvXx
t/wQm38tkWyzUGsNZ6sCIQDR/Ge3+zY+z2Ewkqs4Xa7dsWrNhYUFxmBI2arFkqmH
VQIgRh/f4fnsS1Yqfpgabb3hpPVZGZQg8mu2Rqs4AbFuEkkCIQCy04VwxVtY+jsG
pqFyCzZUTjdqd37lgB+XaudoApKlUQIhAIkWmbnHUXX/FN9SONSWK0P0T5RTG+U+
QxG+VmTaM3jg
-----END PRIVATE KEY-----
```


- Bắt đầu bằng dòng "-----BEGIN PRIVATE KEY-----" và kết thúc bằng "-----END PRIVATE KEY-----".

- Bao gồm các thành phần sau:

Trường			Giá trị
Version			0
Private Key Algorithm Identifier	Rsa Encryption, PKCS #1		1.2.840.113549.1.1.1
	Optional attribute		null
Private Key Wrapper	RSA Private Key	version	0
		Modulus (n)	10341250857561626815012868438579500288909464517570309965157235614824789334981412695340890432885393030108967204617047508086285394454575081916959586538068167
		Public Exponent (e)	65537
		Private Exponent (d)	5331810526527112472027783153632319373212853900067149453326563489707030099470757688084072960382660910192558436035741821000061966714220578170060744282590 33
		Prime 1 (p)	1088789428633267111512870211063908173178517831261361437752300765823308407377 07
		Prime 2 (q)	9497934665422648798995050752041064866686485950548167422471033308897519871778 1
		Exponent 1 ($d \bmod (p-1)$)	3171821684158007796054704681268922111528489621807820402149461559286907260576 9
		Exponent 2 ($d \bmod (q-1)$)	8088541275105190575850462678684039738460810746173830358163378107575695852475 3
		Coefficient ($q^{-1} \bmod p$)	6200679186710446916377662591539988092629298364700692843414943381436656802019 2

- Cách sử dụng chương trình:
    - Sinh khoá bằng OpenSSL:

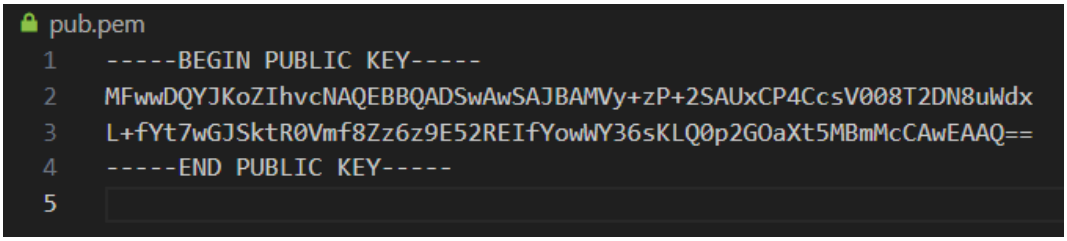
```
openssl genpkey -out priv.pem -algorithm RSA -pkeyopt rsa_keygen_bits:512

openssl pkey -in priv.pem -out pub.pem -pubout
```

    - In các thành phần trong khoá ra màn hình:

```
python read_components.py priv.pem pub.pem
```

pub.pem (Khóa Công Khai): RSA Public Key cũng được lưu dưới định dạng PEM.




```
-----BEGIN PUBLIC KEY-----
MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBAMVy+zP+2SAUxCP4CcsV008T2DN8uWdx
L+fYt7wGJSktR0Vmf8Zz6z9E52REIfYowWY36sKLQ0p2GOaXt5MBmMcCAwEAAQ==
-----END PUBLIC KEY-----
```

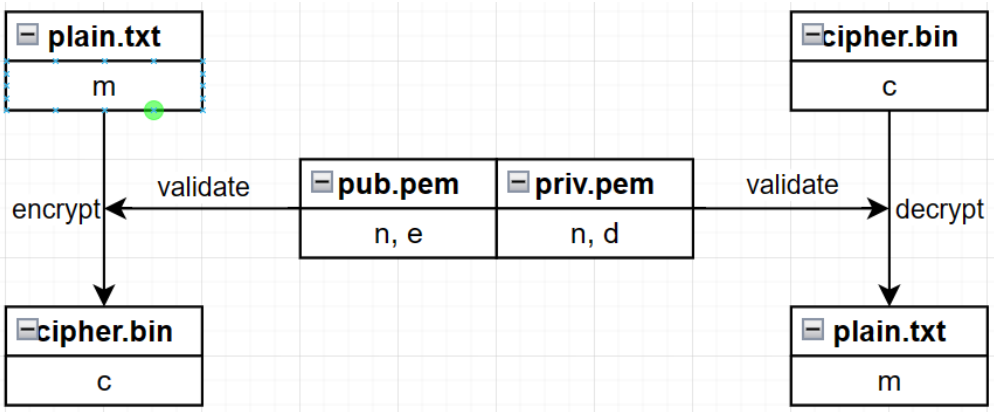
- Bắt đầu bằng dòng "-----BEGIN PUBLIC KEY-----" và kết thúc bằng "-----END PUBLIC KEY-----".
- Bao gồm các thành phần sau:

Trường			Giá trị
RSA Encryption	OID		1.2.840.113549.1.1.1
	Optional parameter		null
Public key Bit String	Public Key components	version	0
		Modulus (n)	10341250857561626815012868438579500288909464517570309965157235614824789334981412695340890432885393030108967204617047508086285394454575081916959586538068167
		Public Exponent (e)	65537

3 Báo cáo tìm hiểu về mã hóa khóa công khai RSA của OpenSSL

- Thông tin về mã nguồn:
  - Ngôn ngữ lập trình: Python
  - Thư viện cần cài đặt: `cryptography`
    - `pip install cryptography`
- Video demo: https://youtu.be/Ef-gAIckU1s
- Cách OpenSSL mã hoá và giải mã:
  - Mã hoá $(m \to c)$:
    - OpenSSL sẽ kiểm tra tính hợp lệ của các giá trị trong file *pub.pem*
    - OpenSSL sẽ sử dụng cặp giá trị (*n*, *e*) trong file *pub.pem* để mã hoá văn bản *m* trong file *plain.txt* theo công thức
    - $c = m^e \bmod n$
  - Giải mã $(c \to m)$:
    - OpenSSL sẽ kiểm tra tính hợp lệ của các giá trị trong file *priv.pem*
    - OpenSSL sẽ sử dụng cặp giá trị (*n*, *d*) trong file *priv.pem* để giải mã bản mã *c* trong file *cipher.bin* theo công thức
      - $m = c^d \bmod n$

    Với (*e*, *d*) thoả $ed \bmod (p-1)(q-1) = 1$ và $pq = n$
  - Sơ đồ:



- Cách sử dụng chương trình:
    - Mã hoá bằng OpenSSL:

```
openssl pkeyutl -in plain_0.txt -out cipher.bin -inkey pub.pem -pubin -encrypt
```

    - Giải mã bằng OpenSSL:

```
openssl pkeyutl -in cipher.bin -out plain_1.txt -inkey priv.pem -decrypt
```

    - Mã hoá bằng chương trình Python:

```
python encrypt.py plain_0.txt cipher.bin pub.pem
```

    - Giải mã bằng chương trình Python:

```
python decrypt.py cipher.bin plain_1.txt priv.pem
```

4 Báo cáo tìm hiểu về chữ ký điện tử RSA của OpenSSL

- Thông tin về mã nguồn:
  - Ngôn ngữ lập trình: Python
  - Thư viện cần cài đặt: `cryptography`
    - `pip install cryptography`
- Video demo: https://youtu.be/uKesmv8nLyc
- Cách OpenSSL ký và xác thực các tệp tin:
  - Ký $(m \to s)$: OpenSSL sẽ sử dụng cặp giá trị (*n*, *d*) trong file *priv.pem* để mã hoá văn bản *m* trong file *mess.txt* theo công thức:
    - $s = m^d \bmod n$
  - Xác thực $(s \to m)$: OpenSSl sẽ sử dụng cặp giá trị (*n*, *e*) *pub.pem* để giải mã chữ ký s trong file *signature.bin* để so sánh với văn bản gốc theo công thức:
    - $m = s^e \bmod n$
- Bằng cách này, OpenSSL có thể xác thực được văn bản gốc đã không bị biến đổi, vẫn giữ được toàn vẹn nội dung hay không.
- Cách sử dụng chương trình:
  - Ký bằng OpenSSL:

```
openssl dgst -sha256 -sign priv.pem -out signature.bin mess.txt
```

  - Xác thực bằng OpenSSL:

```
openssl dgst -sha256 -verify pub.pem -signature signature.bin mess.txt
```

  - Ký bằng chương trình Python:

```
python sign.py mess.txt signature.bin priv.pem
```

  - Xác thực bằng chương trình Python:

```
python verify.py  mess.txt signature.bin pub.pem
```