**MSSV: 21127684**

**Tên: Kuo Yung Sheng**

**Mục lục**

[**2 Tìm hiểu về khóa RSA của OpenSSL** 1](#_Toc152194912)

[**Private key** 1](#_Toc152194913)

[**Public key** 2](#_Toc152194914)

[**Chương trình** 2](#_Toc152194915)

[**3.** **Tìm hiểu về mã hóa khóa công khai RSA của OpenSSL** 2](#_Toc152194916)

[**Cách OpenSSL sử dụng key để mã hóa và giải mã các tệp tin.** 2](#_Toc152194917)

[**Chương trình encrypt** 3](#_Toc152194918)

[**Chương trình decrypt** 3](#_Toc152194919)

[**4. Tìm hiểu về chữ ký điện tử RSA của OpenSSL** 4](#_Toc152194920)

[**Mô tả cách OpenSSL để ký và xác thực các tệp tin.** 4](#_Toc152194921)

[**Chương trình sign** 5](#_Toc152194922)

[**Chương trình verify** 5](#_Toc152194923)

# **2 Tìm hiểu về khóa RSA của OpenSSL**

## **Private key**

-----BEGIN RSA PRIVATE KEY-----

(base64-encoded data) : Là dữ liệu khóa công khai được mã hóa dưới định dạng base64.

-----END RSA PRIVATE KEY-----

Base64 encode bao gồm các thông tin sau:

version Version,

modulus INTEGER, -- n

publicExponent INTEGER, -- e

privateExponent INTEGER, -- d

prime1 INTEGER, -- p

prime2 INTEGER, -- q

exponent1 INTEGER, -- d mod (p-1)

exponent2 INTEGER, -- d mod (q-1)

coefficient INTEGER, -- (inverse of q) mod p

otherPrimeInfos OtherPrimeInfos OPTIONAL

Khóa riêng tư chứa các thông tin như phiên bản của khóa, thông tin về thuật toán (thường là RSA), số nguyên lớn biểu diễn modulo (n), số nguyên lớn biểu diễn khóa riêng tư (d), và các thông tin khác như chỉ mục và các tham số khác.

## **Public key**

-----BEGIN PUBLIC KEY-----: Đánh dấu bắt đầu của phần dữ liệu khóa công khai.

(base64-encoded data): Là dữ liệu khóa công khai được mã hóa dưới định dạng base64.

-----END PUBLIC KEY-----: Đánh dấu kết thúc của phần dữ liệu khóa công khai.

Khóa công khai chứa thông tin tương tự như khóa riêng tư, bao gồm phiên bản, thuật toán, modulo (n), và số nguyên lớn biểu diễn khóa công khai (e).

## **Chương trình**

Ngôn ngữ: python

Thư viện: cryptography

Cài đặt: pip install cryptography

Cách thức chạy:

Usage: python extract.py --pub <filename>.pem

--pub: public key

--priv: private key

Code: extract.py trong thư mực source

Video demo: Read\_info\_from\_key.mp4 trong thư mục Demo

# **3.** **Tìm hiểu về mã hóa khóa công khai RSA của OpenSSL**

## **Cách OpenSSL sử dụng key để mã hóa và giải mã các tệp tin.**

Trong quá trình mã hóa, dữ liệu được mã hóa bằng Modulus (n) và Exponent (e).

1. Dữ liệu được chia thành các khối có kích thước phù hợp với độ dài khóa RSA.
2. Mỗi khối dữ liệu được đặt trong một cấu trúc chuẩn hóa gọi là PKCS#1 v1.5 padding.
3. Padding chứa các thông tin như loại padding, chuỗi ngẫu nhiên và một byte giá trị 0x00 giữa các phần khác nhau.
4. Sau đó, khối dữ liệu được mã hóa bằng cách sử dụng khóa công khai của người nhận và thuật toán RSA.

Giải mã

1. Người nhận sử dụng khóa bí mật (private key) của mình để giải mã các khối dữ liệu.
2. Mỗi khối sau khi giải mã được xử lý để loại bỏ padding và thu được dữ liệu ban đầu.

Chú thích:

Thông điệp được mã hóa khá nhỏ, nếu lớn hơn một mức nào đó sẽ báo lỗi.

Mỗi lần mã hóa file cipher.txt sẽ khác nhau.

A diagram of a key system

Description automatically generated

A screenshot of a computer

Description automatically generated

Ảnh decrypte trên cyberchef

Có 3 chế độ thì để giải mã ở đây em thấy dùng loại encryption scheme: RSAES-PKCS1-V1 5 thì giải mã được. Vậy trong code cũng sẽ dùng loại này để decrypte tương tự.

## **Chương trình encrypt**

Ngôn ngữ: python

Thư viện: cryptography

Cài đặt: pip install cryptography

Cách thức chạy

python encrypt.py <file plain text> <file public key> <file output cipher text>

vd: python encrypt.py plain.txt pub.pem cipher.txt

Video demo: Encrypt.mp4 trong thư mục Demo

## **Chương trình decrypt**

Ngôn ngữ: python

Thư viện: cryptography

Cài đặt: pip install cryptography

Cách thức chạy

python decrypt.py <file cipher> <file private key> <file output>

vd: python decrypt.py cipher.txt priv.pem decoded.txt

Video demo: Decrypt.mp4 trong thư mục Demo

# **4. Tìm hiểu về chữ ký điện tử RSA của OpenSSL**

## **Mô tả cách OpenSSL để ký và xác thực các tệp tin.**

Tạo chữ ký số (digital signature) cho dữ liệu trong tệp plain.txt bằng cách sử dụng khóa bí mật. Dưới đây là các bước cụ thể mà OpenSSL thực hiện trong quá trình này:

1. Load Khóa Bí Mật (Private Key):

OpenSSL sẽ đọc nội dung của tệp priv.pem để lấy khóa bí mật (private key) được sử dụng cho quá trình tạo chữ ký.

1. Hashing (Băm) Dữ Liệu:

OpenSSL sẽ sử dụng một hàm băm (hash function), thường là SHA-256 hoặc một thuật toán băm khác được chỉ định, để tạo ra một giá trị băm (digest) của dữ liệu trong tệp plain.txt.

1. Tạo Chữ Ký Số (Digital Signature):

Khóa bí mật sẽ được sử dụng để ký giá trị băm, tạo ra chữ ký số cho dữ liệu.

Quy trình này thường sử dụng một hàm ký số đặc biệt, phù hợp với thuật toán chữ ký số được chọn. Trong trường hợp này, thuật toán thường là RSA.

1. Ghi Chữ Ký Số Ra Tệp sign.txt:

Chữ ký số được tạo ra sẽ được ghi vào tệp sign.txt để có thể sử dụng sau này để xác thực nguồn gốc của dữ liệu.

Thực hiện quá trình xác thực chữ ký số (digital signature) cho dữ liệu trong tệp plain.txt bằng cách sử dụng khóa công khai. Dưới đây là các bước cụ thể mà OpenSSL thực hiện trong quá trình này:

1. Load Khóa Công Khai (Public Key):

OpenSSL sẽ đọc nội dung của tệp pub.pem để lấy khóa công khai (public key) được sử dụng cho quá trình xác thực chữ ký số.

1. Đọc Chữ Ký Số Từ Tệp sign.txt:

OpenSSL sẽ đọc chữ ký số từ tệp sign.txt để sử dụng trong quá trình xác thực.

1. Hashing (Băm) Dữ Liệu:

OpenSSL sẽ sử dụng cùng một hàm băm (hash function) mà nó đã sử dụng để tạo chữ ký số, để tạo ra một giá trị băm (digest) của dữ liệu trong tệp plain.txt.

1. Xác Thực Chữ Ký Số (Digital Signature):

Khóa công khai sẽ được sử dụng để xác thực chữ ký số cho giá trị băm của dữ liệu.

Quy trình này thường sử dụng một hàm xác thực đặc biệt, phù hợp với thuật toán chữ ký số được chọn (trong trường hợp này, có thể là RSA).

1. Kết Quả Xác Thực:

Nếu quá trình xác thực thành công, OpenSSL sẽ xuất hiện thông báo "Verified OK". Điều này có nghĩa là chữ ký số đã được xác thực thành công, và dữ liệu trong tệp plain.txt không bị thay đổi và đến từ nguồn gốc được xác định.

Tạo chữ ký số cho dữ liệu trong tệp tin, giúp đảm bảo tính toàn vẹn và xác thực của dữ liệu. Chữ ký số có thể được sử dụng sau này để xác nhận rằng dữ liệu không bị thay đổi và đến từ nguồn gốc được xác định.

Tham khảo

<https://datatracker.ietf.org/doc/html/rfc3447#section-3>

<https://www.programcreek.com/python/?CodeExample=load%20pem%20private%20key>

<https://chromium.googlesource.com/chromiumos/platform/ec/+/v1.8.0/util/pem_extract_pubkey.py>

<https://medium.com/rahasak/public-key-cryptography-with-openssl-4909ea423e67>

<https://cryptobook.nakov.com/asymmetric-key-ciphers/rsa-encrypt-decrypt-examples> (encrypte the kiểu OAEP)

<https://gist.github.com/joongh/8ff8e31ee1a2a9585a5a> (code mẫu)