1. Tìm hiểu về khóa RSA của OpenSSL
2. Báo cáo
3. Giới Thiệu

OpenSSL là một thư viện mã nguồn mở cung cấp nhiều chức năng liên quan đến mật mã hóa và bảo mật mạng. Trong quá trình tạo ra các cặp khóa bảo mật, OpenSSL tạo ra hai loại tệp tin quan trọng: một tệp chứa khóa bí mật (priv.pem) và một tệp chứa khóa công khai (pub.pem). Báo cáo này sẽ mô tả cấu trúc của các tệp này và giải thích ý nghĩa của từng thành phần.

2. Cấu Trúc của Tệp Chứa Khóa Bí Mật (priv.pem)

Tệp chứa khóa bí mật (priv.pem) được tạo ra bởi OpenSSL theo định dạng PEM (Privacy Enhanced Mail). Dưới đây là cấu trúc cơ bản của tệp priv.pem:

* Tiêu Đề (Header):

Bắt đầu bằng chuỗi ký tự: "-----BEGIN PRIVATE KEY-----".

Chứa thông tin về loại khóa và các tham số liên quan.

* Dữ Liệu Khóa (Key Data):

Chứa thông tin chi tiết về khóa bí mật, thuật toán mã hóa, và các thông số khác.

* Chân Trang (Footer):

Kết thúc bằng chuỗi ký tự: "-----END PRIVATE KEY-----".

3. Cấu Trúc của Tệp Chứa Khóa Công Khai (pub.pem)

Tệp chứa khóa công khai (pub.pem) cũng tuân theo định dạng PEM và có cấu trúc tương tự như tệp chứa khóa bí mật:

* Tiêu Đề (Header):

Bắt đầu bằng chuỗi ký tự: "-----BEGIN PUBLIC KEY-----".

* Dữ Liệu Khóa (Key Data):

Cũng biểu diễn dưới dạng Base64.

Chứa thông tin về khóa công khai, thuật toán mã hóa, và các thông số khác.

* Chân Trang (Footer):

Kết thúc bằng chuỗi ký tự: "-----END PUBLIC KEY-----".

4. Ý Nghĩa Các Thành Phần

* Tiêu Đề: Xác định loại khóa và định dạng của dữ liệu trong tệp.
* Dữ Liệu Khóa: Chứa thông tin cụ thể về khóa, bao gồm cả thông số thuật toán và giá trị của khóa.
* Chân Trang: Đánh dấu kết thúc của phần dữ liệu, giúp ứng dụng có thể nhận biết nơi kết thúc của mỗi tệp.

1. Chương trình:

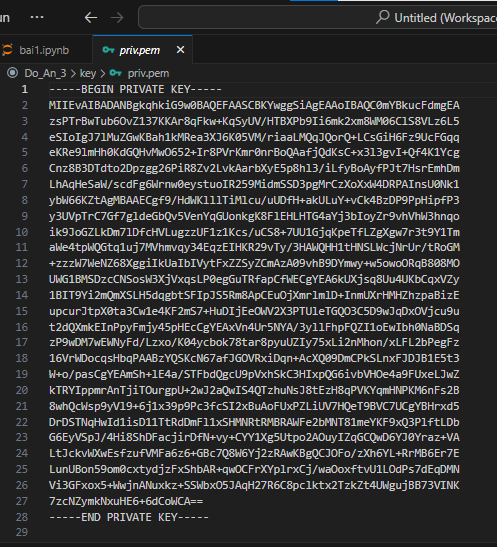
* Ngôn ngữ lập trình: Python
* Các thư viện cần cài đặt: crytography



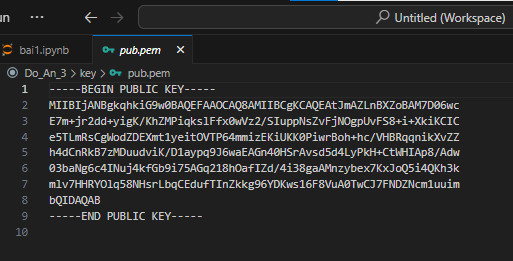
* Cách thức biên dịch, cách thức chạy:

+ Trước tiên tạo 2 file priv.pem và pub.pem bằng openssl

Lệnh tạo file priv.pem 



Lệnh tạo file pub.pem 



+ Khai báo 1 hàm read\_key\_file  dùng để đọc file mã hóa

và 1 hàm display\_key\_components  để hiển thị các thành phần có trong các tệp đó đảm bảo tính hợp lý để tạo thành một bộ khóa RSA hợp lệ.

1. Video demo: <https://youtu.be/XBbhYewwOr8>
2. Tìm hiểu về mã hóa khóa công khai RSA của OpenSSL
3. Báo cáo mô tả cách OpenSSL sử dụng các tệp chứa khóa bí mật và khóa công khai để mã hóa và giải mã các tệp tin

* Xác Định Khóa:

OpenSSL xác định khóa cần sử dụng dựa trên lệnh được cung cấp và loại khóa (bí mật hoặc công khai).

* Mã Hóa và Giải Mã:

Khi mã hóa, OpenSSL sử dụng thuật toán RSA và khóa công khai để mã hóa dữ liệu.

Khi giải mã, OpenSSL sử dụng thuật toán RSA và khóa bí mật để giải mã dữ liệu đã được mã hóa trước đó.

- Khóa Công Khai:

- Đọc khóa công khai từ tệp pub.pem

Mã Hóa:

- Đọc dữ liệu cần mã hóa từ tệp plain.txt

- Sử dụng khóa công khai để mã hóa dữ liệu

- Lưu kết quả vào tệp cipher.txt

Khóa Bí Mật:

- Đọc khóa bí mật từ tệp priv.pem

Giải Mã:

- Đọc dữ liệu đã được mã hóa từ tệp cipher.txt

- Sử dụng khóa bí mật để giải mã dữ liệu

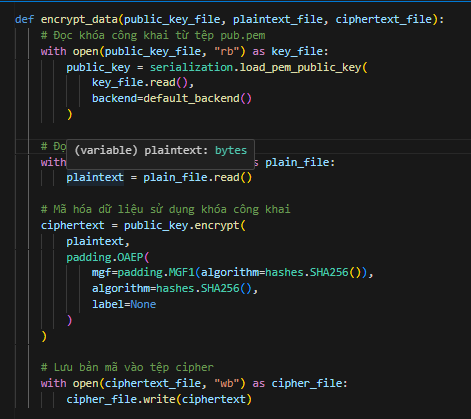
- Lưu kết quả vào tệp plain.txt

1. Chương trình

- Ngôn ngữ lập trình: Python

- Các thư viện cần cài đặt: crytography

1. Chương trình mã hóa đọc tệp chứa khóa công khai và tệp chứa bản rõ , xuất ra tệp chứa bản mã sau khi mã hóa



* Hàm encrypt\_data:

Nhận vào đường dẫn của tệp chứa khóa công khai (pub.pem), tệp chứa bản rõ (plain), và tệp sẽ chứa bản mã sau khi mã hóa (cipher).

Mở và đọc khóa công khai từ tệp pub.pem.

Mở và đọc bản rõ từ tệp plain.

Sử dụng khóa công khai để mã hóa bản rõ sử dụng thuật toán OAEP với hàm băm SHA-256.

Lưu bản mã vào tệp cipher.

* Đọc Khóa Công Khai (pub.pem):

Mở tệp chứa khóa công khai (pub.pem) và đọc nó.

serialization.load\_pem\_public\_key: Chuyển đổi dữ liệu khóa công khai từ định dạng PEM sang đối tượng khóa công khai có thể sử dụng.

* Đọc Bản Rõ (plain):

Mở tệp chứa bản rõ (plain) và đọc nó.

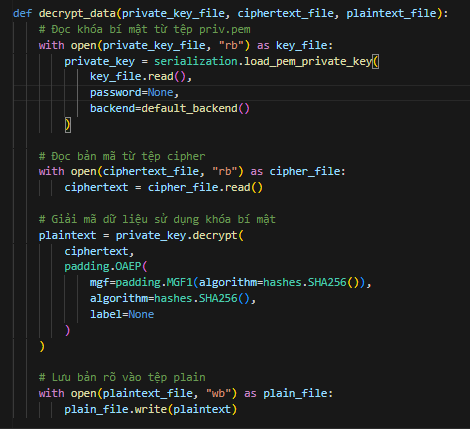
* Mã Hóa Dữ Liệu:

public\_key.encrypt: Sử dụng khóa công khai để mã hóa bản rõ sử dụng thuật toán OAEP với hàm băm SHA-256.

* Lưu Bản Mã (cipher):

Mở tệp sẽ chứa bản mã sau khi mã hóa (cipher) và ghi bản mã vào đó.

1. Chương trình giải mã đọc tệp chứa khóa bí mật và tệp chứa bản mã , xuất ra tệp chứa bản rõ sau khi giải mã .



* Hàm decrypt\_data:

Nhận vào đường dẫn của tệp chứa khóa bí mật (priv.pem), tệp chứa bản mã (cipher), và tệp sẽ chứa bản rõ sau khi giải mã (plain).

Mở và đọc khóa bí mật từ tệp priv.pem.

Mở và đọc bản mã từ tệp cipher.

Sử dụng khóa bí mật để giải mã bản mã sử dụng thuật toán OAEP với hàm băm SHA-256.

Lưu bản rõ vào tệp plain.

* Đọc Khóa Bí Mật (priv.pem):

Mở tệp chứa khóa bí mật (priv.pem) và đọc nó.

serialization.load\_pem\_private\_key: Chuyển đổi dữ liệu khóa bí mật từ định dạng PEM sang đối tượng khóa bí mật có thể sử dụng.

* Đọc Bản Mã (cipher):

Mở tệp chứa bản mã (cipher) và đọc nó.

* Giải Mã Dữ Liệu:

private\_key.decrypt: Sử dụng khóa bí mật để giải mã bản mã sử dụng thuật toán OAEP với hàm băm SHA-256.

* Lưu Bản Rõ:

Mở tệp sẽ chứa bản rõ sau khi giải mã (plain) và ghi bản rõ vào đó.

1. Video demo: <https://youtu.be/yNJM8dgjftk>
2. Tìm hiểu về chữ ký điện tử RSA của OpenSSL
3. Báo cáo

* Xác Định Khóa:

OpenSSL xác định khóa cần sử dụng dựa trên lệnh được cung cấp và loại khóa (bí mật hoặc công khai).

* Ký và Xác Thực:

Khi ký, OpenSSL sử dụng thuật toán RSA và khóa bí mật để tạo chữ ký số cho dữ liệu.

Khi xác thực, OpenSSL sử dụng thuật toán RSA và khóa công khai để kiểm tra tính hợp lệ của chữ ký số.

Khóa Bí Mật:

- Đọc khóa bí mật từ tệp priv.pem

Ký:

- Đọc dữ liệu cần ký từ tệp mess.txt

- Sử dụng khóa bí mật để tạo chữ ký số cho dữ liệu

- Lưu chữ ký số vào tệp sign.txt

Khóa Công Khai:

- Đọc khóa công khai từ tệp pub.pem

Xác Thực:

- Đọc dữ liệu cần xác thực từ tệp mess.txt

- Đọc chữ ký số từ tệp sign.txt

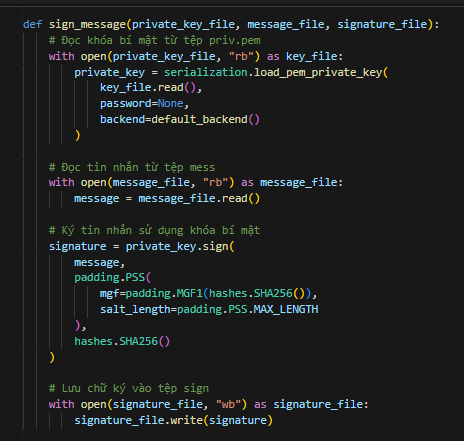
- Sử dụng khóa công khai để xác thực chữ ký số cho dữ liệu

1. Chương trình

- Ngôn ngữ lập trình: Python

- Các thư viện cần cài đặt: crytography

1. Chương trình ký đọc tệp chứa khóa bí mật và tệp chứa tin nhắn cần ký , xuất ra tệp chứa chữ ký .



* Hàm sign\_message:

Nhận vào đường dẫn của tệp chứa khóa bí mật (priv.pem), tệp chứa tin nhắn cần ký (mess), và tệp sẽ chứa chữ ký (sign).

Mở và đọc khóa bí mật từ tệp priv.pem.

Mở và đọc tin nhắn từ tệp mess.

Sử dụng khóa bí mật để ký tin nhắn sử dụng thuật toán PSS với hàm băm SHA-256.

Lưu chữ ký vào tệp sign.

* Đọc Khóa Bí Mật (priv.pem):

Mở tệp chứa khóa bí mật (priv.pem) và đọc nó.

serialization.load\_pem\_private\_key: Chuyển đổi dữ liệu khóa bí mật từ định dạng PEM sang đối tượng khóa bí mật có thể sử dụng.

* Đọc Tin Nhắn (mess):

Mở tệp chứa tin nhắn cần ký (mess) và đọc nó.

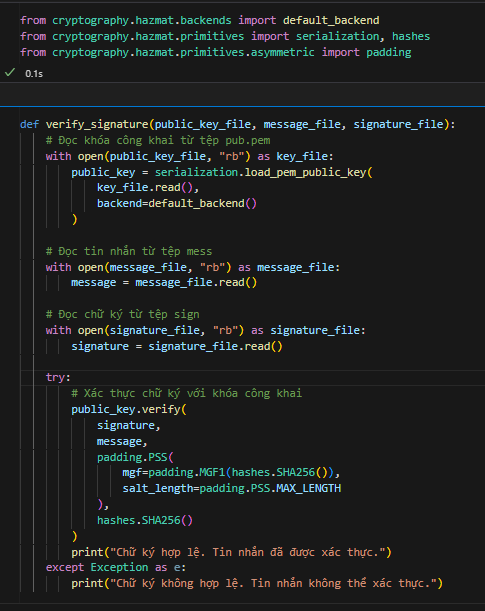
* Ký Tin Nhắn:

private\_key.sign: Sử dụng khóa bí mật để ký tin nhắn sử dụng thuật toán PSS với hàm băm SHA-256.

* Lưu Chữ Ký (sign):

Mở tệp sẽ chứa chữ ký (sign) và ghi chữ ký vào đó.

1. Chương trình xác thực đọc tệp chứa khóa công khai , tệp chứa tin nhắn , và tệp chứa chữ ký , xuất ra câu trả lời tin nhắn được xác thực bởi chữ ký hay không.



* Hàm verify\_signature:

Nhận vào đường dẫn của tệp chứa khóa công khai (pub.pem), tệp chứa tin nhắn cần xác thực (mess), và tệp chứa chữ ký (sign).

Mở và đọc khóa công khai từ tệp pub.pem.

Mở và đọc tin nhắn từ tệp mess.

Mở và đọc chữ ký từ tệp sign.

Sử dụng khóa công khai để xác thực chữ ký sử dụng thuật toán PSS với hàm băm SHA-256.

* Đọc Khóa Công Khai (pub.pem):

Mở tệp chứa khóa công khai (pub.pem) và đọc nó.

serialization.load\_pem\_public\_key: Chuyển đổi dữ liệu khóa công khai từ định dạng PEM sang đối tượng khóa công khai có thể sử dụng.

* Đọc Tin Nhắn (mess):

Mở tệp chứa tin nhắn cần xác thực (mess) và đọc nó.

* Đọc Chữ Ký (sign):

Mở tệp chứa chữ ký (sign) và đọc nó.

* Xác Thực Chữ Ký:

public\_key.verify: Sử dụng khóa công khai để xác thực chữ ký sử dụng thuật toán PSS với hàm băm SHA-256.

Nếu xác thực thành công, in ra "Chữ ký hợp lệ. Tin nhắn đã được xác thực.". Ngược lại, nếu có lỗi hoặc chữ ký không hợp lệ, in ra "Chữ ký không hợp lệ. Tin nhắn không thể xác thực.".

1. Video demo: <https://youtu.be/Fe-fx3gHBE4>