ĐỒ ÁN 03 – LÀM QUEN VỚI OPENSSL

November 27, 2023

1 Tóm tắt đồ án

OpenSSL là một trong nhiều hệ thư viện phần mềm được tạo ra nhằm cung cấp các công cụ mật mã cho các ứng dụng. Do đó, trong đồ án này, sinh viên sẽ tập làm quen với OpenSSL như một cách để hiểu các hệ mật mã được sử dụng trong thực tế như thế nào. Cụ thể, sinh viên sẽ thực hiện các yêu cầu sau:

- (4 điểm) Tìm hiểu về định dạng khóa RSA mà OpenSSL sinh ra.
- (3 điểm) Tìm hiểu về cách mã hóa và giải mã của OpenSSL đối với hệ mã RSA.
- (3 điểm) Tìm hiểu về cách ký và xác thực của OpenSSL đối với hệ mã RSA.

2 (4 điểm) Tìm hiểu về khóa RSA của OpenSSL

2.1 Mô tả

Để sinh khóa bí mật RSA bằng OpenSSL, ta dùng lệnh sau:

```
$ openssl genpkey -out <priv.pem> -algorithm RSA
                  -pkeyopt rsa_keygen_bits:<numbits>
```

Trong đó:

- `<priv.pem>` là tệp tin chứa khóa bí mật RSA mà OpenSSL sinh ra.
- `<numbits>` là kích thước khóa RSA (tức là số bit của N) mà OpenSSL sinh ra.

Khi đó, khóa bí mật mà OpenSSL sinh ra sẽ có dạng như dưới đây. Cần lưu ý rằng khóa bí mật dưới đây chỉ là ví dụ vì mỗi lần thực hiện câu lệnh, OpenSSL sẽ trả ra một khóa khác nhau.

```
-----BEGIN PRIVATE KEY-----
MIIBVgIBADANBgkqhkiG9w0BAQEFAASCAUAwggE8AgEAAkEAv1C/yropIICHb3DZ
bYVmNZeD4iBF9c8crGtGkAP4N9Z2WthmijYII7q6auRXPDiM3U9hWcNoEC5f0aCr
jV50JwIDAQABAkAoUGm53HSiJrNjZlJK49tf26es1MQpQhV2t7xMyKlbinZpWRNs
GNFOJaiTqopNEMGj11sdmCdTrDx9MzXzibaxAiEA4Vz/2Sx7CpMl86cjfmbaljpQ
HaL1GR03ri1q2VlijgOCIQDZUtSSKgHG5iGle/TKAs0GmtJ/bHKkmDaSBvuEvYly
AwIhAL2R1UWBvR5wGQSUG69AJa8o7it/4Fx3z1acrbyXC+ghAiEAn/7sQuOsRTu+
P6//qw3e1eL74BX+XRE289EyMSq9WvMCIQCPPuzhyhrLBdKW9qsOFQx8Ldf90PfH
dPPeKWJNXpSsHg==
-----END PRIVATE KEY-----
```

Sau khi có khóa bí mật, ta sẽ thực hiện việc sinh khóa công khai với OpenSSL, bằng lệnh sau:

```
$ openssl pkey -in <priv.pem> -out <pub.pem> -pubout
```

Trong đó:

- <priv.pem> là tệp tin chứa khóa bí mật RSA mà OpenSSL sinh ra.
- <pub.pem> là tệp tin chứa khóa công khai RSA mà OpenSSL sinh ra.

Khi đó, khóa công khai mà OpenSSL sinh ra sẽ có dạng như dưới đây. Cần lưu ý rằng khóa công khai dưới đây chỉ là ví dụ vì tương ứng với mỗi khóa bí mật, sẽ có một khóa công khai duy nhất.

```
-----BEGIN PUBLIC KEY-----
MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBAL9Qv8q6KSCAh29w2W2FZjWXg+IgRfXP
HKxrRpAD+DfWdlrYZoo2CCO6umrkVzw4jN1PYVnDaBAuX9Ggq41edCcCAwEAAQ==
-----END PUBLIC KEY-----
```

2.2 Yêu cầu

Trong phần này, sinh viên sẽ thực hiện các yêu cầu sau:

- Viết báo cáo mô tả cấu trúc của các tệp chứa khóa bí mật <priv.pem> và khóa công khai <pub.pem> mà OpenSSL sinh ra, từ đó cho biết các thành phần và ý nghĩa của chúng.
- Viết chương trình đọc các tệp chứa khóa bí mật <priv.pem> và khóa công khai <pub.pem> mà OpenSSL sinh ra, xuất ra màn hình các thành phần có trong các tệp đó. Các thành phần xuất ra cần đảm bảo tính hợp lý để tạo thành một bộ khóa RSA hợp lệ.
- Quay video demo cách sử dụng chương trình.

Lưu ý: Cần ghi rõ thông tin liên quan đến mã nguồn (chẳng hạn như loại ngôn ngữ lập trình, các thư viện cần cài đặt, cách thức biên dịch, cách thức chạy) và video demo (chẳng hạn như nơi tải lên) trong báo cáo.

3 (3 điểm) Tìm hiểu về mã hóa khóa công khai RSA của OpenSSL

3.1 Mô tả

1) Sau khi đã có khóa công khai RSA chứa trong tệp <pub.pem>, ta thực hiện việc mã hóa với OpenSSL như sau:

```
$ openssl pkeyutl -in <plain> -out <cipher> -inkey <pub.pem> -pubin -encrypt
```

Trong đó:

- <plain> là tệp chứa bản rõ.
- <cipher> là tệp chứa bản mã sau khi được mã hóa.
- <pub.pem> là tệp tin chứa khóa công khai RSA mà OpenSSL sinh ra.

2) Sau khi đã có khóa bí mật RSA chứa trong tệp <priv.pem>, ta thực hiện việc giải mã với OpenSSL như sau:

```
$ openssl pkeyutl -in <cipher> -out <plain> -inkey <priv.pem> -decrypt
```

Trong đó:

- <cipher> là tệp chứa bản mã,
- <plain> là tệp chứa bản rõ sau khi được giải mã.
- <priv.pem> là tệp tin chứa khóa bí mật RSA mà OpenSSL sinh ra.

3.2 Yêu cầu

Trong phần này, sinh viên sẽ thực hiện các yêu cầu sau:

- Viết báo cáo mô tả cách OpenSSL sử dụng các tệp chứa khóa bí mật `<priv.pem>` và khóa công khai `<pub.pem>` để mã hóa và giải mã các tệp tin. Sinh viên nên kết hợp giữa mô tả bằng lời với các cách trình bày khác như mã giả, lưu đồ thuật toán, sơ đồ để đảm bảo tính rõ ràng trong mô tả.
- Dựa trên báo cáo, viết chương trình mã hóa đọc tệp chứa khóa công khai `<pub.pem>` và tệp chứa bản rõ `<plain>`, xuất ra tệp chứa bản mã sau khi mã hóa `<cipher>`. Sinh viên cần đảm bảo rằng sử dụng khóa bí mật chứa trong tệp `<priv.pem>`, OpenSSL vẫn giải mã thành công được bản mã chứa trong tệp `<cipher>`.
- Dựa trên báo cáo, viết chương trình giải mã đọc tệp chứa khóa bí mật `<priv.pem>` và tệp chứa bản mã `<cipher>`, xuất ra tệp chứa bản rõ sau khi giải mã `<plain>`. Sinh viên cần đảm bảo rằng chương trình vẫn giải mã thành công được bản mã chứa trong tệp `<cipher>` được mã hóa bằng OpenSSL sử dụng khóa công khai chứa trong tệp `<pub.pem>`.
- Quay video demo cách sử dụng chương trình.

Lưu ý: Cần ghi rõ thông tin liên quan đến mã nguồn (chẳng hạn như loại ngôn ngữ lập trình, các thư viện cần cài đặt, cách thức biên dịch, cách thức chạy) và video demo (chẳng hạn như nơi tải lên) trong báo cáo.

4 (3 điểm) Tìm hiểu về chữ ký điện tử RSA của OpenSSL

4.1 Mô tả

1) Sau khi đã có khóa bí mật RSA chứa trong tệp `<priv.pem>`, ta thực hiện việc ký với OpenSSL như sau:

```
$ openssl pkeyutl -in <mess> -out <sign> -inkey <priv.pem> -sign
```

Trong đó:

- `<mess>` là tệp chứa tin nhắn cần ký,
- `<sign>` là tệp chứa chữ ký.
- `<priv.pem>` là tệp tin chứa khóa bí mật RSA mà OpenSSL sinh ra.

2) Sau khi đã có khóa công khai RSA chứa trong tệp `<pub.pem>`, ta thực hiện việc xác thực với OpenSSL như sau:

```
$ openssl pkeyutl -in <mess> -sigfile <sign> -inkey <pub.pem> -pubin -verify
```

Trong đó:

- `<mess>` là tệp chứa tin nhắn,
- `<sign>` là tệp chứa chữ ký cần xác thực.
- `<pub.pem>` là tệp tin chứa khóa công khai RSA mà OpenSSL sinh ra.

4.2 Yêu cầu

Trong phần này, sinh viên sẽ thực hiện các yêu cầu sau:

- Viết báo cáo mô tả cách OpenSSL sử dụng các tệp chứa khóa bí mật `<priv.pem>` và khóa công khai `<pub.pem>` để ký và xác thực các tệp tin. Sinh viên nên kết hợp giữa mô tả bằng lời với các cách trình bày khác như mã giả, lưu đồ thuật toán, sơ đồ để đảm bảo tính rõ ràng trong mô tả.
- Dựa trên báo cáo, viết chương trình ký đọc tệp chứa khóa bí mật `<priv.pem>` và tệp chứa tin nhắn cần ký `<mess>`, xuất ra tệp chứa chữ ký `<sign>`. Sinh viên cần đảm bảo rằng sử dụng khóa công khai chứa trong tệp `<pub.pem>`, OpenSSL vẫn có khả năng xác thực được chữ ký chứa trong tệp `<sign>` cho tin nhắn chứa trong tệp `<mess>`.
- Dựa trên báo cáo, viết chương trình xác thực đọc tệp chứa khóa công khai `<pub.pem>`, tệp chứa tin nhắn `<mess>`, và tệp chứa chữ ký `<sign>`, xuất ra câu trả lời tin nhắn được xác thực bởi chữ ký hay không. Sinh viên cần đảm bảo rằng chương trình vẫn có khả năng xác thực được chữ ký chứa trong tệp `<sign>` được ký bằng OpenSSL sử dụng khóa bí mật chứa trong tệp `<priv.pem>` cho tin nhắn chứa trong tệp `<mess>`.
- Quay video demo cách sử dụng chương trình.

Lưu ý: Cần ghi rõ thông tin liên quan đến mã nguồn (chẳng hạn như loại ngôn ngữ lập trình, các thư viện cần cài đặt, cách thức biên dịch, cách thức chạy) và video demo (chẳng hạn như nơi tải lên) trong báo cáo.

5 Các quy định khác về đồ án

- Đồ án được thực hiện cá nhân.
- Thời gian thực hiện là 3 tuần tính từ lúc đồ án được công bố chính thức bằng thông báo.
- Cấu trúc bài làm như sau:

  ```
  <MSSV>.zip
  ├── Report
  ├── Source
  └── Demo
  ```

  Trong đó:

  - `<MSSV>` là mã số sinh viên của người nộp.
  - `.zip` là định dạng nén cho bài làm (định dạng ZIP).
  - `Report` là thư mục chứa toàn bộ các báo cáo cho đồ án này.
  - `Source` là thư mục chứa toàn bộ các mã nguồn cho đồ án này.
  - `Demo` là thư mục chứa toàn bộ các video demo cho đồ án này.
- Sinh viên được tự do lựa chọn ngôn ngữ lập trình mà mình muốn cho bất kỳ phần nào của đồ án này.
- Trong trường hợp video có kích thước quá lớn, không thể nộp được thì trong thư mục Demo, sinh viên soạn một tệp văn bản ghi đường dẫn tới các video mà sinh viên đã tải lên (chẳng hạn như YouTube).
- Mọi thắc mắc vui lòng gửi về email: `nvqhuy@fit.hcmus.edu.vn`