

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



BÁO CÁO BÀI TẬP LỚN

MẬT MÃ VÀ AN NINH MẠNG

ĐỀ TÀI:

HIỆN THỰC CHƯƠNG TRÌNH MÃ HÓA TẬP TIN VÀ THƯ MỤC

GV: Nguyễn Hữu Hiếu

SVTH: Tạ Chí Tây 51303574

Chu Thanh Văn 51304750

Thành phố Hồ Chí Minh, tháng 03/2017

MỤC LỤC

[**CHƯƠNG 1.** **TÓM TẮT** 3](#_Toc477530647)

[**CHƯƠNG 2.** **GIỚI THIỆU** 4](#_Toc477530648)

[**CHƯƠNG 3.** **NỘI DUNG** 5](#_Toc477530649)

[**CHƯƠNG 4.** **PHÂN TÍCH VÀ KẾT LUẬN** 6](#_Toc477530650)

[**CHƯƠNG 5.** **HƯỚNG PHÁT TRIỂN** 7](#_Toc477530651)

[**CHƯƠNG 6.** **THAM KHẢO** 8](#_Toc477530652)

[PHỤ LỤC 1 9](#_Toc477530653)

[PHỤ LỤC 2 10](#_Toc477530654)

1. **TÓM TẮT**

Trong ngành mật mã học, mã hóa là quá trình dùng để biến đổi thông tin từ dạng này sang dạng khác nhằm ngăn chặn bất cứ ai có thể đọc được thông tin đó khi chưa được phép của bạn. Mã hóa có vai trò rất quan trọng trong ngành điện toán, đặc biệt là trong giao dịch điện tử, nó giúp đảm bảo bí mật, toàn vẹn thông tin khi thông tin được truyền qua mạng, cũng như là nền tảng của chữ ký điện tử, hệ thống PKI,…

Nội dung của bài báo cáo này tập trung vào việc ứng dụng các giải thuật mã hóa vào trong ứng dụng thực tế (chương trình “chatting” giữa 2 người), làm cho việc gửi và nhận dữ liệu (message, files) được an toàn khi đi trong môi trường mạng máy tính. Phần đầu báo cáo sẽ trình bày cách thức tiến hành để mã hóa / giải mã các file dữ liệu, đồng thời chứng minh tính toàn vẹn so với tập tin gốc ban đầu. Tiếp đến là trình bày về một số việc ứng dụng chương trình mã hóa như: chương trình chat, sinh và phân phối khóa. Cuối cùng là đánh giá kết quả và mặt hạn chế cũng như hướng phát triển trong tương lai.

1. **GIỚI THIỆU**
   1. **Phạm vi giới hạn đề tài**

* Hiện thực các giải thuật mã hóa DES, AES, RSA thông qua thư viện Java Crytography Architecture (JCA).
* Ứng dụng các giải thuật mã hóa trên vào chương trình chat (có gửi message và file), quá trình sinh và phân phối khóa, mã hóa/giải mã toàn bộ tập tin được chọn.
  1. **Nội dung đã thực hiện**

Chưa biết ghi gì

1. **NỘI DUNG**
   1. Kiến trúc chương trình

// hình vẽ kiến trúc.

1. Khởi tạo chương trình
   * Bên A sẽ sử dụng chức năng sinh khóa của chương trình hoặc nhập khóa đối xứng (DES, AES) vào file config.properties để xác định khóa sẽ sử dụng cho mã hóa thông tin
   * Trước khi trao đổi dữ liệu với nhau, A sẽ chuyển khóa đối xứng (DES, AES) đã mã hóa bằng RSA qua cho B. Sau khi B nhận được, giải mã để có được khóa phục vụ cho việc giải mã dữ liệu sau này. Trong bài báo cáo này chúng tôi giả định trường hợp A và B đã biết public key của nhau.
2. Ứng dụng của mã hóa vào chương trình chat

Để phục vụ cho việc demo chương trình thể hiện rõ việc mã hóa/giải mã, chúng tôi đã sử dụng 2 cách khác nhau để ứng dụng:

* + Nếu dữ liệu là các tin nhắn: Giả sử A muốn gửi tin nhắn “I love U” cho B, chương trình sẽ tự động mã hóa trước khi gửi. Sau khi B nhận, chương trình sẽ tự động giải mã và trả về chuỗi “I love U” cho B. Đây cũng là ứng dụng của mã hóa trong thực tế, chỉ làm ở tầng bên dưới, không ảnh hưởng tới trải nghiệm người dùng.
  + Nếu dữ liệu là file: nếu A muốn gửi một file “MyLove.jpg” cho B, chương trình sẽ hiện lên để A có thể tùy chọn các thông số như: loại mã hóa, key (AES/DES). Tương tự, sau khi B nhận cũng sẽ chọn các thông số để giải mã dữ liệu: loại mã hóa, giải mã key AES/DES bằng RSA.
  1. Cơ sở lý thuyết các giải thuật mã hóa
* Lý thuyết
* Chương trình sử dụng thư viện lập trình của Java là Java Crytography Architecture (JCA).
  1. Hiện thực các giải thuật mã hóa

1. DES, AES
   * Thư viện JCA có hỗ trợ lớp dùng cho mã hóa là “Cipher”, ta lần lượt truyền (“AES/ECB/NoPadding”, 16) hay (“DES/ECB/NoPadding”,16) tương ứng cho giải thuật AES và DES. Thông số (“AES/ECB/NoPadding”,16) cho biết sẽ sử dụng giải thuật AES và mỗi lần mã hóa một block là 16 byte, thông số NoPadding sẽ được trình bày phía dưới. Tương tự cho DES.
   * Vì DES, AES phục vụ cho mục đích mã hóa dữ liệu (khác với RSA thường chỉ phục vụ cho việc phân phối khóa), dữ liệu có thể rất lớn (hàng GB). Do đó, mỗi khi mã hóa / giải mã các file dữ liệu, chương trình chỉ đọc từng block (mặc định là 16 byte/block) để mã hóa rồi ghi xuống file mã hóa. Quá trình trên được lặp lại tới khi hết file dữ liệu, nhằm đảm bảo việc nếu file quá lớn, bộ nhớ Ram không đủ sẽ gây giảm hiệu suất sử dụng, thậm chí tê liệt chương trình.
   * Việc đọc từng block 16 byte có thể dẫn đến việc block cuối cùng của file không đủ 16 byte để truyền làm thông số cho hàm mã hóa. Ví dụ file có kích thước 1028 byte, block cuối cùng có 1028 % 16 = 4 byte. Vì vậy, ta truyền thông số NoPadding để hàm mã hóa không tự thêm “padding” vào vì có thể ảnh hưởng tới việc giải mã. Để giải quyết vấn đề này, chúng tôi đã tự thêm “padding” vào block cuối cùng trước khi gọi hàm mã hóa của thư viện, sau khi giải mã xong thì bỏ “padding” đó đi trước khi trả về kết quả.
2. RSA
   1. Tính toàn vẹn của dữ liệu trước và sau khi mã hóa
   * Chương trình sử dụng hàm hash MD5 thuộc với MessageDigest của Java để kiểm tra về tính toàn vẹn của dữ liệu theo 2 cách:
     + Sử dụng song song với quá trình mã hóa / giải mã để tiết kiệm chi phí I/O. Tức là, sau khi mã hóa / giải mã xong ta sẽ có 2 chuỗi kết quả để so sánh với nhau. Nếu trùng nghĩa là 2 file đó giống nhau và quá trình mã hóa/giải mã không làm thay đổi file.
     + MD5 nhận vào đối số là một file dữ liệu, ta sẽ load file dữ liệu đó từ đĩa rồi mới thực hiện hash. Sau khi được kết quả, ta so sánh nếu trùng thì quá trình mã hóa thành công.
   1. Các ứng dụng bổ sung
3. Chương trình chat / file sharing

Được xây dựng bằng việc sử dụng socket của Java, trong đó một máy sẽ đóng vai trò server, một máy là client. Khi muốn trao đổi dữ liệu qua lại với nhau thì message đó sẽ được serialization trước, sau đó sẽ được mã hóa rồi mới gửi đi. Bên nhận sẽ giải mã rồi deserialization để nhận được thông tin cần thiết.

1. Quá trình sinh và phân phối khóa

Quá trình này được thực hiện như mô tả ở phần kiến trúc chương trình. Đó là: một bên sẽ sinh khóa đối xứng (AES, DES) rồi gửi cho bên còn lại và sử dụng khóa này để mã hóa/giải mã thông tin trong quá trình trao đổi qua lại. Trong đó, trước khi gửi khóa thì bên gửi sẽ mã hóa nó bằng public key của bên nhận, bên nhận dùng private key của mình để giải mã mới có được key đối xứng mong muốn.

1. Mã hóa/ giải mã toàn bộ tập tin trong một thư mục đã chọn

Cho người dùng chọn một thư mục cần mã hóa, chương trình sẽ nén thư mục thành 1 tập tin duy nhất thông qua class “ZipEntry” và “ZipOutputStream” của namespace java.util.zip. Tiếp đến sẽ đối xử với nó như một file thông thường. Sau khi bên nhận giải mã tập tin đã nhận đó thì chương trình thực hiện giải nén để được thư mục mong muốn.

1. **PHÂN TÍCH VÀ KẾT LUẬN**
   1. **Kết quả đạt được**

* Định dạng mà chương trình có khả năng mã hóa là không giới hạn. Các loại định dạng đã được thử nghiệm là: txt, doc, docx, pdf, mp3, jpg, rar, zip,… Vì sử dụng cách mã hóa/ giải mã như mô tả ở trên nên chương trình theo lý thuyết là không bị giới hạn bởi kích thước file dữ liệu cần mã hóa/giải mã nếu sử dụng độc lập với chương trình chat (vì chương trình chat bị giới hạn kích thước gửi nhận)
* Dẫn chứng một số ví dụ: kích thước lớn nhất đã chạy thử, chạy demo
  1. **Mặt hạn chế**

\

1. **HƯỚNG PHÁT TRIỂN**

* Chuyển phần mã hóa file thành tự động, người dùng không phải tùy chọn mỗi lần gửi/nhận file (do mục đích ban đầu của chương trình để làm rõ quá trình mã hóa) nhằm nâng cao trải nghiệm người dùng.
* Nâng cấp chương trình chat để có thể gửi file theo lý thuyết là không giới hạn kích thước.

1. **THAM KHẢO**

Ghi thêm vài cái vào,

<https://thrift.apache.org/docs/>

<http://docs.oracle.com/javase/6/docs/api/javax/crypto/Cipher.html>

<https://en.wikipedia.org/wiki/Data_Encryption_Standard>

<https://en.wikipedia.org/wiki/Advanced_Encryption_Standard>

<https://en.wikipedia.org/wiki/RSA_(cryptosystem)>

# PHỤ LỤC 1

Bảng đánh giá nhiệm vụ, vai trò của các thành viên tham gia bài tập lớn này:

|  |  |  |  |
| --- | --- | --- | --- |
| Họ và tên sinh viên | Mã số sinh viên | Nhiệm vụ | Đánh giá |
| Tạ Chí Tây | 51303574 | * Tìm hiểu và thực hiện giải thuật RSA, AES. * Tìm hiểu và thực hiện giao diện chương trình với JavaFX sử dụng Scene Builder 2.0. | 50% |
| Chu Thanh Văn | 51304750 | * Tìm hiểu và thực hiện giải thuật DES, AES. * Tìm hiểu và hiện thực chức năng chat và gửi file cho hệ chương trình. | 50% |

# PHỤ LỤC 2

**HƯỚNG DẪN SỬ DỤNG CHƯƠNG TRÌNH**

**©**

Tha hồ chụp hình khúc này