ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA KHOA HỌC & KĨ THUẬT MÁY TÍNH



MẬT MÃ & AN NINH MẠNG

Bài tập lớn1

**GVHD:** NGUYỄN NHẬT NAM

**Danh sách thành viên nhóm**

VY QUỐC VŨ: 51204653

PHẠM PHƯƠNG UYÊN: 51204447

Tóm Tắt

# Giới thiệu

## Yêu cầu đề bài

Nếu bạn hoặc những người khác trong tổ chức của mình sử dụng Dropbox hoặc SugarSync, bạn sẽ biết rằng các dịch vụ lưu trữ đám mây phổ biến này đã mã hóa dữ liệu của bạn, bảo vệ dữ liệu khi truyền và trong khi được lưu trên các máy chủ của họ. Thật không may, cũng chính những dịch vụ đó nắm giữ chìa khóa giải mã, có nghĩa là họ có thể giải mã các tập tin của bạn mà bạn không hề hay biết. Nếu bạn có bất kỳ tập tin thực sự nhạy cảm nào trong dịch vụ lưu trữ đám mây của mình, hãy sử dụng lớp mã hóa thứ hai để giữ cho chúng an toàn khỏi những con mắt tò mò.

Trong assignment này các bạn sẽ thực hiện lớp mã hóa thứ hai để giữ cho các tập tin và thư mục trên các hệ thống lưu trữ đám mây thật sự an toàn. Cụ thể là xây dựng chương trình mã hóa và giải mã các tập tin và thư mục sử dụng các giải thuật mã hóa như DES, RSA…

## Phân tích yêu cầu

Hiện thực một chương trình mã hoá – giải mã file để sử dụng trong hoàn cảnh lưu trữ dữ liệu bí mật:

* Tích hợp được 3 giải thuật mã hoá, trong đó bao gồm giải thuật mã hóa đối xứng, giải thuật mã hóa bất đối xứng và ít nhất một giải thuật mã hóa không được học trên lớp.
* mã hóa được một tập tin bất kì như hình ảnh, âm thanh, doc, pdf...
* Tính năng nâng cao
* Viết bài báo cáo

Phạm vi:

* 3 giải thuật mã hoá, có thể sử dụng thư viện hỗ trợ
* Chương trình phục vụ mục đích mã hoá dữ liệu trước khi lưu trữ trên các kho lưu trữ đám mây.
* Sử dụng ngôn ngữ bất kì

## Công việc đã làm:

* Chương trình tích hợp được 3 giải thuật mã hoá, trong đó bao gồm giải thuật mã hóa đối xứng, giải thuật mã hóa bất đối xứng và ít nhất một giải thuật mã hóa không được học trên lớp.
* Mã hóa một tập tin bất kì như hình ảnh, âm thanh, doc, pdf...
* Tính năng nâng cao
* Viết bài báo cáo (tối thiểu 15 trang)

# Chi tiết thực hiện

## Phân tích và thiết kế chương trình

Ngôn ngữ sử dụng: Java và thư viện thiết kế giao diện Swings

Chương trình phân thành 4 task tương ứng 4 tabs trên giao diện, gồm 3 task hiện thực mã hoá – giải mã cơ bản và 1 task hiện thực tính năng nâng cao.

Cấu trúc chương trình:

Package idea: thực hiện mã hoá – giải mã IDEA.

Package rsa: thực hiện giải thuật RSA.

Package crytographyelgamal: thực hiện giải thuật elgamal.

Package filemanager gồm các class, hàm thực hiện thao tác trên file (đọc, ghi dữ liệu, lấy đường dẫn, phân tích đường dẫn file).

## Sơ lược các thuật toán

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên thuật toán mã hoá | Phân loại | Tác giả | Chi tiết thuật toán | Mô hình đã thực hiện |
| IDEA (International Data Encryption Algorithm) | Mã hoá đối xứng | [James Massey](https://vi.wikipedia.org/w/index.php?title=James_Massey&action=edit&redlink=1)  [Lai Học Gia](https://vi.wikipedia.org/w/index.php?title=Lai_H%E1%BB%8Dc_Gia&action=edit&redlink=1) | Khối dữ liệu: 64 bit  [Độ dài khóa](https://vi.wikipedia.org/wiki/%C4%90%E1%BB%99_d%C3%A0i_kh%C3%B3a): 128 bit  Cấu trúc: Substitution-permutation network  Số chu trình: 8.5 | ECB |
| RSA | Mã hóa bất đối xứng, | [Ron Rivest](https://vi.wikipedia.org/w/index.php?title=Ron_Rivest&action=edit&redlink=1),  [Adi Shamir](https://vi.wikipedia.org/w/index.php?title=Adi_Shamir&action=edit&redlink=1) , [Len Adleman](https://vi.wikipedia.org/w/index.php?title=Len_Adleman&action=edit&redlink=1) | Thuật toán RSA có hai  [khóa](https://vi.wikipedia.org/wiki/Kh%C3%B3a_(m%E1%BA%ADt_m%C3%A3)): [khóa công khai](https://vi.wikipedia.org/wiki/M%E1%BA%ADt_m%C3%A3_h%C3%B3a_kh%C3%B3a_c%C3%B4ng_khai) (hay khóa công cộng) và [khóa bí mật](https://vi.wikipedia.org/w/index.php?title=Kh%C3%B3a_b%C3%AD_m%E1%BA%ADt&action=edit&redlink=1) (hay khóa cá nhân). |  |
| ???? |  |  |  |  |

## Chi tiết từng thuật toán

1. Thuật toán mã hoá IDEA

* Lí do chọn:

Đây là phương pháp mã hoá an toàn, được phát triển và cải tiến qua nhiều giai đoạn (phương pháp này đưa ra như là một sự thay thế cho phương pháp cũ [DEA](https://vi.wikipedia.org/w/index.php?title=DEA_(m%C3%A3_h%C3%B3a)&action=edit&redlink=1) (Data Encryption Standard). Tiền thân của IDEA là phương pháp mã trước đó có tên là [PES](https://vi.wikipedia.org/w/index.php?title=PES_(m%C3%A3_h%C3%B3a)&action=edit&redlink=1) (Proposed Encryption Standard); lúc đầu IDEA còn có tên là IPES (Improved PES). Sau này IDEA đã trở thành tên thương mại), được sử dụng trong PGP.

* Cách thực hiện:

IDEA là hệ mã thao tác trên từng khối 64bit, mã hóa cụm rõ 64bit thành cụm mã 64bit, sử dụng 128bit làm khóa bao gồm 1 chuỗi 8 lần biến đổi liên tiếp và tiếp theo là 1 lần biến đổi đầu ra. Lần biến đổi thứ r sử dụng 6 khối khóa con 16 bit Ki(r), 1<=i<=6, để biến đổi 64bit đầu vào X thành 1 đầu ra với 4 khối 16bit và đầu ra này lại là đầu vào cho lần biến đổi tiếp theo. Đầu ra của lần biến đổi thứ 8 là đầu vào cho lần biến đổi cuối cùng là đầu ra của phương pháp này. Trong lần biến đổi này nó dùng 4 khóa con Ki(9), 1<=i<=4, để tính toán cụm mã cuối cùng Y= (Y1,Y2, Y3, Y4). Tất cả các khóa con nhận được từ K.

* Tính an toàn:

Tính bảo mật trong thiết kế mang tính đột phá của phương pháp IDEA là cách sử dụng hỗn độn các phép toán từ tập 3 phép toán đại số khác nhau của nhóm 2\*\*n phần tử. Các phép toán nhóm thực hiện trên các phần tử biểu diễn bằng 16 bit a và b là: XOR theo từng bit: a ^ b; cộng modulo 2\*\*n: (a+b) mod 2\*\*n, kí hiệu: a [+] b, và nhân modulo 2\*\*n+1, ký hiệu a (\*) b.

Phương pháp này đã được phá sử dụng phương pháp tấn công bicliques với giảm mã dài về thành 2 bits tương tự phương pháp tấn công AES  
Khóa yếu: Khoá phải có độ dài đủ để chống lại các phương pháp vét cạn khoá ( Chống khả năng thử các khoá được sinh ra từ (N)bit khoá cho trước ).

* Thư viện sử dụng:

1. **Thuật toán mã hoá RSA**

Trước đó , vào năm 1973 ,Clifford Cocks , một nhà toán học người Anh làm việc tại GCHQ đã mô tả một thuật toán tương tự . Với khả năng tính toán tại thời điểm đó thì thuật toán này không khả thi và chưa bao giờ được thực nghiệm .Tuy nhiên , phát minh này chỉ được công bố vào năm 1997 vì được xếp vào loại tuyệt mật

Thuật toán RSA được MIT đăng ký bằng sáng chế tại Hoa Kỳ vào năm 1983.Tuy nhiên , do thuật toán đã được công bố trước khi có đăng ký bảo hộ nên sự bảo hộ hầu như không có giá trị bên ngoài Hoa Kỳ .Ngoài ra , nếu như công trình của Clifford Cocks đã công bố trước đó thì bằng sáng chế RSA đã không thể được đăng ký .

Thuật toán RSA có hai khóa: khóa công khai (hay khóa công cộng) và khóa bí mật (hay khóa cá nhân). Mỗi khóa là những số cố định sử dụng trong quá trình mã hóa và giải mã. Khóa công khai được công bố rộng rãi cho mọi người và được dùng để mã hóa. Những thông tin được mã hóa bằng khóa công khai chỉ có thể được giải mã bằng khóa bí mật tương ứng. Nói cách khác, mọi người đều có thể mã hóa nhưng chỉ có người biết khóa cá nhân (bí mật) mới có thể giải mã được.

### ***Tạo khóa***

1. Chọn 2 số nguyên tố lớn p \, và q \, với p \ne q, lựa chọn ngẫu nhiên và độc lập.
2. Tính: n = p q \,.
3. Tính: giá trị hàm số Ơle \phi(n) = (p-1)(q-1) \,.
4. Chọn một số tự nhiên e sao cho 1 < e < \phi(n) \, và là số nguyên tố cùng nhau với \phi(n) \,.
5. Tính: d sao cho d e \equiv 1 \pmod{\phi(n)}.

***Một số lưu ý:***

* Các số nguyên tố thường được chọn bằng phương pháp thử xác suất.
* Các bước 4 và 5 có thể được thực hiện bằng giải thuật Euclid mở rộng (xem thêm: số học môđun).
* Bước 5 có thể viết cách khác: Tìm số tự nhiên x \, sao cho d = \frac{x(p-1)(q-1)+1}{e} cũng là số tự nhiên. Khi đó sử dụng giá trị d \mod{(p-1)(q-1)} \,.
* Từ bước 3, PKCS#1 v2.1 sử dụng \lambda = LCM(p-1, q-1) \, thay cho \phi = (p-1)(q-1) \,).

***Khóa công khai****bao gồm*:

* *n*, môđun, và
* *e*, số mũ công khai (cũng gọi là *số mũ mã hóa*).

***Khóa bí mật****bao gồm*:

* *n*, môđun, xuất hiện cả trong khóa công khai và khóa bí mật, và
* *d*, số mũ bí mật (cũng gọi là *số mũ giải mã*).

***Mã hóa***  c = m^e \mod{n}

### ***Giải mã*** m = c^d \mod{n}

Do  *ed* ≡ 1 (mod *p*-1) và  *ed* ≡ 1 (mod *q*-1),

m^{ed} \equiv m \pmod{p}

và

m^{ed} \equiv m \pmod{q}

Do *p* và *q* là hai số nguyên tố cùng nhau, áp dụng định lý số dư Trung Quốc, ta có:

m^{ed} \equiv m \pmod{pq}.

hay:

c^d \equiv m \pmod{n}.

***Đặc điểm RSA***

Việc tìm ra 2 số nguyên tố đủ lớn p và q thường được thực hiện bằng cách thử xác suất các số ngẫu nhiên có độ lớn phù hợp (dùng phép kiểm tra nguyên tố cho phép loại bỏ hầu hết các hợp số).

P và Q còn cần được chọn không quá gần nhau để phòng trường hợp phân tích n bằng phương pháp phân tích Fermat . Ngoài ra, nếu p-1 hoặc q-1 có thừa số nguyên tố nhỏ thì n cũng có thể dễ dàng bị phân tích và vì thế p và q cũng cần được thử để tránh khả năng này.

Bên cạnh đó, cần tránh sử dụng các phương pháp tìm số ngẫu nhiên mà kẻ tấn công có thể lợi dụng để biết thêm thông tin về việc lựa chọn (cần dùng các bộ tạo số ngẫu nhiên tốt). Yêu cầu ở đây là các số được lựa chọn cần đồng thời ngẫu nhiên và không dự đoán được. Đây là các yêu cầu khác nhau: một số có thể được lựa chọn ngẫu nhiên (không có kiểu mẫu trong kết quả) nhưng nếu có thể dự đoán được dù chỉ một phần thì an ninh của thuật toán cũng không được đảm bảo. Một ví dụ là bảng các số ngẫu nhiên do tập đoàn Rand xuất bản vào những năm 1950 có thể rất thực sự ngẫu nhiên nhưng kẻ tấn công cũng có bảng này. Nếu kẻ tấn công đoán được một nửa chữ số của p hay q thì chúng có thể dễ dàng tìm ra nửa còn lại (theo nghiên cứu của Donald Coppersmith vào năm 1997 )

Một điểm nữa cần nhấn mạnh là khóa bí mật d phải đủ lớn. Năm 1990 , Wiener chỉ ra rằng nếu giá trị của p nằm trong khoảng q và 2q (khá phổ biến) và d < n1/4/3 thì có thể tìm ra được d từ n và e.

Mặc dù e đã từng có giá trị là 3 nhưng hiện nay các số mũ nhỏ không còn được sử dụng do có thể tạo nên những lỗ hổng (đã đề cập ở phần chuyển đổi văn bản rõ). Giá trị thường dùng hiện nay là 65537 vì được xem là đủ lớn và cũng không quá lớn ảnh hưởng tới việc thực hiện hàm mũ.

**Tấn công dựa trên thời gian :**

Vào năm 1995, Paul Kocher mô tả một dạng tấn công mới lên RSA: nếu kẻ tấn công nắm đủ thông tin về phần cứng thực hiện mã hóa và xác định được thời gian giải mã đối với một số bản mã lựa chọn thì có thể nhanh chóng tìm ra khóa d.

Dạng tấn công này có thể áp dụng đối với hệ thống chữ ký điện tử sử dụng RSA. Năm 2003, Dan Boneh và David Brumley chứng minh một dạng tấn công thực tế hơn: phân tích thừa số RSA dùng mạng máy tính (Máy chủ web dùng SSL). Tấn công đã khai thác thông tin rò rỉ của việc tối ưu hóa định lý số dư Trung quốc mà nhiều ứng dụng đã thực hiện.

Để chống lại tấn công dựa trên thời gian là đảm bảo quá trình giải mã luôn diễn ra trong thời gian không đổi bất kể văn bản mã. Tuy nhiên, cách này có thể làm giảm hiệu suất tính toán. Thay vào đó, hầu hết các ứng dụng RSA sử dụng một kỹ thuật gọi là che mắt . Kỹ thuật này dựa trên tính nhân của RSA: thay vì tính cd mod n, Alice đầu tiên chọn một số ngẫu nhiên r và tính (rec)d mod n. Kết quả của phép tính này là rm mod n và tác động của r sẽ được loại bỏ bằng cách nhân kết quả với nghịch đảo của r. Đỗi với mỗi văn bản mã, người ta chọn một giá trị của r. Vì vậy, thời gian giải mã sẽ không còn phụ thuộc vào giá trị của văn bản mã .

**Tấn công lựa chọn thích nghi mã :**

Năm 1981, Daniel Bleichenbacher mô tả dạng tấn công lựa chọn thích nghi bản mã (adaptive chosen ciphertext attack) đầu tiên có thể thực hiện trên thực tế đối với một văn bản mã hóa bằng RSA . Văn bản này được mã hóa dựa trên tiêu chuẩn PKCS #1 v1, một tiêu chuẩn chuyển đổi bản rõ có khả năng kiểm tra tính hợp lệ của văn bản sau khi giải mã. Do những khiếm khuyết của PKCS #1, Bleichenbacher có thể thực hiện một tấn công lên bản RSA dùng cho giao thức SSL (tìm được khóa phiên). Do phát hiện này, các mô hình chuyển đổi an toàn hơn như chuyển đổi mã hóa bất đối xứng tối ưu (Optimal Asymmetric Encryption Padding) được khuyến cáo sử dụng. Đồng thời phòng nghiên cứu của RSA cũng đưa ra phiên bản mới của PKCS #1 có khả năng chống lại dạng tấn công nói trên

1. Thuật toán ???
2. Tính năng nâng cao

Dựa vào bối cảnh đề bài, chúng ta muốn thực hiện thêm một lớp mã hoá trước khi lưu trữ dữ liệu vào các kho lưu trữ đám mây (Dropbox, OneDrive, SugarSync…), nhóm chúng em đã thực hiện thêm chức năng mã hoá toàn bộ thư mục với cấu trúc cây thư mục giống với cấu trúc ban đầu và sao chép tự động vào folder location của kho lưu trữ.

Người dùng duyệt tìm đường dẫn đến kho lưu trữ đích. Đường dẫn này được lưu lại trong một file nội bộ thực hiện mã hoá tự động trong các lần sau

Chọn thư mục muốn lưu trữ

Một thư mục tương ứng được tạo ra trong kho lưu trữ với cấu trúc cây giữ nguyên

Tất cả các file được mã hoá với cùng 1 khoá Key. Khoá này được ghi lại trong file và chứa trực tiếp trong thư mục đã chọn để mã hoá ban đầu.

# Kết quả đạt được và kết luận

1. Thuật toán mã hoá IDEA:

* Đã thực hiện được:

Mã hoá và giải mã được hầu hết các dạng file thông thường: hình ảnh (.png, .jpg,…), văn bản (.docx, .txt, .pdf…), âm thanh (.mp3), video (.flv, .mp4), file nén (.rar, .zip…)

Key do người dùng nhập, sinh tự động hoặc import từ file. Các khoá sử dụng sau khi mã hoá đều được lưu trữ lại trong file để thực hiện giải mã sau này.

Hướng phát triển

Mã hoá theo nhiều mô hình khác nhau với các mô hình làm tăng tính bảo mật của dữ liệu mã hoá (như CBC)

* Screenshot:

1. Thuật toán mã hoá RSA

* Đã thực hiện được: Các file định dạng text cơ bản
* Giới hạn:
* Key: sinh ra hoặc load từ file lên.
* Hướng phát triển: Cải thiện mã hóa các file khác nhau, nâng độ phức tạp lên.
* Screenshots

1. Hgsdfgs

- Đã thực hiện được:

- Giới hạn:

- Hướng phát triển:

- Screenshots

1. Tính năng nâng cao

* Đã thực hiện được

Mã hoá toàn bộ thư mục được chọn, bản mã được lưu tự động trong kho lưu trữ đám mây liên kết với chương trình.

* Giới hạn:

Chương trình chỉ nên sử dụng với cấu trúc cây không đòi hỏi các thành phần liên kết với nhau (chẳng hạn như một project với nhiều package, một chương trình với gói cài đặt nhiều thành phần), để sau này người dùng có thể giải mã bằng tay từng file.

* Hướng phát triển: bổ sung tính năng giải mã lại toàn bộ thư mục đã mã hoá.
* Screenshots

# Hướng phát triển

# Tham khảo

<https://vi.wikipedia.org/wiki/IDEA>

<http://www.kilobooks.com/ma-hoa-an-toan-du-lieu-ma-hoa-idea-international-data-encryption-algorithm-396452>

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

<https://vi.wikipedia.org/wiki/RSA_(m%C3%A3_h%C3%B3a)>