**Tên 1: Nguyễn Đức Duy** – **MSSV 1: 1312084**

**Tên 2:** Từ Giang Hải Đằng – **MSSV 2:** 1312134

Đồ án 1 – Mail Client

Bộ Môn: An Ninh Mạng

GIảng Viên:

+ lê giang thanh

+ Nguyễn thanh quân

2016

Mục lục

[I/ Giải pháp về hệ thống bảo mật mail 1](#_Toc451527324)

[Hàm băm 1](#_Toc451527325)

[Mã hóa đối xứng 1](#_Toc451527326)

[Mã hóa bất đối xứng 2](#_Toc451527327)

[Mã hóa hỗn hợp 2](#_Toc451527328)

[Chữ ký điện tử 2](#_Toc451527329)

[Bảo mật Private Key 2](#_Toc451527330)

[II/ Thiết kế của chương trình 2](#_Toc451527331)

[III/ Cấu trúc dữ liệu trong chương trình 2](#_Toc451527332)

[IV/ Hướng dẫn sử dụng chương trình 2](#_Toc451527333)

# I/ Giải pháp về hệ thống bảo mật mail

## Hàm băm

Chương trình sử dụng 2 loại hàm băm là:

* SHA-256 với các nội dung bình thường.
* SHA-512 khi kết hợp với khóa RSA để tạo chữ ký điện tử.

## Mã hóa đối xứng

Thông tin về việc sử dụng mã hóa đối xứng trong chương trình:

* Thuật toán: thuật toán mã hóa mã hóa đối xứng AES.
* Mode: CBC.
* Padding: PKCS5Padding.
* Độ dài key: 256 bits.
* IV: được khởi tạo ngẫu nhiên đối với mỗi key mã hóa.

Có 2 mode để sử dụng key mã hóa AES này

1. Mã hóa đối xứng: key mã hóa và IV sẽ được tạo và lưu xuống file, ta sẽ sử dụng key này trong suốt quá trình truyền gởi dữ liệu giữa 2 bên. Tạm gọi đây là ***Secret Key***.
2. Mã hóa hỗn hợp: đối với mỗi email, ta sẽ tạo ngẫu nhiên 1 key mã hóa và IV. Key và IV này sẽ được mã hóa bằng RSA và lưu trực tiếp trên mail. Key này chỉ có giá trị với 1 mail duy nhất. Ta gọi key này là ***Session Key***.

## Mã hóa bất đối xứng

Thông tin

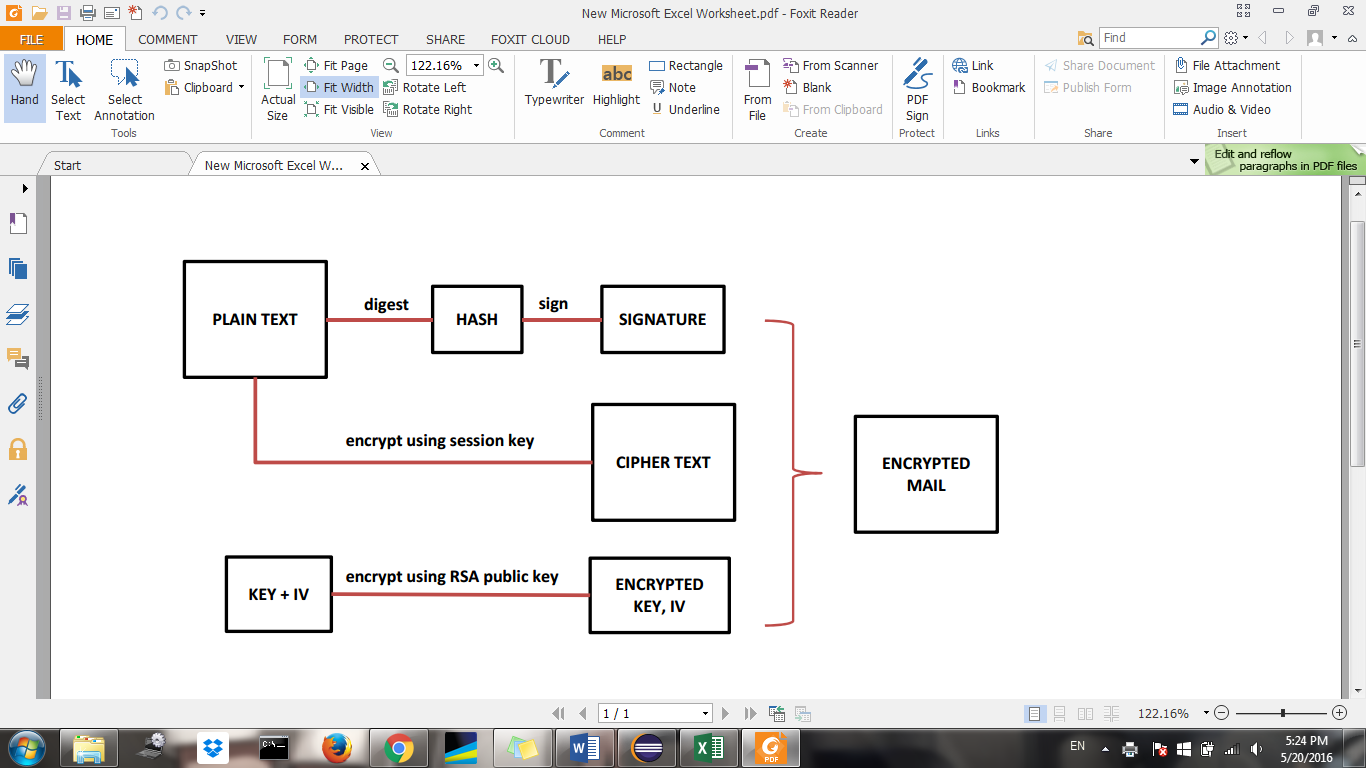
* Thuật toán: RSA.
* Độ dài khóa: 2048 bits.
* Mode: ECB.
* Padding: PKCS1Padding.

Ta sẽ gởi public key cho bên cần liên lạc, khi họ cần gởi thông tin cho ta, họ sẽ mã hóa thông tin đó lại bằng Public Key và gởi đi. Khi nhận được thông tin mà bên gởi mã hóa, ta sẽ giải mã nó ra bằng Private Key

## Mã hóa hỗn hợp

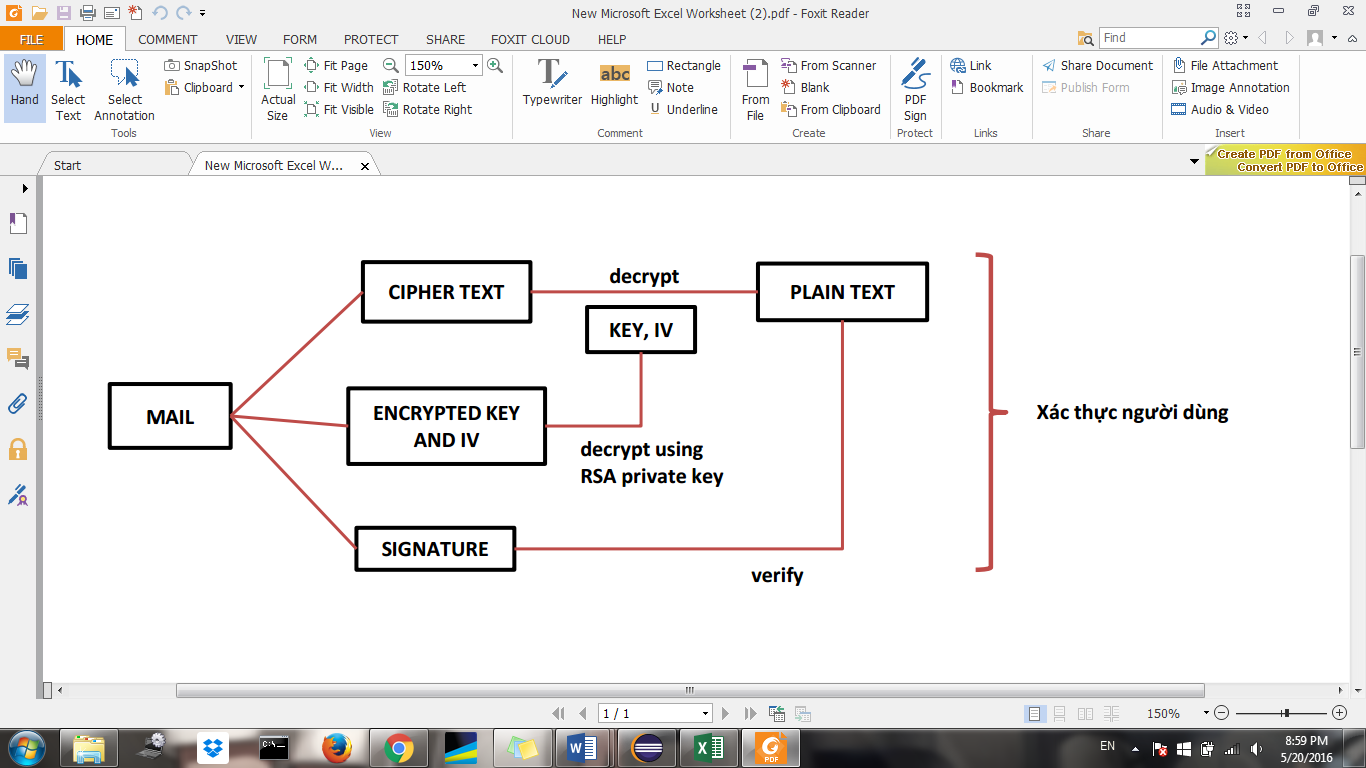
Trong mode mã hóa hỗn hợp, chương trình tuân theo mô hình chuẩn của PGP (Pretty Good Privacy) theo như mô hình sau:

**ENCRYPT MODE:**



Sơ đồ mã hóa theo mô hình PGP

**DECRYPT MODE:**



Sơ đồ giải mã theo mô hình PGP

## Chữ ký điện tử

Thông tin

* Hàm băm: SHA-512
* Thuật toán dùng để ký: RSA-2048 bit

Ta sẽ băm nội dung bức thư bằng hàm SHA-512 bit sau đó ta sẽ dùng private key để ký lên nội dung thư đã băm, sau đó chữ ký này sẽ được chứng thực bằng public key mà bên nhận đang giữ.

## Bảo mật Private Key

Mã hóa các private key bằng passphrase (hàm mã hóa đối xứng đã viết trong chương trình)

# II/ Thiết kế của chương trình

Chương trình có một class MailContent đóng gói các thuộc tính của một lá mail như recveivers, cc, bcc, host, title,…

Chương trình gồm có các hàm hỗ trợ gửi nhận mail và xóa mail như một mail client cơ bản:

Hàm public static void GetSessionObject(Person acc, String host)

* Tham số đầu vào gồm acc là một lớp Person chứa username và password để login, và một chuỗi host chứa host để connect tới
* Hàm này có nhiệm vụ tạo properties, store và kết nói tới host bằng username và password, các hàm khác sẽ gọi hàm này trước khi thực thi chức năng chính.

Hàm public static void fetchMail(Person acc, String host)

* Tham số đầu vào: Tham số này dùng để đưa vào làm tham số cho hàm GetSessionObject
* Nhiệm vụ chính của hàm này là sẽ kết nối tới thư mục INBOX để fetch mail về máy và ghi ra file lưu xuống cơ sở dữ liệu của mail client, nếu mail có file đính kèm hàm này cũng sẽ tự động tải về. Chỉ tải những mail chưa tải

Hàm public static void DeleteMail(Person acc, String host)

* Tham số đầu vào có chức năng tương tự như fetchMail
* Nhiệm vụ chính là xử lý xóa file theo 3 cơ chế, xóa mail trên cơ sở dữ liệu lưu trên client (hàm DeleteLocal), xóa mail trên server (hàm DeleteRemot) và xóa cả 2 (gọi hai hàm)

Hàm public static void sendMail (Person sender, MailContent mailclient)

* Tham số đầu vào gồm sender lớp Person chứa username và password để connect tới server, mailclient là lớp MailContent chứa toàn bộ thông tin cấu trúc của một mail
* Chức năng chính của hàm này là gửi mail từ mail client tới server

Ngoài những hàm chính trên còn có các hàm quản lý folder, phân loại mail như writePart và writeEnvelope hỗ trợ chương trình.

# III/ Cấu trúc dữ liệu trong chương trình

// Có dùng cấu trúc gì ko??? Chém được một ít à ráng chém thêm nha, đặc biệt file mã hóa nằm ở đâu với người nào truy cập thường xuyên gì ấy

Khi đăng nhập một user mới thì chương trình sẽ tạo thư mục dành riêng cho user đó trong thư mục users với tên là địa chỉ email của người đó bỏ đuôi “.” Đi

Trong thư mục của user sẽ chứa mail của người đó và các thư mục chứa file đính kèm có tên giống với tên file mail trong đó có các file đính kèm của mail đó.

# IV/ Hướng dẫn sử dụng chương trình

// Này viết gì giờ