**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN 1**



**BÀI TẬP**

**CƠ SỞ AN TOÀN THÔNG TIN**

***Đề tài: “*Tìm hiểu về giao thức bảo mật PGP: kiến trúc,  
hoạt động và ứng dụng**.***”***

**Giảng viên: HOÀNG XUÂN DẬU**

**Thực hiện : NHÓM 11**

**Lớp : Cơ sở an toàn thông tin nhóm 02**  **Khoá: 2018-2023**

**Thành viên trong nhóm:**

1. **Nguyễn Vũ Hải Thái**
2. **Kiều Văn Thành**
3. **Vũ Tiến Thành**
4. **Trần Quang Thạo**
5. **Ngô Văn Thắng**
6. **Đoàn Văn Thìn**

**Hà Nội, tháng 3 /2021**

Mục lục

|  |
| --- |
| [Chương I: Giới thiệu về giao thức bảo mật PGP 2](#_Toc67492736)  [1. Giới thiệu chung về giao thức PGP 2](#_Toc67492737)  [2. Mục đích sử dụng PGP 2](#_Toc67492738)  [3. Phương thức hoạt động của PGP: 2](#_Toc67492739)  [Chương II: Nội dung 4](#_Toc67492740)  [1. Giải thuật sử dụng trong giao thức bảo mật PGP: 4](#_Toc67492741)  [1.1 Thuật toán mã hóa khóa đối xứng 4](#_Toc67492742)  [1.1.1 IDEA 4](#_Toc67492743)  [1.1.2 3DES 4](#_Toc67492744)  [1.1.3 CAST5 5](#_Toc67492745)  [1.2 Thuật toán mã hóa khóa bất đối xứng 5](#_Toc67492746)  [1.2.1 RSA 5](#_Toc67492747)  [1.2.2 ElGamal 6](#_Toc67492748)  [1.2.3 DSA 6](#_Toc67492749)  [1.3 Hàm băm 7](#_Toc67492750)  [1.3.1 MD5 7](#_Toc67492751)  [1.3.2 SHA-1 7](#_Toc67492752)  [1.4 Thuật toán nén 8](#_Toc67492753)  [2. Mô hình kiến trúc 8](#_Toc67492754)  [3. Cách thức hoạt động của giao thức bảo mật PGP 9](#_Toc67492755)  [3.1 Quy trình mã hóa văn bản cơ bản 9](#_Toc67492756)  [3.2 Chữ ký số 11](#_Toc67492757)  [3.3 Ưu nhược điểm của PGP 12](#_Toc67492758)  [3.3.1 Ưu điểm của mã hóa PGP 12](#_Toc67492759)  [3.3.2 Nhược điểm của PGP 13](#_Toc67492760)  [CHƯƠNG III: ỨNG DỤNG 14](#_Toc67492761)  [1.1 Kleopatra Gpg4win-3.1.15 14](#_Toc67492762)  [1.2 Sử dụng dịch vụ trên web: 17](#_Toc67492763)  [1.3 Sử dụng ProtonMail: 18](#_Toc67492764)  [KẾT LUẬN 19](#_Toc67492765)  [DANH MỤC TÀI LIỆU THAM KHẢO 20](#_Toc67492766) |

**Chương I: Giới thiệu về giao thức bảo mật PGP**

## Giới thiệu chung về giao thức PGP

PGP là viết tắt của từ Pretty Good Privacy (Bảo mật rất mạnh). Mã hóa PGP là một phần mềm máy tính dùng để mật mã hóa dữ liệu và xác thực. Phiên bản PGP đầu tiên do Phil Zimmermann được công bố vào năm 1991. Kể từ đó, phần mềm này đã có nhiều cải tiến và hiện nay tập đoàn PGP cung cấp phần mềm dựa trên nền tảng này.

## Mục đích sử dụng PGP

Mục đích sử dụng PGP là phục vụ cho việc mã hóa thư điện tử, phần mềm mã nguồn mở PGP hiện nay đã trở thành một giải pháp mã hóa cho các công ty lớn, chính phủ cũng như các cá nhân. Các ứng dụng của PGP được dùng để mã hóa bảo vệ thông tin lưu trữ trên máy tính xách tay, máy tính để bàn, máy chủ và trong quá trình trao đổi email hoặc chuyển file, chữ ký số…

## Phương thức hoạt động của PGP:

PGP sử dụng kết hợp mật mã hóa khóa công khai và thuật toán khóa đối xứng cộng thêm với hệ thống xác lập mối quan hệ giữa khóa công khai và chỉ danh người dùng (ID). Phiên bản đầu tiên của hệ thống này thường được biết dưới tên mạng lưới tín nhiệm dựa trên các mối quan hệ ngang hàng (khác với hệ thống X.509 với cấu trúc cây dựa vào nhà cung cấp chứng thực số). Các phiên bản PGP về sau dựa trên các kiến trúc tương tự như hạ tầng khóa công khai.

PGP sử dụng thuật toán mã hóa khóa bất đối xứng. Trong hệ thống này, người sử dụng đầu tiên phải có một cặp khóa: Khóa công khai và khóa bí mật. Người gửi sử dụng khóa công khai của người nhận để mã hóa một khóa chung (còn được gọi là khóa phiên) dùng trong các thuật toán mật mã hóa khóa đối xứng. Khóa phiên này chính là chìa khóa để mật mã hóa các thông tin gửi qua lại trong các phiên giao dịch. Có rất là nhiều khóa công khai của những người sử dụng PGP được lưu trữ trên mác máy chủ khóa PGP trên khắp thế giới.

Một điều vô cùng quan trọng nữa là để phát hiện thông điệp có bị thay đổi hoặc giả mạo người gửi. Để thực hiện mục tiêu trên thì người gửi phải ký văn bản với thuật toán RSA hoặc DSA. Đầu tiên, PGP tính giá trị hàm băm của thông điệp rồi tạo ra chữ ký số với khóa bí mật của người gửi. Khi nhận được văn bản, người nhận tính lại giá trị hàm băm của văn bản đó đồng thời giải mã chữ ký số bằng khóa công khai của người gửi. Nếu hai giá trị này giống nhau thì có thể khẳng định là văn bản chưa bị thay đổi kể từ khi gửi và người gửi đúng là người sở hữu khóa bí mật tương ứng.

Trong quá trình mã hóa cũng như kiểm tra chữ ký, một điều vô cùng quan trọng là khóa công khai được sử dụng thực sự thuộc về người được cho là sở hữu của nó. Nếu chỉ đơn giản download một khóa công khai từ đâu đó sẽ không đảm bảo được điều này. PGP thực hiện việc phân phối khóa thông qua thực chứng số được tạo nên bởi những kỹ thuật mật mã sao cho việc sửa đổi có thể dễ dàng bị phát hiện. Tuy nhiên chỉ điều này thôi thì vẫn chưa đủ vì nó chỉ ngăn chặn được việc sửa đổi sau khi chứng thực được tạo ra. Người dùng còn cần phải trang bị khả năng xem xét khóa công khai có thực sự thuộc về người chủ sở hữu hay không. Từ phiên bản đầu tiên. PGP đã có một cơ chế hỗ trợ điều này được gọi là mạng lưới tín nhiệm.Mỗi khóa công khai đều có thể được một bên thứ 3 xác nhận.

OpenPGP cung cấp các chữ ký tin cậy có thể được sử dụng để tạo ra các nhà cung cấp chứng thực số (CA). Một chữ ký tin cậy có thể chứng tỏ rằng một khóa thực sự thuộc về một người sử dụng và người đó đáng tin cậy để ký xác nhận một khóa của mức thấp hơn. Một chữ ký có mức 0 tương đương với chữ ký trong mô hình mạng lưới tín nhiệm. Chữ ký ở mức 1 tương đương với chữ ký của một CA vì nó có khả năng xác nhận cho một số lượng không hạn chế chữ ký mức 0. Chữ ký ở mức 2 tương tự như chữ ký trong danh sách các CA mặc định trong Internet Explorer; nó cho phép người chủ tạo ra các CA khác.

PGP cũng được thiết kế với khả năng hủy bỏ hoặc thu hồi các chứng thực có khả năng đã bị vô hiệu hóa. Điều này tương đương với danh sách thực chứng

bị thu hồi của mô hình hạ tầng khóa công khai. Các phiên bản PGP gần đây cũng hỗ trợ tính năng hạn của thực chứng.

# Chương II: Nội dung

## 1. Giải thuật sử dụng trong giao thức bảo mật PGP:

### 1.1 Thuật toán mã hóa khóa đối xứng

#### 1.1.1 IDEA

Trong ngành mật mã học, phương pháp IDEA (International Data Encryption Algorithm) là một phương pháp mã khối. Được phát triển bởi Lai Học Gia và James L.Massey của ETH Zurich và được công bố lần đầu tiên vào năm 1991. IDEA là hệ mã thao tác trên từng khối 64 bit, mã hóa cụm rõ 64 bit thành cụm mã 64 bit, sử dụng 128 bit làm khóa bao gồm 1 chuỗi 8 lần biến đổi liên tiếp và 1 lần biến đổi đầu ra. Mỗi lần lặp IDEA sử dụng 3 phép toán khác nhau, mỗi phép toán thao tác trên hai đầu vào 16 bit để sản sinh một đầu ra 16 bit đơn. Ba phép toán đó là:

* XOR theo từng bit: a ^ b
* Cộng modulo 24: (a+b) mod 24, ký hiệu: a [+] b
* Nhân modulo 24+1, ký hiệu a (\*) b

#### 1.1.2 3DES

Thuật toán mã hoá 3DES là một biến thể phụ của DES, như ta đã biết DES vẫn tồn tại nhiều nhược điểm như: Có thể bẻ gãy bằng những máy có mục đích đặc biệt để tìm ra khóa. Thuật toán mã hoá 3DES gồm 3 chìa khoá 64 bit, tức là toàn bộ chiều dài khoá là 192 bit. Trong khi mã hoá riêng tư, chúng ta đơn giản là nhập toàn bộ 192 bit khóa đơn là vào mỗi 3 chìa khoá cá nhân. Thủ tục mã hoá cũng tương tự DES nhưng nó được lặp lại 3 lần tức là tăng lên 3 lần DES. Dữ liệu được mã hoá với chìa khoá đầu tiên, và được giải mã với chìa khoá 2, sau đó mã hoá lần nữa với chìa khóa thứ 3 để thu được dữ liệu mã hoá cuối cùng.

Các mẫu hoạt động của 3DES:

* Triple ECB (Triple Electronic Code Book): Sách mã hoá điện tử
* Triple CBC (Triple Cipher Chaining): Móc nối khối ký số

#### 1.1.3 CAST5

CAST-128 là mật mã Feistel 12 hoặc 16 vòng có kích thước khối là 64 bit và kích thước khóa lên đến 128 bit; nó sử dụng xoay vòng để cung cấp miễn nhiễm nội tại đối với các cuộc tấn công tuyến tính và vi sai; nó sử dụng một hỗn hợp XOR, cộng và trừ (modulo 232) trong hàm xoay vòng; và nó sử dụng ba biến thể của hàm xoay vòng trong suốt mật mã. Cuối cùng, s-box 8x32 được sử dụng trong vòng mỗi hàm xoay vòng có độ phi tuyến tính tối thiểu là 74 và mục nhập tối đa của 2 trong bảng phân phối chênh lệch.

### 1.2 Thuật toán mã hóa khóa bất đối xứng

#### 1.2.1 RSA

Thuật toán RSA được phát minh năm 1978. Thuật toán RSA có hai khóa: khóa công khai (public key) và khóa bí mật (private key). 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 bí mật mới có thể giải mã được.

Thuật toán sử dụng chế độ mã hóa khối P, C là một số nguyên (0, n)

Nhắc lại: C = EPU (P): mã hóa khóa PU

P = DPR(EPU (P)): giải mã khóa PR (ko cho phép tính được PR từ PU)

* Dạng mã hóa/ giải mã:

C = Pe mod n

P = cd mod n = Ped mod n

PU = {e, u} -> Public

PR = {d, n} -> Private

- Người gửi và người nhận biết giá trị của n và e, nhưng chỉ người nhận biết giá trị của d

- Mục đích: tìm các giá trị e, d, n (chọn) để tính P và C

Nhận xét:

- Có thể tìm giá trị của e, d, n sao cho Ped = P mod n với P < n

- Không thể xác định d nếu biết e và n

#### 1.2.2 ElGamal

Mã hóa ElGamal là một hệ thống mật mã khóa công khai. Nó sử dụng mã hóa khóa bất đối xứng để giao tiếp giữa hai bên và mã hóa tin nhắn. Hệ thống mật mã này dựa trên sự khó khăn trong việc tìm kiếm **logarit rời rạc** trong một nhóm tuần hoàn, tức là ngay cả khi chúng ta biết g a và g k , thì việc tính gak cũng vô cùng khó khăn

**Ý tưởng về hệ mật mã ElGamal**  
Giả sử Alice muốn giao tiếp với Bob.

1. Bob tạo khóa công khai và khóa riêng tư:
   * Bob chọn một số rất lớn **q** và một nhóm tuần hoàn **F**q
   * Từ nhóm tuần hoàn **F**q , anh ta chọn bất kỳ phần tử **g** và một phần tử **a** sao cho gcd (a, q) = 1
   * Sau đó, anh ta tính h = g a
   * Bob xuất bản **F** , **h = ga**, **q** và **g** là khóa công khai của mình và giữ lại **a** như khóa riêng
2. Alice mã hóa dữ liệu bằng khóa công khai của Bob:
   * Alice chọn một phần tử **k** từ nhóm tuần hoàn **F** sao cho gcd (k, q) = 1
   * Sau đó cô ấy tính p = g k và s = ​​h k = g ak
   * Cô ấy bội s với M
   * Sau đó, cô ấy gửi (p, M \* s) = (g k , M \* s)
3. Bob giải mã tin nhắn:
   * Bob tính s ′ = p a = g ak
   * Anh ta chia M \* s cho s ′ để thu được M là s = ​​s ′

#### 1.2.3 DSA

DSA là một phiên bản đặc biệt của ElGamal. Đây là phiên bản ElGamal cần một lượng lớn các tính toán đối với con số có độ dài 1024 bit, mặc dù các con số chữ ký được chọn ra là một tập con của 2160 phần tử. Các nhà thiết kế đã thành công khi tạo ra một thủ tục chỉ cần 160 bit để thể hiện nhóm con của các phần tử đó. Điều này đã làm cho các chữ ký được sinh ra có kích thước khá nhỏ, nó chỉ cần hai con số có độ lớn là 160 bit thay vì phải dùng hai số lớn có độ dài 1024 bit.

MD5 được thiết kế bởi Ronald Rivest vào năm 1991 để thay thế cho hàm băm trước đó, MD4. Vào năm 1996, người ta phát hiện ra một lỗ hổng trong MD5; trong khi vẫn chưa biết nó có phải là lỗi nghiêm trọng hay không, những chuyên gia mã hóa bắt đầu đề nghị sử dụng những giải thuật khác, như SHA-1 (khi đó cũng bị xem là không an toàn). Trong năm 2004, nhiều lỗ hổng hơn bị khám phá khiến cho việc sử dụng giải thuật này cho mục đích bảo mật đang bị đặt nghi vấn.

### 1.3 Hàm băm

#### 1.3.1 MD5

Trong mật mã học, MD5 (Message-Digest algorithm 5, giải thuật Tiêu hóa tin 5) là một hàm băm mật mã học được sử dụng phổ biến với giá trị Hash dài 128-bit. Là một chuẩn Internet (RFC 1321), MD5 đã được dùng trong nhiều ứng dụng bảo mật, và cũng được dùng phổ biến để kiểm tra tính toàn vẹn của tập tin. Một bảng băm MD5 thường được diễn tả bằng một số hệ thập lục phân 32 ký tự.

#### 1.3.2 SHA-1

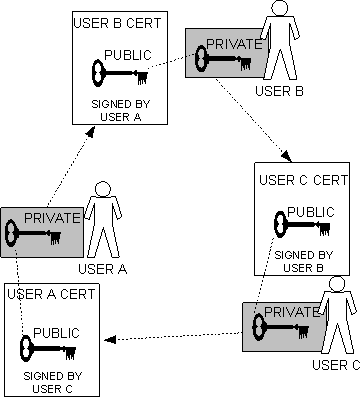
Trong mật mã học, SHA-1 (Secure Hash Algorithm 1) là hàm băm mật mã lấy đầu vào và tạo ra giá trị băm 160 bit (20 byte) được gọi là thông báo đã tiêu hóa - thường được hiển thị dưới dạng số thập lục phân, dài 40 chữ số. Nó được thiết kế bởi Cơ quan An ninh Quốc gia Hoa Kỳ, và là một Tiêu chuẩn Xử lý Thông tin Liên bang Hoa Kỳ. SHA-1 được sử dụng rộng rãi trong nhiều ứng dụng và giao thức an ninh khác nhau, bao gồm TLS và SSL, PGP, SSH, S/MIME, và IPSec. SHA-1 được coi là thuật giải thay thế MD5, một thuật giải băm 128 bit phổ biến khác. Hiện nay, SHA-1 không còn được coi là an toàn bởi đầu năm 2005, ba nhà mật mã học người Trung Quốc đã phát triển thành công một thuật giải dùng để tìm được hai đoạn dữ liệu nhất định có cùng kết quả băm tạo ra bởi SHA-1.

### 1.4 Thuật toán nén

PGP sẽ mặc định nén thông điệp sau khi ký nhưng trước quá trình mã hóa. Điều này có lợi cho việc cất giữ không gian vừa cho truyền thông email vừa cho lưu trữ trên máy tính. PGP sử dụng giải thuật Zip để nén thông điệp. Thực chất giải thuật Zip tìm kiếm những chuỗi ký tự lặp lại trong dữ liệu vào và thay thế những chuỗi như vậy với những mã gọn hơn.

## 2. Mô hình kiến trúc:

Kiến trúc tổng quan



*Hình 1.1:Kiến trúc tổng quan*

Hai dịch vụ chính mà PGP cung cấp cho người dùng là: mã hóa và xác thực thông điệp. Khi thiết kế một ứng dụng bảo mật email, người thiết kế phải đương đầu với hai vấn đề chính, trước hết, phải bảo mật ứng dụng bằng những giải thuật nào?

Trong trường hợp của PGP, những dịch vụ của nó dựa vào ba giải thuật: IDEA (mã hóa khóa bí mật), RSA (mã hóa khóa công khai) và MD5 (Hàm băm an toàn). Trong phần này chúng ta sẽ nghiên cứu toàn bộ những bước thực hiện của PGP trong truyền và nhận thông điệp và những thông báo xử lý thông điệp. Sau đó chúng ta sẽ tìm hiểu chi tiết những bước chính của quá trình xử lý này.

## 3. Cách thức hoạt động của giao thức bảo mật PGP

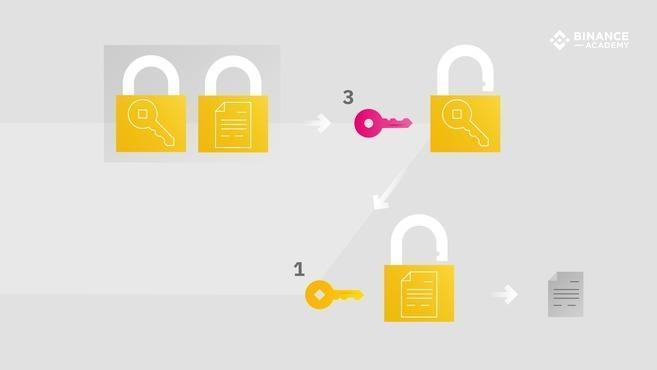
### 3.1 Quy trình mã hóa văn bản cơ bản

Trong một quy trình mã hóa văn bản cơ bản, văn bản thuần túy (dữ liệu có thể đọc được) được chuyển đổi thành văn bản mã hóa (dữ liệu không thể đọc được).Nhưng trước khi quá trình mã hóa diễn ra, hầu hết các hệ thống PGP đều thực hiện nén dữ liệu. Bằng cách nén các tệp văn bản thuần túy trước khi gửi chúng, PGP tiết kiệm cả không gian đĩa và thời gian gửi - đồng thời cải thiện tính bảo mật.

Quá trình mã hóa thực sự bắt đầu sau khi tệp được nén. Ở giai đoạn này, các tệp văn bản thuần túy đã nén được mã hóa bằng khóa sử dụng một lần, được gọi là khóa phiên (**session key**). Khóa này được tạo ngẫu nhiên thông qua việc sử dụng mật mã đối xứng và mỗi phiên giao tiếp PGP có một khóa phiên duy nhất.

Tiếp theo, khóa phiên (1) được mã hóa bằng mã hóa bất đối xứng: người nhận (người A) cung cấp khóa công khai (2) của mình cho người gửi tin nhắn (người B) để người gửi ( người B) có thể mã hóa khóa phiên. Bước này cho phép người gửi ( người B) chia sẻ khóa phiên với ( người A) một cách an toàn thông qua Internet, bất kể tình trạng bảo mật. Việc mã hóa bất đối xứng của khóa phiên thường được thực hiện thông qua việc sử dụng thuật toán RSA. (Hình 1.2)

 *Hình 1.2: Quá trình mã hóa*

 Sau khi văn bản mã của tin nhắn và khóa phiên được mã hóa được gửi đi, người nhận (người B) có thể sử dụng khóa riêng (3) của mình để giải mã khóa phiên, sau đó khóa phiên (1) được sử dụng để giải mã văn bản mã hóa trở lại văn bản gốc. (Hình 1.3)

*Hình 1.3 Quá trình giải mã*

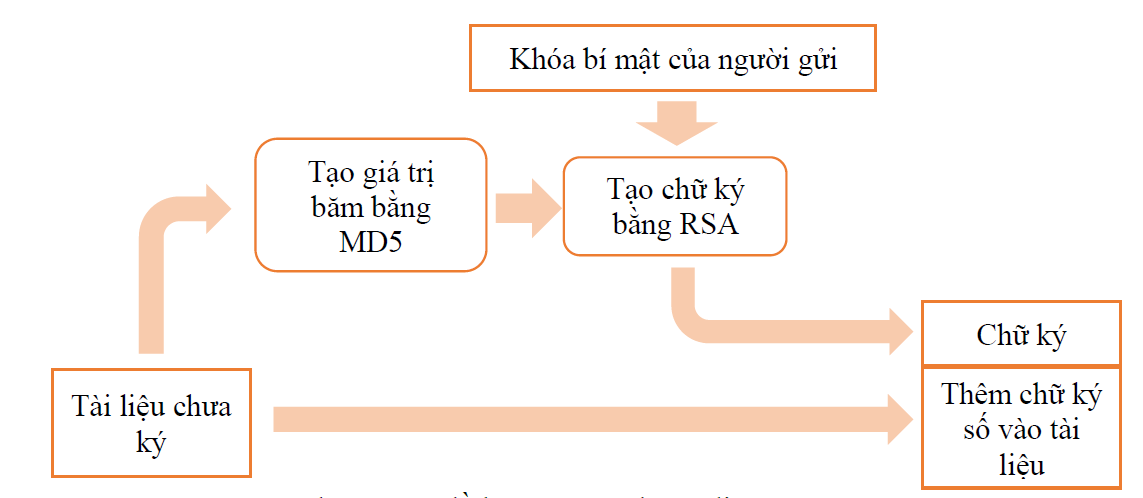
### 3.2 Chữ ký số

Ngoài quy trình mã hóa và giải mã cơ bản, PGP còn hỗ trợ chữ ký số, tính năng này có ít nhất ba chức năng:

* Xác thực: người B có thể xác minh rằng người gửi tin nhắn là A
* Tính toàn vẹn: người B có thể chắc chắn rằng thông điệp được giữ nguyên, không bị thay đổi
* Đảm bảo người gửi không thể thoái thác: sau khi tin nhắn được ký điện tử, người A không thể nói rằng cô ấy chưa gửi nó

Một chữ ký số phục vụ cùng một mục đích như một chữ ký viết tay. Tuy nhiên một chữ ký viết tay rất dễ dàng bị giả mạo. Một chữ ký số cao cấp hơn một chữ ký viết tay là gần như không thể làm giả, và nó là minh chứng cho nội dung của thông tin cũng như danh tính của người ký.

Chữ ký số cho người nhận thông tin xác minh tính xác thực của nguồn gốc thông tin, và cũng xác nhận rằng thông tin còn nguyên vẹn. Một chữ ký số công khai rất quan trọng trong cung cấp chứng thực và toàn vẹn dữ liệu. Cách thức làm việc của chữ ký số được mô tả trong hình 1.4



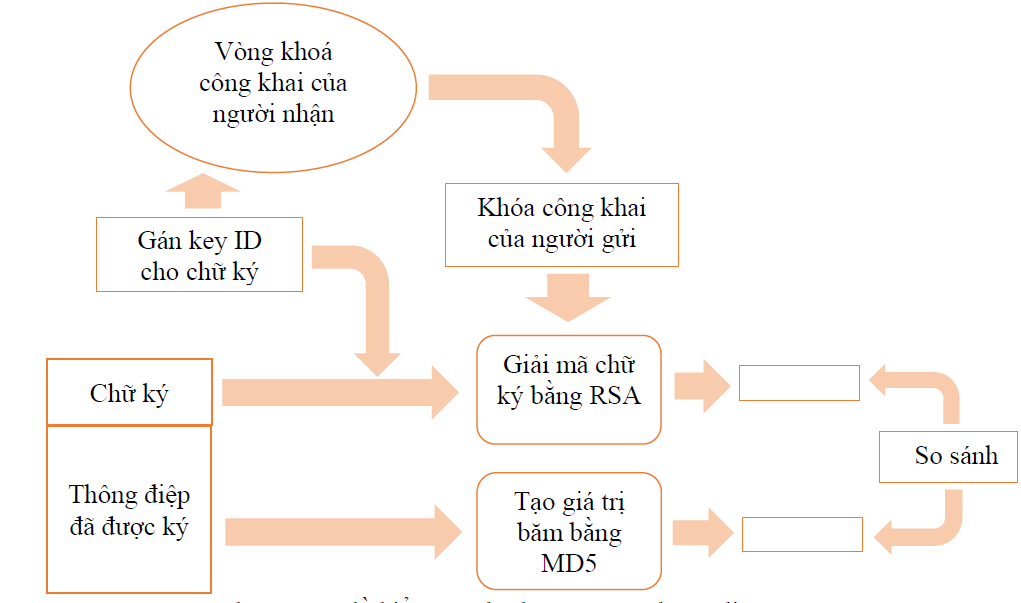
*Hình 1.4: Lược đồ ký trên 1 thông điệp PGP*

Người gửi tạo ra một thông điệp:

1. PGP sử dụng MD5 băm thông điệp tạo ra một mã băm 128 bit

2. Người gửi lấy khóa bí mật trên vòng khóa để sử dụng

3. PGP mã hóa mã băm bằng RSA sử dụng chìa khóa bí mật của người gửi, và gán kết quả vào thông điệp. Key ID của khóa công khai của người gửi tương ứng gắn liền với chữ ký



*Hình 1.5: Lược đồ kiểm tra chữ ký trên một thông điệp*

PGP của người nhận:

1. PGP lấy Key ID được gán trong chữ ký và sử dụng nó để lấy khóa công khai đúng từ vòng khóa công khai.
2. PGP sử dụng RSA với khóa công khai của người gửi để giải mã khôi phục mã băm.
3. PGP tạo ra một mã băm mới cho thông điệp và so sánh nó với mã băm giải mã. Nếu cả hai trùng nhau, thông điệp được xác thực.

Sự kết hợp của MD5 và RSA cung cấp một sơ đồ chữ ký số hiệu quả. Với sức mạnh của RSA, người nhận chắc chắn rằng chỉ người sở hữu riêng với khóa thích hợp mới có thể tạo chữ ký. Với sức mạnh của MD5, người nhận chắc chắn rằng không ai khác có thể tạo ra một thong điệp mới mà mã băm trùng với mã băm của thông điệp gốc và vì vậy không thể trùng với chữ ký của thông điệp gốc.

### 3.3 Ưu nhược điểm của PGP

#### 3.3.1 Ưu điểm của mã hóa PGP

Nhờ sử dụng kết hợp mã hóa đối xứng và bất đối xứng, PGP cho phép người dùng chia sẻ thông tin và khóa mật mã qua Internet một cách an toàn. Là một hệ thống kết hợp, PGP vừa có tính bảo mật của mật mã bất đối xứng và tốc độ của mã hóa đối xứng. Ngoài bảo mật và tốc độ, tính năng chữ ký số đảm bảo tính toàn vẹn của dữ liệu và tính xác thực của người gửi.

Giao thức OpenPGP tạo ra một môi trường cạnh tranh được tiêu chuẩn hóa, và hiện nhiều công ty và tổ chức cung cấp các giải pháp PGP.

Điểm mạnh của mã hóa PGP là về cơ bản nó không thể phá vỡ. Đó là lý do tại sao nó vẫn được sử dụng bởi các nhà báo và nhà hoạt động, và nó thường được coi là cách tốt nhất để cải thiện bảo mật đám mây. Nói tóm lại, về cơ bản không ai - dù là hacker hay thậm chí là NSA - có thể phá vỡ mã hóa PGP.

Mặc dù đã có một số tin bài chỉ ra các lỗ hổng bảo mật trong một số triển khai PGP, chẳng hạn như lỗ hổng Efail , nhưng về cơ bản bản thân PGP vẫn rất an toàn.

#### 3.3.2 Nhược điểm của PGP

Hệ thống PGP có nhược điểm ở chỗ nó không dễ sử dụng và dễ hiểu, đặc biệt là đối với người dùng có ít kiến thức kỹ thuật. những người sử dụng hệ thống cần phải biết cách thức hoạt động của hệ thống, phòng trường hợp họ đưa ra các lỗ hổng bảo mật do sử dụng nó không đúng cách. Điều này có nghĩa là các doanh nghiệp đang cân nhắc chuyển sang PGP sẽ cần phải đào tạo. Ngoài ra, nhiều người cho rằng độ dài của các khóa công khai gây nhiều bất tiện.

Cuối cùng, PGP mã hóa tin nhắn của bạn, nhưng nó không làm cho bạn ẩn danh. Không giống như các trình duyệt ẩn danh sử dụng máy chủ proxy hoặc làm việc thông qua VPN để ẩn vị trí thực của bạn, email được gửi qua PGP có thể được truy tìm từ người gửi và người nhận. Các dòng tiêu đề của chúng cũng không được mã hóa, vì vậy bạn không nên đặt bất kỳ thông tin nhạy cảm nào trong dòng chủ đề.

# CHƯƠNG III: ỨNG DỤNG

Thông thường có 3 cách sử dụng PGP trong thực tế :

* Gửi và nhận Email mã hóa
* Xác minh danh tính của người gửi
* Mã hóa tập tin lưu trữ trên thiết bị hoặc lưu trữ đám mây

Gửi và nhận Email mã hóa được sử dụng phổ biến. Ngày nay, sự phổ biến của PGP ngày càng mở rộng. Càng nhiều người dùng nhận ra những thông tin của họ được các tập đoàn hay chính phủ thu nhập, càng nhiều người dùng ngày nay sử dụng tiêu chuẩn để giữ thông tin riêng tư của mình.

Việc lựa chọn phần mềm PGP còn tùy thuộc vào nhu cầu cá nhân (hoặc doanh nghiệp).Không phải mọi email đều cần mã hóa nên việc tải một tiện ích mở rộng đôi khi không cần thiết. Cần cân nhắc sử dụng dịch vụ PGP trực tuyến để gửi các email quan trọng.

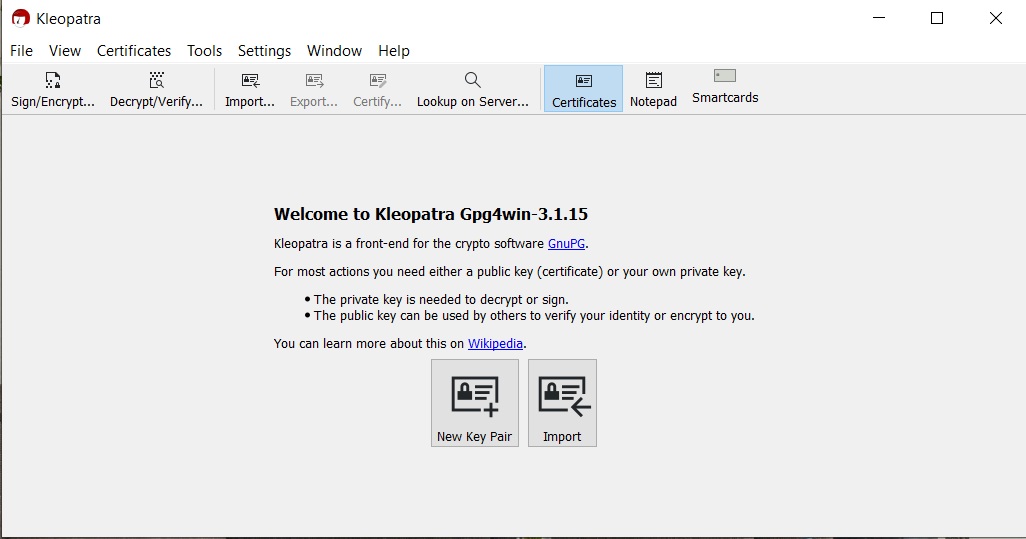
Một số phần mềm chương trình giúp sử dụng mã hóa PGP :

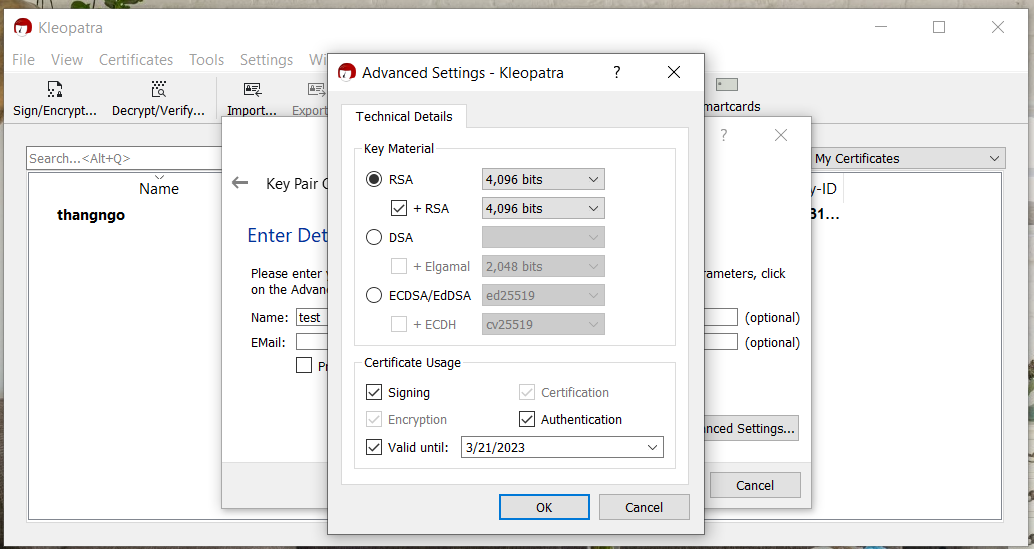
## 1.1 Kleopatra Gpg4win-3.1.15

- Download và cài đặt tại <https://www.gpg4win.org/download.html>

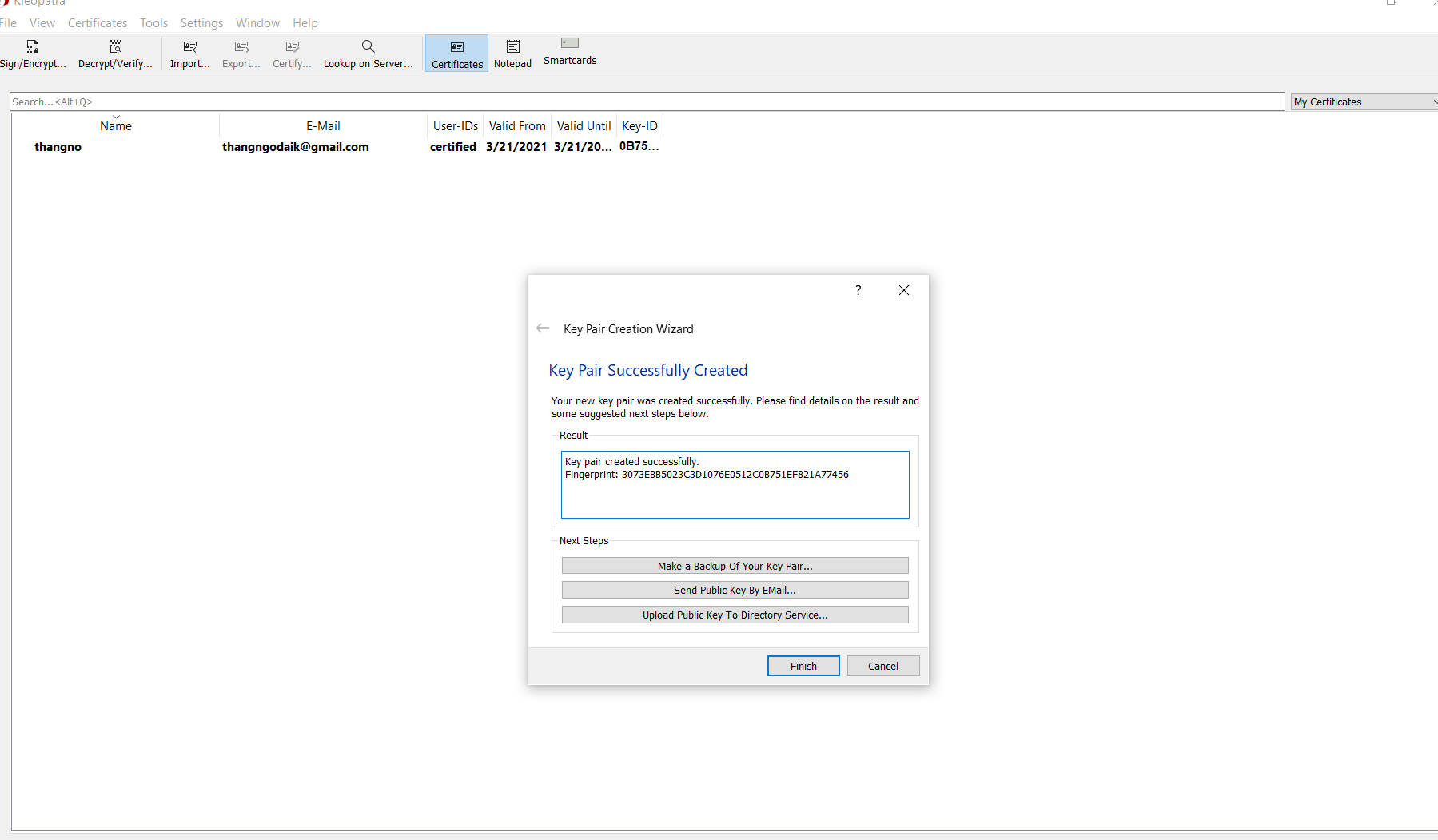
- Lưu ý: lưu file cài đặt trong ổ C( tránh lỗi NSIS)

- Màn hình sau khi cài đặt:



- Tạo mới Key Pair

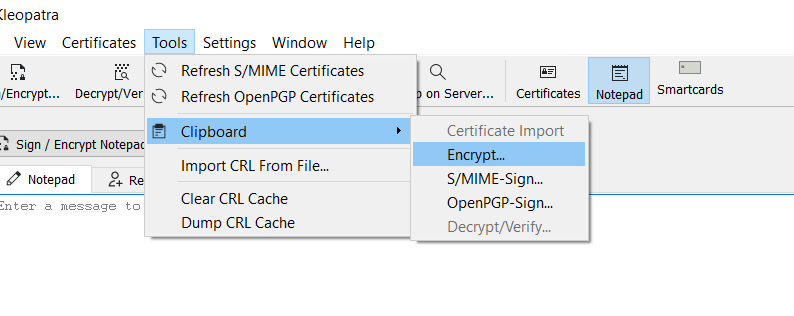
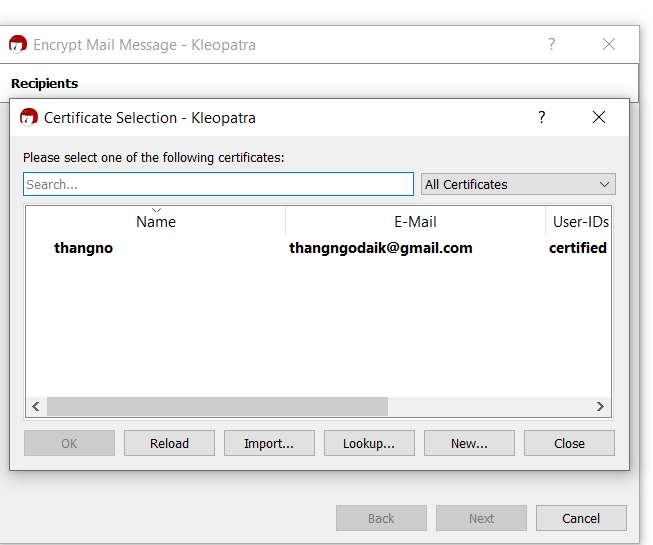
- Private key được tạo:

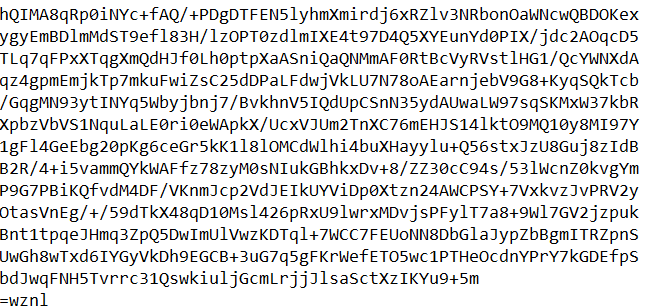


- Chọn Export để tạo được file public key

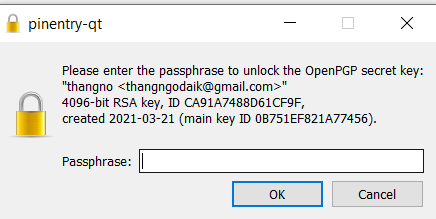
- Như vậy đã có 2 key tương ứng là private và public. Để gửi nhận thông điệp cho người khác ta cần có public key của họ.

- Thử nghiệm tự gửi thông điệp mã hóa và giải mã:

* Copy thông điệp vào clipboard chọn mã hóa
* Chọn người nhận
* Thu được tin nhắn mã hóa:

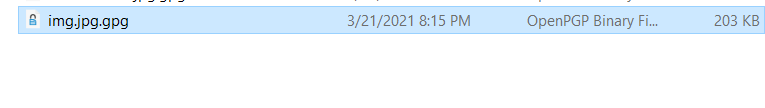


* Để giải mã : copy tin nhắn mã hóa vào clipboard rồi chọn giải mã, nhập passphrase thu được thông điệp bản rõ:



- Một số chức năng khác: mã hóa và giải mã hóa tập tin: chọn tập tin cần mã hóa-> chọn sign and encrypt, chọn những người có thể giải mã.

- Thu được tập tin đã được mã hóa:



## 1.2 Sử dụng dịch vụ trên web:

- Tạo key: [https://www.igolder.com/pgp/](https://www.igolder.com/pgp/encryption/)

- Để tạo thông điệp mật mã hóa: <https://www.igolder.com/pgp/encryption/>

* Nhập publickey của người nhận
* Nhập thông điệp gửi

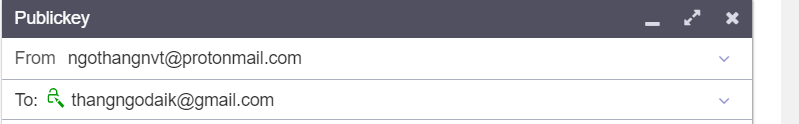
- Để giải mã thông điệp vào trang: <https://www.igolder.com/pgp/decryption/>

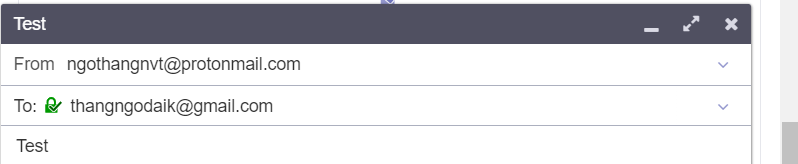
* Nhập private key cùng với passphrase của để giả mã thông điệp mã hóa

## 1.3 Sử dụng ProtonMail:

ProtonMail mang lại cho bạn sự bảo mật mà bạn cần, bằng cách cung cấp một tài khoản email được mã hóa và các cài đặt bảo mật khác để giữ cho email an toàn. Đồng thời, bạn cũng sẽ nhận được các tính năng mong đợi trong một dịch vụ email.ProtonMail có cả trên web và thiết bị di động.

Mặc định những người cùng sử dụng ProtonMail sẽ gửi mail cho nhau được mã hóa PGP một cách tự động. Để gửi email mã hóa PGP từ ProtonMail ra người dùng bên ngoài cần có thiết lập tương tự như gửi khóa công khai với chữ ký số. Email chưa được mã hóa :



Email đã được mã hóa và xác thực chữ ký số : 

# KẾT LUẬN

Trên đây, em đã hoàn thành nội dung báo cáo, tuy chưa thực sự hoàn thiện và đầy đủ nhưng em xin được tóm tắt lại như sau:

Chương 1: Giới thiệu tổng quát về giao thức bảo mật PGP

Chương 2: Nội dung về giao thức bảo mật PGP: giải thuật sử dụng, mô hình kiến trúc, cách thức hoạt động

Chương 3: Ứng dụng giao thức bảo mật PGP.

Nội dung báo cáo vẫn còn nhiều nhược điểm và nhiều phần thiếu sót, chúng em mong nhận được nhận xét, bổ sung của thầy để đề tài hoàn thiện và đầy đủ hơn.

Một lần nữa, chúng em xin chân thành cảm ơn!

**DANH MỤC TÀI LIỆU THAM KHẢO**

**Tài liệu tham khảo**

1. <https://academy.binance.com/vi/articles/what-is-pgp>

2. <https://www.varonis.com/blog/pgp-encryption/>

3. <https://en.wikipedia.org/wiki/Pretty_Good_Privacy>

4. <https://quantrimang.com/pretty-good-privacy-pgp-la-gi-162378#mcetoc_1d5viig8e2>

5. <https://protonmail.com/support/knowledge-base/how-to-use-pgp/>

6. <https://youtu.be/lAblt1Qt_ng>

7. <https://youtu.be/mwyYRFBcEsU>