HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA AN TOÀN THÔNG TIN

-----🙞🙜🕮🙞🙜-----



MẬT MÃ HỌC CƠ SỞ

BÀI TIỂU LUẬN

GIAO THỨC BẢO MẬT PGP

Giảng viên: Đỗ Xuân Chợ

Thành viên:

Nguyễn Hà Thanh (B20DCAT173)

Ngô Quang Tùng (B20DCAT169)

Đỗ Tiến Sĩ (B20DCAT153)

Đỗ Trung Kiên (B20DCAT097)

*HÀ NỘI, 2023*

**MỤC LỤC**

[**I. Giới thiệu tổng quan về giao thức PGP 2**](#_Toc3841309)

[**1.1. Giới thiệu chung về giao thức PGP 2**](#_Toc3841310)

[**1.2. Mục Đích sử dụng PGP 2**](#_Toc3841311)

[**1.3. Phương thức hoạt động của PGP 3**](#_Toc3841312)

[**1.4. OpenPGP - phần mềm bảo mật email/file dựa trên PGP 4**](#_Toc3841313)

[**II. Mô hình giao thức bảo mật PGP 5**](#_Toc3841314)

[**2.1. Giải thuật nổi bật trong giao thức bảo mật PGP 5**](#_Toc3841315)

[**2.1.1. Thuật toán khóa bất đối xứng 5**](#_Toc3841316)

[**2.1.2. Thuật toán khóa đối xứng 7**](#_Toc3841317)

[**2.1.3. Hàm băm mật mã (hash) 9**](#_Toc3841318)

[**2.2. Mô hình kiến trúc và hoạt động của PGP 10**](#_Toc3841319)

[**KẾT LUẬN 19**](#_Toc3841321)

[**TÀI LIỆU THAM KHẢO 19**](#_Toc3841322)

# 

# **I. Giới thiệu tổng quan về giao thức PGP**

## **1.1. Giới thiệu chung về giao thức PGP**

Giao thức PGP được tạo ra bởi nhà khoa học máy tính Phil Zimmerman vào năm 1991, người đã quyết định đặt tên nó theo một cửa hàng tạp hóa địa phương - Ralph's Pretty Good Grocery.  Vào thời điểm này, ông ta đã là một nhà hoạt động chống năng lượng hạt nhân và mục đích tạo PGP là để phục vụ những người có mục tiêu tương tự có thể sử dụng các hệ thống bảng thông báo điện tử (bulletin board) và lưu trữ tệp một cách an toàn. Đối với mục tiêu sử dụng phi thương mại, PGP hoàn toàn miễn phí và toàn bộ mã nguồn được bao gồm trong tất cả sản phẩm. PGP dễ dàng thâm nhập vào [Usenet](https://vi.wikipedia.org/wiki/Usenet) và từ đó vào [Internet](https://vi.wikipedia.org/wiki/Internet). Ngay từ khi mới xuất hiện, PGP đã vướng vào chính sách hạn chế xuất khẩu phần mềm mật mã của [chính phủ Hoa Kỳ](https://vi.wikipedia.org/wiki/Ch%C3%ADnh_quy%E1%BB%81n_Hoa_K%E1%BB%B3). Sau đó nó được công bố công khai cho mọi người sử dụng vào năm 1993 khi Zimmerman phát hành công nghệ cho bất cứ ai muốn sử dụng nó ở Mỹ thông qua FTP.

Mười bốn năm sau, Zimmerman đã bán PGP Inc để bảo mật cho Tập đoàn khổng lồ Symantec, và chính nhà phát triển phần mềm diệt virus hiện chịu trách nhiệm cập nhật PGP để đảm bảo nó đủ để bảo vệ thông tin liên lạc qua email. Công ty cũng đã phát triển một biến thể nguồn mở - OpenPGP, được sử dụng cùng với phiên bản được cấp phép.

## **1.2. Mục Đích sử dụng PGP**

Mục tiêu ban đầu của PGP nhằm vào mật mã hóa nội dung các thông điệp thư điện tử và các tệp đính kèm cho người dùng phổ thông. Bắt đầu từ 2002, các sản phẩm PGP đã được đa dạng hóa thành một tập hợp ứng dụng mật mã và có thể được đặt dưới sự quản trị của một máy chủ. Các ứng dụng PGP giờ đây bao gồm: thư điện tử, chữ ký số, mật mã hóa ổ đĩa cứng máy tính xách tay, bảo mật tệp và thư mục, bảo mật các phiên trao đổi IM, mật mã hóa luồng chuyển tệp, bảo vệ các tệp và thư mục lưu trữ trên máy chủ mạng.

PGP hoạt động trên một số tiêu chuẩn khác nhau, phổ biến nhất trong số đó là tiêu chuẩn OpenPGP nguồn mở. Điều này được sử dụng rộng rãi để bảo đảm các ứng dụng trên máy tính để bàn như Microsoft Outlook và Apple Mail trên Mac.

Ngoài ra còn có một plugin do Google phát triển cho phép sử dụng tiêu chuẩn trên trình duyệt Chrome của nó.

## **1.3. Phương thức hoạt động của PGP**

PGP hoạt động bằng cách đặt các lớp bảo mật được mã hóa ở đầu nội dung dựa trên văn bản của ứng dụng. Trong trường hợp ứng dụng email, PGP đóng chặt nội dung của thư bằng thuật toán mã hóa, xáo trộn văn bản theo cách mà nếu nó bị chặn thì sẽ không thể đọc được.

PGP sử dụng kết hợp [mật mã hóa 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) và [thuật toán khóa đối xứng](https://vi.wikipedia.org/w/index.php?title=M%E1%BA%ADt_m%C3%A3_kh%C3%B3a_%C4%91%E1%BB%91i_x%E1%BB%A9ng&action=edit&redlink=1) 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](https://vi.wikipedia.org/wiki/M%E1%BA%A1ng_l%C6%B0%E1%BB%9Bi_t%C3%ADn_nhi%E1%BB%87m) dựa trên các mối quan hệ ngang hàng (khác với hệ thống [X.509](https://vi.wikipedia.org/wiki/X.509) với cấu trúc cây dựa vào các [nhà cung cấp chứng thực số](https://vi.wikipedia.org/wiki/Nh%C3%A0_cung_c%E1%BA%A5p_ch%E1%BB%A9ng_th%E1%BB%B1c_s%E1%BB%91)). Các phiên bản PGP về sau dựa trên các kiến trúc tương tự như [hạ tầng khóa công khai](https://vi.wikipedia.org/wiki/H%E1%BA%A1_t%E1%BA%A7ng_kh%C3%B3a_c%C3%B4ng_khai).

PGP sử dụng thuật toán mật mã hóa khóa bất đối xứng. Trong các 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 gọi là khóa phiên) dùng trong các thuật toán mật mã hóa khóa đối xứng. Khóa phiên này chính là khóa để mật mã hóa các thông tin được gửi qua lại trong phiên giao dịch. Rất nhiều khóa công khai của những người sử dụng PGP được lưu trữ trên các máy chủ khóa PGP trên khắp thế giới (các máy chủ mirror lẫn nhau). Người nhận trong hệ thống PGP sử dụng khóa phiên để giải mã các gói tin. Khóa phiên này cũng được gửi kèm với thông điệp nhưng được mật mã hóa bằng hệ thống mật mã bất đối xứng và có thể tự giải mã với khóa bí mật của người nhận. Hệ thống phải sử dụng cả hai dạng thuật toán để tận dụng ưu thế của cả hai: thuật toán bất đối xứng đơn giản việc [phân phối khóa](https://vi.wikipedia.org/w/index.php?title=Ph%C3%A2n_ph%E1%BB%91i_kh%C3%B3a&action=edit&redlink=1) còn thuật toán đối xứng có ưu thế về tốc độ (nhanh hơn cỡ 1000 lần).

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.

1.4. OpenPGP - phần mềm bảo mật email/file dựa trên PGP

OpenPGP quy định các cơ chế thương lượng giữa các chương trình PGP ở các phía của đường truyền cũng như thuật toán mã hóa được sử dụng và các tính năng bổ sung khác từ phiên bản PGP 2.x. Tất cả các chương trình tuân theo PGP đều bắt buộc phải thực hiện những quy định này. Vì vậy, không tồn tại những vấn đề tương thích lớn giữa các phiên bản PGP, bất kể nó được lập trình từ đâu: PGP Corp, McAfee, Gnu/FSF (ie, GPG), Hushmail, Veridis, Articsoft, Forum... Các lập trình viên của các chương trình này cũng có mối quan hệ nhất định với nhau. Họ coi những bất tương thích là các lỗi phần mềm và sửa mỗi khi phát hiện ra.

So sánh với RFC 1991 (PGP 2.x), OpenPGP đưa ra nhiều tính năng mới. Nó hỗ trợ khả năng tương thích ngược có nghĩa các phiên bản thực hiện OpenPGP có thể đọc và sử dụng các khóa, chứng thực của các phiên bản trước đó. PGP 2.x không có khả năng tương thích xuôi vì nó không thể sử dụng các văn bản hay khóa tuân theo OpenPGP.

Trong các đặc tả gần đây của OpenPGP, 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 ở 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.

II. Mô hình giao thức bảo mật PGP

2.1. Giải thuật nổi bật trong giao thức bảo mật PGP

2.1.1. Thuật toán khóa bất đối xứng

a) RSA (mã hóa và chữ ký)

Trong mật mã học, RSA là một thuật toán mật mã hóa khóa công khai. Đây là thuật toán đầu tiên phù hợp với việc tạo ra chữ ký điện tử đồng thời với việc mã hóa. Nó đánh dấu một sự tiến bộ vượt bậc của lĩnh vực mật mã học trong việc sử dụng khóa công cộng.

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.

***Thuật toán sinh khoá:*** Người dùng A tạo khoá công khai và khoá bí mật theo phương thức sau:

* Sinh ra hai số nguyên tố lớn ngẫu nhiên p và q
* Tính n = p\*q và ** (n) = (p-1)\*(q-1)
* Chọn e là số ngẫu nhiên sao cho 1 < e < ** (n) và nguyên tố cùng nhau với ** (n)
* Ta tìm được duy nhất d  e-1 mod ** (n) nhờ giải thuật Euclid mở rộng.
* Khoá công khai của A là <n, e>, khóa bí mật là <n, d>.

***Thuật toán sinh chữ ký*:** Người dùng A ký lên thông điệp m theo trình tự như sau:

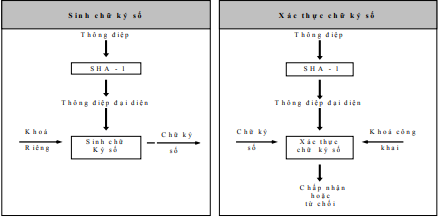
* Số hóa thông điệp Y thành các số nguyên m € [0, n-1]; m = R (Y).
* Tính s = md mod n

Chữ ký của A lên thông điệp m là s, gửi s cho B

***Thuật toán xác nhận chữ ký*:** Để xác thực chữ ký s là của A, B làm như sau:

* Xác thực khóa công khai của A là <n, e>
* Tính m =se mod n
* Xác nhận chữ ký nếu m € MR, ngược lại chối bỏ chữ ký của A.
* Khôi phục lại Y = R-1(m).

b) DSA - Digital Signature Algorithm ( chữ ký)



*Lược đồ chữ ký DSA*

DSA là một phiên bản đăc 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.

***Giải thuật sinh khoá:*** Mỗi một người dùng tạo một khoá công khai và một khoá bí mật tương ứng theo cách sau:

* Chọn một số nguyên tố q sao cho 2159 < q < 2160
* Chọn một số nguyên tố p sao cho 2511+64t < p < 2512+64t ở đó t  [0,8]
* Chọn g là một số nguyên bất kỳ nhỏ hơn p, α = g(p-1)/q mod p
* α khác 1
* Chọn số nguyên a sao cho : 1 ≤ a ≤ q – 1
* Tính β = α a mod p
* Khoá riêng của người dùng là a, khoá công khai là bộ (p,q, α, β)

***Giải thuật sinh chữ ký:*** Khi cần sinh chữ ký cho một thông điệp Z, thực thể phải làm những việc như sau:

* Chọn một số nguyên mật k, 0 < k < q – 1
* Tính γ = (αk mod p) mod q.
* Tính k-1 mod q.
* Tính δ = k-1(h(Z) + aγ) mod q
* Chữ ký của người dùng cho Z là cặp (δ, γ)

***Giải thuật xác nhận chữ ký:*** Khi cần xác nhận chữ ký cho một thông điệp m thực thể phải l àm nhữngviệc như sau:

* Dành lấy khoá công khai của thực thể ký (p,q, α, β)
* Nếu điều kiện : 0 < δ, γ < q không thoả mãn thì từ chối chữ ký
* Tính w = δ -1 mod q và h(Z)
* Tính e1 = w.h(Z) mod q và e2 = γw mod q
* Tính v = (αe1 βe2 mod p) mod q
* Nếu v = γ thì chấp nhận chữ ký ngoài ra thì từ chối.

1. **ElGamal/ Diffie Hellman**

Trong PGP thuật toán Diffie Hellman được gọi là DH và thường được dùng để trao đổi khoá và không được dùng để ký. Vì nếu dùng để ký thì chữ ký sẽ khá lớn. Trong lúc đó, ElGamal có thể dùng để ký và bảo mật mặc dù chữ ký sẽ phải dùng hai số cùng kích thước là 1024 bit trong khi RSA chỉ cần một con số có độ dài là 1024 bit. Đối với DSA thì chỉ cần 2 con số có độ dài là 160 bit.

### **2.1.2. Thuật toán khóa đối xứng**

1. **IDEA**

IDEA ra đời từ những năm 1991 có tên IPES (Improved Proposed Encyption Standard). Đến năm 1992 được đổi tên thành International Data Encrytion Algorithm. Tác giả là Xuejia Lai và James Massey. Thiết kế loại mã này dựa trên phép cộng modulo 2(OR), phép cộng modulo 216 và phép nhân modulo 216+1 (số nguyên tố 65537). Loại mã này rất nhanh về phần mềm (mọi chíp xử lý của máy tính cá nhân có thể thực hiẹn phép nhân bằng một lệnh đơn). Đến nay chưa có cuộc tấn công nào cho phép huỷ được hoàn toàn thuật toán IDEA. Do đó đây là một thuật toán có độ an toàn cao. IDEA là loại mã khối sử dụng một Chìa khóa 128 bit để mã hóa dữ liệu trong những khối 64 bít với 8 vòng lặp. Mỗi lần lặp IDEA sử dụng 3 phép toán khác nhau, mỗi phép toán thao tác trên hai đầu vào 16 bít để sản sinh một đầu ra 16 bít đơn. Ba phép toán đó là:

* Phép XOR theo bít
* Phép cộng modulo 216 với đầu vào và đầu ra là những số nguyên không dấu 16 bít. Hàm này láy hai số nguyên 16 bit làm dầu vào và sản sinh một tổng 16 bít; nếu bị tràn sang bít thứ 17, thì bit này bị vứt bỏ.
* Phép nhân số nguyên theo modulo 216+1. với đầu vào và đầu ra là những số nguyên 16 bít. Trừ trường hợp cả khối đều là 0 thì được xem như 216

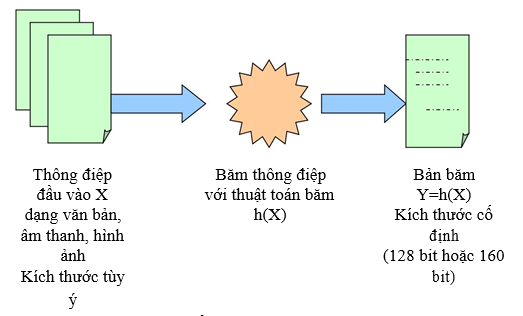
1. **3DES**

Thuật toán DES (Data Encryption Standard) được chính phủ Mỹ tạo ra năm 1977 (NIST và NSA) dựa trên các công việc mà IBM làm. DES thuộc loại mã khối 64 bits với khoá dài 64 bits. Thuật toán DES đầu tiên đã được nghiên cứu trong thời gian dài.

Thuật toán 3DES cải thiện độ mạnh của thuật toán DES bằng việc sử dụng một quá trình mã hóa và giải mã sử dụng 3 khóa. Các chuyên gia xác định rằng 3DES rất an toàn. Nhược điểm của nó là chậm hơn một cách đáng kể so với các thuật toán khác. Bản thân DES đã chậm do dùng các phép hoán vị bit. Lý do duy nhất để dùng 3DES là nó đó được nghiên cứu rất kỹ lưỡng.

### **2.1.3. Hàm băm mật mã (hash)**

Hàm băm là các thuật toán không sử dụng khóa để mã hóa (ở đây ta thường dùng thuật ngữ “băm” thay cho “mã hóa”), nó có nhiệm vụ (băm) thông điệp và đưa vào theo một thuật toán một chiều nào đó, rồi đưa ra một bản băm – văn bản đại diện – có kích thước cố định.



**Sơ đồ mô tả hàm băm thông điệp**

Hàm hash được định nghĩa là một ánh xạ H: X-->{0,1}k

Trong đó X là không gian các bản rõ độ dài tuỳ ý, {0,1}k là tập các dãy số 0,1 có độ dài K cho trước. Hàm Hash được xây dựng sao cho thỏa mãn các tính chất cơ bản sau:

1. Tính chất một chiều

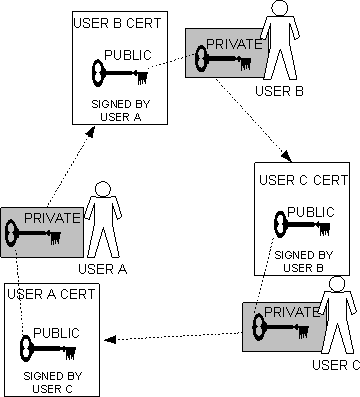
2. Hàm Hash yếu

3. Hàm Hash mạnh

Hàm hash được gọi là thoả mãn tính chất một chiều nếu cho trước giá trị hash (giá trị đã được rút gọn) Z thì về mặt tính toán không thể tìm được giá trị thông báo x sao cho Z = h(x). Tuy nhiên cho trước thông báo x thì việc xác định Z = h(x) lại được thực hiện nhanh chóng. Hàm hash được gọi là có tính chất yếu nếu cho trước một thông báo x thì về mặt tính toán không thể tìm được một thông báo x’≠x (x,x’X) sao cho h(x’)=h(x).Còn hàm hash được gọi là có tính chất mạnh nếu tổng thực hành không thể tìm được 2 thông báo x, x’X khác nhau sao cho h(x’) = h(x). Các phiên bản trước PGP sử dụng hàm băm MD5 để băm dữ liệu còn hiện nay MD5 được thay bằng thuật toán SHA.

## **2.2. Mô hình kiến trúc và hoạt động của PGP**

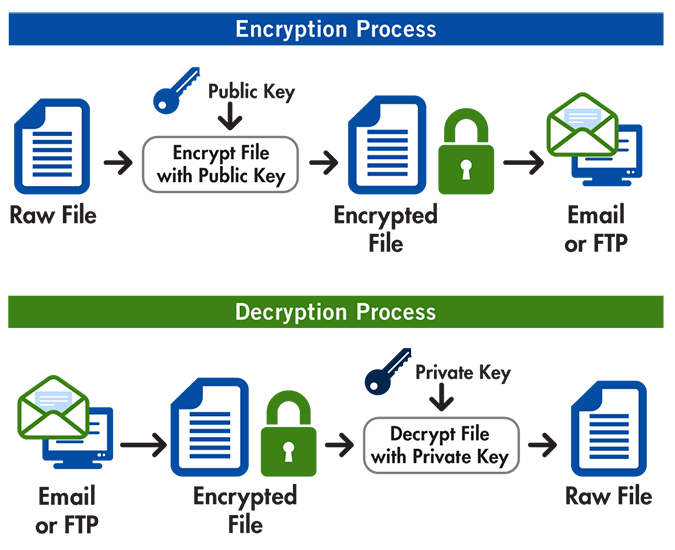
**a) Sơ đồ kiến trúc tổng quan về mô hình PGP**



Hai dịch vụ chính mà PGP cung cấp cho người dùng là: mã hóa và xác thực. 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?

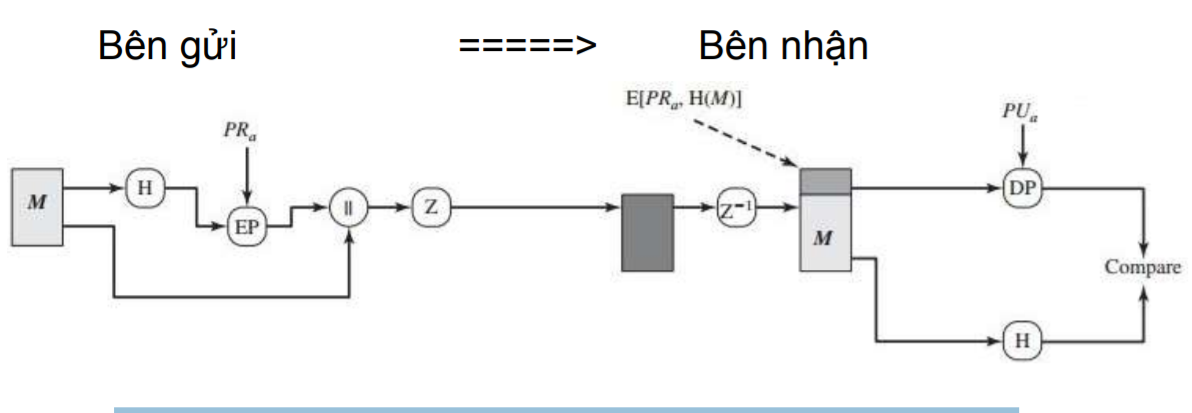
Trong trường hợp của PGP, những dịch vụ của nó dựa vào ba giải thuật: IDEA (mã hóa khóa bí mật), RSA (mã hóa khóa công khai) và MD5 (Hàm băm an toàn). Trong phần này chúng ta sẽ nghiên cứu toàn bộ những bước thực hiên của PGP trong truyền và nhận thông điệp và những thông báo xử lý thông điệp. Sau đó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

**b) Hoạt động PGP**



Đầu tiên ta sẽ quy ước các đối tượng như mô tả dưới đây:

* Ks: khóa phiên được sử dụng trong sơ đồ mã hóa đối xứng
* PRa:khóa riêng của user A, được sử dụng trong lược đồ mã hóa khóa công khai.
* PUa: khóa công khai của user A, được sử dụng trong lược đồ mã hóa khóa công khai.
* EP: mã hóa khóa công khai.
* DP: giải mã khóa công khai.
* EC: mã hóa đối xứng.
* DC: giải mã đối xứng.
* H: Hàm băm.
* ||: ghép.
* Z: nén sử dụng thuật toán ZIP.
* R64: chuyển đổi sang định dạng cơ số 64

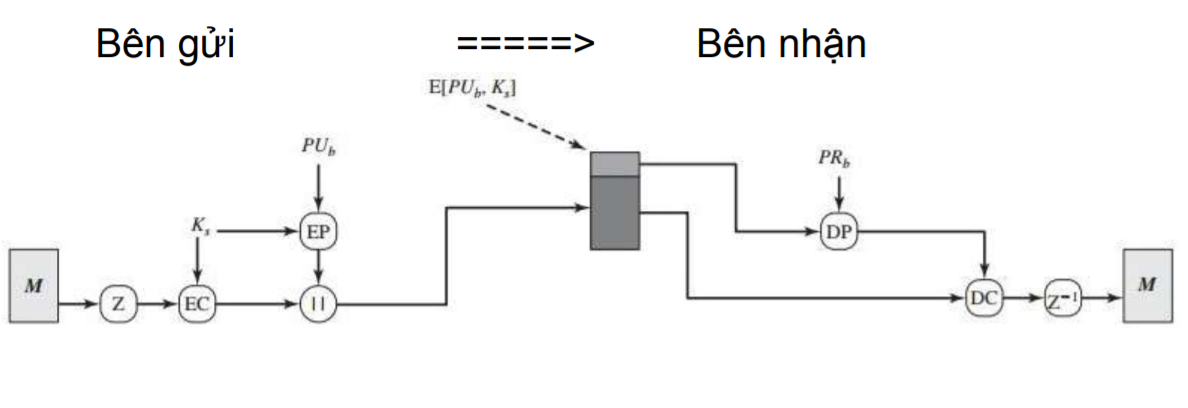


* **Mô hình PGP chỉ đảm bảo xác thực thông điệp(Authentication only):**

PGP cung cấp dịch vụ chữ kí điện tử nhằm đảm bảo tính xác thực thông điệp giữa bên người gửi và bên người nhận

**Hoạt động:**

* Người gửi tạo 1 tin nhắn.
* Giải thuật SHA-1 được sử dụng để tạo 1 hash code 160 bit của tin nhắn.
* Hash code được mã hóa bằng RSA bằng khóa riêng của người gửi và kết quả được gửi trước cho tin nhắn.
* Người nhận sử dụng RSA với khóa công khai của người gửi để giải mã và khôi phục mã băm.
* Người nhận tạo hash code mới cho tin nhắn và so sánh nó với hash code được giải mã. Nếu hai kết quả khớp nhau, tin nhắn được chấp nhận là xác thực.



* **Mô hình PGP chỉ đảm bảo tính bí mật thông điệp(Confidentiality):**

Một trong những dịch vụ khác của PGP là bảo mật(mã hóa tin nhắn để truyền hoặc lưu trữ tệp cục bộ). Trong cả hai trường hợp, thuật toán mã hóa đối xứng CAST-128 có thể được sử dụng. Ngoài ra có thể sử dụng IDEA hoặc 3DES.

chế độ phản hồi mật mã 64 bit (CFB) được sử dụng. Trong PGP, mỗi khóa đối xứng chỉ được sử dụng một lần. Khóa phiên được liên kết với tin nhắn. Để bảo vệ khóa, nó được mã hóa bằng khóa chung của máy thu.

**Hoạt động:**

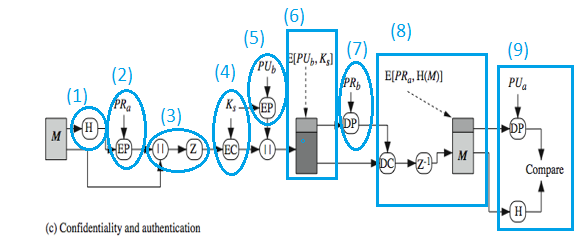
1. Người gửi tạo một tin nhắn và một số 128 bit ngẫu nhiên chỉ được sử dụng làm khóa phiên cho tin nhắn này.

2. Tin nhắn được mã hóa bằng CAST-128 (hoặc IDEA hoặc 3DES) bằng khóa phiên.

3. Khóa phiên được mã hóa bằng RSA bằng khóa công khai của người nhận và được gửi trước tin nhắn.

4. Người nhận sử dụng RSA với khóa riêng của mình để giải mã và khôi phục khóa phiên.

5. Khóa phiên được sử dụng để giải mã tin nhắn.



* **Mô hình PGP đảm bảo tính bí mật và xác thực thông điệp():Kết hợp giữa 2 mô hình ở trên**.

1. Sau khi tin nhắn được viết, 1 hash code được tạo cho nó.

2. Sau đó, nó được mã hóa với khóa riêng của người gửi để tạo ra chữ kí điện tử, vì thế người gửi là người duy nhất có thể tạo ra chữ kí đó. Ở bước này tin nhắn chưa được mã hóa.

3. Giá trị băm đã được mã hóa và tin nhắn được nén lại.

4. Sau đó, chúng được mã hóa bằng 1 khóa bí mật; khóa phiên chỉ được sử dụng trong trường hợp này của giao thức.

5. Khóa phiên này được mã hóa với khóa công khai của người nhận vậy nên chỉ có người nhận mới có thể giải mã nó bằng khóa riêng của người đó.

6. Chuyễn đến người nhận.

7. Khi người nhận nhận được tin nhắn, họ dùng khóa riêng của họ để giải mã khóa phiên.

8. Sau đó họ sử dụng khóa phiên để mở tin nhắn và giá trị băm được mã hóa.

9. Khóa công khai của người gửi được sử dụng để giải mã chữ kí và tìm ra giá trị băm. Người nhận tạo ra giá trị băm từ tin nhắn sau đó so sánh với giá trị băm vừa giải mã được. Nếu tin nhắn không bị sửa đổi, hai giá trị sẽ trùng nhau.

III. Demo:

* Tạo mail server bằng hMailServer

# 

* Tạo domain : netvn.com

# 

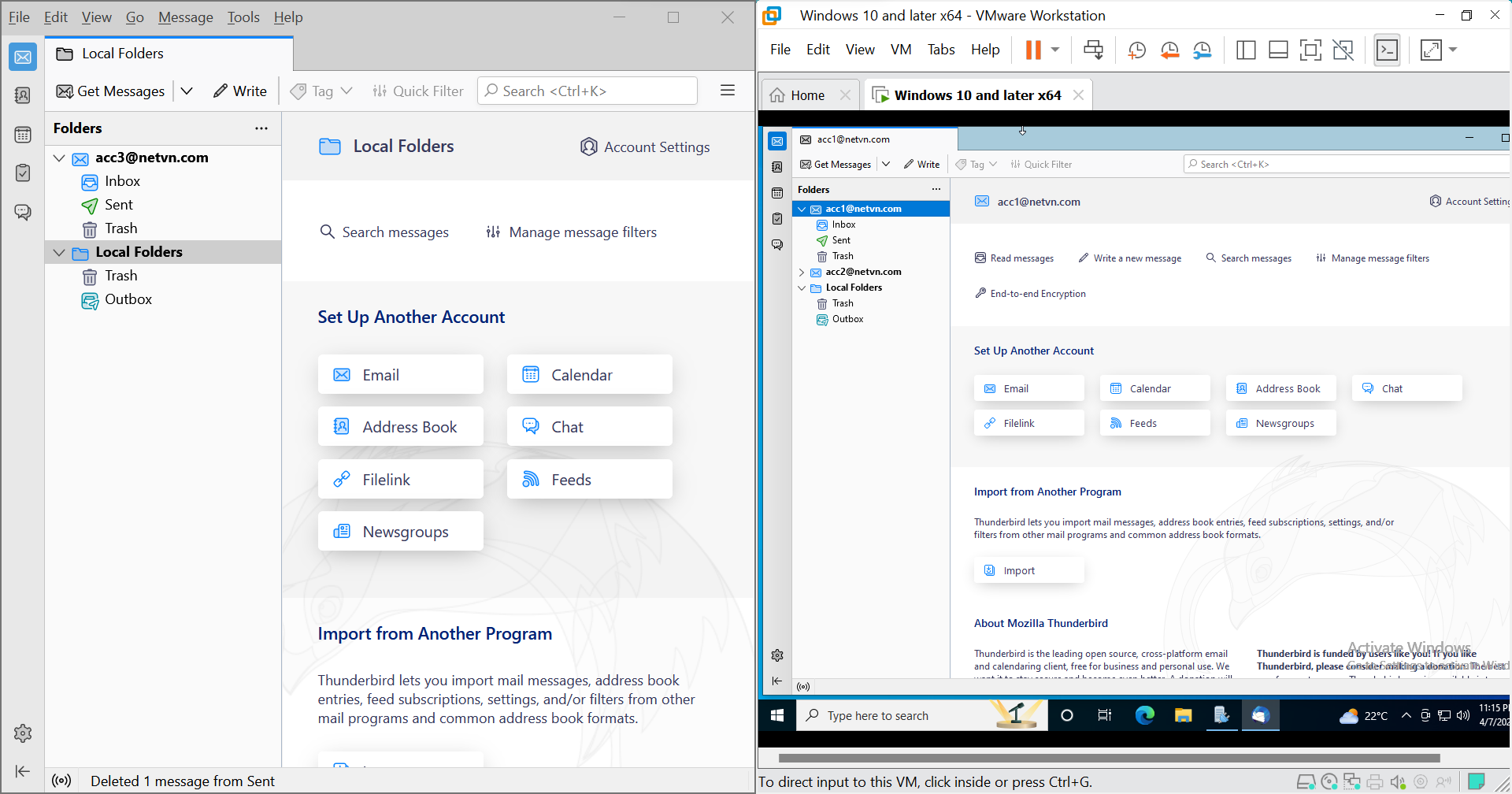
* Tạo ra các tài khoản mail server

# 

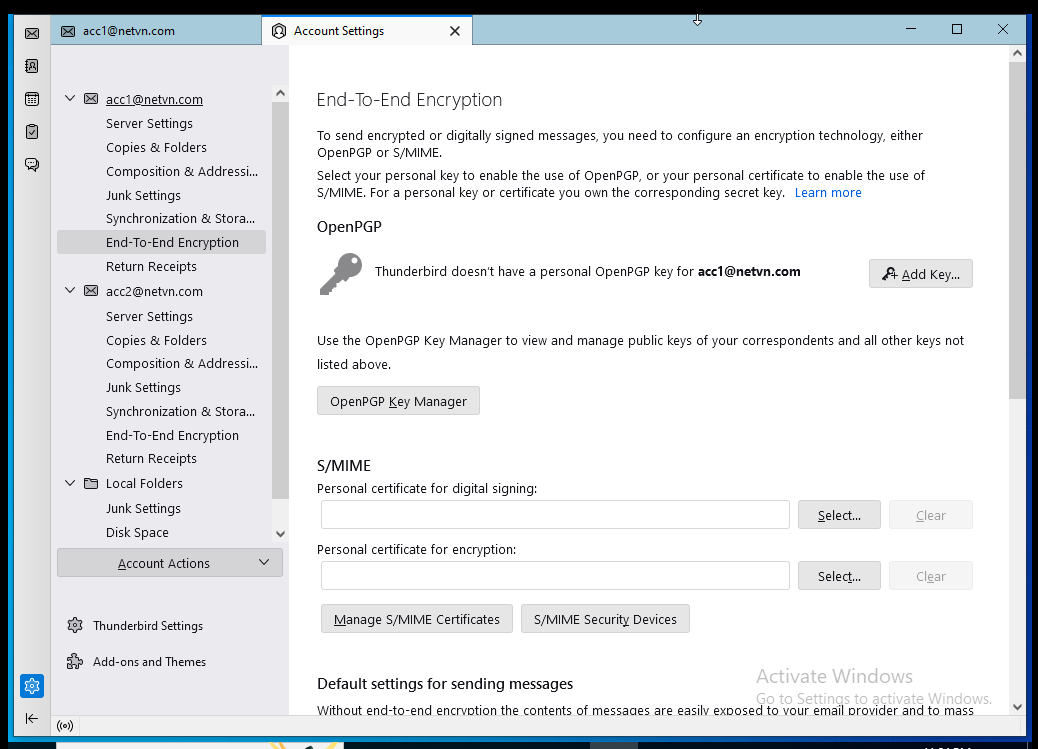
- Sử dụng 2 máy để gửi thư cho nhau gồm máy thật và máy ảo

+ Máy thật sử dụng tài khoản [acc3@netvn.com](mailto:acc3@netvn.com)

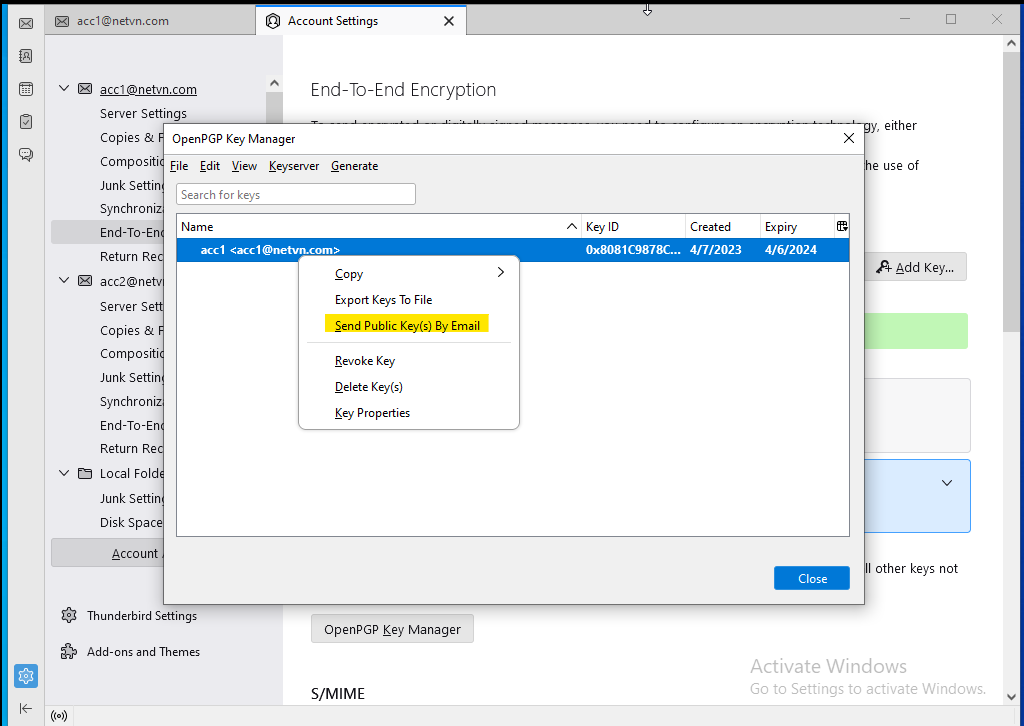
+) Máy ảo sử dụng tài khoản [acc1@netvn.com](mailto:acc1@netvn.com)



* Tạo khóa công khai và khóa bí mật cho tài khoản [acc1@netvn.com](mailto:acc1@netvn.com)

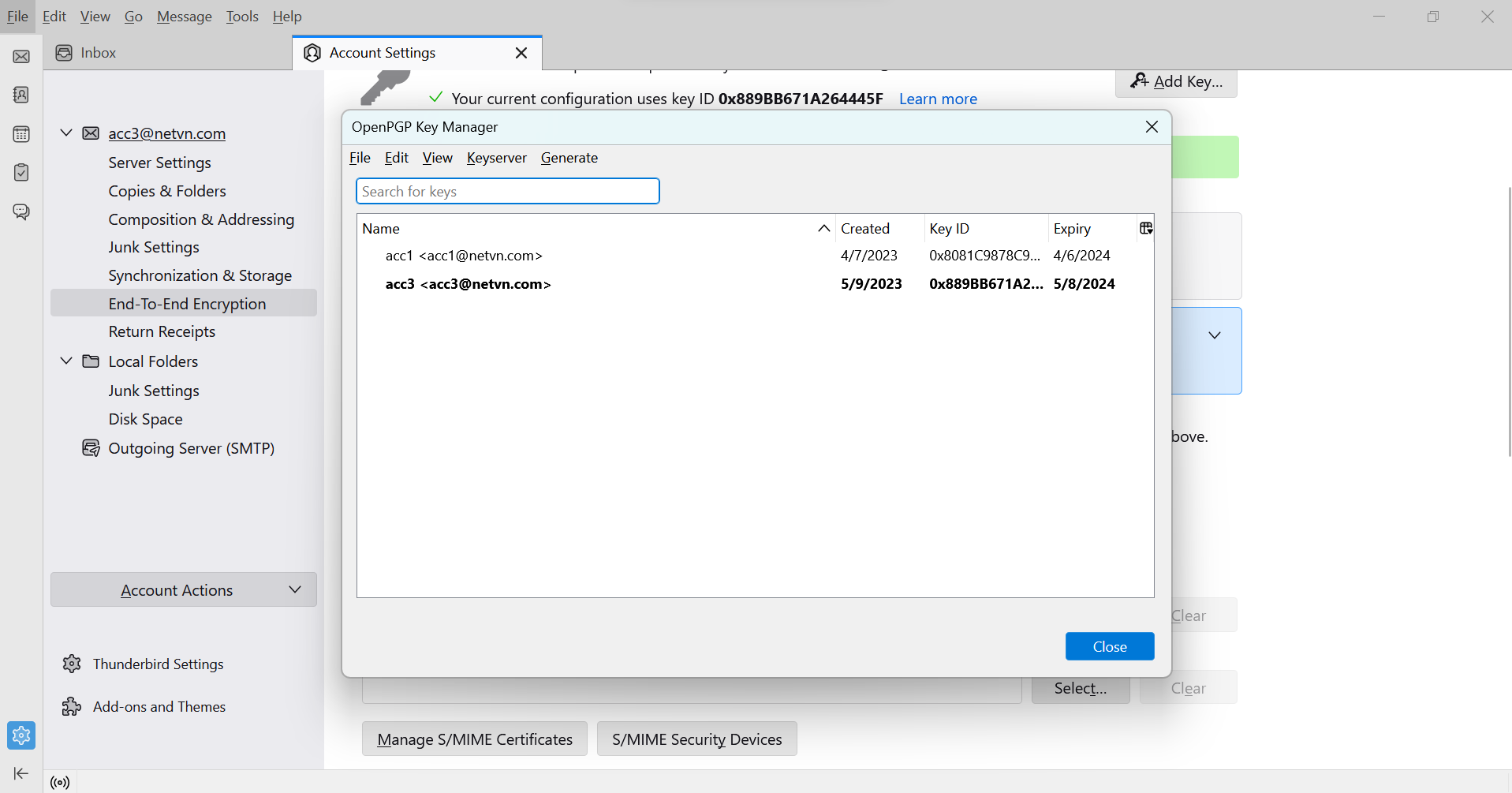


- Gửi khóa công khai của [acc1@netvn.com](mailto:acc1@netvn.com) cho [acc3@netvn.com](mailto:acc3@netvn.com)

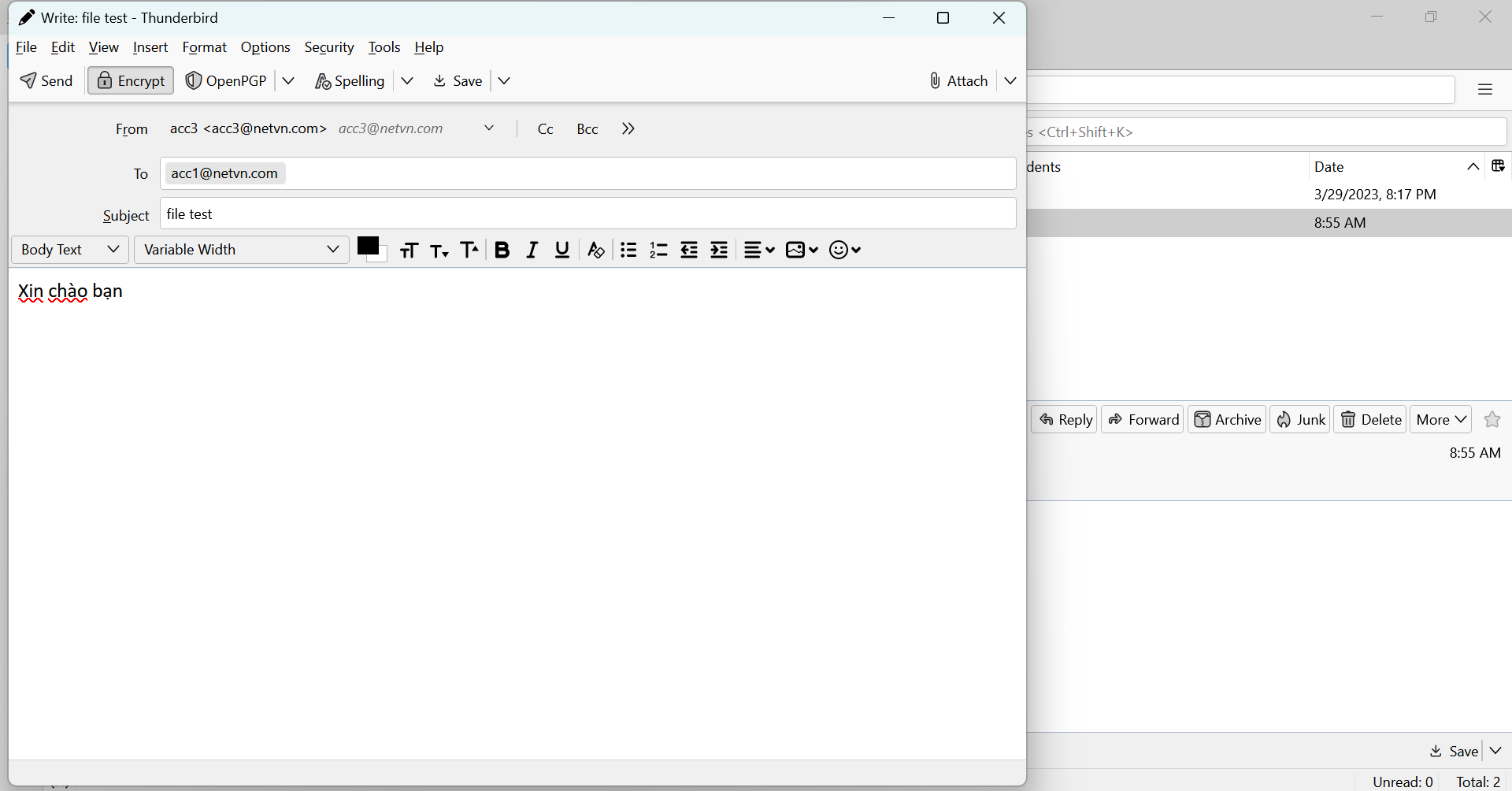


- Lưu lại khóa công khai của tài khoản [acc1@netvn.com](mailto:acc1@netvn.com) vào thư mục public key

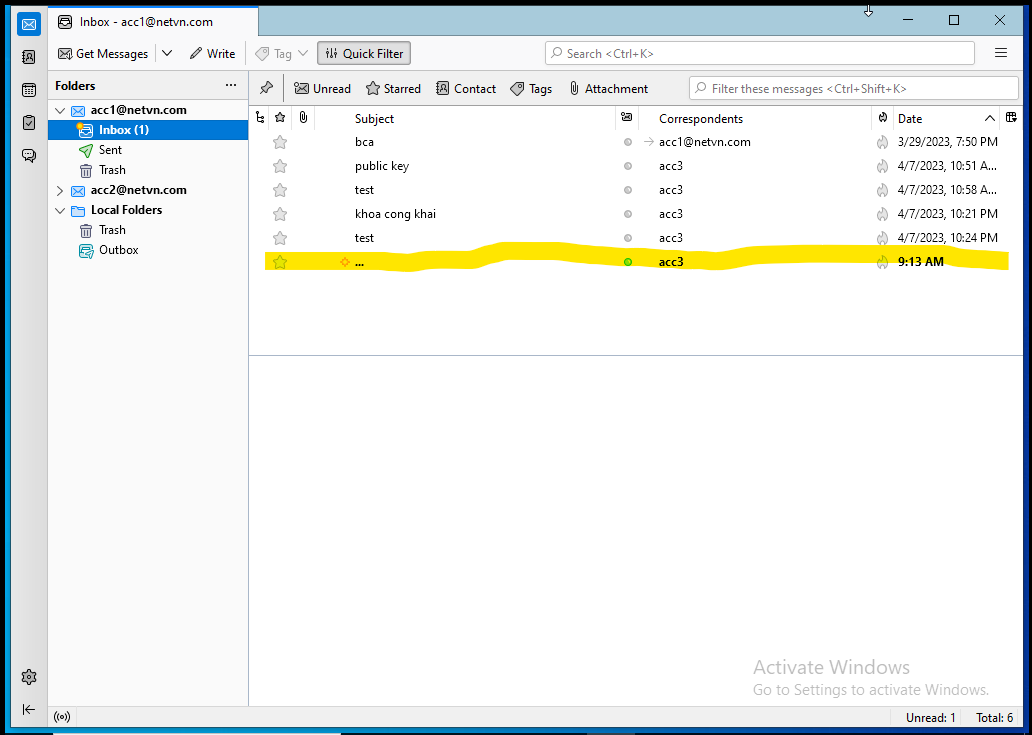
- Thêm khóa công khai của tài khoản [acc1@netvn.com](mailto:acc1@netvn.com) vào OpenPGP Key Manager



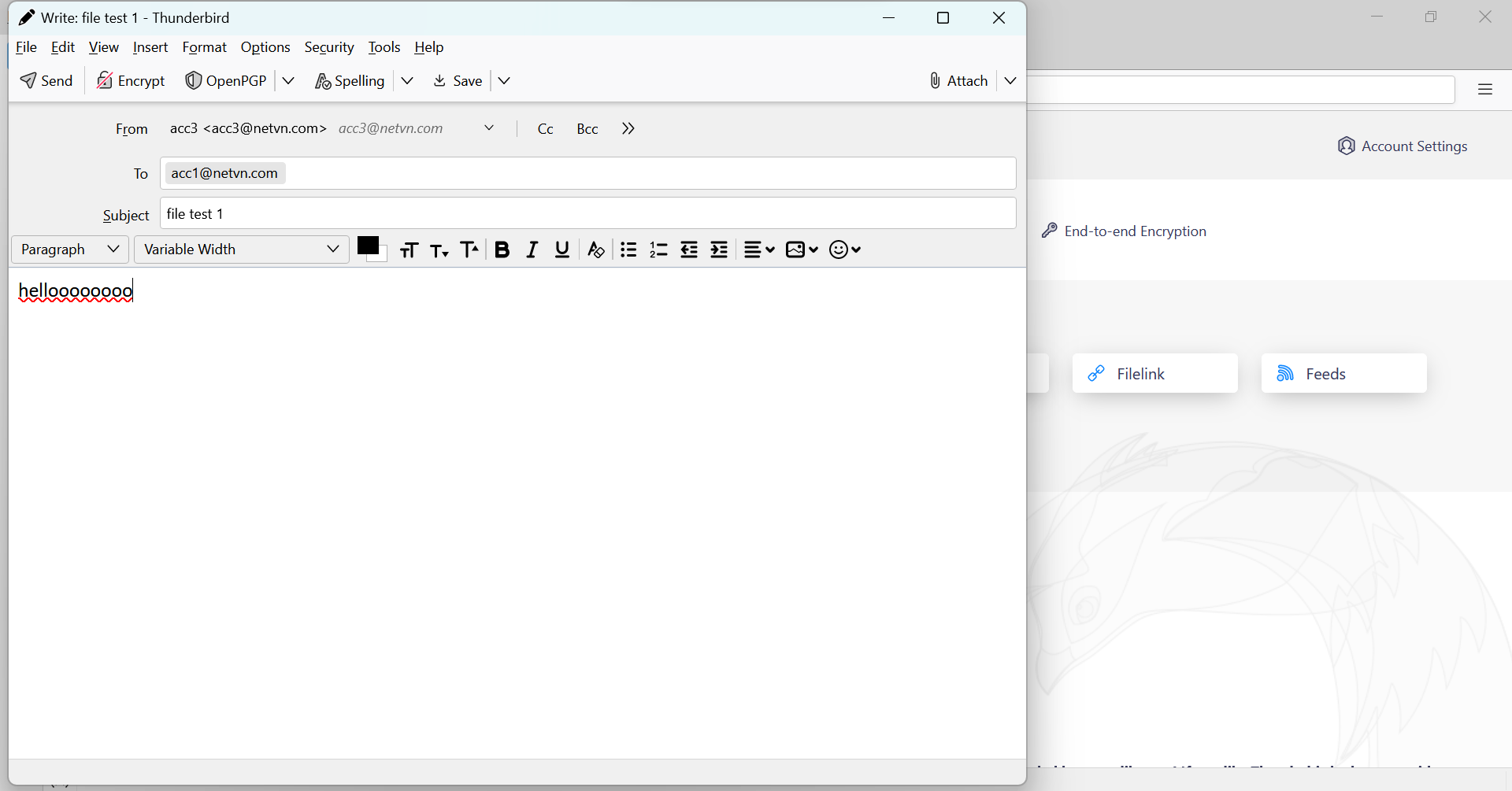
- Gửi thư được mã hóa từ tài khoản [acc3@netvn.com](mailto:acc3@netvn.com) đến tài khoản [acc1@netvn.com](mailto:acc1@netvn.com)

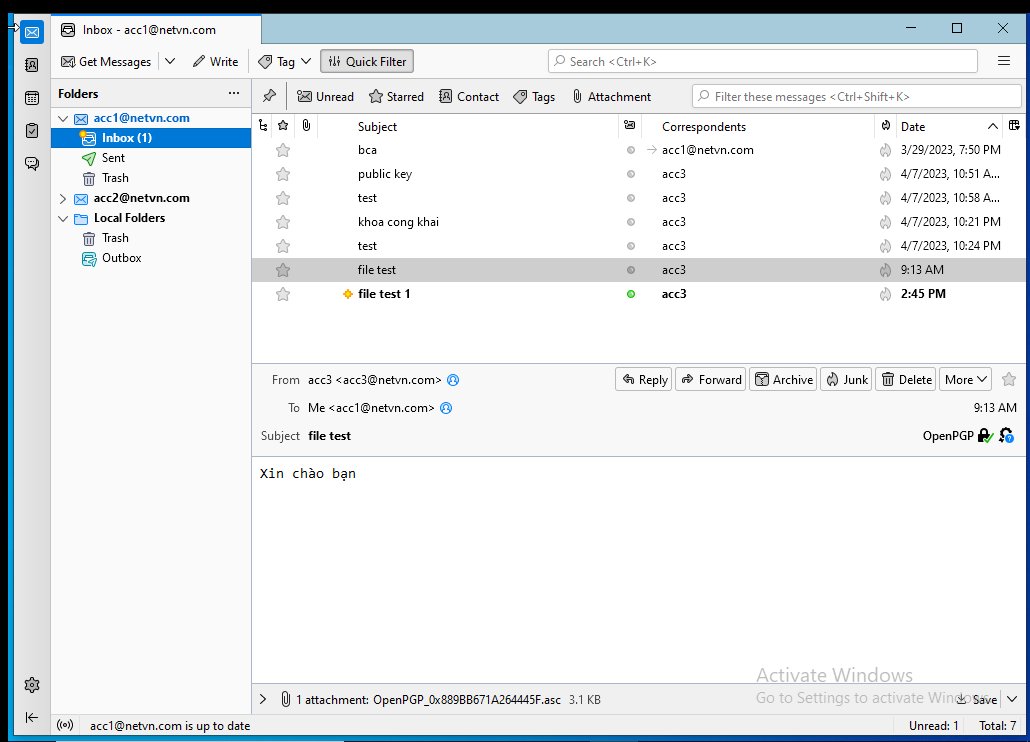


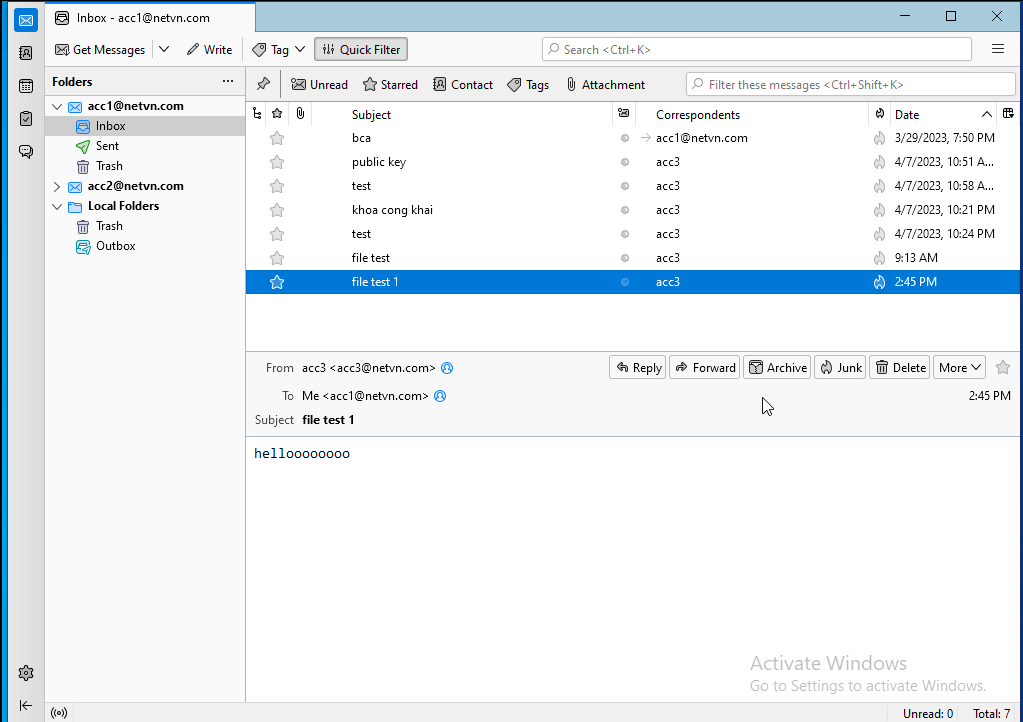
- Xem thư bên [acc1@netvn.com](mailto:acc1@netvn.com)

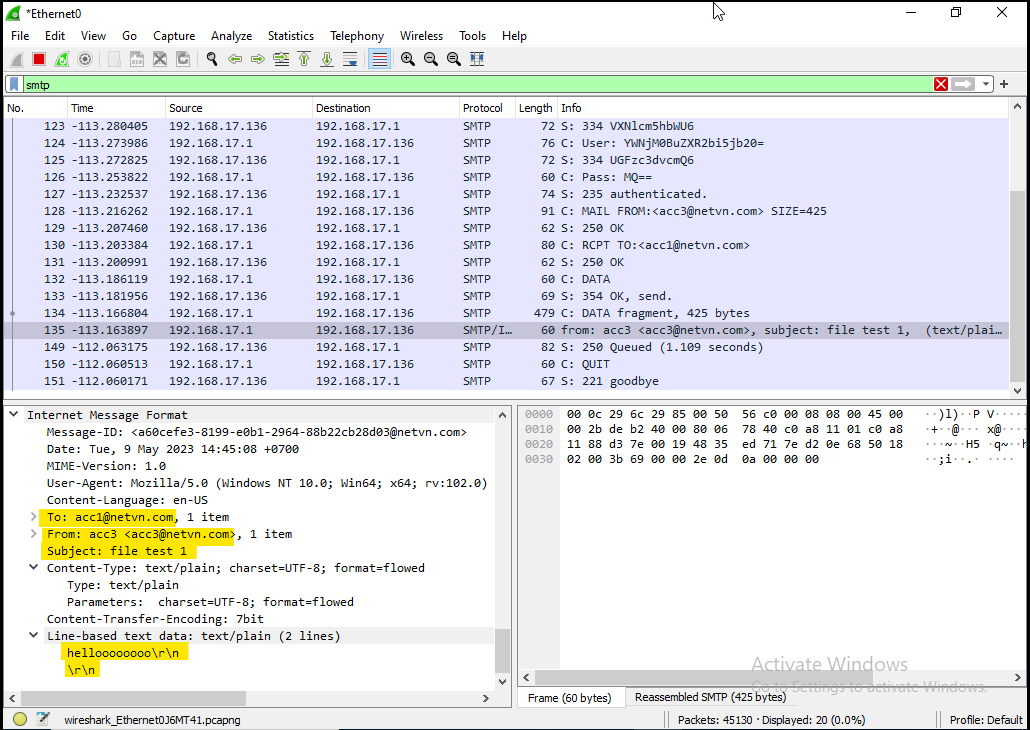


- Khi chưa mã hóa:

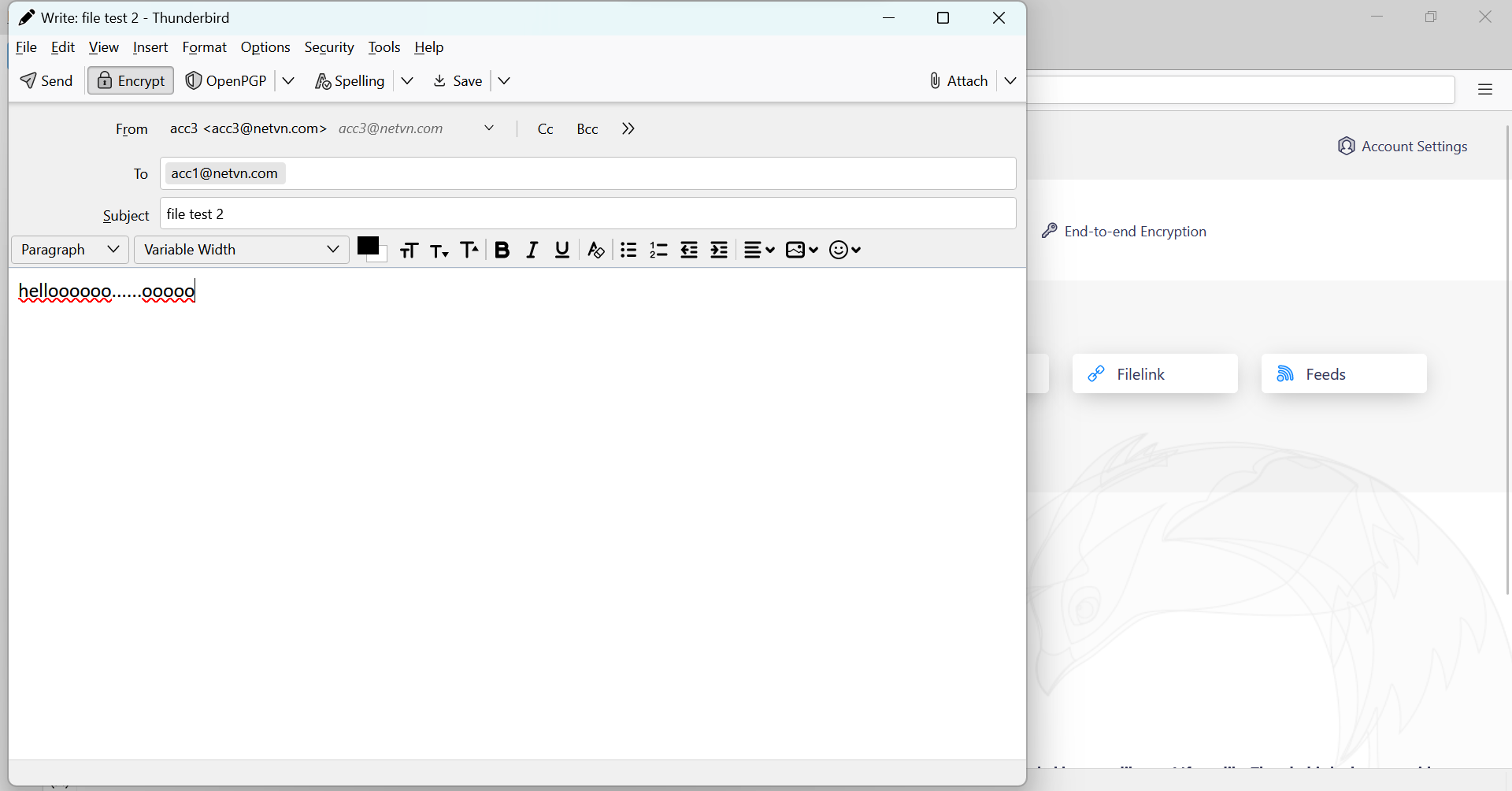


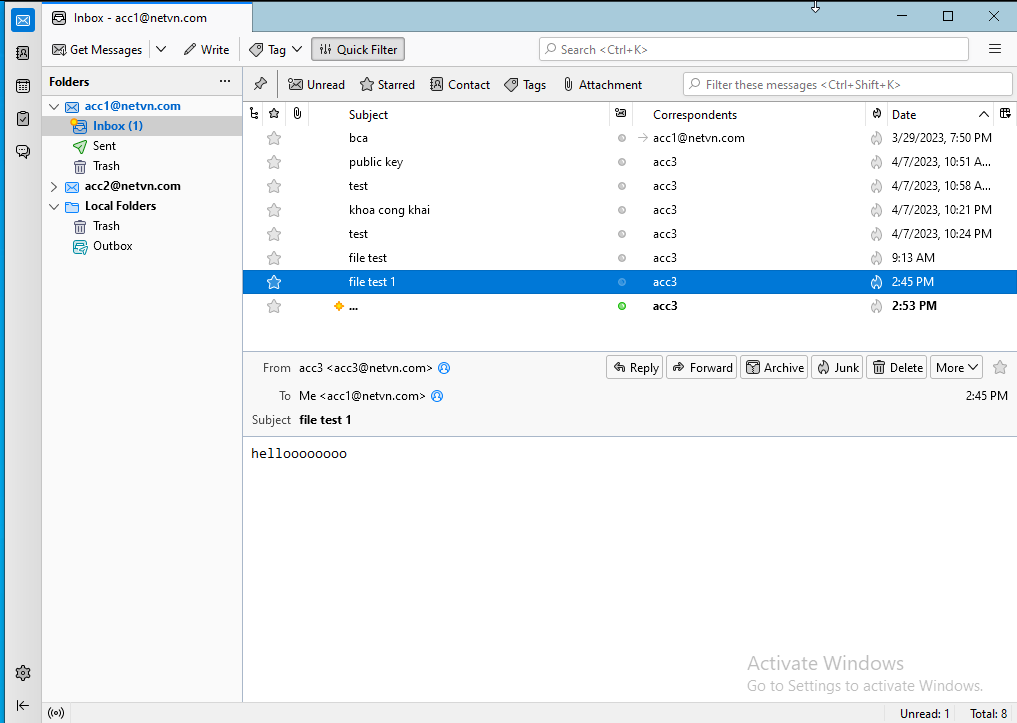


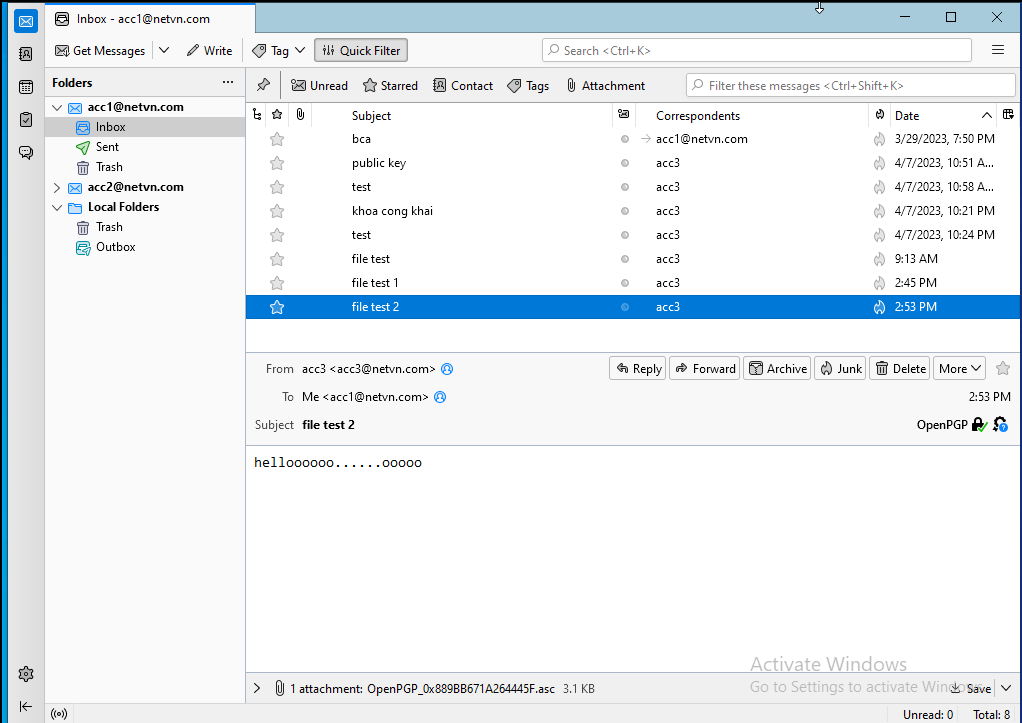


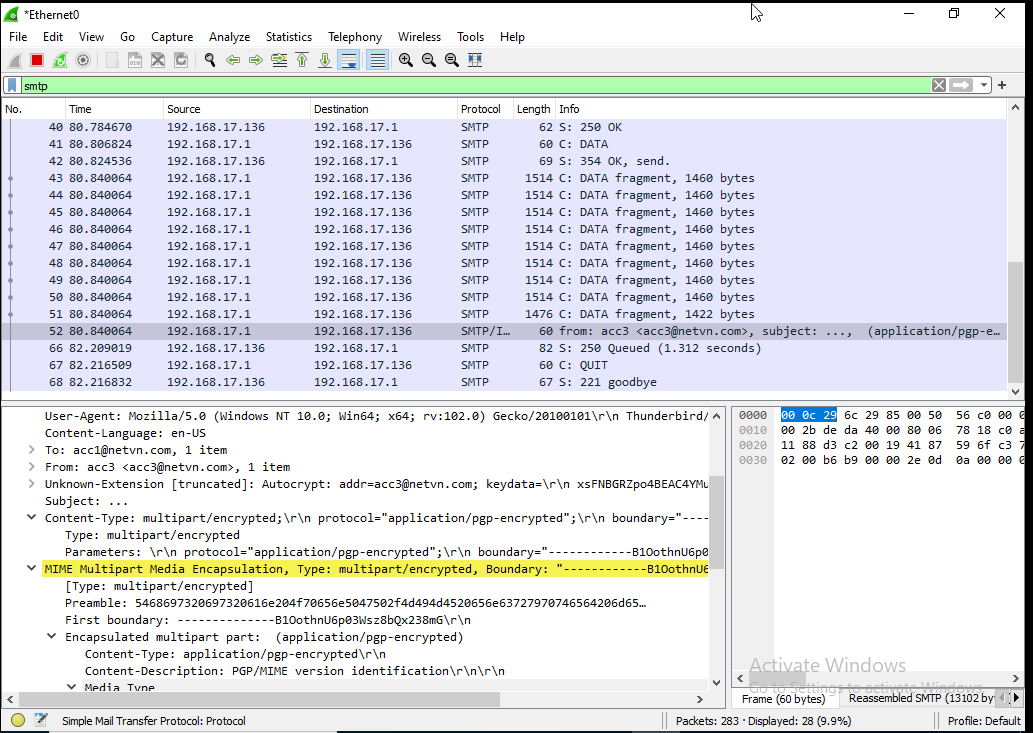


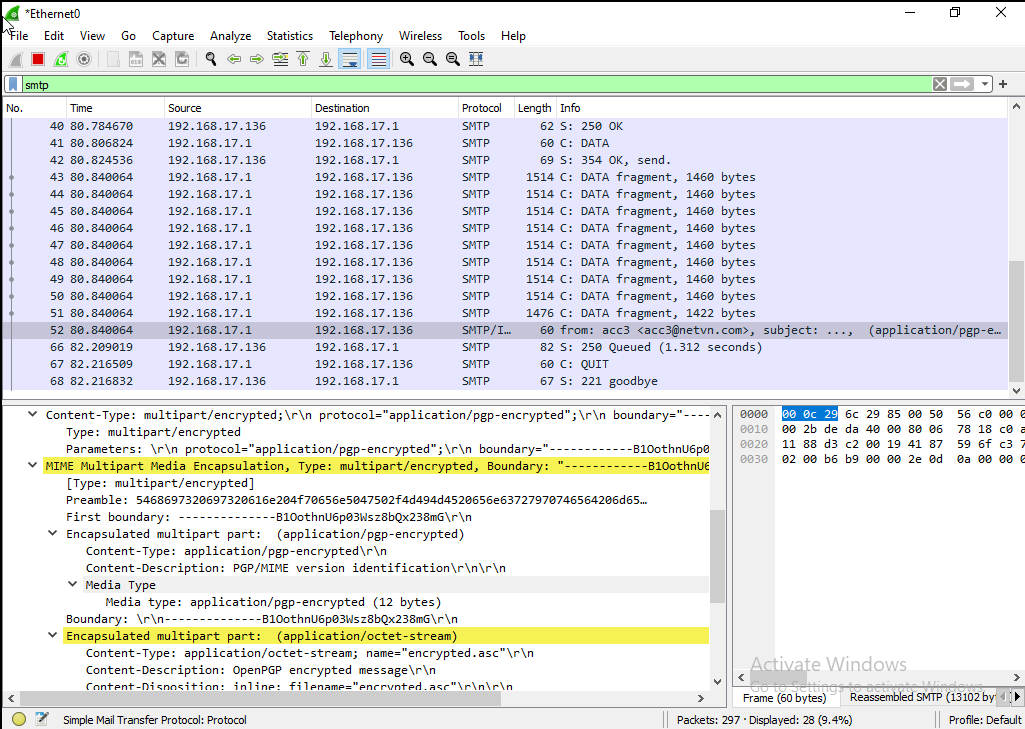
- Sau khi mã hóa

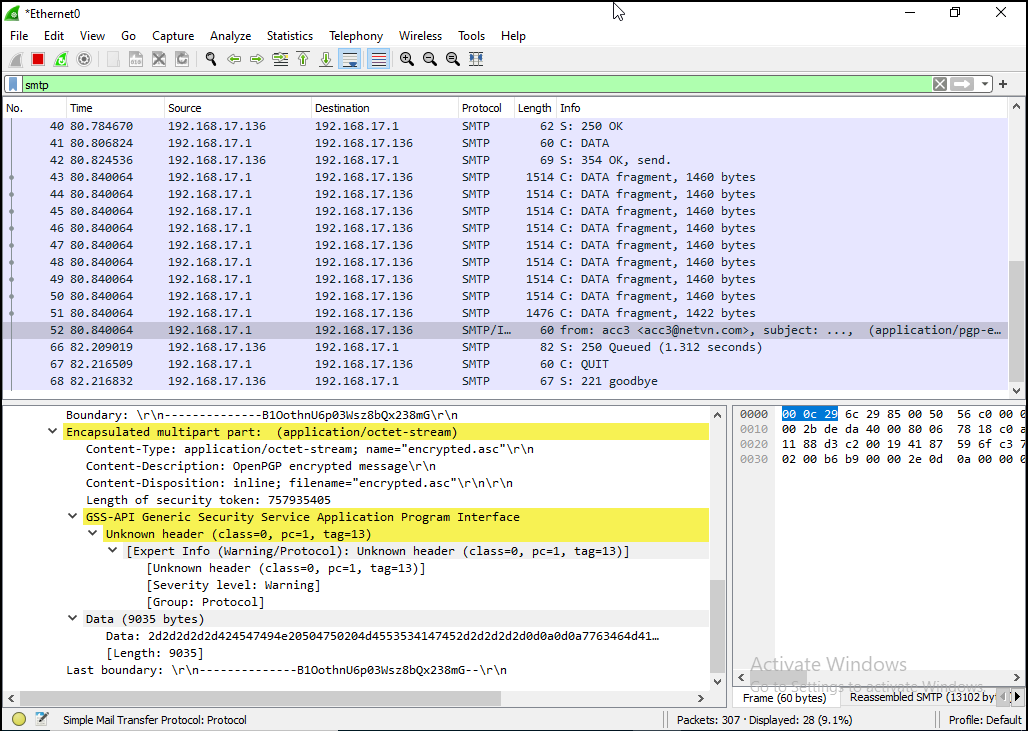












# **KẾT LUẬN**

Trong phần báo nhóm em đã mô tả được nội dung sau :

* Mô hình PGP về các giải thuật đã sử dụng trong PGP
* Mô hình cấu trúc và cách thức hoạt động và sử dụng PGP.
* Cách cài đặt và sử dụng phương thức PGP
* Ứng dụng của PGP trong bảo mật thông tin

Tuy nhiên về mặt hạn chế sau:

* Hạn chế về kiến thức chưa tìm hiểu sâu về PGP, mới chỉ tìm hiểu về tổng quan PGP qua tài liệu thầy cung cấp và tài liệu qua Internet.
* Hạn chế về kiến thức về các thuật toán mã hóa
* Hạn chế tài liệu khi chưa có nhiều kiến thức về bảo mật

TÀI LIỆU THAM KHẢO

[1]. <http://en.wikipedia.org/wiki/Pretty_Good_Privacy>

[2]. http://en.wikipedia.org/wiki/Mật\_mã\_hóa\_khóa\_công\_khai

[3]. <http://www.philzimmermann.com/EN/essays/WhyIWrotePGP.html> - Why I Wrote PGP- Philip Zimmermann

[4]. <http://www.pitt.edu/~poole/PGP.htm> -PGP Installation and use for Dummies

[5]. An toàn thông tin (Mạng máy tính, truyền tin số và truyền dữ liệu) - NXB

Khoa học và Kỹ thuật