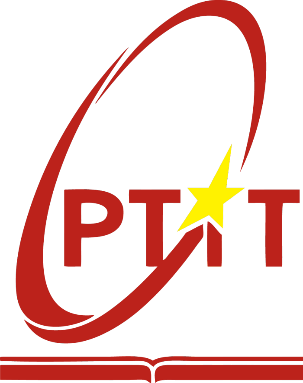
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA AN TOÀN THÔNG TIN**

-----🙞🙜🕮🙞🙜-----



**Báo Cáo Bài Tập Lớn**

**Môn : An Toàn Phần Mềm**

**Lớp :** D18-199

Giáo Viên Hướng Dẫn : Hoàng Xuân Dậu

Nhóm : 11

Đề Tài : Nghiên cứu về giao thức PGP và ứng dụng

Thành Viên Thực Hiện :

* Trần Thế Quân - B18DCAT194
* Vũ Mạnh Phương - B18DCAT189
* Nguyễn Ngọc Quý - b16dcat133
* Bùi Minh Quân - B18DCAT192
* Trịnh Quang Phong - B18DCAT185

Mục Lục

[1](#_Toc68542996)

[Giới Thiệu PGP 3](#_Toc68542997)

[1.1 Giới thiệu chung về giao thức PGP 3](#_Toc68542998)

[1.2 Mục đích sử dụng PGP 3](#_Toc68542999)

[1.3 Phương thức hoạt động của PGP 3](#_Toc68543000)

[I. Mô Hình Hoạt Động của PGP 5](#_Toc68543001)

[1.1. PGP hỗ trợ các mô hình hoạt động nào ? 5](#_Toc68543002)

[1.2. Mô hình PGP chỉ đảm bảo tính xác thực thông điệp 5](#_Toc68543003)

[1.3. Mô hình PGP chỉ đảm bảo tính bí mật thông điệp. 6](#_Toc68543004)

[1.4. Mô hình PGP đảm bảo tính bí mật và xác thực thông điệp. 7](#_Toc68543005)

[II. Nền tảng mã hoá khoá bí mật cho PGP (DES, 3-DES, AES) 8](#_Toc68543006)

[2.1 DES. 8](#_Toc68543007)

[2.2 3DES 14](#_Toc68543008)

[2.3. AES 17](#_Toc68543009)

[3. Mã Hóa Bất Đối Xứng. 23](#_Toc68543010)

[4.1 Các thuật toán 23](#_Toc68543011)

[4.1 Hàm hash 25](#_Toc68543012)

[4.2 Mô hình kiến trúc 25](#_Toc68543013)

[4.3 Hoạt động 26](#_Toc68543014)

[5.Chữ Ký Số. 30](#_Toc68543015)

[5.1 Nén 31](#_Toc68543016)

[5.2 Mã hóa và giải mã thông điệp 31](#_Toc68543017)

[Tài liệu tham khảo: 34](#_Toc68543018)

# Giới Thiệu PGP

## 1.1 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.

## 1.2 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 đổ email hoặc chuyển file, chữ ký số…

## 1.3 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 dự 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.

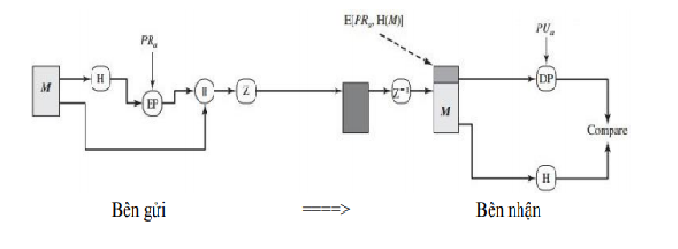
OpenPGP cung cấp các chữ ký tin cậy có thể được sử dụng để tạo ra các nhà cung cấp chứng thực số (CA). Một chữ ký tin cậy có thể chứng tỏ rằng một khóa thực sự thuộc về một người sử dụng và người đó đáng tin cậy để ký xác nhận một khóa của mức thấp hơn. Một chữ ký có mức 0 tương đương với chữ ký trong 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 rong 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.

1. Mô Hình Hoạt Động của PGP
   1. PGP hỗ trợ các mô hình hoạt động nào ?

PGP hỗ trợ 3 mô hình hoạt động, bao gồm (1) Mô hình PGP chỉ đảm bảo tính xác thực thông điệp, (2) Mô hình PGP chỉ đảm bảo tính bí mật thông điệp và (3) Mô hình PGP đảm bảo tính bí mật và xác thực thông điệp. Để thuận tiện cho mô tả hoạt động của các mô hình PGP, gọi H là hàm băm một chiều, EC là hàm mã hóa khóa đối xứng, DC là hàm giải mã khóa đối xứng, EP là hàm mã hóa khóa bất đối xứng, DP là hàm giải mã khóa bất đối xứng, Z là hàm nén, Z-1 là hàm giải nén, PUa là khóa công khai của bên A, PRa là khóa riêng của bên A, PUb là khóa công khai của bên B, PRb là khóa riêng của bên B và Ks là khóa phiên.

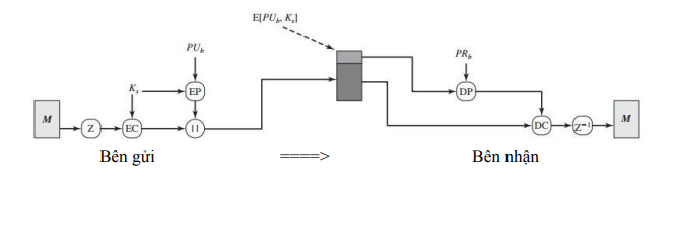
* 1. Mô hình PGP chỉ đảm bảo tính xác thực thông điệp



Hình 1: Mô hình PGP chỉ đảm bảo tính xác thực thông điệp.

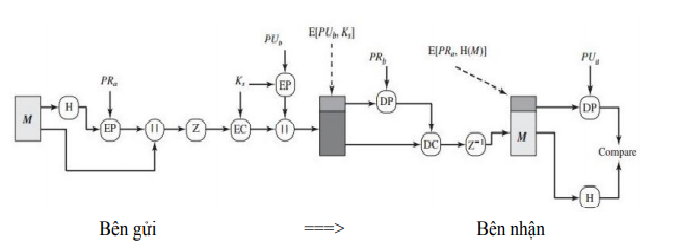
Hình 1 biểu diễn mô hình PGP chỉ đảm bảo tính xác thực thông điệp truyền. Theo đó, mô hình này sử dụng chữ ký số để xác thực tính toàn vẹn và chủ thể gửi thông điệp. Điều kiện thực hiện mô hình này là bên gửi A phải sở hữu cặp khóa (khóa công khai PUa và khóa riêng PRa). Quá trình thực hiện gửi/nhận thông điệp M đảm bảo tính xác thực tại mỗi bên như sau: - Bên gửi A: + Tính toán giá trị băm (giá trị đại diện) của thông điệp M sử dụng hàm băm H; + Sử dụng khóa riêng PRa để mã hóa (ký) giá trị băm của M tạo thành chữ ký số; + Ghép chữ ký số vào thông điệp M; + Nén thông điệp và chữ ký số sử dụng hàm nén Z; + Gửi bản dữ liệu đã nén cho người nhận. - Bên nhận B: + Giải nén dữ liệu nhận được sử dụng hàm Z-1 ; + Tách chữ ký số khỏi thông điệp M và sử dụng khóa công khai của bên gửi PUa để kiểm tra (giải mã) chữ ký số để khôi phục giá trị băm h1. Bên gửi A có thể sử 111 dụng các phương pháp trao đổi khóa công để chuyển khóa công khai PUa cho bên nhận; + Tính toán giá trị băm h2 của thông điệp M sử dụng hàm băm H; + So sánh 2 giá trị băm h1 và h2, nếu h1 = h2 thì thông điệp truyền là toàn vẹn và thông điệp được gửi bởi bên gửi A. Nếu h1 ≠ h2 thì thông điệp M có thể đã bị sửa đổi, hoặc không được ký và gửi bởi bên gửi A.

* 1. Mô hình PGP chỉ đảm bảo tính bí mật thông điệp.



Hình 2 : Mô hình PGP chỉ đảm bảo tính bí mật thông điệp.

* Biểu diễn mô hình PGP chỉ đảm bảo tính bí mật thông điệp truyền. Theo đó, mô hình này sử dụng kết hợp giữa mã hóa khóa đối xứng và mã hóa khóa bất đối xứng để đảm bảo tính bí mật của thông điệp. Điều kiện thực hiện mô hình này là bên nhận B phải sở hữu cặp khóa (khóa công khai PUb và khóa riêng PRb). Quá trình thực hiện gửi/nhận thông điệp M đảm bảo tính bí mật tại mỗi bên như sau: - Bên gửi A: + Nén thông điệp M sử dụng hàm nén Z; + Sinh khóa phiên Ks và sử dụng khóa Ks để mã hóa thông điệp M sử dụng hàm mã hóa đối xứng EC; + Sử dụng khóa công khai PUb của bên nhận B để mã hóa khóa phiên Ks sử dụng hàm mã hóa bất đối xứng EP. Bên nhận B có thể sử dụng các phương pháp trao đổi khóa công khai đã nêu ở mục 3.5.3 để chuyển khóa công khai PUb cho bên gửi; + Ghép chữ bản mã của Ks vào bản mã của thông điệp M; + Gửi bản mã dữ liệu cho người nhận. - Bên nhận B: + Tách bản mã của Ks vào bản mã của thông điệp M; + Giải mã bản mã Ks sử dụng hàm giải mã khóa bất đối xứng DP và khóa riêng PRb để khôi phục Ks; + Sử dụng khóa phiên Ks và hàm giải mã khóa đối xứng DC để giải mã khôi phục thông điệp đã nén M; + Giải nén khôi phục thông điệp M sử dụng hàm Z-1 ;
  1. Mô hình PGP đảm bảo tính bí mật và xác thực thông điệp.



Hình 3: Mô hình PGP đảm bảo tính bí mật và xác thực thông điệp

* Biểu diễn mô hình PGP đảm bảo tính xác thực và bí mật thông điệp truyền. Theo đó, mô hình này sử dụng chữ ký số để xác thực tính toàn vẹn và chủ thể gửi thông điệp. Đồng thời mô hình sử dụng kết hợp giữa mã hóa khóa đối xứng và mã hóa khóa bất đối xứng để đảm bảo tính bí mật của thông điệp. Điều kiện thực hiện mô hình này là bên gửi A phải sở hữu cặp khóa (khóa công khai PUa và khóa riêng PRa) và bên nhận B phải sở hữu cặp khóa (khóa công khai PUb và khóa riêng PRb). Mô hình này là sự kết hợp của mô hình PGP chỉ đảm bảo tính xác thực và mô hình PGP chỉ đảm bảo tính bí mật. Theo đó, bên gửi A thực hiện ký và mã hóa thông điệp, còn bên nhận B thực hiện giải mã và kiểm tra chữ ký của thông điệp.

1. Nền tảng mã hoá khoá bí mật cho PGP (DES, 3-DES, AES)
   1. DES.

**2.1.1. Giới thiệu**

Thuật toán mã hoá DES được IBM phát triển vào những năm 1970 sau đó được Uỷ ban tiêu chuẩn Quốc gia Hoa Kỳ (The National Bureau of Standard). DES đã trở thành chuẩn mã hoá dữ liệu chính thức cho Chính phủ Hoa Kỳ và năm 1977 và trở thành hệ mật mã được sử dụng rộng rãi nhất trên thế giới. Thuật toán mã hoá DES có thể thoả mãn các yêu cầu sau:

- Thuật toán phải có độ an toàn cao.

- Thuật toán phải được định nghĩa đầy đủ và hoàn toàn dễ hiểu.

- Độ an toàn phải nằm ở khóa, không phụ thuộc vào tính bí mật của thuật toán.

- Thuật toán phải sẵn sàng cung cấp cho mọi người dùng.

- Thuật toán phải thích nghi được với việc dùng cho các ứng dụng khác nhau.

- Thuật toán phải được cài đặt được một cách tiết kiệm trong các thiết bị điên từ.

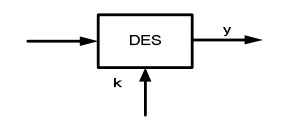
- Thuật toán khi sử dụng phải phát huy tối đa hiệu quả.

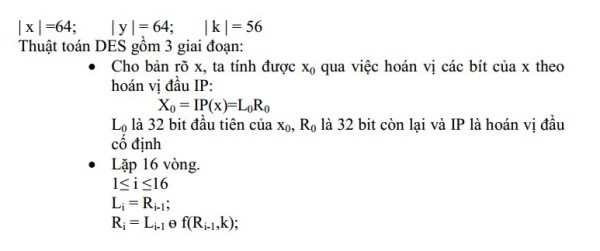
- Thuật toán phải có khả năng hợp thức hoá.

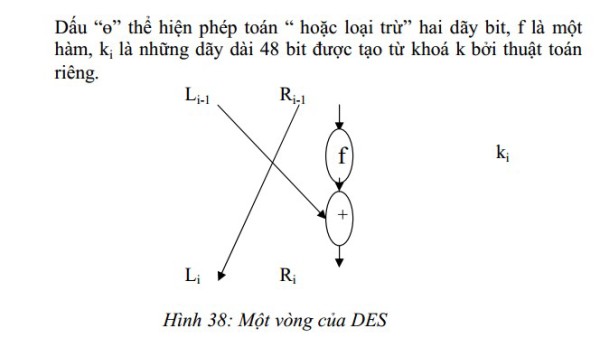
- Thuật toán phải có tính thương mại.

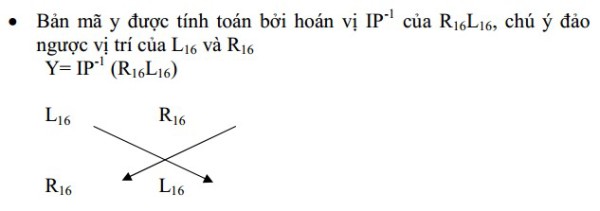
**2.2. Mô tả thuật toán DES**

Một mô tả đầy đủ về DES được nêu ra trong Công báo về chuẩn xử lý thông tin Liên bang số 46 ngày 15-1-1977. DES mã hoá một dòng bit rõ x có độ dài 64 với khoá K là dòng 56 bit, đưa ra bản mã y cũng là một dãy bit có độ dài 64.

1. 
2. Mô tả DES







Các mẫu hoạt động của DES: như ta đã thấy, đầu vào của DES chỉ có 8 byte, vậy mà văn bản cần mã lại có thể rất dài, cỡ vài kbyte chẳng hạn. Để giải quyết vấn đề này, người ta đã đề ra 4 mẫu hoạt động cho DES là:

- Electronic CodeBook mode (ECB).

- Cippher FeedBack mode (CFB).

- Cipher Block Chaining mode (CBC).

- Output FeedBack mode (OFB).

**2.3. Ưu và nhược điểm của DES**

- Ưu điểm: Thuật toán mã hoá DES tốc độ mã hoá dữ liệu rất nhanh.

- Nhược điểm: Do DES có kích cỡ của không gian khoá 256 là quá nhỏ, không đủ an toàn, cho nên những máy có mục đích đặc biệt có thể sẽ bẻ gãy và dò ra khoá rất nhanh.

**2.4. Ứng dụng của thuật toán DES trong thực tế**

Một ứng dụng rất quan trọng của DES là ứng dụng cho các văn bản trong giao dịch ngân hang sử dụng các tiêu chuẩn được hiệp hội các ngân hang Mỹ phát triển. DES được sử dụng để mã hoá các số nhận dạng cá nhân (Pins) và các văn bản về tài khoản được máy thu ngân tự động thực hiện (ATMs)…

Thuật toán mã hóa khối DES (Data Encryption Standard) là một thuật toán khối với kích thước khối 64 bít và kích thước khóa 56 bít, được Tổ chức Tiêu chuẩn xử lý thông tin liên bang Hoa Kỳ (FIPS) công bố chính thức vào tháng 11/1976 và được xuất bản trong tài liệu có tên là [FIPS PUB 46](http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf) vào tháng 01/1977. DES lần lượt được cập nhật vào năm 1988 ([FIPS-46-1](http://www.bitpipe.com/tlist/FIPS-PUB-46-1.html)), 1993 ([FIPS-46-2](http://www.itl.nist.gov/fipspubs/fip46-2.htm)), 1998 ([FIPS-46-3](http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf)) Thuật toán mã hóa khối 3 lần 3DES (Triple Data Encryption Standard) là tên chung của việc áp dụng mã hóa một khối dữ liệu qua 03 lần mã hóa DES với 03 khóa khác nhau. 3DES công bố trong tài liệu tiêu chuẩn của Viện Tiêu chuẩn Quốc gia Hoa Kỳ ANSI ([X 9.52–1998](http://www.techstreet.com/products/1327219)), tiêu chuẩn ISO/IEC ([ISO/IEC 18033-3:2010](http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=54531)) (ISO/IEC là Tổ chức tiêu chuẩn hoá quốc tế/[Ủy ban kỹ thuật điện quốc tế](http://vi.wikipedia.org/wiki/%E1%BB%A6y_ban_k%E1%BB%B9_thu%E1%BA%ADt_%C4%91i%E1%BB%87n_qu%E1%BB%91c_t%E1%BA%BF) (International Organization for Standardization/International Electrotechnical Commission)). Đối với nhiều tài liệu kỹ thuật, người ta phân biệt khái niệm DES (là một tiêu chuẩn) và thuật toán DEA (Data Encryption Algorithm, hay Thuật toán Mã hóa Dữ liệu) - thuật toán dùng trong chuẩn DES. Tương tự với 3DES là một tiêu chuẩn và TDEA (Triple Data Encryption Alogorithm) là thuật toán trong 3DES.

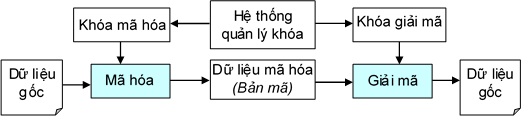
Tiền thân của thuật toán DES là thuật toán Lucifer, một thuật toán do IBM phát triển. Cuối năm 1976, DES được chọn làm chuẩn mã hóa dữ liệu của Hoa Kỳ, sau đó được sử dụng rộng rãi trên toàn thế giới. DES cùng với mã hóa bất đối xứng đã mở ra một thời kì mới cho ngành mã hóa thông tin. Đồng thời, việc nghiên cứu mã hóa thông tin cũng không còn là bí mật nữa mà đã trở thành một ngành khoa học máy tính bình thường. Tuy nhiên thuật toán DES không được xem là an toàn do độ dài 56 bít của khóa là quá nhỏ, nhiều kết quả nghiên cứu, phân tích cho thấy việc mã hóa có thể bị phá khóa, thuật toán 3DES (thực hiện DES 3 lần) có tính an toàn cao hơn và được sử dụng trong thực tế nhiều hơn.

Đặc điểm kỹ thuật

***Một số khái niệm về mã hóa***

Mã hóa dữ liệu là sử dụng một phương pháp biến đổi dữ liệu từ dạng bình thường sang một dạng khác, mà một người không có thẩm quyền, không có phương tiện giải mã thì không thể đọc hiểu được.

Giải mã dữ liệu là quá trình ngược lại, là sử dụng một phương pháp biến đổi dữ liệu đã được mã hóa về dạng thông tin ban đầu. Có thể mô tả quy trình thực hiện mã hóa dữ liệu và giải mã dữ liệu như sau:



***Hình 1: Quy trình mã hóa dữ liệu***

(Nguồn: *Handbook of Applied Cryptography*, A. Menezes, P . van Oorschot and S. V anstone)

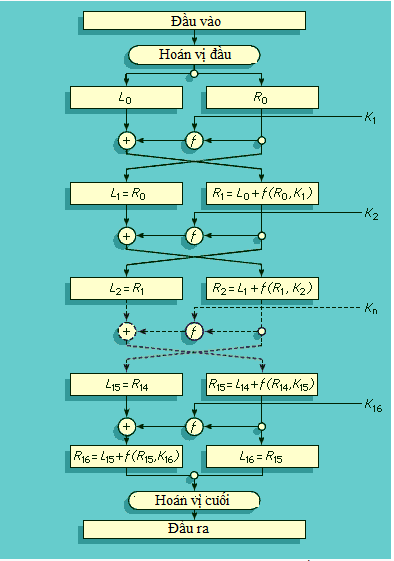
Trong hệ mật mã cổ điển có hai kiểu phổ biến là mã hóa dòng (mã hóa trên từng bít của dòng dữ liệu) và mã hóa khối, trong đó mã hóa khối (block cipher) là kiểu mã hóa từng khối nhiều ký tự cùng một lúc, hai tham số ảnh hưởng tới mã hóa theo khối là kích thước khối và kích thước khóa. Thuật toán mã hóa khối xác định độ dài cố định của khối dữ liệu đầu vào và khối dữ liệu đầu ra. Khi thực hiện mã hóa, khối dữ liệu có độ dài tùy ý phải chia thành nhiều đơn vị có độ dài xác định trước sau đó áp dụng thuật toán mã hóa nhiều lần. Kết quả là các khối dữ liệu đầu ra được kết hợp theo một sơ đồ mật mã nào đó để thu được khối dữ liệu được mã hóa. Thuật toán DES là một hệ điển hình mã hóa khối, chúng ta sẽ tiếp tục tìm hiểu trong phần sau.

***Đặc tả thuật toán DES***

Thuật toán DES được thiết kế để mã hóa và giải mã các khối dữ liệu 64 bít với một khóa có độ dài 64 bít. Việc giải mã phải được thực hiện bởi một khóa tương ứng như trong quá trình mã hóa, mỗi khóa 64 bít sẽ chứa 56 bít sinh phát ngẫu nhiên và sử dụng như là các bít khóa trong thuật toán. Tám bít còn lại không được sử dụng bởi thuật toán, được dùng để phát hiện lỗi. Tám bít phát hiện lỗi được thiết lập để kiểm tra lỗi tương ứng với 8 byte trong khóa.

Quy trình mã hóa của DES được mô tả trong hình phía dưới. Theo hình 2, ta thấy mã hóa DES được thực hiện qua 16 vòng. Thông tin đầu vào là 64 bít, sẽ được chia thành 2 khối (block) trái (L) và phải (R). Sau đó từ khóa (56 bít) người ta tạo ra các khóa con (subkey) 48 bít gọi là Ki. Hàm f ở trên thực chất là 1 hàm hoán vị.

Trong quá trình mã hóa, dữ liệu đầu vào phải thực hiện quá trình hoán vị đầu (initial permutation) và hoán vị cuối (final permutation) sau vòng thứ 16. Việc thực hiện hoán vị phục vụ cho quá trình đưa thông tin vào và lấy thông tin ra từ các khối phần cứng, tạo điều kiện cho việc cài đặt phần cứng. Hàm cơ sở f cho phép đảm bảo tính bảo mật trong thuật toán DES này.



***Hình 2: Quá trình mã hóa của DES***

*(Nguồn:* [*Data Encryption Standard (DES), FIPS 46-3*](http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf)*)*

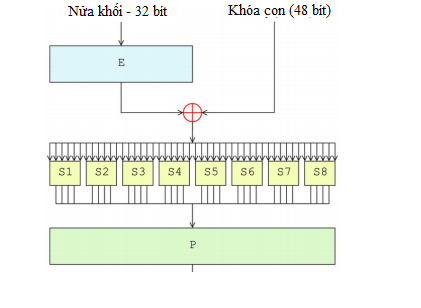
Việc thực hiện nhiều lần các bước lặp với tác dụng của f là nhằm tăng cường tính hỗn loạn, rắc rối (confusion) và tính khuếch tán (diffusion) đã có trong hàm f.

Cấu trúc của vòng lặp DES thực hiện theo công thức sau:

**(Li,Ri) = (Ri-1, Li-1 XOR f (Ri-1,Ki))**

Trong đó (Li,Ri) là nửa trái và nửa phải lấy được của phép biến đổi vòng lặp thứ i.

Minh họa sơ đồ biến đổi cụ thể của hàm f được trong hình 3, trong hình này 32 bít của thành phần Ri-1 được mở rộng thành 48 bít rồi thực hiện biến đổi E (expansion: mở rộng với sự lặp lại một số bít). Thực hiện phép XOR với 48 bít của khóa Ki. Thu được kết quả 48 bít phân thành 8 nhóm 6 bít. Mỗi nhóm này sẽ đi vào một biến đổi đặc biệt gọi là biến đổi Si (có 8 Si khác nhau) và cho ra kết quả là 8 nhóm 4 bít. Từ đó, 32 bít hợp thành (sau khi qua 8 Si khác nhau) sẽ được hoán vị lại theo hàm hoán vị P để đưa ra kết quả cuối cùng của hàm f (tức nhân của Fi).



***Hình 3: Hàm f dùng trong DES***

(Nguồn*:* [*Data Encryption Standard (DES), FIPS 46-3*](http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf))

## 2.2 3DES

Thuật toán 3DES sử dụng một nhóm khóa bao gồm 03 khóa DES K1, K2 và K3, mỗi khóa có giá trị 56 bít. Thuật toán mã hóa thực hiện như sau:

*Quá trình mã hóa*

**Bản mã= EK3(DK2(EK1(Bản rõ)))**

Trước tiên, thực hiện mã hóa DES với khóa K1, tiếp tục giải mã DES với khóa K2 và cuối cùng mã hóa DES với khóa K3 (E – Encryption: quá trình mã hóa; D - Decryption: quá trình giải mã; Bản rõ: Dữ liệu đầu vào của phép mã hóa hoặc dữ liệu đầu ra của phép giải mã; Bản mã: Dữ liệu đầu ra của phép mã hóa hoặc dữ liệu đầu vào của phép giải mã).

*Quá trình giải mã*

**Bản rõ = DK1(EK2(DK3(Bản mã))**

Quá trình giải mã với việc giải mã với khóa K3, sau đó mã hóa với khóa K2, và cuối cùng giải mã với khóa K1..

3DES mã hóa một khối dữ liệu có giá trị 64 bít (bản rõ) thành một khối dữ liệu mới có giá trị 64 bít (bản mã). Các tiêu chuẩn chỉ ra phương thức lựa chọn nhóm khóa (K1, K2, K3) như sau:

1. Lựa chọn 1: K1, K2, K3 là các khóa độc lập
2. Lựa chọn 2: K1, K2 là hai khóa độc lập và K3 = K1
3. Lựa chọn 3: K1=K2=K3

Lựa chọn 1 là phương thức mã hóa mạnh nhất với 168 bít khóa độc lập (168=3x56). Lựa chọn 2 ít bảo mật hơn với 112 bít khóa ( 2x56=112 bít) và lựa chọn 3 chỉ tương đương với việc mã hóa DES 1 lần với 56 bít khóa. Mỗi khóa DES thông thường được lưu trữ và truyền đi trong 8 byte, vì vậy một nhóm khóa yêu cầu 8 hoặc 16, 24 byte cho việc lưu trữ khóa.

*Yêu cầu lưu trữ khóa*

Đối với một khóa riêng lẻ hay một nhóm khóa gồm 03 khóa sẽ phải:

* Đảm bảo duy trì tính bảo mật;
* Việc cấp phát khóa sử dụng một phương pháp đã được chấp thuận dựa trên việc sinh khóa ngẫu nhiên;
* Nhóm khóa độc lập với các nhóm khóa khác;
* Phải có tính toàn vẹn do mỗi khóa trong nhóm khóa không thể thay đổi theo thời gian trong quá trình sinh phát khóa, truyền tải và lưu trữ khóa;
* Sử dụng một cách tuần tự theo thứ tự xác định trước;
* Số lượng khóa là cố định với mỗi khóa đơn lẻ không thể làm sai lệch.

*Khóa yếu*

Có nhiều khóa được xem là khóa yếu trong thuật toán DES, đó là các khóa dễ có khả năng bị đối tượng phá khóa do một số bít của khóa lặp lại và dễ dự đoán trước. Việc sử dụng khóa yếu có thể làm giảm tính bảo mật của DES, do đó tránh sử dụng các khóa yếu này. Cụ thể các khóa yếu là:

* + 01010101 01010101
  + FEFEFEFE FEFEFEFE
  + E0E0E0E0 F1F1F1F1
  + 1F1F1F1F 0E0E0E0E

Một số cặp khóa yếu nên tránh sử dụng:

* 011F011F010E010E và 1F011F010E010E01
* 01E001E001F101F1 và E001E001F101F101
* 01FE01FE01FE01FE và FE01FE01FE01FE01
* 1FE01FE00EF10EF1 và E01FE01FF10EF10E
* 1FFE1FFE0EFE0EFE và FE1FFE1FFE0EFE0E

48 khóa được coi là “có thể yếu” nên hạn chế sử dụng:

01011F1F01010E0E 1F1F01010E0E0101 E0E01F1FF1F10E0E

0101E0E00101F1F1 1F1FE0E00E0EF1F1 E0E0FEFEF1F1FEFE

0101FEFE0101FEFE 1F1FFEFE0E0EFEFE E0FE011FF1FE010E

011F1F01010E0E01 1FE001FE0EF101FE E0FE1F01F1FE0E01

011FE0FE010EF1FE 1FE0E01F0EF1F10E E0FEFEE0F1FEFEF1

011FFEE0010EFEF1 1FE0FE010EF1FE01 FE0101FEFE0101FE

01E01FFE01F10EFE 1FFE01E00EFE01F1 FE011FE0FE010EF1

FE01E01FFE01F10E 1FFEE0010EFEF101 FE1F01E0FE0E01F1

01E0E00101F1F101 1FFEFE1F0EFEFE0E FE1FE001FE0EF101

01E0FE1F01F1FE0E E00101E0F10101F1 FE1F1FFEFE0E0EFE

01FE1FE001FE0EF1 E0011FFEF1010EFE FEE0011FFEF1010E

01FEE01F01FEF10E E001FE1FF101FE0E FEE01F01FEF10E01

01FEFE0101FEFE01 E01F01FEF10E01FE FEE0E0FEFEF1F1FE

1F01011F0E01010E E01F1FE0F10E0EF1 FEFE0101FEFE0101

1F01E0FE0E01F1FE E01FFE01F10EFE01 FEFE1F1FFEFE0E0E

1F01FEE00E01FEF1 E0E00101F1F10101 FEFEE0E0FEFEF1F1

***Tấn công bằng phương pháp vét cạn***

DES có **256=1017** khóa, nếu như biết một cặp bản gốc-bản gốc thì có thể tiến hành phép thử 1017 để tìm ra khóa phù hợp. Giả thiết một phép thử mất 10-6 giây trên một máy tính cá nhân thì sẽ mất 1011 giây để tìm khóa (gần 7300 năm). Tuy nhiên với sức mạnh của các máy tính chuyên dụng với chip mã có tốc độ lên tới 4.5x107 bít/s, việc phá mã trở nên nhanh chóng. Năm 1997, công ty bảo mật RSA tài trợ một số cuộc thi với giải thưởng lên tới 10.000 đô la Mỹ cho đội đầu tiên phá được thông điệp mã hóa bằng DES. Đội chiến thắng là dự án DESCHALL với những người dẫn đầu bao gồm Rocke Verser, Matt Curtin và Justin Dolske. Họ đã sử dụng hàng nghìn máy tính nối mạng để phá mã DES. Năm 1998, tổ chức Electronic Frontier Foundation (EFF), một tổ chức hoạt động cho quyền công dân trên Internet, xây dựng một hệ thống chuyên biệt để phá mã DES với giá thành 250.000 đô la Mỹ. DES đã bị giới nghiên cứu phê bình do sử dụng khóa có kích thước quá ngắn, do đó, thực tế 3DES được áp dụng phổ biến hơn với việc thực hiện 3 lần DES làm tăng độ phức tạp và tổng kích thước của khóa hơn.

* 1. AES

Tiêu chuẩn Advanced Encryption Standard (AES) - Tiêu chuẩn mã hóa tiên tiến là một thuật toán tiêu chuẩn của chính phủ Hoa Kỳ nhằm mã hóa và giải mã dữ liệu do Viện Tiêu chuẩn và Công nghệ quốc gia Hoa Kỳ (National Institute Standards and Technology– NIST) phát hành ngày 26/11/2001 và được đặc tả trong Tiêu chuẩn Xử lý thông tin Liên bang 197 (Federal Information Processing Standard – FIPS 197) sau quá trình kéo dài 5 năm trình phê duyệt, AES tuân theo mục 5131 trong Luật Cải cách quản lý công nghệ thông tin năm 1996 và Luật An toàn máy tính năm 1997.

AES là một thuật toán “mã hóa khối” (block cipher) ban đầu được tạo ra bởi hai nhà mật mã học người Bỉ là Joan Daemen và Vincent Rijmen. Kể từ khi được công bố là một tiêu chuẩn, AES trở thành một trong những thuật toán mã hóa phổ biến nhất sử dụng khóa mã đối xứng để mã hóa và giải mã (một số được giữ bí mật dùng cho quy trình mở rộng khóa nhằm tạo ra một tập các khóa vòng). Ở Việt Nam, thuật toán AES đã được công bố thành tiêu chuẩn quốc gia TCVN 7816:2007 năm 2007 về Thuật toán mã hóa dữ liệu AES.

**Đặc điểm kỹ thuật**

AES là một thuật toán mã hóa khối đối xứng với độ dài khóa là 128 bít (một chữ số nhị phân có giá trị 0 hoặc 1), 192 bít và 256 bít tương ứng dọi là AES-128, AES-192 và AES-256. AES-128 sử dụng 10 vòng (round), AES-192 sử dụng 12 vòng và AES-256 sử dụng 14 vòng.

Vòng lặp chính của AES thực hiện các hàm sau: SubBytes(), ShiftRows(), MixColumns() và AddRoundKey(). Ba hàm đầu của một vòng AES được thiết kế để ngăn chặn phân tích mã bằng phương thức “mập mờ“ (confusion) và phương thức “khuếch tán“ (diffusion), còn hàm thứ tư mới thực sự được thiết kế để mã hóa dữ liệu. Trong đó “khuếch tán“ có nghĩa là các kiểu mẫu trong bản rõ (Dữ liệu đầu vào của phép mã hóa hoặc dữ liệu đầu ra của phép giải mã) được phân tán trong các bản mã (Dữ liệu đầu ra của phép mã hóa hoặc dữ liệu đầu vào của phép giải mã), “mập mờ“ nghĩa là mối quan hệ giữa bản rõ và bản mã bị che khuất. Một cách đơn giản hơn để xem thứ tự hàm AES là: Trộn từng byte (SubBytes), trộn từng hàng (ShiftRows), trộn từng cột (MixColumns) và mã hóa (AddRoundKey).

**Đặc tả thuật toán AES**

Đối với thuật toán AES, độ dài của khối đầu vào, khối đầu ra và trạng thái là 128 bít, số các cột (các từ có độ dài 32 bít) tạo nên trạng thái là Nb = 4.

Trong thuật toán AES, độ dài khóa mã K có thể là 128, 192 hay 256 bít. Độ dài khóa được biểu diễn bằng Nk = 4, 6 hoặc 8 thể hiện số lượng các từ 32 bít (số cột) của khóa mã.

Đối với thuật toán AES, số vòng được thay đổi trong quá trình thực hiện thuật toán phụ thuộc vào kích cỡ khóa. Số vòng này được ký hiệu là Nr. Nr = 10 khi Nk = 4, Nr = 12 khi Nk = 6 và Nr = 14 khi Nk = 8.

Các tổ hợp khóa-khối-vòng phù hợp đối với tiêu chuẩn này thể hiện trong Bảng 1. Việc thực hiện cụ thể thuật toán có liên quan đến độ dài khóa, kích cỡ khối và số vòng.

***Bảng 1: Tổ hợp Khóa-khối-vòng***

|  |  |  |  |
| --- | --- | --- | --- |
|  | Độ dài khóa(Nk từ) | Độ dài khối  (Nb từ) | Số vòng  (Nr) |
| AES-128 | 4 | 4 | 10 |
| AES-196 | 6 | 4 | 12 |
| AES-256 | 8 | 4 | 14 |

Đối với phép mã hóa và phép giải mã, thuật toán AES sử dụng một hàm vòng gồm bốn phép biến đổi byte như sau: phép thay thế byte (một nhóm gồm 8 bít) sử dụng một bảng thay thế (Hộp-S), phép dịch chuyển hàng của mảng trạng thái theo các offset (số lượng byte) khác nhau, phép trộn dữ liệu trong mỗi cột của mảng trạng thái, phép cộng khóa vòng và trạng thái. Các phép biến đổi này (cũng như các phép nghịch đảo tương ứng của chúng) được mô tả trong phần dưới đây.

**Phép mã hóa**

Tại thời điểm bắt đầu phép mã hóa, đầu vào được sao chép vào mảng trạng thái sử dụng các quy ước. Sau phép cộng khóa vòng khởi đầu, mảng trạng thái được biến đổi bẳng cách thực hiện một hàm vòng liên tiếp với số vòng lặp là 10, 12 hoặc 14 (tương ứng với độ dài khóa), vòng cuối cùng khác biệt không đáng kể với Nr-1 vòng đầu tiên. Trạng thái cuối cùng được chuyển thành đầu ra. Hàm vòng được tham số hóa bằng cách sử dụng một lược đồ khóa – mảng một chiều chứa các từ 4 byte nhận từ phép mở rộng khóa.

Phép biến đổi cụ thể gồm SubBytes(), ShiftRows(), MixColumns() và AddRoundKey() dùng để xử lý trạng thái.

**SubBytes()**

Phép biến đổi dùng trong phép mã hóa áp dụng lên trạng thái (kết quả mã hóa trung gian, được mô tả dưới dạng một mảng chữ nhật của các byte) sử dụng một bảng thay thế byte phi tuyến (Hộp S – bảng thay thế phi tuyến, được sử dụng trong một số phép thay thế byte và trong quy trình mở rộng khóa, nhằm thực hiện một phép thay thế 1-1 đối với giá trị mỗi byte) trên mỗi byte trạng thái một cách độc lập.

**ShiftRows()**

Phép biến đổi dùng trong phép mã hóa áp dụng lên trạng thái bằng cách chuyển dịch vòng ba hàng cuối của trạng thái theo số lượng byte các offset khác nhau.

**MixColumns()**

Phép biến đổi trong phép mã hóa thực hiện bằng cách lấy tất cả các cột trạng thái trộn với dữ liệu của chúng (một cách độc lập nhau) để tạo ra các cột mới.

**AddRoundKey()**

Phép biến đổi trong phép mã hóa và phép giải mã. Trong đó, một khóa vòng (các giá trị sinh ra từ khóa mã bằng quy trình mở rộng khóa) được cộng thêm vào trạng thái bằng phép toán XOR (phép toán hoặc và loại trừ). Độ dài của khóa vòng bằng độ dài của trạng thái.

**Mở rộng khóa**

Thuật toán AES nhận vào một khóa mã K và thực hiện phép mở rộng khóa để tạo ra một lược đồ khóa. Phép mở rộng khóa tạo ra tổng số Nb(Nr+1) từ. Thuật toán yêu cầu một tập khởi tạo gồm Nb từ và mỗi trong số Nr vòng đòi hỏi Nb từ làm dữ liệu khóa đầu vào. Lược đồ khóa kết quả là một mảng tuyến tính các từ 4 byte.

**Phép giải mã**

Các phép biến đổi trong phép mã hóa có thể được đảo ngược và sau đó thực hiện theo chiều ngược lại nhằm tạo ra phép giải mã trực tiếp của thuật toán AES. Các phép biến đổi sử dụng trong phép giải mã gồm: InvShiftRows(), InvSubBytes(), InvMixColumns() và AddRoundKey().

**InvSubBytes()**

Phép biến đổi InvSubBytes() là nghịch đảo của phép thay thế theo byte SubBytes(), trong đó sử dụng một hộp-S nghịch đảo áp dụng cho mỗi byte của trạng thái.

**InvShiftRows()**

Phép biến đổi InvShiftRows() là phép biến đổi ngược của ShiftRows(). Các byte trong ba từ cuối của trạng thái được dịch vòng theo số byte khác nhau. Ở hàng đầu tiên (r=0) không thực hiện phép chuyển dịch, ba hàng dưới cùng được dịch vòng Nb-shift(r,Nb) byte.

**InvMixColumns()**

Phép biến đổi InvMixColumns() là phép biến đổi ngược của MixColumns(). Nó thao tác theo từng cột của trạng thái, xem mỗi cột như một đa thức bốn hạng tử.

**Biến đổi nghịch AddRoundKey()**

Phép biến đổi AddRoundKey() là phép biến đổi thuận nghịch vì nó chỉ áp dụng một phép toán XOR nên nó được thực hiện như nhau ở cả phép mã hóa và phép giải mã.

Ngoài các phép giải mã trên, thuật toán AES còn cho phép thực hiện một phép giải mã tương đương có cùng thứ tự các phép biến đổi như trong phép mã hóa (các biến đổi được thay bằng các phép biến đổi ngược). Có thể thực hiện được điều này là nhờ một thay đổi trong lược đồ khóa. Hai tính chất tạo nên một phép giải mã tương đương là: Tính giao hoán giữa hai phép biến đổi SubBytes() và ShiftRows() (tính chất này cũng đúng với phép nghịch đảo InvSubBytes() và InvShiftRows()), Các phép toán trộn cột MixColumns() và InvMixColumns() là tuyến tính đối với đầu vào cột. Các tính chất này cho phép đảo ngược thứ tự của các phép biến đổi InvSubBytes() và InvShiftRows(). Thứ tự của các phép biến đổi AddRoundKey() và InvMixColumns() cũng có thể đảo ngược với điều kiện đảm bảo rằng các cột của lược đồ khóa giải mã được chỉnh sửa bằng cách sử dụng phép biến đổi InvMixColumns().

**Vấn đề thực hiện khóa**

*Yêu cầu về độ dài khóa*

Việc thực hiện khóa của thuật toán AES sẽ hỗ trợ ít nhất một trong ba độ dài khóa là 128 bít, 192 bít và 256 bít. Việc thực hiện khóa có thể tùy chọn hỗ trợ hai hoặc ba độ dài khóa, nhằm tăng thêm tính tương tác cho các thực hiện thuật toán.

*Tham số hóa độ dài khóa, kích thước khối và số vòng*

AES quy định cụ thể các giá trị được phép dùng cho chiều dài khóa, kích thước khối và số vòng. Tuy nhiên, các giá trị này có thể thay đổi trong tương lai. Do đó, những nhà triển khai thuật toán AES có thể lựa chọn thiết kế linh hoạt với mong muốn của họ.

**Ứng dụng**

Thuật toán AES cho phép thực hiện hiệu quả bằng cả phần mềm và phần cứng. Thông thường với những ứng dụng không yêu cầu cao về hiệu năng và tốc độ thì AES được thực hiện ở dạng phần mềm. Với việc thực hiện trên phần mềm, thuật toán AES có thể được viết bằng nhiều ngôn ngữ lập trình như Assembler, C/C++, Visual Basic, Java, C#... và có thể vận hành trên nhiều hệ điều hành như Windows, Linux/Unix, Solaris.... Khi thực hiện trên phần cứng, thuật toán AES hỗ trợ thực hiện hai dòng: dòng thiết bị thứ nhất dựa vào một hệ vi xử lý phụ kết hợp với hệ vi xử lý chính của máy tính, dòng thiết bị thứ hai thường được thiết kế ở dạng thẻ thông minh (smart card) hoặc các thiết bị cắm qua cổng USB (Universal Serial Bus). Trong Thông tư số 01/2011/TT-BTTTT ngày 04/01/2011 của Bộ Thông tin và Truyền thông Công bố Danh mục tiêu chuẩn kỹ thuật về ứng dụng công nghệ thông tin trong cơ quan nhà nước quy định Khuyến nghị áp dụng tiêu chuẩn AESvà được xếp vào nhóm Tiêu chuẩn về an toàn thông tin.

1. Mã Hóa Bất Đối Xứng.

### 4.1 Các thuật toán

#### **a) RSA**

Thuật toán RSA được phát minh năm 1978. 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 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

#### b) 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.

1. **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.

### 4.1 Hàm hash

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

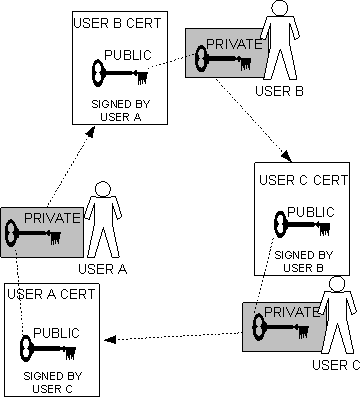
2. Hàm Hash yếu

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 h 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.

### 4.2 Mô hình kiến trúc

Kiến trúc tổng quan



*Hình 1:Kiến trúc tổng quan*

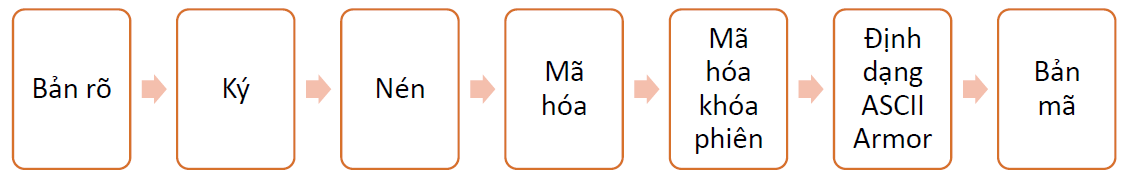
Hai dịch vụ chính mà PGP cung cấp cho người dùng là: mã hóa và xác thực thông điệp. Khi thiết kế một ứng dụng bảo mật email, người thiết kế phải đương đầu với hai vấn đề chính, trước hết, phải bảo mật ứng dụng bằng những giải thuật nào?

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 hiên của PGP trong truyền và nàận 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.

### 4.3 Hoạt động

#### 4.3.1 Mã hóa

Hoạt động thực tế của PGP để gửi và nhận thông điệp bao gồm năm dịch vụ: chữ ký số, mã hóa thông điệp trong PGP. Quy trình thực hiện theo các bước sau:

****

Hình 1: Qúa trình mã hóa 1 thông điệp trong PGP

***a, Chuẩn bị file***

Mỗi lần thực hiện, PGP chỉ xử lý một file. Những file được xử lý bởi PGP nói chung thường là văn bản. Đây là dạng phổ biến nhất của truyền thông email. Nhưng PGP có thể chấp nhận bất kỳ file nào, kể cả file nhị phân, file PICT... Một trong những dịch vụ tiện lợi do PGP cung cấp cho phép người dùng gửi file theo đường email bình thường.

***b, Chữ ký số***

Khi nhận file đầu vào, bước đầu tiên của PGP là tạo một chữ ký số để gán vào file. Đây chỉ là một dịch vụ lựa chọn. Nếu người gửi yêu cầu chữ ký số, PGP sẽ tạo một mã băm của file và sau đó mã hóa mã băm với RSA sử dụng cho khóa riêng tư người gửi. Kết quả mã hóa mã băm là chữ ký số cho file này. Chữ ký số bảo đảm file này là của người gửi và file đó không bị biến dạng.

***c, Nén***

Việc nén lại sẽ giúp tiết kiệm thời gian truyền, không gian đĩa và quan trọng hơn là giúp tăng cường tính bảo mật của mật mã. Hầu hết các kỹ thuật phân tích mã hóa được tìm thấy trong bản rõ để phá mật mã. Nén làm giảm bớt đi các mô hình này, qua đó giúp tăng cường khả năng chống giải mã. Tuy nhiên người dùng có thể lựa chọn dùng nén hoặc không.

***d, Mã hóa***

Đầu tiên người dùng sẽ sử dụng thuật toán mã hóa đối xứng mã hóa bản rõ bằng một khóa chung (còn gọi là khóa phiên). Tiếp theo người dùng sẽ sử dụng cặp khóa công khai bí mật được tạo bởi thuật toán mã hóa bất đối xứng. Sử dụng khóa công khai trong cặp khóa công khai – bí mật mã hóa khóa phiên được tạo ra sau quá trình mã hóa bản rõ bằng thuật toán mã hóa đối xứng.

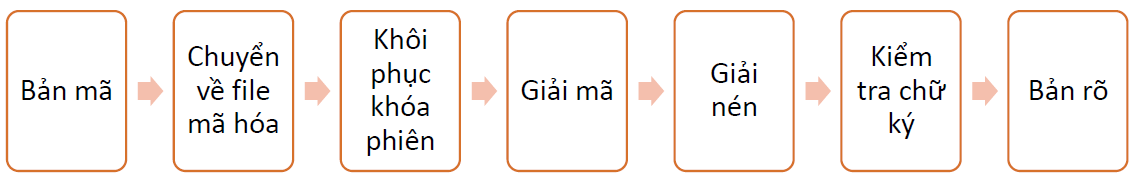
Phần mã hóa thông điệp gửi đi của PGP sử dụng cả hai thuật toán mã hóa đối xứng và mã hóa bất đối xứng để tận dụng ưu thế của cả hai. Thuật toán mã hóa bất đối xứng đảm bảo việc phân phối khóa phiên trong hệ thống với độ bảo mật cao còn thuật toán mã hóa bí mật có ưu thế về tốc độ mã hóa và giải mã (nhanh hơn cỡ 1000 lần).

***e, Tính tương thích Email***

Nếu ký, nén hoặc mã hóa được thực hiện trên file gốc thì khối dữ liệu được sản sinh ra là những dữ liệu nhị phân. Tuy nhiên, nhiều hệ thống email không thể xử lý với dữ liệu nhị phân mà chỉ có thể xử lý những file văn bản. Khắc phục hạn chế này, PGP chuyển đổi dữ liệu nhị phân thành những ký tự có thể in được. PGP sử dụng khuôn dạng ASCII armor để chuyển đổi dữ liệu.

#### 4.3.2 Giải mã

Hình dưới mô tả quá trình giải mã một thông điệp trong PGP. Về cơ bản, để giải mã, PGP chỉ cần thực hiện đảo ngược các bước của quá trình mã hóa.

****

Hình 2: Quá trình giải mã một thông điệp trong PGP

Đầu tiên PGP sẽ thực hiện việc chuyển file bản mã về lại dạng nhị phân để thực hiện giải mã. Tiếp theo người dùng sẽ sử dụng khóa riêng tư của mình trong cặp khóa công khai – riêng tư để thực hiện việc giải mã khóa phiên. Sau khi có được khóa phiên thực hiện việc quá trình giải mã bản rõ. Việc giải nén sẽ được thực hiện để khôi phục đầy đủ các mô hình trong văn bản. Cuối cùng là việc kiểm tra chữ ký để xem văn bản có bị sửa đổi hay xâm phạm trong quá trình truyền đi hay chưa.

#### 4.3.3 Khóa

Khóa là một giá trị làm việc với một thuật toán mã hóa để tạo ra một bản mã cụ thể. Về cơ bản khóa là những con số rất lớn. Kích thước của khóa được đo bằng bit. Trong các thuật toán mã hóa, khóa càng lớn thì tính bảo mật càng cao.

Tuy nhiên kích thước của cặp khóa công khai – bí mật so với khóa thông thường là không hề liên quang với nhau. Như một khóa thông thường 80 bit có sức mạnh tương đương với một khóa công khai 1024 bit. Kích thước khóa là quan trọng cho sự an toàn, nhưng các thuật toán được sử dụng cho từng loại là rất khác nhau.. Vì thế không thể so sánh chỉ kích thước khóa của các hệ mật mã với nhau.

Nền tảng những thao tác của PGP là yêu cầu mỗi người dùng có một cặp khóa công khai – bí mật cũng như các bản sao chép các khóa công khai của người nhận. Mặc dù một cặp khóa công khai – bí mật về mặt toán học là có liên quan đến nhau, nó rất khó để có thể suy ra được một khóa bí mật nếu như chỉ có khóa công khai. Tuy nhiên, vẫn có thể suy ra được khóa bí mật nếu có đủ thời gian và khả năng tính toán. Điều này dẫn đến một vấn đề rất quan trọng là làm sao để chọn ra được một khóa đúng kích cỡ, tức là đủ lớn để có thể đảm bảo an toàn và đủ nhỏ để có thể áp dụng một cách nhanh chóng. Ngoài ra bạn cũng cần phải xem xét những ai có thể cố gắng đọc các tập tin của bạn, họ có bao nhiêu thời gian và khả năng họ có thể.

Khóa được lưu trữ ở dạng mã hóa. PGP lưu trữ các khóa trong hai tập tin trên đĩa cứng của bạn. Một cho khóa công cộng và một cho khóa bí mật. Những tập tin này được gọi là một vòng khóa.

***A. Khóa công khai***

PGP thường lưu lại những chìa khóa công khai mà người dùng thu được. Các khóa này được tập hợp và lưu lại trên vòng khóa công khai. Mỗi mục vòng gồm các phần:

* Khóa công khai
* User ID chủ nhân của khóa công khai này, tên đặc trưng của chủ nhân
* Một keyID, là định danh cho khóa này
* Thông tin khác liên quan đến độ tin cậy của khóa và chủ nhân của nó.

***B. Khóa bí mật***

Để sử dụng PGP, người dùng cần phải có một khóa bí mật. Nếu muốn người dùng có thể tạo nhiều khóa bí mật. Vòng khóa bí mật chứa đựng thông tin của mỗi khóa.

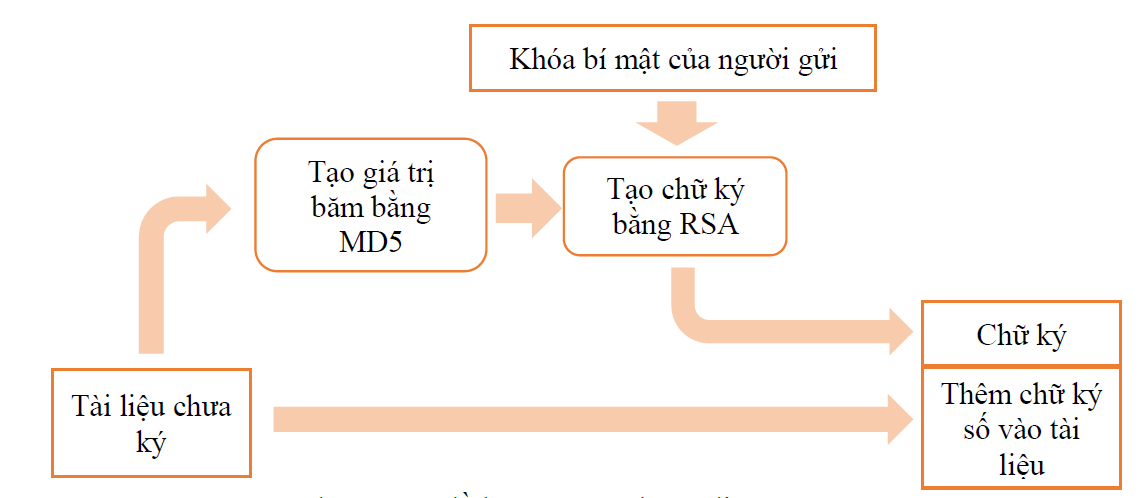
* Khóa riêng gồm 128 bit được sinh ra nhờ một passphrase và hàm băm MD5
* User ID.
* Key ID của khóa công khai tương ứng.

## 5.Chữ Ký Số.

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.



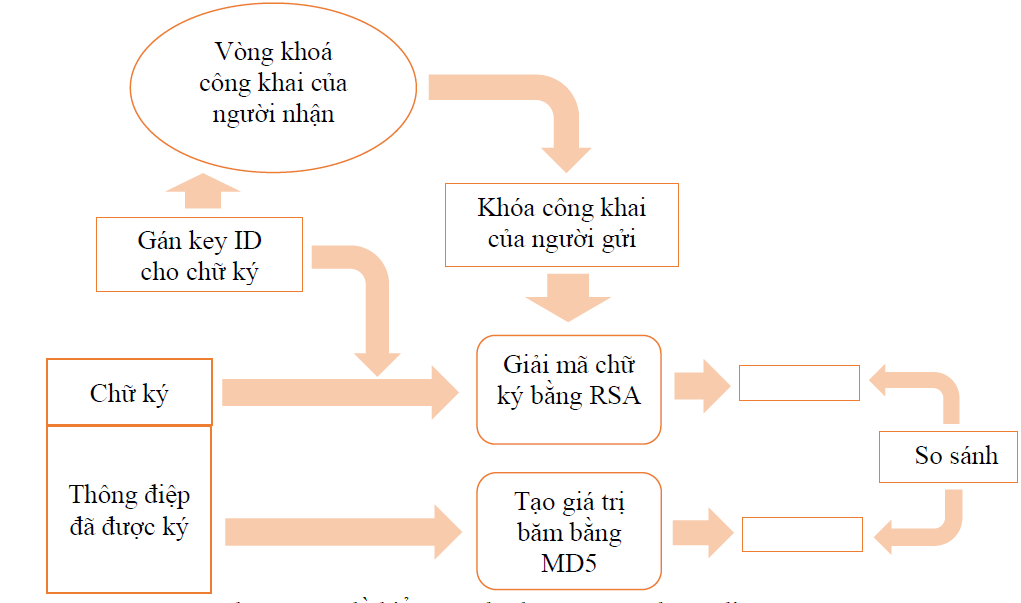
Hình 3: Lược đồ ký trên 1 thông điệp PGP

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 4: Lược đồ kiểm tra chũ 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.

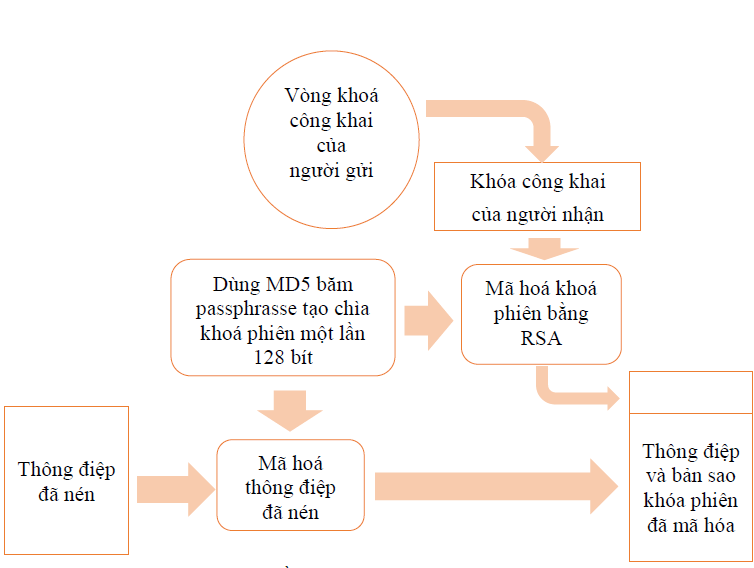
### 5.1 Nén

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.

### 5.2 Mã hóa và giải mã thông điệp

Một dịch vụ cơ bản khác của PGP cung cấp là mã hóa những thông điệp để truyền đi hoặc cất giữ trên máy tính. Trong cả hai trường hợp đều sử dụng giải thuật mã hóa truyền thống IDEA. Những phiên bản mới nhất, PGP sử dụng thuật toán AES thay vì IDEA.

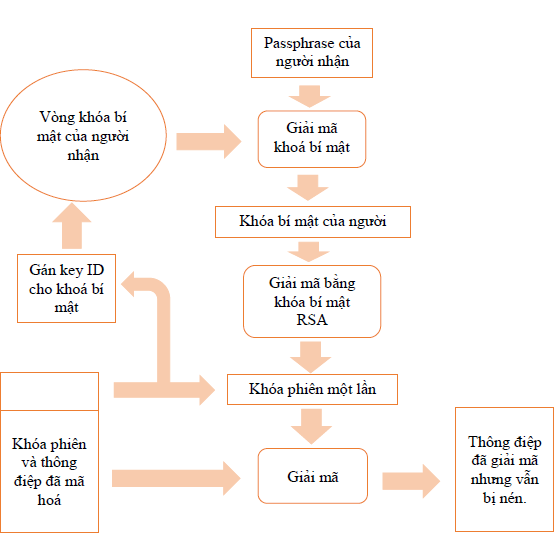
Trong khi các thuật toán mã hóa luôn chú trọng vào vấn đề phân phối khóa. Thì với PGP mỗi khoá truyền thống chỉ được sử dụng một lần; với mỗi thông điệp chỉ có một khóa 128 bít ngẫu nhiên được tạo ra. Vì chỉ được sử dụng một lần, nên khoá phiên được gắn vào thông điệp và truyền cùng với thông điệp. Để bảo vệ khoá phiên, PGP sử dụng RSA với khoá công cộng của người nhận.



Hình 5: Lược đồ mã hóa thông điệp trong PGP

Hình 6 minh họa vấn đề này bao gồm các bước sau:

1. PGP chỉ tạo một số 128 bit ngẫu nhiên nhờ việc băm passphrase của người gửi bằng MD5 và sử dụng nó làm khóa phiên cho thông điệp.
2. PGP mã hóa thông điệp sử dụng khóa phiên.
3. PGP mã hóa khóa phiên với RSA. Sử dụng khóa công khai của người nhận được gắn vào khóa phiên đó mã hóa.



Hình 6: Lược đồ giải mã thông điệp trong PGP

1. PGP lấy key ID được gán vào thông điệp và sử dụng nó để lấy khóa bí mật đúng từ vòng khóa bí mật. Một người dùng có thể có hơn một khóa riêng.
2. Người nhận cung cấp một passphrase. Nó cho phép PGP giải mã khóa riêng của người nhận.
3. PGP sử dụng RSA với khóa riêng để giải mã và khôi phục khóa phiên.
4. PGP sử dụng khóa phiên giải mã thông điệp.

## Tài liệu tham khảo:

1. <https://vi.wikipedia.org/wiki/Pretty_Good_Privacy>
2. <https://academy.binance.com/vi/articles/what-is-pgp>
3. Giáo trình Cơ Sở ATTT-PTIT
4. <https://123doc.net//document/2519337-tai-lieu-huong-dan-ma-hoa-voi-pgp.htm>