MỤC LỤC

# MỞ ĐẦU

Hiện nay trên thế giới, mạng máy tính đang ngày càng đóng vai trò thiết yếu trong mọi lĩnh vực hoạt động của toàn xã hội, nó đã và đang trở thành phương tiện trao đổi thông tin dữ liệu thì nhu cầu bảo mật thông tin được đặt lên hàng đầu. Nhu cầu này không chỉ có ở các bộ máy An ninh, Quốc phòng, Quản lý nhà nước mà đã trở thành bức thiết trong nhiều hoạt động kinh tế xã hội: tài chính, ngân hàng, thương mại … và trong cả hoạt động thường ngày như thư điện tử, thanh toán, tín dụng …

Trên thế giới hiện nay có khá nhiều giải pháp mã hóa thông tin theo công nghệ mới dựa trên các thuật toán có độ phức tạp cao và sản phẩm loại này cũng bắt đầu thương mại hóa. Tuy nhiên mức độ bảo mật và tốc độ xử lý của các loại sản phẩm rất khác nhau. Mặt khác dù có thuật toán tốt nhưng chúng ta không nắm bắt được mọi khía cạnh của công nghệ bảo mật sẽ không có cách nào bịt hết được mọi kẽ hở mà các tin tặc dễ dàng tấn công.Vì vậy để bảo mật các thông tin "nhậy cảm" thì giải pháp là tự xây dựng những chương trình bảo mật thông tin cho chính mình.

Nhu cầu đòi hỏi trên đặt ra cho các chuyên gia CNTT những thách thức mới: làm thế nào để vừa thỏa mãn các yêu cầu đòi hỏi về tốc độ xử lý, dải thông đường truyền truy cập của người sử dụng, đồng thời đảm bảo an toàn và bảo mật hệ thống thông tin, với việc mở rộng kết nối tới các hệ thống khác không nằm trong tầm kiểm soát của mình, để đảm bảo cho tốc độ phát triển chung của việc khai thác các tiềm năng, hiệu quả to lớn do mạng máy tính đem lại.

# CHƯƠNG 1: TÌM HIỂU HỆ ĐIỀU HÀNH MẠNG LINUX

## 1.1. Hệ điều hành mạng

### 1.1.1 Hệ điều hành Linux

Linux bắt nguồn từ một hệ điều hành lớn hơn có tên là UNIX. UNIX là một trong những hệ điều hành được sử dụng rộng rãi nhất trên thế giới do tính ổn định và khả năng hỗ trợ của nó. Về nguyên tắc hệ điều hành cũng là một software, nhưng đây là một software đặc biệt – được dùng để điều phối các tài nguyên (resource) của hệ thống (bao gồm cả hardware và software khác). Linux còn được gọi là Open Source Unix (OSU), Unix – like Kernel, clone of the UNIX operating system.

Linux là phiên bản UNIX được cung cấp miễn phí, ban đầu được phát triển bởi Linus Torvalds năm 1991 (sinh viên trường đại học Helsinki, Phần Lan). Khi Linus tung ra phiên bản miễn phí đầu tiên của Linux (0.02) trên Internet đã tạo ra một làn sóng phát triển phần mềm lớn nhất từ trước đến nay trên phạm vi toàn cầu. Hiện nay Linux được phát triển và bảo trì bởi một nhóm hàng nghìn lập trình viên cộng tác chặt chẽ với nhau qua Internet.

Tháng 11/1991 Linus đưa ra bản chính thức đầu tiên của Linux (0.02), nó có thể chạy bash và gcc (trình dịch C GNU – GNU's Not UNIX). Nhưng hệ thống chưa có các hỗ trợ người dùng và tài liệu hướng dẫn. Tháng 3/1994 phiên bản Linux 2.2.6, có thể làm việc trên môi trường đồ họa với các ứng dụng cao cấp như các tiện ích đồ họa và các tiện ích khác. Linux khó có thể thành công được như hiện nay nếu không có các công cụ GNU của tổ chức phần mềm miễn phí (Free Software Foundation). Trình dịch gcc của GNU đã giúp cho việc viết mã của Linux dễ dàng hơn rất nhiều. Hiện nay Linux là một hệ điều hành UNIX đầy đủ và độc lập. Nó có thể chạy X Window, TCP/IP, Emacs, Web, thư điện tử và các phần mềm khác.

### 1.1.2 Linux và UNIX

UNIX là một hệ điều hành mạnh, UNIX đã qua thử thách và chạy trên các máy chủ ở môi trường xí nghiệp rộng rãi trong một thời gian rất dài. Hệ điều hành UNIX đến nay vẫn chưa có đối thủ có thể đứng ngang với nó về tầm vóc cũng như sự chịu đựng về thời gian. Windows của Microsoft trước đây chỉ dùng cho các máy để bàn (desktop). Họ sản phẩm của Microsoft chưa bao giờ mang các tính năng của một máy chủ (server) thực thụ cho đến khi Windows NT và Windows 2000 ra đời. Tuy nhiên UNIX, NT và Windows 2000 đều là sản phẩm có bản quyền.

Linux trở lên phổ biến rộng rãi và được sự ủng hộ của rất nhiều lập trình viên trên thế giới. Điểm nổi bật của Linux là mã nguồn mở và tính ổn định do kế thừa từ kiến trúc UNIX đã qua thử thách.

Linux chỉ là hạt nhân cung cấp các chức năng cần thiết tối thiểu của một hệ điều hành tựa UNIX. Vì UNIX không có phiên bản chạy trên PCs theo kiến trúc bộ vi xử lý Intel nên Linux được xem là một sản phẩm rất giá trị.

### 1.1.3 Ưu điểm khi sử dụng Linux

Linux là hệ điều hành mã nguồn mở, được cung cấp miễn phí cho người sử dụng. Nó có khả năng đa nhiệm, đa xử lý, hỗ trợ mạng, khả năng tương thích phần cứng và nhiều tính năng khác:

- Tính ổn định: Linux có tính ổn định cao, ít bị lỗi khi sử dụng so với hầu hết các hệ điều hành khác. Người sử dụng Linux không phải lo lắng đến việc máy tính của mình bị "treo cứng" khi đang sử dụng nữa.
- Tính bảo mật: Linux là hệ điều hành đa nhiệm, đa người dùng (nhiều người sử dụng có thể vào phiên làm việc của mình trên cùng một máy tại cùng một thời điểm). Linux cung cấp các mức bảo mật khác nhau cho người sử dụng . Mỗi người sử dụng chỉ làm việc trên một không gian tài nguyên dành riêng, chỉ có người quản trị mới có quyền thay đổi trong máy.

- Tính hoàn chỉnh: Bản thân Linux đã được kèm theo các trình tiện ích cần thiết. Tất cả các trình tiện ích mà người sử dụng mong đợi đều có sẵn hoặc ở một dạng tương đương rất giống. Trên Linux, các trình biên dịch như C, C++, … các hạt nhân hay TCP/IP đều được chuẩn hóa.
- Tính tương thích: Linux tương thích hầu như hoàn toàn với một số chuẩn UNIX như IEEE POSIX.1, UNIX System V và BSD UNIX. Trên Linux cũng có thể tìm thấy các trình giả lập của DOS và Window, cho phép chạy các ứng dụng quen thuộc trên DOS và Window. Linux cũng hỗ trợ hầu hết các phần cứng máy PC.
- Hệ điều hành 32 bit đầy đủ: chúng ta không còn phải lo lắng về giới hạn bộ nhớ, các trình điều khiển EMM hay các bộ nhớ mở rộng… khi sử dụng Linux.
- Dễ cấu hình: Người sử dụng hầu như toàn quyền điều khiển về cách làm việc của hệ thống, không phải bận tâm về các giới hạn 640K và tiến hành tối ưu hóa bộ nhớ mỗi lần cài đặt một trình điều khiển mới.
- Khả năng làm việc trên nhiều loại máy: Cấu hình phần cứng tối thiểu chỉ là chip 80386, 2MB bộ nhớ, 10-20 MB không gian đĩa để bắt đầu. Linux có khả năng chạy trên nhiều dòng máy khác nhau như Apple Macintosh, Sun, Dec Alpha và Power PC.

Giống như UNIX, Linux là một hệ điều hành đa nhiệm, sử dụng sự quản lý bộ nhớ tinh vi và điều khiển tất cả các tiến trình, nếu một chương trình nào đó bị hỏng chúng ta có thể loại bỏ nó và tiếp tục làm các công việc khác. Linux gần như miễn dịch với sự tấn công của các loại virus.

Với sự phát triển và thay đổi liên tục về công nghệ và giao diện, Linux đã làm cho nhiều công ty, chính phủ, các tổ chức và người sử dụng quan tâm đến. Hàng ngàn website trên thế giới đã sử dụng Linux như một webserver, nhiều công ty lớn như HP, Ericssion đã sử dụng Linux như một hệ điều hành

chạy trên những sản phẩm của họ mà chúng ta đã biết thuật ngữ Linux Embedded.

Gần đây chính phủ của nhiều quốc gia đã chọn Linux trong chiến lược phát triển công nghệ bảo mật (security) chống lại sự tấn công của các tin tặc. Có nhiều phiên bản Linux đã phát triển thành những sản phẩm thương mại có độ ổn định cao, đáp ứng nhiều công việc như Red Hat Linux, Caldera, SuSE.

Với những việc làm trên, ta thấy Linux sẽ là hệ điều hành của tương lai đáp ứng được nhiều yêu cầu của người tiêu dùng trên khắp thế giới.

## 1.2. Một số đặc điểm của hệ điều hành mạng Linux

### 1.2.1 Đặc điểm của hệ thống

- Các version khác nhau của Linux: thông thường các nhân Linux (Linux kernel) có một số hiệu phiên bản riêng. Tại mỗi thời điểm có hai phiên bản mới nhất là phiên bản ổn định (Stable) và phiên bản phát triển (development). Phiên bản ổn định dành cho hầu hết người dùng, còn phiên bản phát triển thay đổi rất nhanh và được chạy thử bởi các nhà phát triển trên Internet. Phiên bản ổn định thường có số hiệu nhỏ hơn.
- Các đặc tính của hệ thống:
  - ✓ Linux là hệ điều hành đa nhiệm và đa người dùng.
  - ✓ Tương thích gần như hoàn toàn với các bản UNIX như chuẩn IEEE POSIX.1, System V, BSD.
  - ✓ Có thể được cài đặt cùng với các hệ điều hành khác như Windows 95/98, Windows NT, OS/2, hoặc các phiên bản khác của Linux. Chương trình tải hệ thống Linux (LILO) cho phép chọn hệ điều hành khi khởi động.
  - ✓ Linux có thể chạy trên nhiều cấu trúc CPU khác nhau như Intel, SPAERC, Alpha, Power PC, MIPS và m68k.

- ✓ Hỗ trợ nhiều hệ thống file khác nhau như: hệ thống file mở rộng (ext2fs) dành riêng, MS DOS file, Windows, OS2, Apple, …
- ✓ Mạng là một trong những điểm mạnh của Linux. Linux cũng cung cấp đủ các dịch vụ giao thức mạng TCP/IP, bao gồm các Drive thiết bị cho card Ethernet, PPP và SLIP, PLIP (Parallel Line Internet Protocol) và NFS (Network file system). Hỗ trợ các dịch vụ như FTP, Telnet, NNTP và SMTP (Simple Mail Transfer Protocol). Kernel Linux hỗ trợ bức tường lửa (firewall) cho mạng, người sử dụng có thể đặt một cấu hình một máy Linux bất kỳ như một filewall.

- Kernel: là phần chính, là trái tim của hệ điều hành, nó điều khiển giao diện giữa chương trình người sử dụng với các thiết bị phần cứng, xếp lịch các tiến trình và nhiều tác vụ khác của hệ thống. Kernel không phải là một tiến trình chạy riêng biệt trong hệ thống mà là các tập trình đơn nằm trong bộ nhớ, mọi tiến trình đều gọi đến chúng. Trên nhiều cấu trúc máy tính, kernel có thể mô phỏng các lệnh dấu phẩy động (PFU) nếu bộ nhớ không có bộ đồng xử lý toán học. Kernel Linux cũng hỗ trợ các kỹ thuật như: phân trang bộ nhớ, bộ nhớ ảo, cache đĩa, …

### 1.2.2 Các đặc điểm phần mềm

- Các lệnh cơ bản và các tiện ích: tất cả các tiện ích và các lệnh cơ bản của UNIX đều được chuyển sang Linux. Các lệnh cơ bản như ls, awk, tr, sed, bs, more, và các phần mềm như Perl, Python, Java Deverlopment Kit. Các trình soạn thảo văn bản như: vi, ex, GNU emacs.

Một trong những tiện ích quan trọng nhất trong Linux là shell. Shell là một chương trình cho phép đọc và thực hiện các lệnh của người dùng. Trong shell người ta có thể viết các shell script, tương tự như file Bat trong MSDOS, đó

là các tệp chứa các chương trình ngôn ngữ lệnh Shell trong Linux như C shell, Bash shell (GNU Bourne Again shell), ksh (Korn shell).

- Các ngôn ngữ lập trình: Linux cung cấp một môi trường lập trình UNIX đầy đủ, bao gồm mọi thư viện chuẩn, các công cụ lập trình, trình dịch và gỡ rối. Hai ngôn ngữ lập trình phổ biến nhất là C và C++ được hỗ trợ trong Linux với trình dịch gcc của GNU. Bộ Java Deverlopment Kit của Sun cũng được đưa vào Linux. Các ngôn ngữ lập trình khác như Smalltalk, Fortran, Pascal, Lisp,…
- Hệ thống X Window: là giao diện đồ họa chuẩn cho các máy UNIX. Phiên bản X Window trên Linux là XFree86. Các ứng dụng chuẩn trên X Window là xterm (bộ mô phỏng đầu cuối dùng cho các ứng dụng ở các chế độ text), xdm (quản lý việc vào ra hệ thống của người dùng), xclock (đồng hồ), xman (bộ đọc trang hướng dẫn đồ họa).

Giao diện X Window được điều khiển bởi chương trình window manager (đặt các cửa sổ, cho phép thay đổi kích thước, đặt biểu tượng, di chuyển cửa sổ,đặt kiểu của khung cửa sổ). Giao diện X Window được đảm nhiệm bởi chương trình XFree86. Chương trình XFree86 chứa chương trình window manager chuẩn MIT twm, các thư viện lập trình và các file includes cho các nhà phát triển có thể phát triển các ứng dụng trên X Window. X Window cũng hỗ trợ Athena, Openlook, Xaw3D, các công cụ đồ họa 3 chiều như PEX, Mesa (phiên bản cài đặt miễn phí của OpenGL 3D).

- KDE và GNOME: là hai dự án quan trọng trong thế giới Linux. Hầu hết các phiên bản Linux đều cho phép đặt cấu hình một cách tự động một trong hai chương trình trên. Mục tiêu chính của KDE là dễ sử dụng, ổn định và giao diện người dùng tương thích với các môi trường khác trong khi GNOME chú ý đến giao diện đẹp mắt và có khả năng đặt cấu hình tối đa.

- Giao tiếp với Windows và MS-DOS: Linux có rất nhiều tiện ích cho phép có thể giao tiếp với Windows và MS-DOS như Wine – trình giả lập Microsoft Windows trên X Window trong Linux cho phép các ứng dụng trên windows có thể chạy trên Linux, trình giả lập MS-Dos trên Linux cho phép chạy các ứng dụng dưới DOS trên Linux.

### 1.2.3 Linux và mạng

Linux là một trong những hệ điều hành mạng mạnh nhất, hỗ trợ hai giao thức cơ bản cho các hệ thống UNIX: TCP/IP và UUCP.

Hầu hết các mạng TCP/IP đều sử dụng card mạng Ethernet để kết nối. Linux hỗ trợ rất nhiều card Ethernet thông dụng cũng như các loại Fast Ethernet, Gigabit Ethernet, ATM, ISDN, mạng Lan không dây, Token Ring, packet ratio và các giao diện mạng hiệu năng cao khác.

Linux cũng hỗ trợ PPP và SLIP, cho phép kết nối Internet qua modem, hỗ trợ các trình duyệt Web như: Netscape và các web server như Apache.

Samba là gói phần mềm cho phép các máy tính Linux hoạt động như các file server và các print server trên Windows. NFS cho phép hệ thống có thể chia sẻ các tệp giữa các máy tính với nhau trên mạng. Với NFS cho phép nhìn các tệp ở xa giống như trên chính máy tính của người sử dụng. Giao thức FTP (File Transfer Protocol) cho phép truyền các tệp giữa các máy tính trên mạng với nhau.

Các dịch vụ truyền thư điện tử như: Send mail, exim, Smail, các dịch vụ telnet, rlogin, ssh và rsh cho phép truy nhập và làm việc trên một máy tính khác trên mạng. Linux cũng hỗ trợ TCP/IP và cung cấp một giao diện lập trình socket chuẩn. Transmission Control Protocol và Internet Protocol là hai giao thức chính của họ TCP/IP

## 1.3. Tìm hiểu nhân của hệ điều hành Linux

### 1.3.1 Bộ phân thời cho tiến trình (Process Scheduler - SCHED)

Các hệ điều hành đa nhiệm cho phép nhiều chương trình chạy cùng một lúc bằng cách chuyển quyền thực thi qua lại giữa các chương trình thật nhanh, làm cho chúng ta có cảm giác các chương trình chạy cùng một lúc với nhau.

### 1.3.2 Bộ quản lý bộ nhớ (Memory Manager - MM)

Bộ nhớ quy ước (conventional memory) của PC chỉ có 640K do chương trình BIOS chỉ quản lý được tới FFFFF, mà vùng nhớ cao (High memory từ A0000 trở lên) dùng để ánh xạ (map) BIOS, Video card memory và các thiết bị ngoại vi khác, vùng nhớ còn xài được (Low memory) là từ 9FFFF trở xuống. Ở chế độ bảo vệ (protect mode) của CPU 32 bit đưa ra khái niệm virtual memory (bộ nhớ ảo). Lúc này mỗi process được cấp cho 4G virtual memory từ 00000000-FFFFFFFF. Nhưng kernel sẽ giữ một table mô tả ánh xạ từng page của virtual memory với physical memory. Physical memory bây giờ bao gồm cả RAM và swap disk space. Tất nhiên 4G virtual memory không bao giờ được ánh xạ đầy đủ. Phần lớn mặc dù có đánh địa chỉ nhưng chỉ khi ta đọc hoặc ghi lên đó thì kernel mới định phần từ physical memory.

### 1.3.3 Hệ thống file ảo (Virtual File System - VFS)

Hệ thống này không chỉ cung cấp truy xuất đến hệ thống file trên harddisk mà còn cho tất cả các thiết bị ngoại vi. Ý tưởng này bắt nguồn từ UNIX và các hệ điều hành sau này đều thiết lập theo hướng đấy.

### 1.3.4 Giao diện mạng (Network Interface - NET)

Linux dựng sẵn TCP/IP trong kernel.

### 1.3.5 Bộ truyền thông nội bộ (Inter Process Communication IPC)

Cung cấp các phương tiện truyền thông giữa các tiến trình trong cùng hệ thống Linux.

## 1.4. Các cấu trúc dữ liệu hệ thống

Hệ điều hành Linux hoạt động nhờ vào các dữ liệu này:

- ✓ Task List (danh sách tác vụ): SCHED lưu một bộ dữ liệu cho mỗi tiến trình đang hoạt động. Các bộ dữ liệu này làm thành một danh sách liên kết gọi là danh sách tác vụ. SCHED còn có một con trỏ current để chỉ tác vụ nào đang hoạt động.
- ✓ Memory map (ánh xạ bộ nhớ): MM cần một ánh xạ từ bộ nhớ vật lý cho bộ nhớ ảo 4G của mỗi tiến trình. Ngoài ra còn các thông tin để chỉ cách lấy và thay cho từng trang cụ thể. Tất cả các thông tin này chứa trong memory map và memory map được chứa trong task list.
- ✓ I-nodes: VFS dùng I-nodes để định vị các file. Cấu trúc dữ liệu i-nodes dùng để ánh xạ các file block thành các địa chỉ vật lý ở trường hợp đĩa cứng và đĩa mềm là các sector, cyclinder và head.
- ✓ Data connection: mô tả network connection đang mở

## 1.5. Cấu trúc của SCHED

Đây là bộ phận trung tâm của hệ điều hành. SCHED được chia thành 4 module:

- ✓ Module luật định thời (scheduling policy): chịu trách nhiệm phân xử xem process nào được quyền truy xuất CPU. Hệ thống hoạt động có thông suốt hay không nhờ vào bộ luật này, tránh trường hợp 1 process lợi dụng sơ hở của điều luật mà chiếm thời gian hệ thống quá nhiều làm các process bị đóng băng (freeze).

- ✓ Module phụ thuộc kiến trúc (architecture-specific): module gồm các code assembly phụ thuộc vào mỗi loại CPU dùng để suspend hay assume process.
- ✓ Module độc lập kiến trúc (architecture-independent): Module gọi các hàm từ module phụ thuộc kiến trúc và module luật để chuyển đổi giữa các process đồng thời nó còn gọi các hàm ở MM để thiết lập virtual memory cho các process được bắt đầu lại.
- ✓ Module hàm gọi hệ thống (system call): là các hàm mà user có thể dùng để tương tác với SCHED.

# CHƯƠNG 2: MẬT MÃ KHÓA CÔNG KHAI

## 2.1. Một số khái niệm cơ bản

### 2.1.1 Số học modulo

***Định nghĩa 1***:

Giả sử a và b là các số nguyên và n là một số nguyên dương. Khi đó ta viết $a \equiv b \pmod n$ nếu n chia hết cho a-b. Mệnh đề $a \equiv b \pmod n$ được gọi là "a đồng dư với b theo modulo n", số nguyên n được gọi là modulus.

Giả sử chia a và b cho n ta thu được các thương nguyên và phần dư nằm giữa 0 và n-1, nghĩa là $a = q_1n+r_1$ và $b = q_2n+r_2$ trong đó $0 \leq r_1, r_2 \leq n-1$. Khi đó có thể thấy rằng $a \equiv b \pmod n$ khi và chỉ khi r1 = r2.

***Định nghĩa 2***: Số học modulo n

$Z_n$ được coi là tập hợp {0, …, n-1} được trang bị hai phép toán cộng và nhân. Phép toán cộng và nhân trong $Z_n$ được thực hiện giống như cộng và nhân các số thực, ngoại trừ một điểm các kết quả được rút gọn theo modulo n.

Phép cộng và phép nhân trên $Z_n$ thỏa mãn các tính chất sau:

$\forall a, b \in Z_n \rightarrow a + b \in Z_n$

$\forall a, b \in Z_n \rightarrow a + b = b + a$

$\forall a, b, c \in Z_n \rightarrow (a + b) + c = a + (b + c)$

$\forall a \in Z_n , 0 \in Z_n$ mà $a + 0 = 0 + a = a$

$\forall a \in Z_n , n\text{-}a \in Z_n$ mà $a + (n - a) = (n - a) + a = 0$

$\forall a, b \in Z_n \rightarrow ab \in Z_n$

$\forall a, b \in Z_n \rightarrow ab = ba$

$\forall a, b, c \in Z_n \rightarrow (ab)c = a(bc)$

$\forall a \in Z_n , \exists 1 \in Z_n$ mà $a1 = 1a = a$

$\forall a, b, c \in Z_n \rightarrow (a+b)c = ac + bc$ và $a(b+c) = ab+ac$

$Z_n$ thỏa mãn các tính chất trên là một vành.

### 2.1.2 Hàm Euler

***Định lý 1***:

Đồng dư thức $ax \equiv b \pmod n$ chỉ có một nghiệm duy nhất $x \in Z_n$ với mọi $b \in Z_n$ khi và chỉ khi UCLN(a,n)=1.

***Định nghĩa 3***:

Giả sử $a \geq 1$ và $n \geq 2$ là các số nguyên. Nếu UCLN(a,n) = 1 thì ta nói rằng a với n là nguyên tố cùng nhau. Số các số nguyên trong $Z_n$ nguyên tố cùng nhau với n ký hiệu là $\phi(n)$ (hàm này được gọi là hàm Euler).

***Định lý 2***:

Giả sử $n = \prod_{i=1}^{m} p^{ei}_i$ trong đó các số nguyên tố $p_i$ khác nhau và $e_i > 0$, $1 \leq i \leq m$.

Khi đó $\phi(n) = \prod_{i=1}^{m} (p^{ei}_i - p^{ei\,-1}_i)$.

***Định nghĩa 4***:

Giả sử $a \in Z_n$ phần tử nghịch đảo (theo phép nhân) của a là phần tử $a^{-1} \in Z_n$ sao cho $aa^{-1} \equiv a^{-1}a \equiv 1 \pmod n$.

Ta thấy rằng $a \in Z_n$ có nghịch đảo theo modulo n khi và chỉ khi UCLN(a,n)=1.

### 2.1.3 Thuật toán Euclide

Ta có $Z_n$ là một vành với một số nguyên dương bất kỳ n. Ta cũng biết $b \in Z_n$ có phần tử nghịch đảo của phép nhân khi và chỉ khi UCLN(b,n) = 1 và các số nguyên dương nhỏ hơn n mà nguyên tố cùng nhau với n bằng $\phi(n)$ (tổng quát, giả sử $n = \prod_{i=1}^{m} p^{ei}_i$ trong đó các số nguyên tố $p_i$ khác nhau và $e_i > 0$, $1 \leq i \leq m$. Khi đó $\phi(n) = \prod_{i=1}^{m} (p^{ei}_i - p^{ei\,-1}_i)$ ).

Tập các thặng dư theo modun n và nguyên tố cùng nhau với n ký hiệu là $Z^*_n$ đều có phần tử nghịch đảo.

Trước hết ta xem thuật toán Euclide thông thường được dùng để tính UCLN của 2 số nguyên dương $r_0$ và $r_1$ với $r_0 > r_1$. Thuật toán Euclide bao gồm thực hiên dãy các phép chia sau:

$r_0 = q_1r_1 + r_2$ $\quad 0<r_2<r_1$

$r_1 = q_2r_2 + r_3$ $\quad 0<r_3<r_2$

.....

$r_{m-2} = q_{m-1}r_{m-1} + r_m$ $\quad 0<r_m<r_{m-1}$

$r_{m-1} = q_mr_m$ $\quad 0<r_m<r_{m-1}$

Khi đó ta có UCLN($r_0$,$r_1$) = UCLN($r_1$,$r_2$) = … = UCLN($r_{m-1}$,$r_m$) = $r_m$ vì vậy UCLN($r_0$,$r_1$) = $r_m$.

***Định lý 3***:

Giả sử cho dãy số $t_1$, $t_1$, …, $t_m$ xác định theo công thức truy toán sau:

$t_0 = 0$

$t_1 = 1$

$t_t = t_{j-2} – q_{j-1}t_{j-1} \bmod r_0$ nếu j>=2

Nới $0 \le j \le m$ ta có $r_j \equiv t_jr_j \pmod{r_0}$ trong đó các giá trị $q_j$, $r_j$ được xác định theo thuật toán Euclide.

Chứng minh:

Ta chứng minh bằng quy nạp toán học theo j, định lý hiển nhiên đúng với j =0 và j=1.Giả sử định lý cũng đúng với j=i-1 và j=i-2 trong đó $i \ge 2$.Ta đi chứng minh định lý đúng với i=j. Theo quy nạp ta có:

$r_{i-2} \equiv t_{i-2}\ r_1 \pmod{r_0}$.

$r_{i-1} \equiv t_{i-1}r_1 \pmod{r_0}$.

Ta có : $r_i = r_{i-2}-q_{i-1}r_{i-1.}$

$= t_{i-2}r_1-q_{i-1}t_{i-1}r_1 \pmod{r_0}$.

$= (t_{i-2}-q_{i-1}t_{i-1})r1 \pmod{r_0}$.

$= t_ir_1 \pmod{r_0}$.

Định lý được chứng minh.

***Hệ quả 1***:

Giả sử UCLN($r_0$,$r_1$)=1, khi đó $t_m = r_1^{-1} \bmod r_0$.

Thật vậy theo định lý trên ta có $r_m \equiv t_m r_1 \pmod{r_0}$, mà UCLN($r_0$,$r_1$)=1=$r_m$, vậy $1 \equiv t_m r_1 \pmod{r_0}$, suy ra $t_m = r_1^{-1} \pmod{r_0}$.

Một thuật toán tính phần tử nghịch đảo $t_m$ gọi là thuật Euclide mở rộng.

### 2.1.4 Các kiến thức cần thiết khác

***Định nghĩa 5:***

Với G là một nhóm nhân hữu hạn, cấp của phần tử g Є G là số nguyên dương m bé nhất sao cho $g^m = 1$.

***Định lý 4:***

Giả sử G là một nhóm cấp n và g Є G. Khi đó cấp của g là ước của n

***Hệ quả 2:***

Nếu b Є $Z^*_n$ thì $b^{\phi(n)} \equiv 1 \pmod n$

Chứng minh: Ta có $Z^*_n$ có cấp là $\phi(n)$ suy ra $b^{\phi(n)} \equiv 1 \pmod n$ theo định lý trên

***Hệ quả 3:*** *(Ferma)*

Giả sử p là số nguyên tố và b Є $Z_p$. Khi đó $b^p \equiv b \pmod p$

Chứng minh: do $\phi(p)=p-1$ theo hệ quả trên ta có $b^{\phi(p)} \equiv 1 \pmod p$

hay $b^{p-1} \equiv 1 \pmod p$ vậy $b^p \equiv b \pmod p$.

Ta biết rằng nếu p là số nguyên tố thì $Z_p^*$ là một nhóm cấp p-1 và một phần tử bất kỳ trong nhóm $Z_p^*$ sẽ có bậc là ước của p-1. Tuy nhiên nếu p là số nguyên tố thì nhóm $Z_p^*$ là nhóm cyclic tồn tại một phần tử $\alpha \in Z_p^*$ có cấp bằng p-1.

***Định lý 6:***

Nếu p là số nguyên tố thì $Z_p^*$ là nhóm cyclic.

Một phần tử $\alpha$ có cấp p-1 được gọi là phần tử nguyên thủy modulo p xét thấy $\alpha$ là một phần tử nguyên thủy khi và chỉ khi: $\{\alpha^i : 0 \le i \le p-1\} = Z_p^*$

Giả sử p là nguyên tố và $\alpha$ là phần tử nguyên thủy modulo. Một phần tử bất kỳ $\beta \in Z_p^*$ có thể được viết như sau: $\beta = \alpha^i$ trong đó $0 \leq i \leq p-2$ (theo một cách duy nhất). Không khó khăn để chứng tỏ $\beta = \alpha^i$ là: $\frac{p-1}{UCLN(p-1,i)}$

Vậy bản thân $\beta$ sẽ là phần tử nguyên thủy khi và chỉ khi UCLN(p-1,i) = 1 dẫn đến số các phần tử theo modulo p bằng $\phi(p-1)$.

## 2.2. Khái niệm mã hóa bằng khóa công khai

- Khóa công khai:

Đối với hệ mật khóa bí mật yêu cầu phải có thông tin trước về khóa K giữa A và B qua một kênh an toàn trước khi gửi một bản mã bất kỳ nhưng rất khó đảm bảo nếu A và B ở cách xa nhau.

Ý tưởng một hệ mật khóa công khai là tìm một hệ mật không có khả năng tính toán để xác định $d_k$ nếu đã biết $e_k$. Nếu thực hiện được như vậy thì quy tắc mã $e_k$ có thể được công khai bằng cách công bố nó trong một danh bạ. Ưu điểm của hệ mật khóa công khai là A (hoặc bất kỳ ai) gửi một bản tin đã mã cho B mà không cần thông tin trước về khóa bằng cách dùng luật mã công khai $e_k$. B là người duy nhất có thể giải được bản mã này bằng cách sử dụng luật giải mã bí mật $d_k$ của mình.

Một hệ mật khóa công khai không bao giờ có thể đảm bảo được độ mật tuyệt đối. Vì khi nghiên cứu một bản mã kẻ thám mã có thể mã lần lượt các bản rõ có thể bằng luật mã công khai $e_k$ cho tới khi tìm được bản rõ duy nhất x đảm bảo $y = e_k(x)$. Bởi vậy ta chỉ nghiên cứu về độ mật về mặt tính toán của các hệ này.

Khi nghiên cứu về hệ mật khóa công khai ta cần quan tâm đến khái niệm hàm cửa sổ sập một chiều: Hàm mã hóa công khai $e_k$ của B phải là một hàm dễ tính toán nhưng việc tính hàm ngược lại phải rất khó khăn (đối với bất kỳ ai không phải là B). Đặc tính dễ tính toán nhưng khó tính ngược gọi là đặc

tính một chiều. Vì vậy cần thiết $e_k$ là một hàm một chiều. Trong thực tế nhiều hàm được coi là hàm một chiều nhưng cho tới nay vẫn không tồn tại một hàm nào có thể chứng minh được là một chiều.

Để xây dựng một hệ mật khóa công khai thì việc tìm được hàm một chiều vẫn chưa đủ. B phải có một cửa sập chứa thông tin bí mật cho phép dễ dàng tìm được $e_k$. Vì vậy hàm được coi là cửa sập một chiều và trở nên dễ tính ngược nếu biết một cửa sập nhất định.

- Tiêu chuẩn của một hệ mật khóa công khai:

Trong phương pháp mật mã dùng khóa công khai, mỗi người tham gia mạng có hai khóa, một khóa bí mật riêng gọi là khóa riêng (ký hiệu KR), một khóa công khai cho mọi người gọi là khóa công khai (ký hiệu KU).

Một bản tin nếu được mã hóa bằng một trong hai khóa thì chỉ có thể được giải mã bằng khóa còn lại.

Mỗi hệ mật khóa công khai đều phải đạt được các yêu cầu sau:

Mỗi thực thể B tham gia mạng, dễ dàng có được một cặp khóa KUb, KRb, khi một thực thể A muốn gửi một thông báo bí mật X đến thực thể B nó phải dễ dàng thực hiện mã hóa bằng hàm cửa sổ sập một chiều để sinh ra bản mã: $Y = E_{KUb}(X)$.

Khi thực thể B nhận được bản mã Y được gửi đến thì nó phải dễ dàng giải mã Y thành X bằng khóa riêng KRb của mình:

$X = D_{KRb}(Y) = D_{KRb}(E_{KUb}(X))$.

Đối phương không thể tìm ra được KR nếu biết KU trong thời gian cho phép.

Với KU và bản mã $Y = E_{KU}(X)$ đối phương không thể tìm ra được X.

Hàm mã hóa và giải mã có thể được sử dụng theo thứ tự ngược lại:

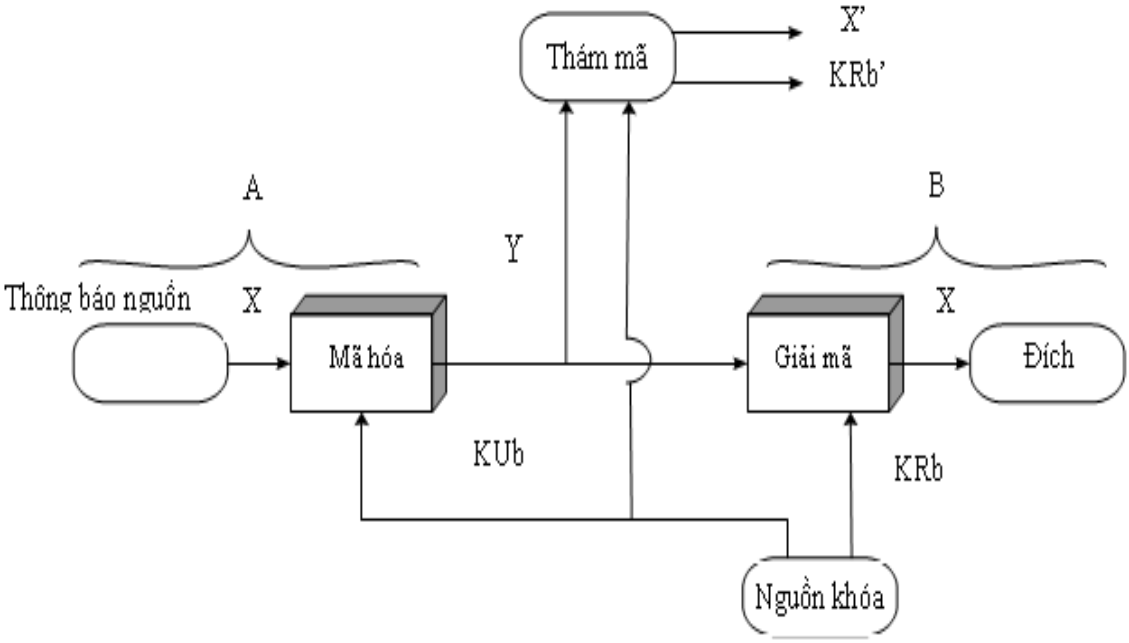
$M = D_{KRb}(E_{KUb}(M))$, $M = E_{KRb}(D_{KUb}(M))$.

## 2.3. Mô hình bảo vệ thông tin của mật mã khóa công khai

### 2.3.1 Một số mô hình bảo vệ thông tin

***a. Mô hình bí mật (secrecy)***

Giả xử A và B là 2 thành viên trong hệ thống mật mã khóa công khai và A muốn gửi cho B một thông báo đòi hỏi phải được giữ bí mật. Giả sử A là một thành viên nào đó trong mật mã khóa công khai, cặp khóa của A ký hiệu là KRa (khóa riêng) và KUa (khóa công khai). Nếu biết khóa công khai của A không thể tìm được khóa riêng của A theo nghĩa độ phức tạp tính toán.



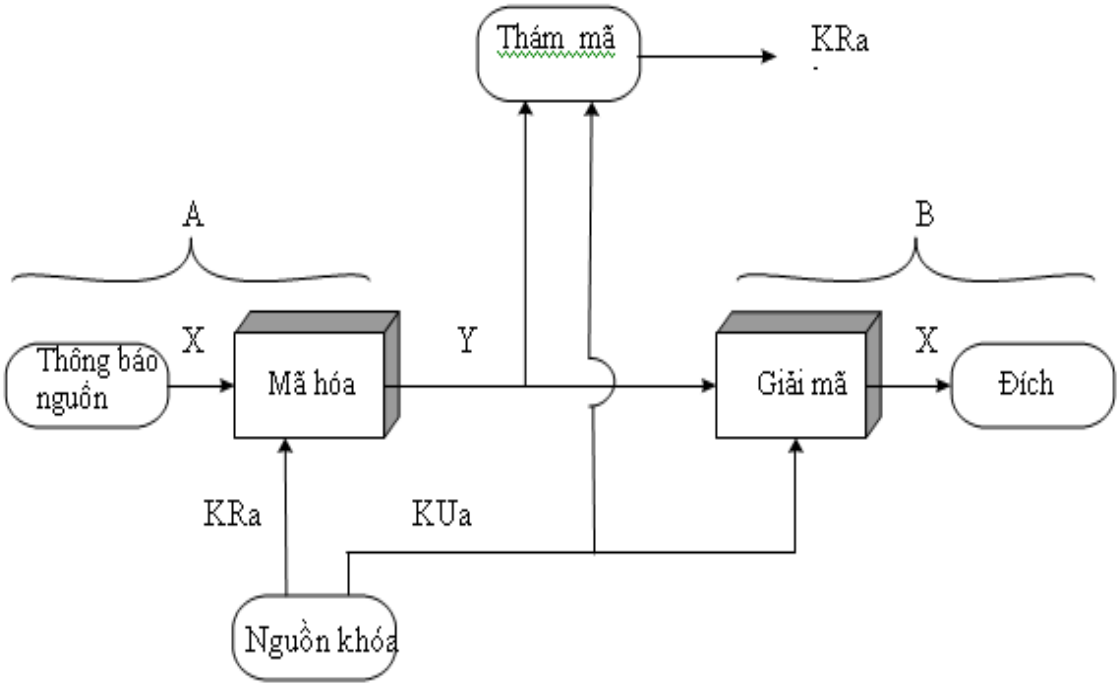

Hình 3.1 Khóa công khai – Mô hình bí mật

A sinh ra thông báo X ở dạng rõ, mã thông báo X bằng khóa công khai KUb để nhận được bản mã $Y=E_{KUb}(X)$, gửi bản mã Y cho B.

B nhận bản mã Y, giải mã Y bằng khóa riêng KRb. Nếu thám mã chỉ quan tâm đến X thì sẽ cố sinh ra bản rõ ước lượng X' của X, nếu thám mã muốn đọc các thông báo tiếp theo thì phải khôi phục KRb bằng việc sinh ra ước lượng KRb' của KRb.

***b. Mô hình xác thực (authentication)***

Khi A muốn gửi cho B một thông báo muốn xác thực:




Hình 3.2 Khóa công klhai – Mô hình xác thực

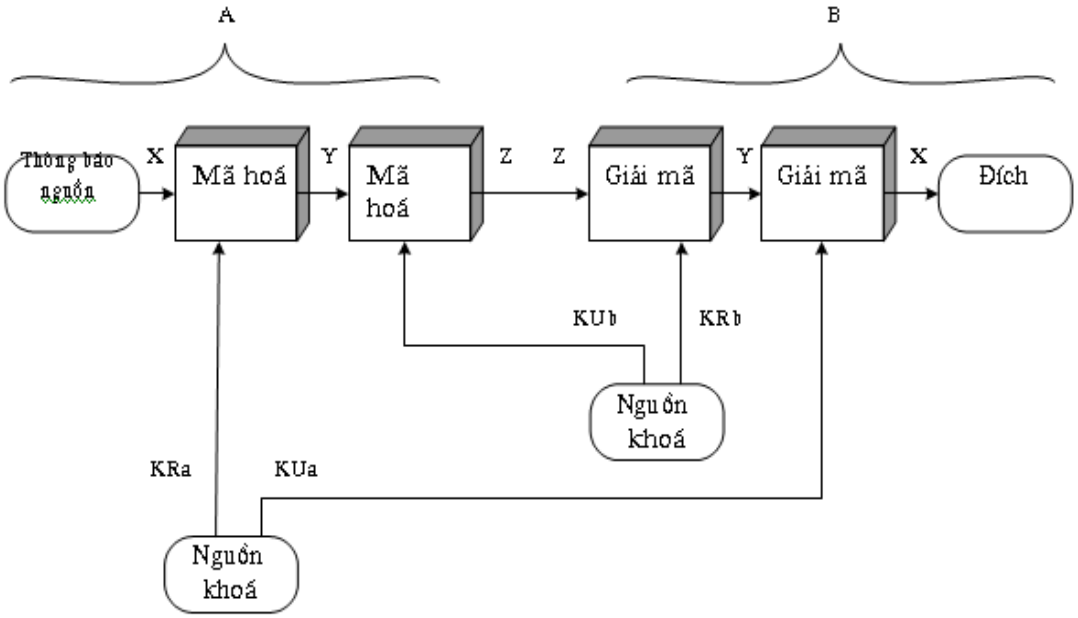
A sinh ra một bản rõ X, mã hóa bằng khóa riêng KRa của mình để nhận được bản mã $Y=E_{KRa}(X)$, gửi bản mã Y cho B.

B nhận bản mã Y và giải mã bằng khóa công khai KUa cua A để nhận được $X=D_{KUa}(Y)$

Trong mô hình này chỉ A mới có KRa tạo ra được Y, dùng khóa công khai của A để giải mã. Nếu thực hiện mã toàn bộ thông báo sẽ đòi hỏi không gian lưu trữ lớn và tốc độ hạn chế, để hiệu quả người ta chỉ mã một khối nhỏ các bit được tạo ra bởi một hàm biến đổi trên bản rõ, được gọi là hàm Hash.

***c. Mô hình bí mật và xác thực (secrecy and authentication)***

Kết hợp cả hai mô hình trên.




Hình 3.3 Khóa công khai – Mô hình bí mật, xác thực

A tạo thông báo X, mã hóa X bằng khóa riêng được bản mã $Y=E_{KRa}(X)$, rồi mã hóa Y bằng khóa công khai của B được bản mã $Z=E_{KUb}(Y)$, gửi Z cho B.

B nhận bản mã Z, giải mã bằng khóa riêng của mình nhận được $Y=D_{KRb}(Z)$ giải mã Y bằng khóa công khai của A nhận được bản mã $X=D_{KUa}(Y)$.

### 2.3.2 Các ứng dụng của mật mã khóa công khai

Hệ thống mật mã khóa công khai được đặc trưng bởi việc dùng một thuật toán mã với hai khóa riêng và khóa công khai. Phụ thuộc vào các ứng dụng người gửi dùng khóa công khai của người nhận hoặc dùng khóa riêng của người gửi hoặc dùng cả hai để mã hóa thông báo. Người nhận giải mã theo cách ngược lại.

Có 3 loại ứng dụng của mật mã khóa công khai trong bảo mật thông tin trên mạng:

- ✓ Mã hóa và giải mã: người gửi mã thông báo bằng khóa công khai của người nhận, người nhận giải mã bằng khóa riêng của mình.
- ✓ Chữ ký điện tử: người gửi ký một thông báo bằng khóa riêng của mình, chữ ký thu được bởi việc mã thao tác trên thông báo hoặc một khối bit được tạo ra từ thông báo bởi hàm Hash. Người nhận giải mã bằng cách dùng khóa công khai của người gửi.
- ✓ Trao đổi khóa: người gửi và người nhận sử dụng mã khóa công khai để trao đổi khóa phiên.

### 2.3.3 Yêu cầu đối với mật mã khóa công khai

Hai người dùng A, B có nhu cầu trao đổi thông tin với nhau.

- ✓ A và B dễ dàng sinh ra cặp khóa công khai KU và khóa riêng KR
- ✓ A dễ dàng tính toán để thu được bản mã $Y=E_{KUb}(X)$ khi biết KUb và X.
- ✓ B dễ dàng tính toán để thu được $X=D_{KRb}(Y)$.
- ✓ Kẻ tấn công không thể tính toán thu được KRb từ KUb.
- ✓ Kẻ tấn công không thể tính toán thu được X khi biết KUb và bản mã Y.
- ✓ Các hàm mã và giải mã thỏa mãn $X=E_{KUb}(D_{KRb}(X))$.

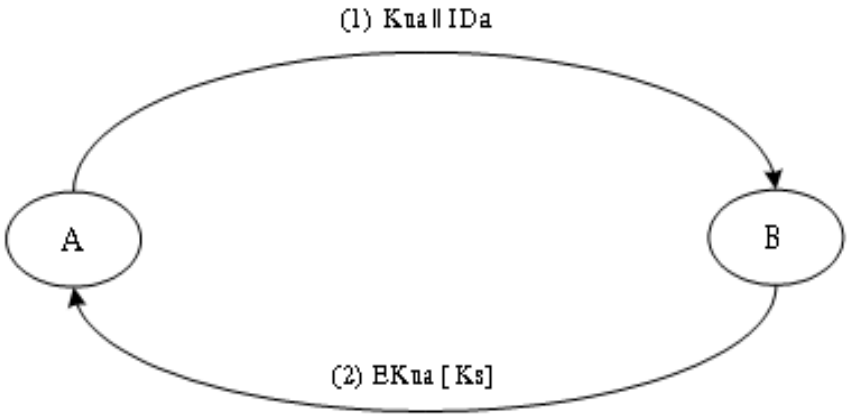
## 2.4. Các phương pháp phân phối khóa công khai

- ✓ Thông báo khóa công khai (announcement)
- ✓ Thư mục khóa công khai (directory)
- ✓ Thẩm quyền khóa công khai (authority)
- ✓ Chứng nhận khóa công khai (certificates)

## 2.5. Dùng mật mã khóa công khai phân phối khóa bí mật

### 2.5.1 Phân phối khóa bí mật đơn giản

Hai thành viên A, B muốn truyền thông với nhau dùng mật mã khóa bí mật, A muốn B gửi cho A một khóa phiên Ks bằng cách dùng mật mã khóa công khai.




Hình 3.8 phân phối khóa bí mật đơn giản

Thủ tục:

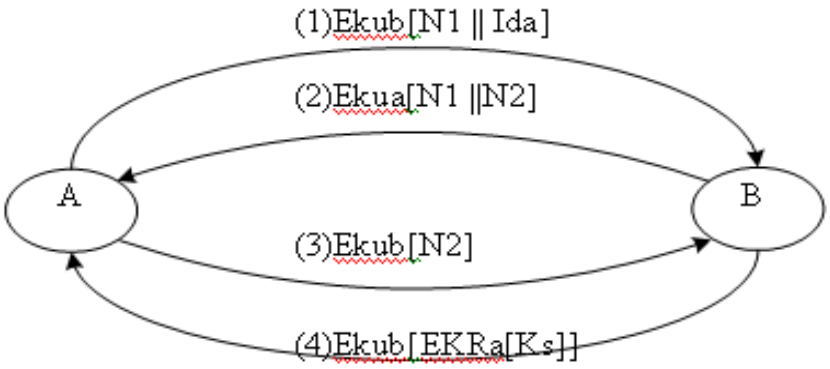
- ✓ A sinh ra một cặp khóa công khai/riêng (KUa, KRa) và truyền thông báo (1) tới B bao gồm KUa và định danh IDa của A
- ✓ B sinh một khóa bí mật Ks, mã Ks bằng khóa công khai của A và gửi cho A bản mã EKUa[Ks].
- ✓ A giải mã DKRa[EKUa[Ks]] để khôi phục khóa bí mật Ks. Vì chỉ có A có thể giải mã thông báo nên chỉ có A và B biết khóa Ks.
- ✓ A hủy KUa, KRa và B hủy KUa.

Bây giờ A và B có thể truyền thông an toàn dùng khóa Ks, kết thúc phiên liên lạc cả A và B hủy Ks.

Cách phân phối này đơn giản, không có thông tin nào tồn tại trước và sau khi truyền thông. Chính vì vậy rủi ro về dàn xếp khóa là nhỏ. Tuy nhiên cách phân phối này dễ dàng bị tấn công xen vào giữa thực hiện thành công.

### 2.5.2 Phân phối khóa bí mật có bí mật và xác thực

Hai thành viên muốn truyền thông với nhau dùng mã khóa bí mật. A muốn B gửi cho A một khóa phiên Ks một cách bí mật và xác thực bằng cách dùng mật mã khóa công khai. Giả thiết A và B đã trao đổi khóa công khai với nhau trước đó.




Hình 3.9 Phân phối khóa bí mật có bí mật và xác thực

Các bước tiến hành:

- ✓ A dùng khóa công khai của B là EKUb lập mã thông báo có chứa thông tin IDa và nonce N1 (giá trị ngẫu nhiên không lặp lại).
- ✓ B gửi cho A bản mã EKUa[N1 || N2], trong đó có giá trị nonce N2 của B.
- ✓ A gửi cho B N2 được mã hóa bằng khóa công khai của B để đảm bảo với B người đáp ứng là A.
- ✓ A chọn một khóa Ks và gửi cho B bản mã Y=EKUb[EKRa[Ks]], trong đó KRa là khóa riêng của A và Ks là khóa bí mật chung của A và B.
- ✓ B tính DKUa[DKRb[Y]] để khôi phục khóa bí mật.

## 2.6. Trao đổi khóa DIFFIE – HELLMAN

Trong các sơ đồ phân phối khóa riêng dùng mật mã khóa công khai ở trên, khóa phiên được tạo ra bởi một bên tham gia truyền thông sau đó được mã bởi khóa công khai và truyền cho bên kia. Điều này có thể dẫn đến lộ khóa bởi bên sinh khóa hoặc trên đường truyền.

Trong thuật toán trao đổi khóa của Diffie – Hellman, hai bên truyền thông cung cấp cho nhau các thông tin bí mật để tạo ra khóa phiên chung, mục đích giúp trao đổi khóa một cách an toàn để mã và giải mã các thông báo.

Dùng giao thức Diffie – Hellman để trao đổi khóa K, giao thức thực hiện như sau:

Giả sử đã chọn được trước một số nguyên tố p và một phần tử nguyên thủy α của $Z_p$ , các bước của giao thức là:

1. A chọn ngẫu nhiên Xa thỏa mãn $0 \leq Xa \leq p-2$, giữ kín Xa, tính $Ya = \alpha^{Xa} \bmod p$ và gửi Ya cho B.
2. B chọn ngẫu nhiên Xb thỏa mãn $0 \leq Xb \leq p-2$, giữ kín Xb, tính $Yb = \alpha^{Xb} \bmod p$ và gửi Yb cho A.
3. Cả A và B đều tính được khóa chung $K = \alpha^{XaXb} \bmod p$, A tính $K = Yb^{Xa} \bmod p$, B tính $K = Ya^{Xb} \bmod p$.

Kẻ tấn công muốn có khóa K phải tính được Xa hoặc Xb, do đó phải đối mặt với bài toán logarit rời rạc trên $Z_p$.

Thuật toán Diffie - Hellman có hai đặc trưng sau:

- Các khóa bí mật chỉ được tạo khi cần thiết, không phải giữ khóa bí mật trong thời gian dài.
- Việc thỏa thuận dựa trên các tham số chung

Tuy nhiên thuật toán Diffie – Hellman có một số điểm yếu sau:

- Nó không cung cấp thông tin bất kỳ về các định danh của các bên.

- Nó an toàn đối với việc tấn công thụ động nghĩa là người thứ ba biết Ya, Yb sẽ không tính được K, tuy nhiên giao thức là không an toàn đối với việc tấn công chủ động bằng cách đánh tráo giữa đường. Trong đó người C mạo danh là B khi truyền thông với A và mạo danh A khi truyền thông với B. Cả A và B đều thỏa thuận với C, sau đó C có thể nghe các thông tin được trao đổi giữa A và B.

## 2.7. Các hệ mật dùng khóa công khai

Hệ mật RSA: độ bảo mật của hệ RSA dựa trên độ khó của việc phân tích ra thừa số nguyên tố các số nguyên lớn.

Hệ mật xếp ba lô Merkle – Hellman: dựa trên tính khó giải của bài toán tổng hợp các tập con (bài toán NP đầy đủ).

Hệ mật McEliece: dựa trên lý thuyết mã đại số - bài toán giải mã cho các mã tuyến tính.

Hệ mật ElGamal: dựa trên tính khó giải của bài toán logarit rời rạc trên các đường hữu hạn.

Hệ mật Chor – Rivest : đây cũng là một loại hệ mật xếp ba lô.

Hệ mật trên các đường cong Elliptic: là biến tướng của các hệ mật nhưng chúng làm việc trên các đường cong Elliptic. Hệ mật này đảm bảo độ mật với khóa số nhỏ hơn các hệ mật khóa công khai khác.

# CHƯƠNG 3: THIẾT KẾ VÀ XÂY DỰNG ỨNG DỤNG TRÊN LINUX

## 3.1. Phát triển ứng dụng trên Linux

### 3.1.1 GNU và các sản phẩm miễn phí

Cộng đồng mã nguồn mở GNU (*GNU's Not UNIX*) đã xây dựng rất nhiều ứng dụng có khả năng chạy trên UNIX và Linux gồm: trình soạn thảo, trò chơi, đồ họa, ứng dụng Internet, trình chủ web, các ngôn ngữ lập trình, trình biên dịch, thông dịch…

GNU cung cấp bộ công cụ biên dịch C/C++ gồm:

| | |
|---|---|
| gcc | Trình biên dịch C |
| g++ | Trình biên dịch C++ |
| gdb | Trình gỡ lỗi |
| GNU make | Trình quản lý mã nguồn và trợ giúp biên dịch |
| GNU Emacs | Trình soạn thảo văn bản (hỗ trợ cho việc chỉnh sửa nguồn khi lập trình) |
| bash | Hệ vỏ Shell hỗ trợ các dòng lệnh của hệ điều hành |
| Bision | Bộ phân tích tương thích với yacc của UNIX |

### 3.1.2 Lập trình trên Linux

C là ngôn ngữ lập trình có vai trò quan trọng trên UNIX và Linux, vì nguyên thủy UNIX được viết từ C và phần lớn các ứng dụng của UNIX cũng dùng C để viết. Tuy nhiên có thể dùng nhiều ngôn ngữ khác như Java, JavaScript, SQL, Pascal, Prolog, Fortran… trong đó C/C++ và pascal có khả năng biên dịch mạnh và gần gũi nhất. Trình biên dịch C và Pascal trên Linux hoàn toàn có khả năng biên dịch cả mã nguồn viết bằng ngôn ngữ máy Assembler. Vì vậy trong luận văn này C là ngôn ngữ được chọn để phát triển ứng dụng.

### 3.1.3 Chương trình UNIX và Linux

Chương trình ứng dụng chạy trên UNIX và Linux tồn tại ở hai dạng: dạng thực thi (file nhị phân) và dạng thông dịch script. File chương trình thực thi ở dạng mã máy nhị phân tương tự như file .exe, file script tương tự như các file .bat của DOS.

Hầu như script và chương trình nhị phân đều có khả năng và sức mạnh ngang nhau. Khó phân biệt được đâu là lệnh gọi chương trình nhị phân và đâu là lệnh gọi chương trình ứng dụng script trên UNIX và Linux (trừ khi xem nội dung của nó). Chúng có thể hoán chuyển cho nhau, một chương trình script có thể chuyển thành chương trình nhị phân bằng ngôn ngữ biên dịch C hay Pascal. Chương trình trong UNIX/Linux chỉ được thực hiện khi bạn có quyền.

## 3.2. Hệ mật khóa công khai RSA (Rivest, Shamir và Adlemam)

***a. Hệ mật RSA:*** sử dụng các tính toán trong $Z_n$, trong đó n là tích của hai số nguyên tố phân biệt p và q $\Rightarrow \phi(n) = (p-1)(q-1)$. Mô tả hình thức của hệ mật như sau:

Cho n=pq trong đó p, q là các số nguyên tố. Đặt P = C = $Z_n$ và định nghĩa: K = {(n, p, q, a, b) | n = pq, p, q là các số nguyên tố, $ab \equiv 1 (\text{mod } \phi(n))$, $0 < b < \phi(n)$ và UCLN(b, $\phi(n)$) = 1}

Với K = (n, p, q, a, b) ta xác định:

$$e_K(x) = x^b \bmod n = y$$

và $$d_K(x) = y^a \bmod n$$

($x,y \in Z_n$), các giá trị n và b được công khai và các giá trị p, q, a được bí mật. Phép mã và phép giải mã là phép toán nghịch đảo của nhau vì:

$ab \equiv 1 (\text{mod } n) \leftrightarrow ab = t\,\phi(n) + 1$ với mọi t nguyên lớn hơn 1.

Giả sử x Є $Z^*_n$ khi đó ta có:

$((x)^b)^a \leftrightarrow x^{t\phi(n)+1} \pmod n$

$\leftrightarrow (x^{\phi(n)})^t x \pmod n$

$\leftrightarrow (1)^t x \pmod n$ (theo hệ quả định lý lagrage)

$\leftrightarrow x \pmod n$

Với x Є $Z_n \backslash Z^*_n$ hoàn toàn tương tự

***b. Độ mật của RSA***

Độ mật của hệ RSA dựa trên hàm mã $e_K(x) = x^b$ mode n là hàm một chiều, thám mã không có khả năng về mặt tính toán để giải mã. Nếu hai số p, q được chọn là lớn cỡ chừng 100 chữ số thập phân, b được chọn sao cho $0<b<\phi(n)$ và UCLN(b, $\phi(n)$) = 1. Chọn b phải thỏa mãn $2^b > n$, nếu không, có thể xảy ra khả năng $x^b < n$ như vậy để tìm x chỉ cần khai căn bậc b thông thường của y là tìm được x.

Nếu ta chọn các số p và q chừng 100 chữ số thập phân thì n khoảng 200 chữ số thập phân. Để phân tích một số nguyên cỡ lớn như thế với những thuật toán nhanh nhất và những máy tính hiện đại nhất cũng phải mất hàng tỷ năm.

Thực tế thấy RSA an toàn nhưng cần chú ý chọn p, q sao cho p-1, q-1 không chỉ toàn các ước nguyên tố nhỏ, ngoài ra UCLN(p-1)(q-1) là số nhỏ, p và q phải có các chữ số trong khai triển thập phân khác nhau không nhiều.

***c. Thực hiện hệ mật RSA***

Để thiết lập hệ thống người nhận B sẽ thực hiện các bước sau:

- ✓ Bước 1: B tạo hai số nguyên tố lớn p và q
- ✓ Bước 2: B tính n=pq và $\phi(n)=(p-1)(q-1)$
- ✓ Bước 3: B tạo một số ngẫu nhiên b: $0<b<\phi(n)$ và UCLN(b, $\phi(n)$)=1
- ✓ Bước 4: B tính $a=b^{-1} \bmod \phi(n)$ dùng thuật toán Euclide mở rộng
- ✓ Bước 5: B công bố n và b trong danh bạ và dùng chúng làm khóa công khai

## 3.3. Mô hình thanh toán bằng tiền điện tử

Ở Việt Nam hiện nay Internet và các phần mềm cũng như dịch vụ trực tuyến hầu như mới bắt đầu nên việc xây dựng và triển khai một hệ thống thanh toán đồng bộ là hoàn toàn thực hiện được. Ngoài ra do phát triển sau nên có thể áp dụng công nghệ và học qua các yếu điểm của các hệ thống thanh toán khác trên thế giới nên hệ thống xây dựng cần kết hợp ưu điểm của các hệ thống khác.

Giải pháp đề xuất ở đây là xây dựng hệ thống thanh toán và phát hành tiền điện tử dạng prepaid card (thẻ trả tiền trước) cho các thanh toán trong nước, bằng chung gian chuyển thẻ này thành thẻ tín dụng cho thanh toán ngoài nước và hệ thống thu tiền cho các chủ hàng dựa trên web base. Hệ thống sẽ triển khai công nghệ bảo mật dựa trên hệ mật khóa công khai RSA.

***a. Tiền điện tử (eCash)***

Hiện nay người dân đã quen thuộc sử dụng các thẻ trả tiền trước như thẻ cước điện thoại di động (vina card, mobi card), thẻ truy cập Internet. Đăc tính chung cho các thẻ này là sử dụng riêng cho từng dịch vụ. Việc xây dựng tiền điện tử không khác nhiều về bản chất các thẻ trên nhưng yêu cầu được sử dụng thay thế cho tiền mặt nghĩa là nó có thể dùng mua hàng tại các quầy hàng trên Internet, thanh toán cho các dịch vụ khác nhau và lý tưởng nhất là phải có khả năng rút ra tiền mặt.

Mô tả: Tiền điện tử giống các thẻ số cào, mỗi thẻ ứng với một ID 20 chữ số nằm dưới lớp bột than chì, kèm theo ngày hết hạn, mã số thẻ, các mệnh giá.

Để sử dụng được thẻ khi mua xong đại lý cần kích hoạt mã thẻ vào hệ thống, sau đó người sử dụng cào và nạp 20 chữ số vào hệ thống. Để đảm bảo an toàn mỗi thẻ được thiết kế tương ứng với một số PIN do người dùng tự tạo ra và quản lý. Hệ thống không biết số PIN này

***b. Chức năng của tiền điện tử***

- ✓ Dễ sử dụng, không cần thông báo thông tin cá nhân, không cần điều kiện ràng buộc, có thể chuyển tiền từ thẻ này sang thẻ khác.
- ✓ Thanh toán trên các website chấp nhận thẻ này.
- ✓ Thuê thẻ tín dụng để mua hàng trên các site khác không chấp nhận tiền điện tử của Việt Nam bằng cách lưu và gửi đơn hàng đến hệ thống.
- ✓ Có thể mua tiền điện tử trực tuyến bằng thẻ tín dụng hoặc bằng chính tiền điện tử.
- ✓ Có thể rút tiền mặt từ tài khoản đã được kích hoạt.

***c. Phát hành tiền điện tử***

Do tổ chức phát hành thẻ thực hiện. Hệ thống trước hết phải có chương trình phát hành thẻ: các thẻ bảo đảm phải là duy nhất ID không trùng nhau trong suốt thời gian lưu hành. Mã ID phải xây dựng trên thuật toán không thể mò ra sau một số lần nhập nhất định (sử dụng hàm Randomize()). Khi mua thẻ để sử dụng được hệ thống khóa thẻ này bằng một phương tiện riêng bảo đảm an toàn. Sau khi đã khóa xong thẻ mới được kích hoạt và lưu số thẻ đã mã hóa cùng giá trị tiền hiện hành của thẻ vào danh mục các chủ thẻ.

## 3.4. Mô tả các yêu cầu đối với hệ thống

- ✓ Bảo mật về tài khoản của người mua.
- ✓ Bảo đảm khi thanh toán trên trang web bán hàng, người mua phải tin tưởng rằng đã trả đúng địa chỉ.
- ✓ Khi thanh toán hệ thống vẫn phải đảm bảo bí mật hoàn toàn các tài khoản của cả hai bên.

Cần có 3 bên tham gia:

- ✓ Tổ chức chuyên phát hành tiền điện tử.
- ✓ Các đơn vị bán hàng trên Internet.
- ✓ Chủ tài khoản tiền điện tử.

### 3.4.1 Đối tượng phục vụ

Khách hàng sử dụng tiền điện tử: là bất kỳ ai làm chủ tiền điện tử mua qua các đại lý hoặc trực tuyến. Hệ thống yêu cầu khách hàng phải đăng ký các thông tin riêng. Để sử dụng tiền điện tử và thực hiện mua bán hàng trên mạng phải có máy tính, kết nối Internet và thông tin về thẻ điện tử. Ngoài ra cần cài đặt online phần mềm đồng bộ với hệ thống thanh toán và chủ hàng.

Các chủ hàng: là chủ nhân của các website cung cấp dịch vụ bán hàng trên mạng. Hệ thống phải cung cấp một công cụ tích hợp dễ dàng với các quầy hàng dạng webbase, một đoạn code html của hệ thống gắn vào site của chủ hàng. Khi khách hàng nhấn thanh toán thì đoạn code này sẽ khởi động một phần mềm thanh toán (đồng bộ với trung tâm và khách hàng sử dụng tiền điện tử), đồng thời các dữ liệu như tổng số thanh toán và thông tin về thẻ điện tử (bí mật ngay cả đối với chủ hàng) sẽ được chuyển về trung tâm sau khi chủ hàng xác nhận về số tiền. Yêu cầu đối với hệ thống thanh toán là các chủ hàng cũng dễ dàng cài đặt trực tuyến. Tiền của khách hàng sẽ được chuyển đến tài khoản của chủ hàng cũng nằm trong hệ thống trung tâm và sẽ thanh toán bằng tiền mặt hoặc chuyển khoản theo cam kết giữa hai doanh nghiệp

Các đại lý phát hành thẻ: là những địa điểm phân phối thẻ, nơi khách hàng có thể tìm mua. Các đại lý phải kết nối với hệ thống thanh toán. Chức năng của họ là quản lý các thẻ phát hành, nhập vào hệ thống thẻ nào đã bán được và có thể là nơi thanh toán rút tiền mặt từ các tài khoản. Các giao diện với khối đại lý là riêng biệt. Hệ thống cần có phần mềm để quản lý thu chi của các đại lý cũng như điều phối việc phát hành thẻ.

### 3.4.2 Chức năng và thành phần của hệ thống

Phần mềm:

- ✓ Hệ thống phải có chương trình phát hành thẻ.
- ✓ Phần mềm quản lý thẻ và phát hành.
- ✓ Phần mềm quản lý các tài khoản lưu hành.
- ✓ Phần mềm quản lý các chủ hàng.
- ✓ Phần mềm quản lý các đại lý.
- ✓ Phần mềm quản lý việc rút tiền mặt.
- ✓ Phần mềm quản lý thanh toán ngoài nước thuê thẻ tin dụng trung gian.
- ✓ Phần mềm quản lý việc trả tiền bằng thẻ tín dụng cho các chủ hàng.

Cơ sở dữ liệu: CSDL thẻ, CSDL tài khoản, CSDL chủ hàng, CSDL đại lý.

Thiết bị: hệ thống phải có máy chủ với kết nối Internet đủ mạnh để bảo đảm phục vụ khách hàng 24/24.

Triển khai: hệ thống có thành công hay không phụ thuộc rất lớn vào khâu tổ này. Có thể tận dụng hệ thống phân phối các thẻ Mobicard và Vinacard hiện nay. Ngoài ra việc thuyết phục các chủ hàng chấp nhận phương thức thanh toán qua hệ thống cũng cần triển khai rộng rãi. Có thể kết hợp với các công ty đang cung cấp dịch vụ Internet hoặc thiết kế website hiện nay.

## 3.5. Mô hình ứng dụng RSA trong thanh toán

Cơ chế bảo mật của hệ thống ứng dụng hệ mật khóa công khai RSA trong các chức năng mã hóa, giải mã và xác minh chủ hàng, chủ thẻ trước khi thực hiện các nghiệp vụ tài chính như kiểm tra số dư tài khoản và chuyển tiền giữa hai tài khoản khác nhau.

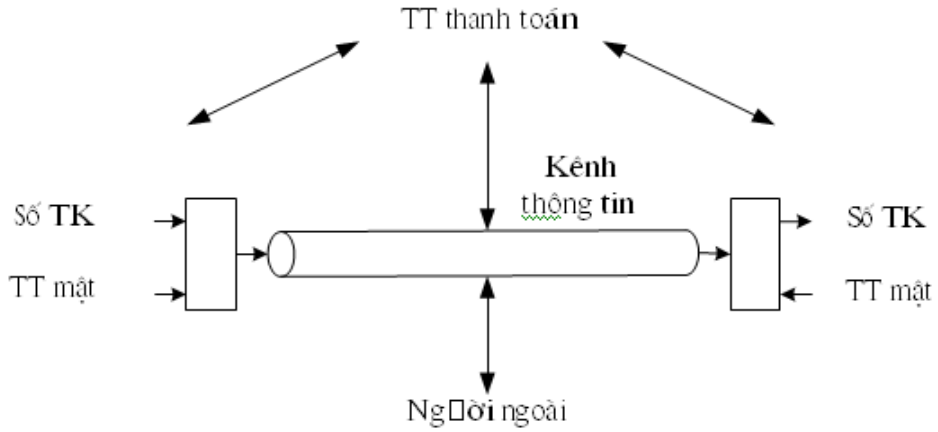
- ✓ Kiểm tra số dư tài khoản: để truy cập vào tài khoản của A, phần mềm sẽ mã hóa ID tài khoản bằng khóa riêng của A, sau đó mã hóa bằng khóa công khai của trung tâm. Trung tâm giải mã bằng khóa riêng của

mình, sau đó giải mã bằng khóa công khai của A. Điều này xác minh trước khi cho phép A mở xem tình trạng tài khoản của mình.

- ✓ Thanh toán giữa chủ thẻ A và chủ hàng B: phần mềm được kích hoạt sẽ mã hóa tài khoản và số tiền cần chuyển bằng khóa riêng được bản mã M1. Để bảo mật tài khoản với người giao dịch, A mã hóa M1 bằng khóa công khai của trung tâm được bản mã M2, A mã hóa M2 bằng khóa công khai của B được bản mã M3, gửi M3 cho B băng đường truyền công cộng.

B giải mã M3 bằng khóa riêng được M2, đính kèm $ID_B$, số tiền thụ hưởng và thực hiện mã hóa bằng khóa riêng của B được bản mã M4. Để bảo mật B mã hóa M4 bằng khóa công khai của trung tâm được bản mã M5 và gửi M5 cho trung tâm trên đường truyền công cộng.

Trung tâm giải mã M5 bằng khóa riêng của trung tâm được M4, giải mã M4 bằng khóa công khai của B được M2 và thông tin B đính kèm, xác nhận người thụ hưởng là B. Trung tâm giải mã M2 bằng khóa riêng của trung tâm được M1, giải mã M1 bằng khóa công khai của A được nội dung thông tin cần chuyển của A và xác nhận người chuyển tiền là A.




Hình 4.1 Mô hình thanh toán trên mạng

Tại trung tâm, sau khi giải mã, xác thực được người thụ hưởng là B và tài khoản của B, xác thực được người chuyển tiền là A và tài khoản của

A, cùng số tiền T mà B được thụ hưởng từ A. Nếu thỏa mãn một số điều kiện ràng buộc như giá trị tiền gửi và nhận là như nhau, số tiền còn lại trong tài khoản A vượt quá mức cần chuyển … thì trung tâm sẽ thực hiện lệnh thanh toán bằng cách trừ vào tài khoản của A một khoản T và cộng vào tài khoản B số tiền tương ứng.

## 3.6. Phạm vi ứng dụng

Khả năng ứng dụng thực tế:

- ✓ Dễ sử dụng
- ✓ Các khái niệm dễ hiểu
- ✓ Không yêu cầu khách hàng hay chủ hàng khai báo trước khi sử dụng
- ✓ An toàn cao cho khách hàng cũng như chủ hàng
- ✓ Chi phí thấp, cho phép thanh toán với cả những giao dịch nhỏ, có thể ẩn danh, cài đặt trực tuyến và rất thuận tiên đơn giản với người mua hàng, có khả năng phát triển độc lập không bị quá phụ thuộc vào các đối tác cung cấp hạ tầng.

## 3.7. Chương trình ứng dụng

Trong phạm vi luận văn này, chương trình ứng dụng được xây dựng với mục đích thể hiện việc mã hóa số thẻ của tiền điện tử và truyền an toàn trên hạ tầng Internet thông thường và thực hiên các thao tác giao dịch thông dụng nhất của tài khoản đó là chuyển khoản.

Mô tả: trên giao diện web-based, chủ thẻ tiền điện tử có tài khoản $ID_A$ chuyển một số tiền x đến một chủ thẻ có tài khoản $ID_B$. Hệ thống thanh toán trung tâm phải hoàn thành việc thay đổi tài khoản giữa hai chủ thẻ và bảo đảm dữ liệu truyền trên kênh công cộng là bí mật đối với người ngoài.

Tiến trình thực hiện như sau:

1. A mã hóa tài khoản tiền điện tử của mình $ID_A$ và số tiền cần chuyển bằng khóa riêng được bản mã M1.
2. Để bảo mật tài khoản đối với người giao dịch, A mã hóa M1 bằng khóa công khai của trung tâm được bản mã M2.
3. A mã hóa M2 bằng khóa công khai của B được bản mã M3. Gửi M3 cho B bằng đường truyền công cộng.
4. B giải mã M3 bằng khóa riêng được M2, đính kèm $ID_B$ và số tiền thụ hưởng và thực hiện mã hóa bằng khóa riêng của B được bản mã M4.
5. Để bảo mật, B mã hóa M4 bằng khóa công khai của trung tâm được bản mã M5 và gửi M5 đến trung tâm bằng đường truyền công cộng
6. Trung tâm giải mã M5 bằng khóa riêng của trung tâm được M4, giải mã M4 bằng khóa công khai của B được M2 và thông tin B đính kèm, xác nhận người thụ hưởng là B.
7. Trung tâm giải mã M2 bằng khóa riêng của trung tâm được M1, giải mã M1 bằng khóa công khai của A được nội dung thông tin cần chuyển của A và xác nhận người chuyển tiền là A.
8. Trung tâm thực hiện giao dịch theo lệnh của A tới tài khoản của B vừa được xác minh ở bước trước

Để thực hiện được quá trình này, cả 3 đối tượng: A, B, TT đều được cài đặt phần mềm mã hóa và giải mã RSA.

# KẾT LUẬN

Trên đây là toàn bộ báo cáo đồ án tốt nghiệp. Trong luận văn này em đã tìm hiểu về hệ điều hành mã nguồn mở Linux, lý thuyết mã khóa công khai và xây dựng một ứng dụng mã khóa công khai dùng hệ mật RSA trong môi trường mã nguồn mở Linux, thực hiện thiết lập hệ mật, mã hóa, giải mã để bảo mật, xác thực trong mô hình thanh toán bằng tiền điện tử.

Qua luận văn này em thấy thanh toán bằng tiền điện tử qua mạng Internet là một xu thế tất yếu, nó cần được phát triển hoàn thiện và được ứng dụng trong thực tế ở nước ta, để phát triển nền kinh tế và hội nhập với các nước trên thế giới.

Do thời gian có hạn và trình độ bản thân còn hạn chế nên em rất mong được sự góp ý, giúp đỡ và sự chỉ bảo tận tình của các thầy cô giáo cùng toàn thể các bạn để em có thể hoàn thiện chương trình tốt hơn nữa.

Cuối cùng em xin chân thành cảm ơn các thầy cô giáo. Đặc biệt em xin tỏ lòng biết ơn tới thầy giáo ThS Võ Văn Tùng, trong thời gian qua thầy đã giành nhiều thời gian và tâm huyết để hướng dẫn em hoàn thành đề tài này.

# TÀI LIỆU THAM KHẢO


1. Nguyễn Thúc Hải (1999), *Mạng máy tính và các hệ thống mở,* Nhà xuất bản giáo dục, Hà Nội.
2. Nguyễn Thanh Thủy, Nguyễn Quang Huy, Nguyễn Hữu Đức, Đinh Lan Anh (2000), *Nhập môn hệ điều hành Linux,* Nhà xuất bản khoa học và kỹ thuật, Hà Nội.
3. VN-GUIDE (2000), *Linux toàn tập,* Nhà xuất bản thống kê, TP HCM.
4. Phạm Huy Điển, Hà Huy Khoái (2004), *Mã hóa thông tin cơ sở toán học & ứng dụng,* Nhà xuất bản Đại học quốc gia HN, Hà Nội.
5. William Stalling (1999), *Cryptography and Network Security*, Prentic Hall.
6. Website http://quantrimang.com