

## CHƯƠNG 1: TỔNG QUAN VỀ AN TOÀN THÔNG TIN

### 1.1. Các khái niệm cơ bản

Dữ liệu (Data) là các giá trị của thông tin định lượng hoặc định tính của các sự vật, hiện tượng trong cuộc sống. Trong tin học, dữ liệu được dùng như một cách biểu diễn hình thức hoá của thông tin về các sự kiện, hiện tượng thích ứng với các yêu cầu truyền nhận, thể hiện và xử lý bằng máy tính.

Thông tin (Information) là dữ liệu đã được xử lý, phân tích, tổ chức nhằm mục đích hiểu rõ hơn sự vật, sự việc, hiện tượng theo một góc độ nhất định.

Hệ thống thông tin (Information System) là một hệ thống gồm con người, dữ liệu và những hoạt động xử lý dữ liệu và thông tin trong một tổ chức.

#### **Tài sản của hệ thống bao gồm:**

- Phần cứng
- Phần mềm
- Dữ liệu
- Các truyền thông giữa các máy tính của hệ thống
- Môi trường làm việc
- Con người

#### **Bảo mật hệ thống thông tin (Information System Security)**

- Bao hàm một lĩnh vực rộng lớn các hoạt động trong một tổ chức.
- Nó bao gồm cả những sản phẩm và những quy trình nhằm ngăn chặn truy cập trái phép, hiệu chỉnh, xóa, phá hủy, làm lộ và làm gián đoạn thông tin và hoạt động của hệ thống một cách trái phép.

An toàn thông tin (ATTT) là an toàn kỹ thuật cho các hoạt động của các cơ sở hạ tầng thông tin (HTTT), trong đó bao gồm an toàn phần cứng và phần mềm theo các tiêu chuẩn kỹ thuật do nhà nước ban hành; duy trì các tính chất bí mật, toàn vẹn, sẵn sàng của thông tin trong lưu trữ, xử lý và truyền dẫn trên mạng (theo định nghĩa trong Nghị định 64-2007/NĐ-CP).

Mục tiêu hướng tới của ATTT là bảo vệ các tài sản thông tin. Tuy nhiên, các sản phẩm và hệ thống thường luôn tồn tại những điểm yếu dẫn đến những rủi ro có thể xảy ra. Các đối tượng tấn công (tín tặc) có chủ tâm đánh cắp, lợi dụng hoặc phá hoại tài sản của các chủ sở hữu, tìm cách khai thác các điểm yếu để tấn công, tạo ra các nguy cơ và các rủi ro cho các hệ thống thông tin.

Đảm bảo ATTT là đảm bảo an toàn kỹ thuật cho hoạt động của các cơ sở HTTT, trong đó bao gồm đảm bảo an toàn cho cả phần cứng và phần mềm hoạt động theo các tiêu chuẩn kỹ thuật do nhà nước ban hành; ngăn ngừa khả năng lợi dụng mạng và các

cơ sở HTTT để thực hiện các hành vi trái phép; đảm bảo các tính chất bí mật, toàn vẹn, sẵn sàng của thông tin trong lưu trữ, xử lý và truyền dẫn trên mạng.

Với các biện pháp đảm bảo ATTT người dùng có được công cụ trong tay để nhận thức được các điểm yếu, giảm thiểu các điểm yếu, ngăn chặn các nguy cơ tấn công, làm giảm các yếu tố rủi ro.

Như vậy, các biện pháp và kỹ thuật đảm bảo ATTT chính là mang lại sự tin cậy cho các sản phẩm và hệ thống thông tin.

## **1.2.Các nguyên tắc nền tảng của an toàn thông tin**

ATTT nhằm đảm bảo 4 đặc điểm quan trọng nhất của thông tin (hình 1), đó là:

- Tính bí mật;
- Tính toàn vẹn;
- Tính sẵn sàng.
- Tính không chối bỏ

Bốn nguyên tắc này là tiêu chuẩn cho tất cả các hệ thống an ninh. Tùy thuộc vào ứng dụng và hoàn cảnh cụ thể mà nguyên tắc nào được xem xét quan trọng hơn.

### **1.2.1.Tính bí mật (Confidentiality)**

Tính bí mật là bảo vệ dữ liệu không bị lộ ra ngoài một cách trái phép.

Một giải pháp đảm bảo an toàn là xác định quyền được truy cập đối với thông tin đang tìm kiếm, đối với một số lượng người sử dụng nhất định và một số lượng thông tin là tài sản nhất định. Trong trường hợp kiểm soát truy cập, nhóm người truy cập sẽ được kiểm soát xem họ đã truy cập những dữ liệu nào. Tính bí mật là sự đảm bảo rằng các chức năng kiểm soát truy cập có hiệu lực.

Đối với an ninh mạng thì tính bí mật rõ ràng là điều đầu tiên được nói đến và nó thường xuyên bị tấn công nhất

Ví dụ: Trong hệ thống quản lý sinh viên, một sinh viên được phép xem thông tin kết quả học tập của mình nhưng không được phép xem thông tin của sinh viên khác.

### **1.2.2.Tính toàn vẹn (Integrity)**

Tính toàn vẹn chỉ những người dùng được ủy quyền mới được phép chỉnh sửa dữ liệu.

Có ba mục đích chính của việc đảm bảo tính toàn vẹn:

- Ngăn cản sự làm biến dạng nội dung thông tin của những người sử dụng không được phép.
- Ngăn cản sự làm biến dạng nội dung thông tin không được phép hoặc không chủ tâm của những người sử dụng được phép.

- Duy trì sự toàn vẹn dữ liệu cả trong nội bộ và bên ngoài.

Ví dụ: Trong hệ thống quản lý sinh viên, không cho phép sinh viên được phép tự thay đổi thông tin kết quả học tập của mình.

### **1.2.3. Tính sẵn sàng (Availability)**

Tính sẵn sàng đảm bảo dữ liệu luôn sẵn sàng khi những người dùng hoặc ứng dụng được ủy quyền yêu cầu.

Tính sẵn sàng của thông tin cũng là một đặc tính rất quan trọng. Tính sẵn sàng bảo đảm các người sử dụng hợp pháp của hệ thống có khả năng truy cập đúng lúc và không bị ngắt quãng tới các thông tin trong hệ thống và tới mạng.

Tính sẵn sàng đảm bảo độ ổn định đáng tin cậy của thông tin, cũng như đảm nhiệm chức năng là thước đo, xác định phạm vi tới hạn an toàn của một hệ thống thông tin.

Ví dụ: Trong hệ thống quản lý sinh viên, cần đảm bảo rằng sinh viên có thể truy vấn thông tin kết quả học tập của mình bất cứ lúc nào.

### **1.2.4. Tính chống thoái thác (Non-repudiation)**

Tính chống thoái thác là khả năng ngăn chặn việc từ chối một hành vi đã thực hiện.

Ví dụ: Trong hệ thống quản lý sinh viên, có khả năng cung cấp bằng chứng để chứng minh một hành vi sinh viên đã làm, như đăng ký học phần, hủy học phần.

## **1.3. Các loại hình tấn công và nguy cơ mất ATTT**

### **1.3.1. Các khái niệm tấn công**

Hiện nay khái niệm "tấn công" (xâm nhập, công kích) vẫn đang được hiểu ở nhiều ý nghĩa khác nhau. Mỗi chuyên gia trong lĩnh vực ATTT luận giải thuật ngữ này theo ý hiểu của mình. Ví dụ, "xâm nhập - là tác động bất kỳ đưa hệ thống từ trạng thái an toàn vào tình trạng nguy hiểm", "là sự phá huỷ chính sách ATTT" hoặc "là tác động bất kỳ dẫn đến việc phá huỷ tính toàn vẹn, tính bí mật, tính sẵn sàng của hệ thống và thông tin xử lý trong hệ thống".

Tấn công (attack) là hoạt động có chủ ý của kẻ phạm tội lợi dụng các thương tổn của hệ thống thông tin và tiến hành phá vỡ tính sẵn sàng, tính toàn vẹn và tính bí mật của hệ thống thông tin.

Tấn công HTTT là các tác động hoặc là trình tự liên kết giữa các tác động với nhau để phá huỷ, dẫn đến việc hiện thực hoá các nguy cơ bằng cách lợi dụng đặc tính dễ bị tổn thương của các hệ thống thông tin này.

Phân loại một số hình thức tấn công:

- **Tấn công chủ động (active attack):** tấn công ngăn chặn thông tin, tấn công chặn bắt thông tin, tấn công sửa đổi thông tin, chèn thông tin giả mạo.

- **Tấn công thụ động (passive attack):** tấn công chặn bắt thông tin (nghe lén, khai thác nội dung thông điệp, phân tích dòng dữ liệu).

### **Tấn công ngăn chặn thông tin (interruption)**

Tài nguyên thông tin bị phá hủy, không sẵn sàng phục vụ hoặc không sử dụng được. Đây là hình thức tấn công làm mất khả năng sẵn sàng phục vụ của thông tin.

### **Tấn công chặn bắt thông tin (interception)**

Kẻ tấn công có thể truy nhập tới tài nguyên thông tin. Đây là hình thức tấn công vào tính bí mật của thông tin.

### **Tấn công sửa đổi thông tin (Modification)**

Kẻ tấn công truy nhập, chỉnh sửa thông tin. Đây là hình thức tấn công vào tính toàn vẹn của thông tin.

### **Chèn thông tin giả mạo (Fabrication)**

Kẻ tấn công chèn các thông tin và dữ liệu giả vào hệ thống. Đây là hình thức tấn công vào tính xác thực của thông tin.

### **Tấn công bị động**

Mục đích của kẻ tấn công là thu thập được thông tin truyền trên mạng. Có hai kiểu tấn công bị động là khai thác nội dung thông điệp và phân tích dòng dữ liệu.

Tấn công bị động rất khó bị phát hiện vì nó không làm thay đổi dữ liệu và không để lại dấu vết rõ ràng. Biện pháp hữu hiệu để chống lại kiểu tấn công này là ngăn chặn (đối với kiểu tấn công này, ngăn chặn tốt hơn là phát hiện).

### **Tấn công chủ động (active attacks)**

Tấn công chủ động được chia thành 4 loại sau:

- Giả mạo (Masquerade): Một thực thể (người dùng, máy tính, chương trình...) đóng giả thực thể khác.
- Dừng lại (replay): Chặn bắt các thông điệp và sau đó truyền lại nó nhằm đạt được mục đích bất hợp pháp.
- Sửa thông điệp (Modification of messages): Thông điệp bị sửa đổi hoặc bị làm trễ và thay đổi trật tự để đạt được mục đích bất hợp pháp.
- Từ chối dịch vụ (Denial of Service - DoS): Ngăn cấm việc sử dụng bình thường hoặc làm cho truyền thông ngừng hoạt động.

## **1.3.2. Một số kỹ thuật tấn công mạng**

### **1.3.2.1. Tấn công thăm dò**

Thăm dò là việc thu thập thông tin trái phép về tài nguyên, các lỗ hổng hoặc dịch vụ của hệ thống.

Tấn công thăm dò thường bao gồm các hình thức:

- Sniffing
- Ping Sweep
- Port Scanning

#### **1.3.2.2. Tấn công từ chối dịch vụ (Denial of Service)**

Về cơ bản, tấn công từ chối dịch vụ là tên gọi chung của kiểu tấn công làm cho một hệ thống nào đó bị quá tải không thể cung cấp dịch vụ, gây ra gián đoạn hoạt động hoặc làm cho hệ thống ngừng hoạt động.

Tùy theo phương thức thực hiện mà nó được biết dưới nhiều tên gọi khác nhau.

Khởi thủy là lợi dụng sự yếu kém của giao thức TCP (Transmission Control Protocol) để thực hiện tấn công từ chối dịch vụ DoS (Denial of Service), mới hơn là tấn công từ chối dịch vụ phân tán DDoS (Distributed DoS), mới nhất là tấn công từ chối dịch vụ theo phương pháp phản xạ DRDoS (Distributed Reflection DoS).

#### **1.3.2.3. Tấn công sử dụng mã độc (malicious code)**

Khái niệm: Mã độc là những chương trình khi được khởi chạy có khả năng phá hủy hệ thống, bao gồm Virus, sâu (Worm) và Trojan,...

Tấn công bằng mã độc có thể làm cho hệ thống hoặc các thành phần của hệ thống hoạt động sai lệch hoặc có thể bị phá hủy.

#### **1.3.2.4. Tấn công xâm nhập (Intrusion attack)**

Là hình thức tấn công, nhằm truy nhập bất hợp pháp vào các HTTT.

Kiểu tấn công này được thực hiện với mục đích đánh cắp dữ liệu hoặc thực hiện phá hủy bên trong HTTT.

#### **1.3.2.5. Tấn công sử dụng kỹ nghệ xã hội (Social engineering)**

Là một nhóm các phương pháp được sử dụng để đánh lừa người sử dụng tiết lộ các thông tin bí mật.

Là phương pháp tấn công phi kỹ thuật, dựa trên sự thiếu hiểu biết của người dùng để lừa gạt họ cung cấp các thông tin nhạy cảm như password hay các thông tin quan trọng khác.

### **1.3.3. Xu hướng tấn công HTTT**

#### ***Sử dụng các công cụ tấn công tự động***

Những kẻ tấn công sẽ sử dụng các công cụ tấn công tự động có khả năng thu thập thông tin từ hàng nghìn địa chỉ trên Internet một cách nhanh chóng, dễ dàng và hoàn toàn tự động.

Các HTTT có thể bị quét từ một địa điểm từ xa để phát hiện ra những địa chỉ có mức độ bảo mật thấp. Thông tin này có thể được lưu trữ, chia sẻ hoặc sử dụng với mục đích bất hợp pháp.

### ***Sử dụng các công cụ tấn công khó phát hiện***

Một số cuộc tấn công được dựa trên các mẫu tấn công mới, không bị phát hiện bởi các chương trình bảo mật, các công cụ này có thể có tính năng đa hình, siêu đa hình cho phép chúng thay đổi hình dạng sau mỗi lần sử dụng.

### ***Phát hiện nhanh các lỗ hổng bảo mật***

Thông qua các lỗ hổng bảo mật của hệ thống, phần mềm kẻ tấn công khai thác các lỗ hổng này để thực hiện các cuộc tấn công.

Hàng năm, nhiều lỗ hổng bảo mật được phát hiện và công bố, tuy nhiên điều này cũng gây khó khăn cho các nhà quản trị hệ thống để luôn cập nhật kịp thời các bản vá. Đây cũng chính là điểm yếu mà kẻ tấn công tận dụng để thực hiện các hành vi tấn công, xâm nhập bất hợp pháp.

### ***Tấn công bất đối xứng và tấn công diện rộng***

Tấn công bất đối xứng xảy ra khi bên tấn công mạnh hơn nhiều so với đối tượng bị tấn công.

Tấn công diện rộng thực hiện khi kẻ tấn công tạo ra một mạng lưới kết hợp các hoạt động tấn công.

### ***Thay đổi mục đích tấn công***

Thời gian trước, các tấn công chỉ từ mục đích thử nghiệm, hoặc khám phá hệ thống an ninh. Hiện nay, mục đích tấn công với nhiều lý do khác nhau như về tài chính, giả mạo thông tin, phá hủy, và đặc biệt nguy hiểm đó là mục đích chính trị, chính vì vậy mà độ phức tạp của các cuộc tấn công đã tăng lên và tác hại lớn hơn rất nhiều so với trước đây, cụ thể:

- 1985-1994: virus máy tính, phát tán thông qua các ổ mềm (floppy disk)
- 1995-1999: internet viruses, worms phát tán qua email, môi trường mạng

Trong các giai đoạn trên, động cơ phá hoại chủ yếu mang tính chất của cá nhân.

- 2000-2006: Worms, Spyware, Bots, Phishing ... tấn công thông qua email, môi trường mạng, website
- 2007-nay: Social networking, tấn công ứng dụng, gián điệp ...

Các giai đoạn tấn công hiện nay đều có mục đích rõ ràng và thường hướng vào mục tiêu tài chính.

### 1.3.4. Các nguy cơ mất ATTT

**Cơ sở hạ tầng mạng:** cơ sở hạ tầng không đồng bộ, không đảm bảo yêu cầu thông tin được truyền trong hệ thống an toàn và thông suốt.

**Thông tin:** dữ liệu chưa được mô hình hóa và chuẩn hóa theo tiêu chuẩn về mặt tổ chức và mặt kỹ thuật. Yếu tố pháp lý chưa được chú trọng trong truyền đưa các dữ liệu trên mạng, nghĩa là các dữ liệu được truyền đi trên mạng phải đảm bảo tính hợp pháp về mặt tổ chức và mặt kỹ thuật.

**Công nghệ:** chưa chuẩn hóa cho các loại công nghệ, mô hình kiến trúc tham chiếu nhằm đảm bảo cho tính tương hợp, tính sử dụng lại được, tính mở, an ninh, mở rộng theo phạm vi, tính riêng tư vào trong HTTT.

**Con người:** sự hiểu biết của những người trực tiếp quản lý, vận hành các HTTT, xây dựng và phát triển hệ thống phần mềm, hệ thống thông tin còn chưa đồng đều và chưa theo quy chuẩn của các cơ quan tổ chức đó.

#### **Quy trình, quản lý:**

Chưa chuẩn hóa qui trình nghiệp vụ trong vận hành HTTT.

Chưa chuẩn hóa các thủ tục hành chính, các qui định pháp lý trong việc đảm bảo ATTT.

Tổ chức quản lý thay đổi hệ thống, ứng dụng chưa đúng cách, chưa chuẩn hóa và có chế tài mang tính bắt buộc thực hiện.

Như vậy để đảm bảo ATTT thì các cơ quan tổ chức phải làm tốt và hạn chế các yếu tố trên.

### 1.4. Giải pháp đảm bảo an toàn thông tin

**Các biện pháp an toàn HTTT được phân loại thành 3 lớp như sau:**

- Các biện pháp công nghệ (Technology): bao hàm tất cả các biện pháp phần cứng, các phần mềm, phần lõi cũng như các kỹ thuật công nghệ liên quan được áp dụng nhằm đảm các yêu cầu an toàn của thông tin trong các trạng thái của nó.
- Các biện pháp về đào tạo, tập huấn, nâng cao nhận thức (Education, training & Awareness): Các biện pháp công nghệ hay các biện pháp về tổ chức thích hợp phải dựa trên các biện pháp đào tạo, tập huấn và tăng cường nhận thức để có thể triển khai đảm bảo an toàn thông tin từ nhiều hướng khác nhau. Các nhà nghiên cứu và các kỹ sư cũng cần phải hiểu rõ các nguyên lý an toàn hệ thống thông tin, thì sản phẩm và hệ thống do họ làm ra đáp ứng được các nhu cầu về độ an toàn của cuộc sống hiện tại đặt ra.
- Biện pháp hợp tác quốc tế: hợp tác với các quốc gia có kinh nghiệm, kế thừa những thành tựu khoa học của các quốc gia đi trước trong vấn đề đảm bảo ATTT.



---

Xây dựng các quy chế phối hợp với các cơ quan tổ chức quốc tế trong ứng phó các sự cố về ATTT.

### **Các bước cơ bản trong bảo mật thông tin**

- **Xác định các mối đe dọa (threat)**

Các mối đe dọa bảo mật (security threat) là những sự kiện có ảnh hưởng đến an toàn của hệ thống thông tin.

Các mối đe dọa được chia làm 4 loại:

- Xem thông tin một cách bất hợp pháp
- Chỉnh sửa thông tin một cách bất hợp pháp
- Từ chối dịch vụ
- Từ chối hành vi

- **Lựa chọn chính sách bảo mật (security policy)**

Việc bảo mật hệ thống cần có một chính sách bảo mật rõ ràng.

Cần có những chính sách bảo mật riêng cho những yêu cầu bảo mật khác nhau

Xây dựng và lựa chọn các chính sách bảo mật cho hệ thống phải dựa theo các chính sách bảo mật do các tổ chức uy tín về bảo mật định ra (compliance)

NIST, SP800, ISO17799, HIPAA

Chính sách bảo mật phải cân bằng giữa 3 yếu tố: khả dụng, bảo mật và hiệu suất.

- **Lựa chọn cơ chế bảo mật (security mechanism)**

Xác định cơ chế bảo mật phù hợp để hiện thực các chính sách bảo mật và đạt được các mục tiêu bảo mật đề ra

Có 4 cơ chế bảo mật:

- Điều khiển truy cập (Access control)
- Điều khiển suy luận (Inference control)
- Điều khiển dòng thông tin (Flow control)
- Mã hóa (Encryption)

## **1.5.Các bài toán an toàn thông tin cơ bản**

### **1.5.1.Ví dụ:**

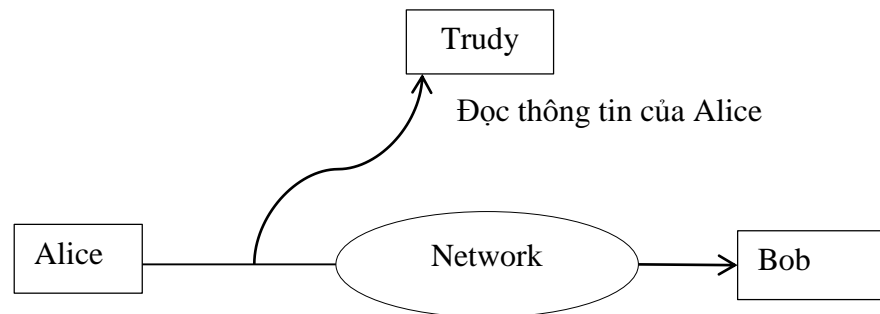
Alice và Bob thực hiện trao đổi thông tin với nhau, còn Trudy là kẻ xấu, đặt thiết bị can thiệp vào kênh truyền tin giữa Alice và Bob. Một số hành động tấn công của Trudy mà ảnh hưởng đến quá trình truyền tin giữa Alice và Bob:



- Xem trộm thông tin (Release of Message Content)
- Thay đổi thông điệp (Modification of Message)
- Mạo danh (Masquerade)
- Phát lại thông điệp (Replay)

- **Xem trộm thông tin (Release of Message Content)**

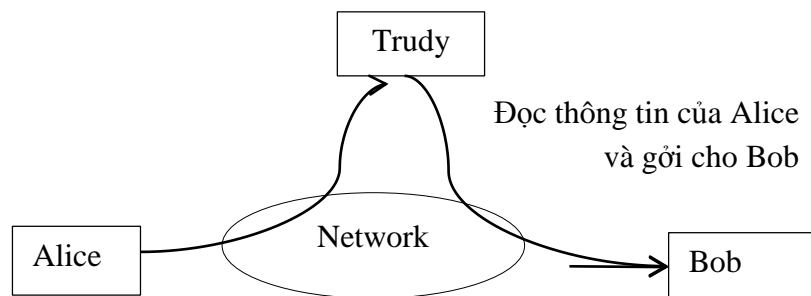
Trong trường hợp này Trudy chặn các thông điệp Alice gửi cho Bob, và xem được nội dung của thông điệp.



Hình 1.1. Xem trộm thông tin

- **Thay đổi thông điệp (Modification of Message)**

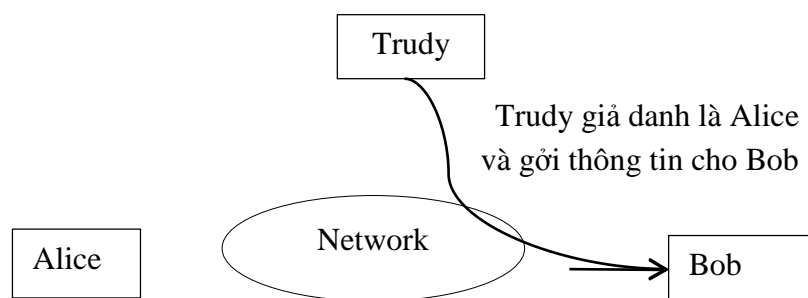
Trudy chặn các thông điệp Alice gửi cho Bob và ngăn không cho các thông điệp này đến đích. Sau đó Trudy thay đổi nội dung của thông điệp và gửi tiếp cho Bob. Bob nghĩ rằng nhận được thông điệp nguyên bản ban đầu của Alice mà không biết rằng chúng đã bị sửa đổi.



Hình 1.2. *Chỉnh sửa thông tin*

- **Mạo danh (Masquerade)**

Trong trường hợp này Trudy giả là Alice gửi thông điệp cho Bob. Bob không biết điều này và nghĩ rằng thông điệp là của Alice.



Hình 1.3. Mạo danh

- **Phát lại thông điệp (Replay)**

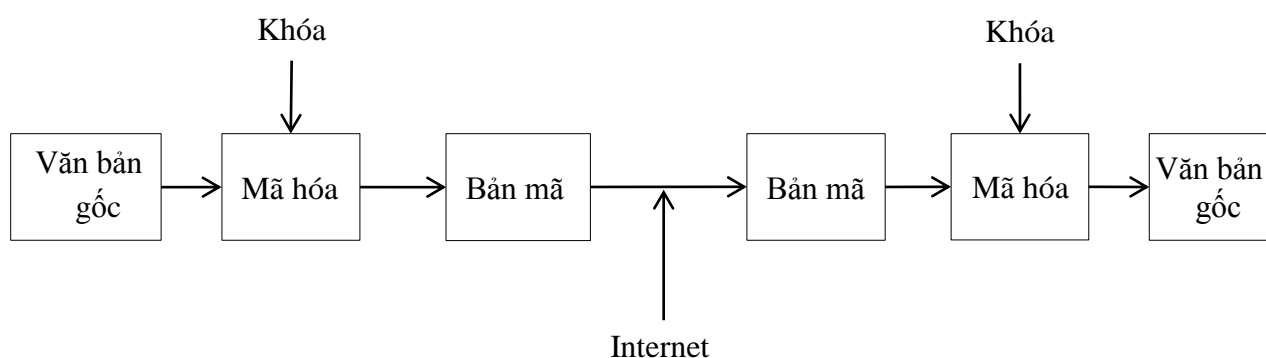
Trudy sao chép lại thông điệp Alice gửi cho Bob. Sau đó một thời gian Trudy gửi bản sao chép này cho Bob. Bob tin rằng thông điệp thứ hai vẫn là từ Alice, nội dung hai thông điệp là giống nhau.

Thoạt đầu có thể nghĩ rằng việc phát lại này là vô hại, tuy nhiên trong nhiều trường hợp cũng gây ra tác hại không kém so với việc giả mạo thông điệp.

Xét tình huống sau: giả sử Bob là ngân hàng còn Alice là một khách hàng. Alice gửi thông điệp đề nghị Bob chuyển cho Trudy 1000\$. Alice có áp dụng các biện pháp như chữ ký điện tử với mục đích không cho Trudy mạo danh cũng như sửa thông điệp. Tuy nhiên nếu Trudy sao chép và phát lại thông điệp thì các biện pháp bảo vệ này không có ý nghĩa. Bob tin rằng Alice gửi tiếp một thông điệp mới để chuyển thêm cho Trudy 1000\$ nữa.

#### 1.5.2.1. Bài toán bảo mật: mã hóa và phong bì số

**Mã hóa:** là quá trình biến đổi thông tin ở văn bản gốc sang dạng bản mã (không đọc được, không hiểu được). Quá trình này thông thường có sự tham gia của khóa. Khóa bên gửi và bên nhận có thể khác nhau hoặc giống nhau tùy vào mục đích sử dụng.



Hình 1.4. Sơ đồ mã hóa cơ bản

**1.5.2.2.Chữ ký số (Digital signature):**

Là thông điệp (có thể là văn bản, hình ảnh, hoặc video...) đi kèm với văn bản gốc nhằm xác minh văn bản đó, chữ ký số được tạo ra dựa trên nền tảng mã hóa khóa công khai (xem chương 2). Quy trình tạo và xác minh chữ ký số được trình bày ở chương 6.

**Mục đích của chữ ký số:**

Xác thực: xác định ai là chủ của thông điệp

Tính toàn vẹn: kiểm tra xem thông điệp có bị thay đổi

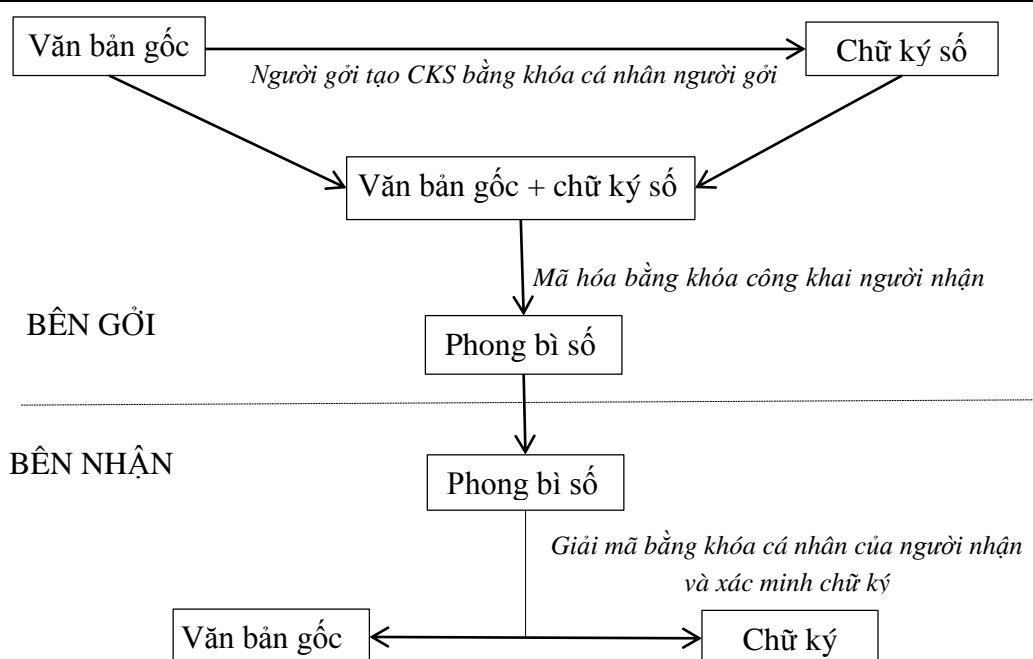
Tính chống thoái thác: ngăn chặn việc người dùng từ chối đã tạo ra và gửi thông điệp

**Quá trình tạo chữ ký số**

- Alice viết một văn bản và muốn gửi cho Bob
- Alice **tạo chữ ký** bằng **khóa cá nhân** của Alice → Chữ ký số
- Alice gửi **văn bản gốc và chữ ký số** cho Bob qua đường truyền mạng
- Bob nhận được văn bản gốc và chữ ký số
- Bob dùng **khóa công khai** của Alice để **xác minh chữ ký**
- Nếu việc xác minh là chính xác, thì văn bản được gửi kèm chính là do Alice tạo ra, ngược lại văn bản đã có sự biến đổi.

**Qui trình tạo chữ ký số- phong bì số**

- Giả sử Alice và Bob trao đổi với nhau thông qua phong bì số, trong đó Alice là người gửi, còn Bob là bên nhận. Yêu cầu của quá trình này thông tin cần bảo mật, chứng thực, toàn vẹn. Các bước thực hiện của quá trình này như sau:
- Alice tạo chữ ký số (CKS) đối với văn bản gốc bằng khóa cá nhân của Alice
- Văn bản gốc và chữ ký số được mã hóa bằng khóa công khai của Bob. Kết quả quá trình mã hóa này gọi là Phong bì số. Phong bì số được gửi cho Bob qua kênh công cộng.
- Bob nhận được phong bì số thực hiện việc giải mã bằng khóa cá nhân, thu được văn bản gốc và chữ ký số. Sau đó thực hiện việc xác minh chữ ký số để kiểm tra tính chứng thực và toàn vẹn của văn bản.



Hình 1.5. Minh họa quá trình gửi và nhận của phong bì số

### 1.5.3. Bài toán chứng thực và toàn vẹn: chữ ký số và mã chứng thực

**Chữ ký số** là thông tin gửi kèm với văn bản gốc, được tạo ra dựa trên nền tảng mã hóa khóa công khai sử dụng khóa cá nhân của người gửi, nhằm đảm bảo cho người nhận định danh, xác thực đúng nguồn gốc và tính toàn vẹn của tài liệu.

#### Sử dụng chữ ký điện tử giúp:

- Chứng thực
- Đảm bảo tính toàn vẹn
- Chống thoái thác

**Chứng thực (authentication):** là quá trình nhận dạng người dùng, là một phần quan trọng trong ĐỊNH DANH và CHỨNG THỰC (Identification & Authentication – I & A).

Ba yếu tố của chứng thực :

- Những gì bạn biết: username, password, pincode ...
- Những gì bạn có: smart card, chứng chỉ ...
- Cái bạn sở hữu: dấu vân tay, móng mắt, giọng nói ...

**Chứng thực số (digital certificate):** Còn gọi là chứng thực khóa công khai (public key certificate), là một tài liệu điện tử dùng để xác minh một khóa công khai là của ai.

Trong mô hình hạ tầng khóa công khai (public key infrastructure - PKI), nhà cung cấp chứng thực số (Certificate Authority - CA) đóng vai trò bên thứ ba để hỗ trợ cho quá trình trao đổi được an toàn và thường là trung tâm trong nhiều mô hình PKI.

Mỗi chứng thực số bao gồm các thông tin cơ bản sau:

- Tên và URL của CA cung cấp chứng thực
- Khóa công khai
- Tên sở hữu: cá nhân, tổ chức, máy chủ
- Thời hạn sử dụng
- CA sẽ chịu trách nhiệm ký lên mỗi chứng thực số

**Một số phương thức chứng thực thông dụng:**

- Dùng username/password
- Giao thức chứng thực thử thách, bắt tay (Challenge HandShake Authentication Protocol - CHAP)
- Chứng chỉ số: Certificate Authority (CA)
- Bảo mật bằng token
- Phương pháp chứng thực Kerberos
- Chứng thực đa yếu tố
- Chứng thực bằng thẻ thông minh (Smart card)
- Chứng thực bằng sinh trắc học

## 1.6.Pháp luật về an toàn thông tin

Con người hiện nay mở rộng phạm vi hoạt động của mình bằng việc kết nối mạng, điều đó có thể có những tác động tích cực và tiêu cực trong một xã hội kết nối với nhau. Mọi cá nhân, tổ chức đều phải nhận thức được trách nhiệm của mình đối với cộng đồng

Với việc kết nối máy tính vào mạng, con người có thể mở rộng phạm vi hoạt động của mình thì điều đó cũng có nghĩa là những tác hại có thể được nhân lên qua mạng. Vì thế trong một xã hội "nối mạng", mọi cá nhân phải nhận thức được trách nhiệm với cộng đồng. Trong phần này đề cập những khái niệm cơ bản về tin tặc, tội phạm kỹ thuật cùng các quy định để đảm bảo an toàn thông tin.

### 1.6.1.Tin tặc, tội phạm kỹ thuật

**Tin tặc (Hacker):** Là một người hay nhóm người sử dụng sự hiểu biết của mình về cấu trúc máy tính, hệ điều hành, mạng, các ứng dụng trong hệ thống... để tìm lỗi, lỗ hổng, điểm yếu và tìm cách xâm nhập, thay đổi hay chỉnh sửa với các mục đích khác nhau (tốt hoặc xấu).

Hiện nay phổ biến hai loại hacker:

- **Hacker mũ trắng** là những người mà hành động tấn công, xâm nhập và thay đổi, chỉnh sửa hệ thống phần cứng, phần mềm với mục đích tìm ra các lỗi, lỗ hổng, điểm yếu bảo mật và đưa ra giải pháp ngăn chặn và bảo vệ hệ thống chẳng hạn như những nhà phân tích An ninh mạng.

- **Hacker mũ đen** là những người mà hành động tấn công, xâm nhập, thay đổi, chỉnh sửa hệ thống phần cứng, phần mềm với mục đích phá hoại, hoặc vi phạm pháp luật.

#### 14.6.1. Một số tội phạm tin học liên quan đến Internet

- **Mạo danh, xâm nhập** máy tính trái phép để đánh cắp và huỷ hoại thông tin.
- **Lừa đảo qua mạng** (Phishing): là xây dựng những hệ thống lừa đảo nhằm đánh cắp thông tin có giá trị như tên đăng nhập, mật khẩu, thông tin thẻ tín dụng, loại hình này trở thành hiểm hoạ đe dọa thương mại điện tử, làm giảm lòng tin vào các giao dịch điện tử.
- **Thư rác** (Spamming) là hình thức lạm dụng gửi bừa bãi thư không mong muốn hàng loạt đến người dùng, làm cho người dùng bị cản trở, khó khăn trong việc tiếp cận các tài liệu của công việc. Hiện nay, tại Việt Nam loại hình này khá phổ biến do sự nhu cầu quảng bá sản phẩm và sự hạn chế của luật phát trong việc xử lý loại hình này.
- **Tấn công từ chối dịch vụ** (Denial of Service): là kiểu tấn công làm cho hệ thống máy tính, hệ thống mạng bị quá tải ... dẫn đến hệ thống quá tải, hoặc không thể hoạt động bình thường. Khi đó, các máy chủ dịch vụ bị quá tải do có quá nhiều kết nối yêu cầu truy vấn làm cho khả năng truy vấn bị hạn chế.

#### 1.6.2. Vấn đề sở hữu trí tuệ và bản quyền

**Sở hữu trí tuệ** được hiểu là là tài sản trí tuệ, là những sản phẩm sáng tạo của bộ óc con người. Đó có thể là tác phẩm văn học, âm nhạc, phần mềm máy tính, phát minh, sáng chế, giải pháp hữu ích, kiểu dáng công nghiệp..

Các loại đối tượng của quyền sở hữu trí tuệ: Bản quyền, bằng sáng chế, thương hiệu, kiểu dáng công nghiệp, sơ đồ bố trí mạch tích hợp, chỉ dẫn địa lý.

Phần mềm máy tính gồm chương trình, tài liệu mô tả chương trình, tài liệu hỗ trợ, cơ sở dữ liệu... Trong lĩnh vực công nghệ thông tin, việc vi phạm bản quyền (sao chép, nhân bản phần mềm bất hợp pháp, sử dụng không xin phép ...) được diễn ra phổ biến, gây ra thất thoát lớn về tiền bạc, lòng tin, môi trường đầu tư. Ở nước ta, Luật sở hữu trí tuệ (2005) đã đưa các nội dung về bảo vệ bản quyền phần mềm máy tính vào trong luật.

Tuy nhiên, tình trạng vi phạm bản quyền, sở hữu trí tuệ ở nước ta vẫn rất phổ biến. Một số nguyên nhân có thể kể đến như:

- Sự hiểu biết của xã hội về sở hữu trí tuệ, bản quyền còn hạn chế, văn hóa tôn trọng bản quyền, sở hữu trí tuệ chưa được hình thành.
- Việc thực thi các văn bản pháp luật cho các hành vi vi phạm còn hạn chế, mặc dù các văn bản khá đầy đủ nhưng việc xử lý chưa đến nơi đến chốn làm cho việc ngăn chặn vi phạm bản quyền, sở hữu trí tuệ nói chung và trong lĩnh vực phần mềm nói riêng chưa phát huy được hiệu quả.

### 1.6.3. Luật về tội phạm tin học ở Việt Nam

Ở nước ta, nhận thức được tính nghiêm trọng của các tội phạm tin học, Quốc hội nước CHXHCN Việt Nam đã ban hành một số luật liên quan đến các vấn đề liên quan đến tội phạm tin học, cụ thể có thể kể đến:

- Luật Công nghệ thông tin năm 2006 (số 67/2006/QH11 ngày 29/6/2006)
- Bộ Luật hình sự năm 1999 (sửa đổi, bổ sung năm 2009) tiếp cận dưới góc độ tội phạm công nghệ cao là tội phạm được thực hiện và gây hậu quả trên môi trường ảo, thế giới ảo do thành tựu của khoa học công nghệ tin học đem lại và nó hoàn toàn khác với các loại tội phạm truyền thống trước kia.
- Bộ luật Hình sự năm 2009 sửa đổi bổ sung Bộ luật Hình sự năm 1999 có sửa đổi, bổ sung các tội danh có liên quan đến máy tính, mạng máy tính, gồm các điều:
  - Điều 224: Tội phát tán vi rút, chương trình tin học có tính năng gây hại cho hoạt động của mạng máy tính, mạng viễn thông, mạng Internet, thiết bị số.
  - Điều 225: Tội cản trở hoặc gây rối loạn hoạt động của mạng máy tính, mạng viễn thông, mạng Internet, thiết bị số.
  - Điều 226: Tội đưa hoặc sử dụng trái phép thông tin mạng máy tính, mạng viễn thông, mạng Internet.
  - Điều 226a: Tội truy cập bất hợp pháp vào mạng máy tính, mạng viễn thông, mạng Internet hoặc thiết bị số của người khác.
  - Điều 226b: Tội sử dụng mạng máy tính, mạng viễn thông, mạng Internet hoặc thiết bị số thực hiện hành vi chiếm đoạt tài sản.

### 1.7. Các phần mềm độc hại

Phần mềm độc hại là bất kỳ phần mềm hay ứng dụng nào được thiết kế để gây hại máy tính, thiết bị, phần mềm mà máy tính hay thiết bị sử dụng. Phần mềm độc hại bao gồm cả hành vi cài đặt phần mềm mà không cần sự đồng ý của người dùng.

Phần mềm độc hại hiện nay phát triển rất nhanh do các mục đích khác nhau, trong đó có kể đến các yếu tố về kinh tế, chính trị, phá hoại và các mục đích khác. Tiêu biểu có thể kể đến: Trojan horse, Virus, Worm, Malicious Mobile Code, Attacker Tool , Backdoor (Trapdoor), Keylogger, Rootkits, Web Browser Plug-in, Email Generator .

### 1.8. Câu hỏi và bài tập

1. Trình bày các khái niệm về tính bí mật, tính sẵn sàng và tính an toàn trong ATTT? Cho ví dụ minh họa.
2. Trình bày một số kiểu tấn công mạng?
3. Trình bày và phân tích các nguy cơ về ATTT? Cho ví dụ minh họa.
4. Trình bày và phân tích các giải pháp bảo đảm ATTT? Cho ví dụ minh họa.
5. Trình bày tổng quan về thực trạng ATTT trên thế giới và tại Việt Nam?
6. Trình bày các khái niệm về tin tặc và tội phạm kỹ thuật.



7. Trình bày một số vấn đề tội phạm tin học liên quan đến lạm dụng mạng.
8. Trình bày sự thay đổi mục đích tấn công hệ thống thông tin qua các giai đoạn. Cho ví dụ minh họa.
9. Trình bày một số vấn đề về sở hữu trí tuệ và luật bản quyền? Nêu một số biện pháp sinh viên có thể thực hiện trong việc tôn trọng bản quyền và sở hữu trí tuệ trong học tập và công việc.
10. Trưởng phòng Quản trị mạng máy tính của công ty gọi điện thoại yêu cầu nhân viên phòng Quan hệ khách hàng cung cấp username và password để làm báo cáo tổng hợp gửi Giám đốc. Theo sinh viên, nhân viên phòng Quan hệ khách hàng có cung cấp thông tin username và password hay không? Giải thích lí do?
11. Tuấn nhận được một email lạ chúc mừng sinh nhật từ một người lạ, trong đó đính kèm thiệp. Một người bạn của Tuấn khuyên xóa ngay email đó mà không cần hỏi ý kiến của bộ phận IT, lời khuyên đó đúng hay sai? Vì sao?
12. Màn hình desktop của bạn đột ngột có sự di chuyển con trỏ của chuột, sự di chuyển này không có tác động của người sử dụng và bất thường. Sinh viên nghĩ đến vấn đề gì?
13. Việc trao đổi, chia sẻ trên facebook hiện nay là trào lưu của thanh niên, trong đó có việc chia sẻ thông tin cá nhân và các thành viên trong gia đình, các hoạt động thường nhật. Sinh viên đánh giá như thế nào về hành động trên về mặt an toàn thông tin.
14. Phương phát hiện mật khẩu của email cá nhân bị tiết lộ ra bên ngoài, theo sinh viên, việc làm nào Phương cần thực hiện đầu tiên?
15. Trình bày các lợi ích của việc thực hiện tiêu chuẩn về an toàn thông tin như ISO/IEC 27001:2013
16. Cho biết sự khác nhau giữa Hacker và Cracker.
17. Một server cung cấp dịch vụ web ra ngoài internet. Server này nằm trong vùng DMZ. Người quản trị hệ thống cần mở port nào tại tường lửa để bên ngoài truy cập được vào server này?
18. Liệt kê một số yêu cầu về tính an toàn đối với một hệ thống phân tán?
19. Trình bày cách sử dụng danh sách kiểm soát truy cập (access control lists) được sử dụng đại diện cho kiểm soát truy cập ma trận (access control matrices). Liệt kê các môi trường mà chúng được sử dụng cùng ưu, khuyết điểm.
20. Trình bày nguyên tắc leo thang đặc quyền.
21. Trong cơ quan A thường xuyên làm việc với các văn bản mật (bao gồm lưu trữ hồ sơ, công văn đi, đến). Anh/chị đề xuất 5 biện pháp (kỹ thuật, phi kỹ thuật) để tăng mức độ an toàn cho các hoạt động của cơ quan trên.

---

## CHƯƠNG 2: MÃ ĐỘC

---

Chương 2 giới thiệu về mã độc (malware), phân loại mã độc, các kỹ thuật lây nhiễm, phá hoại của mã độc và phương pháp phát hiện mã độc.

### 2.1. Giới thiệu Malware

Mã độc (Malicious Software - Malware) là những chương trình máy tính độc hại với nhiệm vụ chủ yếu là đánh cắp thông tin, phá hủy hay làm hư hỏng hệ thống. Malware xâm nhập hệ thống một cách trái phép mà không có sự cho phép của người quản trị. Chính vì thế định nghĩa malware đã bao gồm cả: Virus, Worm, Trojan, Spyware, Adware...

### 2.2. Phân loại Malware

#### 2.2.1. Virus

Virus là chương trình phần mềm có khả năng sao chép chính nó từ đối tượng lây nhiễm này sang đối tượng khác. Khi người dùng mở chương trình thực thi này thì virus cũng hoạt động theo.

Virus chèn code nó vào những vị trí thích hợp, đầu hoặc cuối file bị lây nhiễm. Nếu chèn vào cuối file thì nó thêm một vài lệnh ở đoạn byte đầu file để thực hiện lệnh “nhảy” từ đoạn mã này tới đoạn code virus. Nó cũng có thể chèn toàn bộ nó vào đoạn đầu của file bị lây nhiễm.

Phần lớn virus chỉ làm việc khi bị kích hoạt vào file đã bị lây nhiễm sau đó sẽ trả lại quyền làm việc lại cho chương trình ban đầu. Một số virus lây nhiễm sau khi được chạy và thực thi một tác vụ và lây nhiễm sang những khu vực khác khi người dùng kích hoạt vào chúng, có thể là một ngày, một khoảng thời gian, hoặc một sự kiện đặc biệt nào đó.

Virus System hoặc Boot sector: sao chép code vào vị trí ban đầu của MBR và di chuyển MBR đến vị trí khác trên đĩa cứng. Khi hệ thống khởi động, code Virus được thực thi đầu tiên.

Virus File: lây nhiễm các file như EXE, COM, SYS, OVL, OBI, PRG, MNU, BAT. Virus file có thể thường trú hoặc không thường trú trong bộ nhớ. Các tập tin hoặc chương trình khi chạy sẽ tải virus trong bộ nhớ và thực hiện các chức năng được định sẵn để lây nhiễm cho hệ thống. Ví dụ cho loại này có thể kể đến như: Black Monday, Virus Sunday, Virus Cascade ...

Virus Multipartite: là loại virus đa phương lây nhiễm trên nhiều nền tảng hệ điều hành và các mục tiêu khác nhau, thực hiện cả chức năng của Virus Boot sector và Virus File. Tức là nó thực hiện tấn công cả boot sector và các chương trình thực thi cùng một lúc. Ví dụ: Ghostball, Invader, Flip, ...

**Virus Macro:** hầu hết các Virus macro được viết bằng ngôn ngữ macro Visual Basic (VBA). Virus Macro lây nhiễm các file tài liệu thông thường hoặc đính kèm vào các chúng. Ví dụ: Virus W97M/Class, Virus Concept, Melissa Worm, ...

**Virus Cluster:** thay đổi đường dẫn trỏ đến các code virus thay vì chương trình thực tế. Nó sẽ khởi động chính nó đầu tiên hoặc kiểm soát thay vì chương trình gốc. Virus này lây lan rất nhanh, chỉ cần một bản sao của Virus trên đĩa sẽ lây nhiễm cho tất cả các chương trình trong hệ thống máy tính. Ví dụ: Virus DIR-II

**Virus Overwriting File hoặc Virus Cavity:** ghi đè lên 1 phần của file bị nhiễm với 1 hằng số (null), mà không thay đổi độ dài hoặc chức năng của file. Ví dụ: SHHS

**Virus File Extension:** thay đổi phần mở rộng của file, ngoại trừ phần mở rộng .TXT an toàn vì nó chỉ ra một tập tin văn bản thuần túy. Đây là một virus được viết bằng Visual Basic Script. Biện pháp đối phó là để tắt "phần mở rộng tập tin ẩn" trong Windows. Ví dụ: virus VBS.Gum

**Virus Terminate hoặc Stay Resident (STR):** virus này thường trú trên bộ nhớ, sử dụng kỹ thuật TSR (Terminate and Stay Resident) để lưu lại trên bộ nhớ memory cho đến khi một vài sự kiện nhỏ xảy ra (ví dụ file.exe được mở) và sau đó chúng sẽ lây nhiễm file đó. Ví dụ: Azusa

**Virus Direct Action hoặc Virus Transient:** được tải với các chương trình máy chủ vào bộ nhớ máy tính. Sau khi nhận được kiểm soát, nó tìm kiếm đối tượng mới để lây nhiễm bằng cách tìm kiếm cho các tập tin mới. Virus này thường sử dụng một chuỗi FindFirst, FindNext để tìm một tập hợp các ứng dụng nạn nhân để tấn công. Thông thường virus này chỉ gây nhiễm một vài tập tin khi thực hiện, nhưng một số virus lây nhiễm tất cả mọi thứ cùng một lúc bằng cách liệt kê tất cả các thư mục của các nạn nhân. Ví dụ: Virdem

**Virus Shell:** code virus này tạo thành một shell xung quanh mục tiêu code của chương trình, làm cho bản thân chương trình ban đầu phụ thuộc vào nó. Hầu hết các Virus boot là Virus Shell. Ví dụ: Virus: Bash/1

**Virus Metamorphic:** là thế hệ sau của các dòng virus đa hình. Các virus này có khả năng tự động biến đổi mã lệnh của nó, tạo ra các biến thể khác nhau trong mỗi lần lây nhiễm. Loại virus này thường kết hợp nhiều kiểu đa hình chồng chéo, sinh ra các thế hệ virus "con cháu" F1, F2, ... Ví dụ: W95.Matrix.SCR

**Virus Intrusive:** hoạt động bằng cách thay thế một phần hoặc tất cả code chương trình gốc với code virus. Việc thay thế có thể là lựa chọn, nhưng phần lớn các chương trình chủ được thay thế hoàn toàn bằng các mã virus.

**Virus Companion hoặc Camouflage:** là một loại virus máy tính phức tạp, không giống như các virus truyền thống, không sửa đổi bất kỳ tập tin. Thay vào đó, nó tạo ra

một bản sao của tập tin và đặt một phần mở rộng khác nhau về nó, thường .com. Ví dụ: AIDS

**Virus Polymorphic:** là virus có khả năng tự động biến đổi mã lệnh để tạo ra các loại virus khác nhau trong mỗi lần lây nhiễm. Ví dụ: MtE (The Dark Avenger Mutation Engine)

**Virus Add-on:** chèn thêm code virus vào code chương trình gốc hoặc di dời code chương trình gốc để tự chèn thêm code của chúng. Virus này thường lây nhiễm các trình duyệt, làm thay đổi cookie, history, trang chủ ..., chuyển chúng tới trang web chứa virus hoặc các trang quảng cáo. Ví dụ: Coupon Addon

**Virus Sparse Infector:** virus này sử dụng kỹ thuật lây nhiễm thưa thớt. Ví dụ nó có thể chỉ gây nhiễm mỗi lần thứ 20 một tập tin được thực hiện, nó chỉ có thể lây nhiễm các tập tin có độ dài nằm trong phạm vi hạn hẹp hoặc có tên bắt đầu bằng chữ cái trong một phạm vi nhất định của bảng chữ cái.

**Virus Encrytion:** nó sẽ mã hóa lộn xộn mã chương trình của mình để làm cho nó khó khăn để phát hiện. Mỗi lần nó lây nhiễm, nó sẽ tự động mã hóa bản thân khác nhau, do đó mã của nó là không bao giờ giống nhau. Ví dụ: Cascade.1701

**Virus stealth hoặc virus tunneling:** là một virus có thể sử dụng các cơ chế khác nhau để tránh bị phát hiện bởi các phần mềm chống virus. Ví dụ: FRODO-A

### 2.2.2.Worm

Worm là chương trình độc hại lây lan thông qua các kết nối mạng 1 cách độc lập mà không tương tác với con người. Worm không cần lây nhiễm để kích hoạt vì nó có khả năng tự đóng gói, nên antivirus sẽ không diệt được worm trong hệ thống vì worm sẽ không lây nhiễm file hoặc một vùng nào đó trên đĩa cứng.

Worm thực hiện các công việc cơ bản sau:

- Tự sao chép vào các thư mục của hệ thống.
- Ghi thông tin khởi động vào hệ thống: để mỗi lần khởi động nó có thể làm việc.
- Lây truyền

Vì sử dụng hệ thống mạng và bộ nhớ để làm việc: nên worm luôn chiếm tài nguyên sử dụng của RAM và đường truyền mạng.

Phân loại:

**Network Service Worm:** lan truyền bằng cách lợi dụng các lỗ hổng bảo mật của mạng, của hệ điều hành hoặc của ứng dụng. Ví dụ: Worm/Kazaa

**Mass Mailing Worm:** tấn công qua dịch vụ mail bằng cách tự đóng gói để tấn công và lây nhiễm chứ không bám vào vật chủ là email. Ví dụ: Worm.Pikachu.AuExec

### 2.2.3.Trojan

Trojan là loại mã độc ẩn trong các chương trình hợp pháp, có khả năng gây hại cho người dùng. Thông thường, Trojan thực hiện trực tiếp các công việc gây hại cho người dùng khi ta kích hoạt nó hoặc tự động “nằm vùng” trong máy tính sau đó kích hoạt nhằm mục đích lấy thông tin của người dùng như tài khoản cá nhân (email, username, password, số tài khoản ...)

Một số loại Trojan cơ bản:

- Remote Access Trojan (RAT): cho phép kẻ tấn công (attacker) truy cập từ xa vào hệ thống.
- Data-Sending Trojan: dùng để đánh cắp dữ liệu trên hệ thống và gửi về cho attacker.
- Destructive Trojan: dùng để phá hủy tập tin trên hệ thống.
- Denial of Service Trojan: dùng để phát động các đợt tấn công từ chối dịch vụ.
- Proxy Trojan: dùng để tạo ra các vỏ bọc truyền thông (tunnel) hay phát động tấn công từ một hệ thống khác.
- FTP Trojan: dùng để tạo ra dịch vụ FTP nhằm sao chép dữ liệu lên hệ thống bị nhiễm.
- Security Software Disabler Trojan: dùng để tắt các dịch vụ phòng chống virus, trojan hoặc các tính năng bảo mật khác.

Ví dụ: Back Orifice, Deep Throat, NetBus, Whack-a-mole, Netbus 2 Pro, GrilFriend, Masters Paradise ...

### 2.2.4.Backdoor

Backdoor là các phần mềm độc hại thường trú và đợi lệnh điều khiển từ các cổng dịch vụ TCP hoặc UDP. Attacker thông qua chương trình này để xâm nhập vào hệ thống ở các lần sau.

Phân loại:

- Zoombie (bot): là một chương trình được cài đặt lên hệ thống nhằm mục đích tấn công hệ thống khác. Những thiết bị bị lây nhiễm sẽ bị sử dụng tấn công DOS mà người dùng không biết. Ví dụ: Alpha Strike
- Remote Administration Tool: là công cụ của hệ thống cho phép thực hiện quyền quản trị từ xa. Attacker sử dụng tính năng này để có thể theo dõi màn hình, bàn phím, hoặc tác động vào cấu hình của hệ thống. Ví dụ: TR/BackDoor.Bo

### 2.2.5.Keylogger

Keylogger là phần mềm bí mật ghi lại các thao tác bằng bàn phím, thao tác chuột, hoặc screen rồi gửi tới hacker. Keylogger có thể ghi lại nội dung của email, của văn bản, user name, password, thông tin bí mật, ...

Nguyên tắc hoạt động: Khi bấm 1 phím trên bàn phím máy tính, bàn phím sẽ chuyển nó thành tín hiệu chuyển vào CPU. CPU sẽ chuyển tín hiệu đến hệ điều hành để xử lý. Phần mềm keylogger sẽ ghi nhận các tín hiệu ghi vào tập tin log. Đồng thời nó còn có thể theo dõi cả màn hình và thao tác chuột.

Phân loại:

- Keylogger phần mềm: Đây là những chương trình máy tính được thiết kế để làm việc trên máy tính, được sử dụng cho mục đích theo dõi việc sử dụng mạng của gia đình, con cái .... Tuy nhiên, nhiều cá nhân có thể lợi dụng keylogger trên máy tính công cộng để ăn cắp mật khẩu, thông tin cá nhân, thẻ tín dụng ... Hầu hết các keylogger mềm là trojan hay backdoor. Ví dụ: DanuSoft Free Keylogger, Revealer Keylogger, ...

- Keylogger phần cứng: là loại tồn tại ở mức độ phần cứng trong một hệ thống máy tính. Ví dụ: Wireless keyboard sniffers, Acoustic keyloggers, ...

### 2.2.6.Rootkit

Rootkit là công cụ phần mềm được viết ra để che giấu sự tồn tại và hoạt động của các tiến trình, tập tin ... Attacker thường sử dụng rootkit khi chiếm quyền truy cập vào hệ thống. Đặc điểm Rootkit là có khả năng ẩn các tiến trình, file, và cả dữ liệu trong registry, vì thế người dùng sẽ không phát hiện được có bị nhiễm malware hay không.

Phân loại:

- Rootkit hoạt động ở mức ứng dụng: Ở mức này Rootkit được coi như phần mềm ứng dụng, sử dụng một số kỹ thuật như code inject, tạo file giả ... để can thiệp vào các ứng dụng khác nhằm thực hiện mục đích che giấu tiến trình, file, registry ... Ví dụ: Koutodoor

- Rootkit hoạt động trong nhân của hệ điều hành: hoạt động cùng mức với các driver. Đây là mức thấp của hệ thống, do vậy rootkit có quyền rất lớn với hệ thống. Ví dụ: TDSS

### 2.2.7.Adware/Spyware

Adware tạo ra trình đơn quảng cáo popup mà không có sự cho phép của người dùng. Adware thường được cài đặt bởi một thành phần của phần mềm miễn phí. Adware thường gây khó chịu cho người dùng vì thường xuyên xuất hiện quảng cáo trên màn hình, khi hoạt động adware sử dụng tài nguyên máy tính. Ví dụ: TR.Spectre #1

Spyware là một phần mềm thực hiện đánh cắp thông tin từ máy tính mà người dùng không hề hay biết, đây là phần mềm gián điệp. Các phần mềm không rõ nguồn gốc hiện nay chứa rất nhiều spyware. Ví dụ: Gliss #1

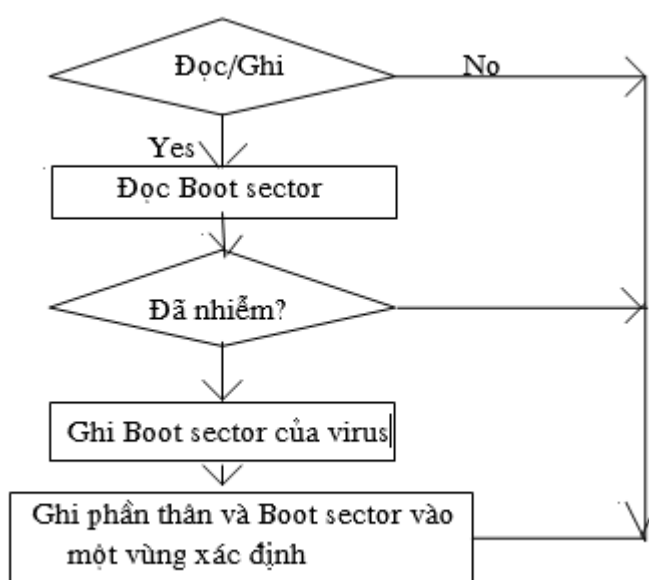
### 2.2.8.Attacker Tool

Attacker Tool là những bộ công cụ có thể được sử dụng để đẩy các phần mềm độc hại vào trong hệ thống. Các bộ công cụ này giúp cho kẻ tấn công có thể truy nhập bất hợp pháp vào hệ thống hoặc làm cho hệ thống bị lây nhiễm mã độc.

## 2.3.Các kĩ thuật lây nhiễm và phá hoại trong Malware

### 2.3.1.Kĩ thuật lây nhiễm

Việc lây nhiễm chiếm một phần lớn mã lệnh của chương trình. Để đảm bảo việc lây nhiễm với đĩa cứng, virus sẽ chiếm ngắt 13h. Lược đồ như sau:



Hình 2.1. Lược đồ lây nhiễm virus

Việc can thiệp vào ngắt 13h sẽ dẫn đến việc giảm tốc độ truy xuất một cách đáng kể vì vậy dễ gây nên sự nghi ngờ. Việc lây nhiễm bắt đầu bằng cách đọc Boot sector lên nếu chưa bị nhiễm, virus sẽ tạo một Boot sector có tham số tương ứng, còn Boot sector vừa đọc lên sẽ được ghi vào một vùng xác định trên đĩa.

### 2.3.2.Kĩ thuật phá hoại

Có thể chia làm 2 kĩ thuật chính:

- Định thời: virus sẽ dựa vào một giá trị nào đó (ví dụ ngày, tháng, số lần lây, số lần khởi động máy tính ...). Khi giá trị này bằng hoặc lớn hơn một giá trị cho trước, virus sẽ bắt đầu phá hoại. Do chỉ phá hoại một lần nên virus này thường rất nguy hiểm. Để kiểm tra giá trị, virus thường dùng cách sau:



• Chiếm ngắt 8 để đếm giờ: mỗi giá trị 0FFFFh của timetick count = 1 giờ. Ví dụ là virus Disk Killer, sau khi PC chạy được 48h, toàn bộ partition boot sẽ bị mã hóa toàn bộ.

• Chiếm ngắt 21h để lấy ngày tháng: ví dụ là virus Joshi, nó sẽ kiểm tra ngày tháng, nếu vào đúng ngày 5 tháng 1, nó sẽ bắt user đánh vào một câu chúc mừng “Happy birthday Joshi” trước khi có thể làm thêm bất kì một điều gì.

• Đếm số lần lây cho đĩa khác: cách này dễ thực hiện, khi một virus lây nhiễm, bộ đếm của nó tự động tăng lên một đơn vị.

- Ngẫu nhiên và liên tục: không giống như định thời, sau khi lây nhiễm, virus sẽ bắt đầu phá hoại. Do tính chất này, virus không mang tính phá hoại mà đơn giản là gây ra một số hiệu ứng ở loa, chuột, màn hình, hoặc lấy cắp thông tin, reset máy hoặc format đĩa, .... Ví dụ là virus Pingpong, sau khi xâm nhập xong, vào phút sau sẽ thấy trên màn hình xuất hiện một trái banh chuyển động và tuân theo các định luật phản xạ khi gặp đường biên.

## 2.4. Tổng quan các kỹ thuật phát hiện Malware

Thông thường có 2 loại phân tích:

- Dynamic analysis (Phân tích động): là quá trình này có nghĩa là quan sát hành vi của mã độc khi nó đang chạy. Cụ thể là những công việc như: Khởi chạy mã độc trên một môi trường ảo, quan sát xem khi mà mã độc chạy nó sẽ làm những gì,.... Debug mã độc sử dụng một công cụ Debugger nào đó: Ví dụ winDBG, OllyDBG để quan sát từng bước hành vi của mã độc khi nó đang được thực thi bởi bộ nhớ và load trong RAM.

- Static analysis (Phân tích tĩnh): Là quá trình dịch ngược mã độc (RCE - reverse engineering). Dịch ngược mã độc từ mã máy ra ngôn ngữ mà con người có thể đọc hiểu (asm, IL,...). Trong quá trình này các nhà phân tích sẽ sử dụng những công cụ dịch ngược như IDA, khi thực hiện dịch ngược IDA không load mã độc vào RAM như ollyDBG, IDA.

## 2.5. Câu hỏi và bài tập

1. Trình bày ảnh hưởng của mã độc đối với an toàn thông tin. Nếu ví dụ 10 loại cụ thể.
2. Nêu một số biện pháp phòng tránh mã độc nói chung và virus nói riêng.
3. Loại virus nào lây nhiễm giữa các tài liệu word và excel?
4. Một ứng dụng được tải về từ internet với mục đích dọn dẹp ổ đĩa và xóa các tập tin không cần thiết, ứng dụng này cũng ghi lại các thao tác của người dùng trên máy tính và gửi đến một địa chỉ cố định cho trước. Theo sinh viên, ứng dụng nào phù hợp với mô tả trên?
5. Chọn mô tả về mã độc trong danh sách (a) phù hợp nhất cho các mô tả (b). Giải thích.

(a) In the wild, anti-virus software, back door, hybrid virus, social engineering, logic bomb, spambot, Trojan horse, malware, data miner, denial of service, macro virus, botnet, adware, e-mail virus, ethical worm, executable, spyware, executable, zoo, DDoS attack, IM worm, payload, hybrid virus/worm, password cracker, probe, ethical worm, port scan, key logger.

(b)

- Loại phần mềm với mục đích quảng cáo đồng thời thường xuyên theo dõi hành vi của người dùng.
- Loại tấn công mà nhiều hệ thống/ thiết bị đồng thời tấn công một mục tiêu, khiến người dùng bị từ chối các dịch vụ bình thường.
- Loại file có chứa chương trình để chạy ứng dụng của nó; virus thường được truyền theo cách này.
- Loại mã độc được kích hoạt bởi một số trình kích hoạt, chẳng hạn như một ngày cụ thể.
- Loại chương trình được sử dụng để tự động vá lỗi bảo mật.
- Mã độc tự sao chép lan truyền qua mạng tin nhắn tức thời
- Công cụ truy cập vào một hệ thống máy tính được đưa ra bởi người dùng hoặc cracker
- Cách tiếp cận phi kỹ thuật để truy cập trái phép mật khẩu, thường là bằng cách lừa người dùng thiếu hiểu biết.
- Một đoạn mã có thể sao chép chính nó và làm hỏng hệ thống hoặc phá hủy dữ liệu.
- Mạng máy tính bị nhiễm phần mềm độc hại và được điều khiển theo nhóm mà không có sự hiểu biết của chủ sở hữu, ví dụ như gửi thư rác.
- Bí mật thu thập mật khẩu, tên người dùng, thông tin tài khoản ngân hàng, số thẻ tín dụng đã được sử dụng nhập vào thiết bị, sau đó truyền lại cho bọn tội phạm.
- Một chương trình tự động thu nhận thông tin cá nhân như email, tin nhắn SMS, tin nhắn trên mạng xã hội. Trong vài trường hợp, nó có thể crack mật khẩu và gửi tin nhắn từ tài khoản của người dùng.
- Chương trình ẩn trong các ứng dụng hợp pháp, khi kích hoạt nó cho phép attacker truy cập trái phép vào máy tính hoặc thiết bị di động.
- Tập hợp các thiết bị bị nhiễm mã độc (hàng nghìn đến hàng chục nghìn thiết bị) có thể được kiểm soát bởi một máy chủ. Toàn bộ (hoặc một phần) có thể được bán cho các tội phạm khác sử dụng với mục đích như thư rác, trộm danh tính hoặc tấn công từ chối dịch vụ phân tán.

6. Chọn mô tả về mã độc trong danh sách (a) phù hợp nhất cho các mô tả (b).  
Giải thích.

(a) Backdoor, Spyware, Adware, Worm, Trojan horse, Rootkit, Logic bomb, Parasitic, Boot sector, Data file.

(b) Nguyên nhân có thể dẫn đến leo thang đặc quyền (người dùng có quyền nhiều hơn những gì mà họ được cấp)

- Người dùng phàn nàn máy tính của họ thường bị treo và phán đoán có ai đó sử dụng tài khoản của họ để đăng nhập email. Họ sử dụng phần mềm diệt virus nhưng không thu được kết quả. (Nguyên nhân?)
- Người dùng mở một tài liệu Microsoft Word và nhận thấy rằng các tập tin khác mà họ đang mở đột ngột đóng lại. Khi mở lại các tệp này thì phát hiện thông tin trong đó đã được sửa đổi.

---

## CHƯƠNG 3: MÃ HÓA

---

Mã hóa là kỹ thuật lưu trữ an toàn, bảo mật truyền tin trong môi trường không an toàn nhằm ngăn chặn việc đọc thông tin của những người không hợp pháp. Mã hóa hiện nay là công cụ không thể thiếu trong việc bảo vệ thông tin trong hệ thống máy tính. Trong chương này giới thiệu một số khái niệm lý thuyết cơ bản về mật mã sử dụng trong bảo mật máy tính cũng như các hoạt động bên trong một số tình huống cụ thể. Danh sách các bài tập, câu hỏi ôn tập cuối chương giúp sinh viên ôn tập lại các kiến thức và rèn luyện kỹ năng giải quyết các vấn đề đặt ra, đồng thời người học cũng có thể nâng cao kiến thức về bảo mật thông qua việc giải quyết các bài toán nâng cao được chọn lọc trong các cuộc thi chuyên sâu về bảo mật máy tính.

### 3.1. Khái niệm về mã hóa

**Kỹ thuật mật mã (cryptology)** là ngành khoa học nghiên cứu 2 lĩnh vực: mã hóa và phân tích mật mã.

**Mật mã học (cryptography)** là ngành khoa học nghiên cứu các phương pháp và kỹ thuật đảm bảo an toàn và bảo mật dữ liệu trong việc truyền tin. Trong thời đại công nghệ hiện nay, ứng dụng của mật mã được triển khai rộng khắp trong nhiều lĩnh vực, một số lĩnh vực cụ thể như:

- Trong các cơ quan chính phủ: bảo vệ thông tin mật, các thông tin quân sự, ngoại giao, ...
- Trong lĩnh vực kinh tế: bảo mật thông tin tài khoản ngân hàng, giao dịch thanh toán, thông tin khách hàng, ...
- Trong y tế: bảo vệ thông tin cá nhân,
- Trong bảo vệ thông tin cá nhân: thông tin riêng tư, tài khoản email, an toàn trên mạng xã hội, ...

**Phân tích mật mã (cryptanalysis)**: ngành khoa học nghiên cứu các phương pháp, kỹ thuật nhằm phá vỡ hệ thống mã hóa. Trong sự phát triển của mật mã thì lĩnh vực mật mã và phân tích mật mã phát triển song hành với nhau, tuy nhiên trong học tập, nghiên cứu thì lĩnh vực mật mã học được quan tâm rộng rãi hơn do các ứng dụng thực tiễn, hiệu quả mà nó đem lại.

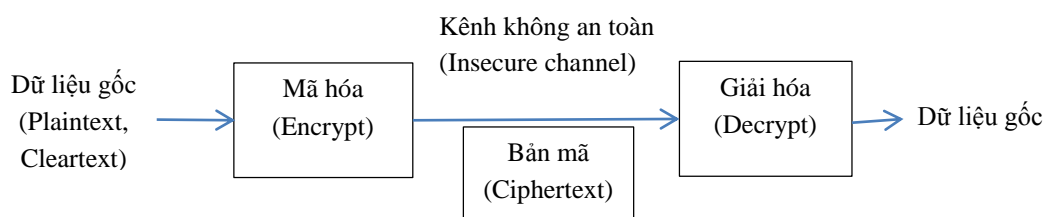
**Giao thức mật mã (cryptographic protocol)** là tập hợp các quy tắc, trình tự thực hiện sơ đồ mã hóa.

**Độ an toàn của hệ mã hóa**: là khả năng chống lại việc thám mã, trong nhiều trường hợp được tính bằng số phép toán cần thực hiện để thám mã sử dụng thuật toán tối ưu nhất.

Hệ thống mật mã (cryptosystem) là hệ thống đảm bảo an toàn dữ liệu sử dụng công cụ mã hóa. Hệ thống mật mã bao gồm: sơ đồ, giao thức mật mã, quy tắc tạo và phân

phối khóa. Khái niệm hệ thống mật mã có thể hiểu đơn giản hơn là bao gồm: thuật toán (algorithm) và giá trị mật (key).

### 3.2. Sơ đồ mã hóa



Hình 3.1. Sơ đồ mã hóa cơ bản

Mô hình toán học tổng quát:

$$\text{Mã hóa: } C = E_{K_1}(P)$$

$$\text{Giải mã: } P = D_{K_2}(C)$$

với

Plaintext (P): dữ liệu gốc, dữ liệu trước khi biến đổi,

Encrypt (E): quá trình mã hóa dữ liệu, biến đổi dữ liệu gốc sang dạng dữ liệu đã mã hóa.

Ciphertext (C): bản mã, dữ liệu nhận được sau khi biến đổi dữ liệu gốc.

Decrypt (D): quá trình giải mã, biến đổi bản mã sang dữ liệu ban đầu.

Key (K): khóa là thành phần tham gia trong thuật toán mã hóa, nếu  $K_1 \equiv K_2$  thì sơ đồ trên là mã hóa đối xứng, ngược lại là mã hóa phi đối xứng.

### 3.3. Phân loại mã hóa

Tùy theo nhu cầu sử dụng trong thực tiễn mà có nhiều phương pháp mã hóa được hình thành và phát triển, theo thời gian có thể chia mật mã thành:

- Mã hóa cổ điển (classical cryptographic): đây là kỹ thuật được hình thành từ xa xưa, dựa trên ý tưởng bên gửi sử dụng thuật toán mã hóa cổ điển dựa trên hai kỹ thuật cơ bản: thay thế (substitution) và hoán vị (transposition), bên nhận dựa vào thuật toán của bên gửi để giải mã mà không cần dùng khóa. Do đó, độ an toàn của kỹ thuật này không cao do chỉ dựa vào sự che giấu thuật toán, hiện nay mã hóa cổ điển ít được sử dụng trong thực tế.
- Mã hóa hiện đại (modern cryptography): mã hóa đối xứng (symmetric cipher, secret key cryptography – 1 khóa), bất đối xứng (asymmetric cipher, public key cryptography – 2 khóa), hàm băm (hash functions – không có khóa).

Ngoài ra, dựa theo cách thức xử lý dữ liệu đầu vào (data input) vào người ta phân chia thành 2 loại:

- Mã hóa khối (block cipher): xử lý dữ liệu đầu vào theo khối, cho kết quả theo khối ở đầu ra.
- Mã hóa luồng (stream cipher): xử lý tuần tự các phần tử đầu vào và cho kết quả từng phần tử ở đầu ra.

### 3.3.1. Mã hóa cổ điển

Mã hóa cổ điển dựa trên kỹ thuật thay thế (thay thế kí tự hoặc các kí tự này bằng kí tự hoặc các kí tự khác tương ứng) và hoán vị (thay đổi trật tự, vị trí các ký tự) trong văn bản gốc. Các kỹ thuật này có thể áp dụng đối với một ký tự (monoalphabetic) hoặc nhiều ký tự (polyalphabetic) tùy vào mục đích sử dụng.

#### 3.3.1.1. Mã Caesar (Caesar cipher)

Mã Caesar (tên gọi khác, mật mã dịch chuyển) là mã hóa thay thế thuộc dạng monoalphabetic: thay thế mỗi kí tự trong văn bản bằng 1 kí tự khác trong bảng chữ cái với 1 khoảng cách cố định cho trước.

#### Ví dụ 1.1:

Bảng chữ cái tiếng Anh có 26 ký tự: ABCDEFGHIJKLMNOPQRSTUVWXYZ

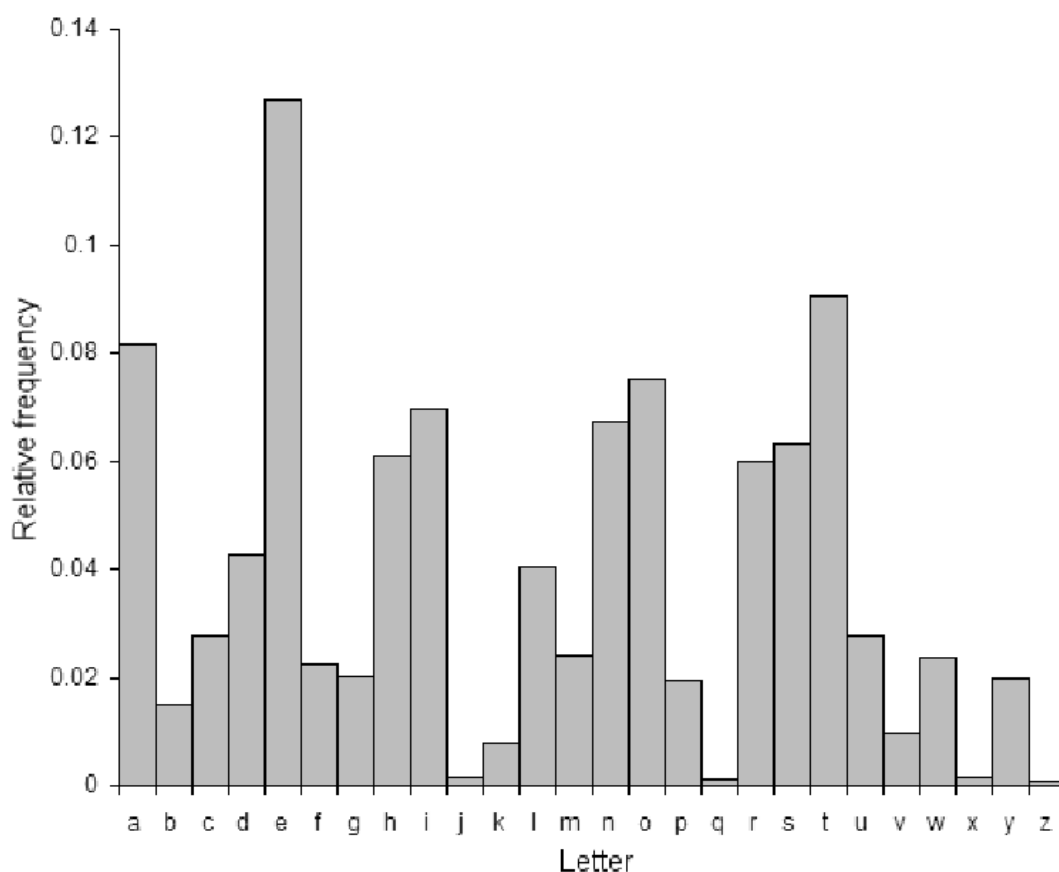
Mã hóa kí tự  $x$  với khoảng cách dịch chuyển 1 đoạn  $n = 13$  (ROT13) theo quy tắc:  
 $y = (x + 13) \bmod 26$ ,

Giải mã:  $x = (y - 13) \bmod 26$ .

Khi đó 1 văn bản gốc chứa thông tin: **GUIDELINES FOR TERM PAPERS** sau khi biến đổi ROT13 có kết quả: **THVQRYVARF SBE GREZ CNCREF**

Để tấn công hệ mật Caesar có thể sử dụng một số kỹ thuật sau:

- Vét cạn (brute-force): thử tất cả các khả năng biến đổi có thể xảy ra để tìm được quy tắc thay thế, do hệ mã Caesar chỉ có 26 ký tự (tương ứng 25 quy tắc - khóa) nên việc giải mã không mất nhiều thời gian trong điều kiện hiện nay.
- Tần số xuất hiện kí tự (Character frequencies): dựa vào thống kê xuất hiện của các kí tự trong bản mã, đối chiếu với bảng tần số được khảo sát trước của từng ngôn ngữ.



Hình 3.2. Bảng thống kê tần số xuất hiện các ký tự trong thư.

### 3.3.1.2. Mã hóa Vigenère Cipher (Vigenère cipher)

Mã hóa Vigenere được hình thành trên mã hóa Caesar có sử dụng khóa (chuỗi các chữ cái) trên văn bản gốc (gồm các chữ cái). Mã hóa Vigenere là sự kết hợp của nhiều phép mã hóa Caesar với các bước dịch chuyển khác nhau. Để mã hóa, sử dụng bảng mã Vigenere (Hình 1.3) với cột dọc là chuỗi khóa (khóa được lặp đi lặp lại để chiều dài tương ứng với văn bản gốc), cột ngang – văn bản gốc, giao giữa ký tự tương ứng cột chứa khóa và văn bản gốc chính là ký tự mã của thuật toán.

**Ví dụ 1.2:** Cho từ khóa “IUH” với văn bản cần mã ở ví dụ 1.1.

Plaintext: **GUIDELINESFORTERMPAPERS**

Key: **IUHIUHIUHIUHIUHIUHIU**

Ciphertext: **OOPLYSQHLAZVZNLZGWIJLZM**

		PLAINTEXT																									
KEY	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	
	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	
	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	
	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	
	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	
	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	



G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	
W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T		
X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	
Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W		

Hình 3.3. Bảng mã Vigenere

Một cách biểu diễn khác về mã hóa Vigenere ở dạng toán học, cho văn bản gốc  $P = p_0, p_1, \dots, p_{n-1}$  và khóa  $K = k_0, k_1, \dots, k_{m-1}$  với  $m < n$ , mỗi ký tự tương ứng với 1 con số (ví dụ: A=0, B=1,...,Z=25). Khi đó bản mã  $C = C_0, C_1, \dots, C_{n-1}$  được tính theo công thức<sup>1</sup>:

$$C = C_0, C_1, \dots, C_{N-1} = E(K, P) = (p_0 + k_0) \bmod 26, (p_1 + k_1) \bmod 26, \dots, (p_{m-1} + k_{m-1}) \bmod 26, \\ (p_m + k_0) \bmod 26, (p_{m+1} + k_1) \bmod 26, \dots, (p_{2m-1} + k_{m-1}) \bmod 26, \dots$$

$$\text{với } C_i = (p_i + k_{i \bmod n}) \bmod 26,$$

$$\text{Giải mã: } p_i = (C_i - k_{i \bmod n}) \bmod 26$$

Độ an toàn của mã hóa Vigenere phụ thuộc vào độ dài của khóa. Khi đó, kẻ tấn công sẽ các định chiều dài của khóa trước khi thực hiện các bước tiếp theo, như việc phân tích tần số cho các bản mã Caesar khác nhau.

Trong trường hợp chiều dài của khóa bằng chiều dài của văn bản gốc (mã Vernam) thì việc bẻ khóa sẽ khó khăn hơn nhiều. Tuy nhiên nếu độ dài văn bản đủ lớn thì kẻ tấn

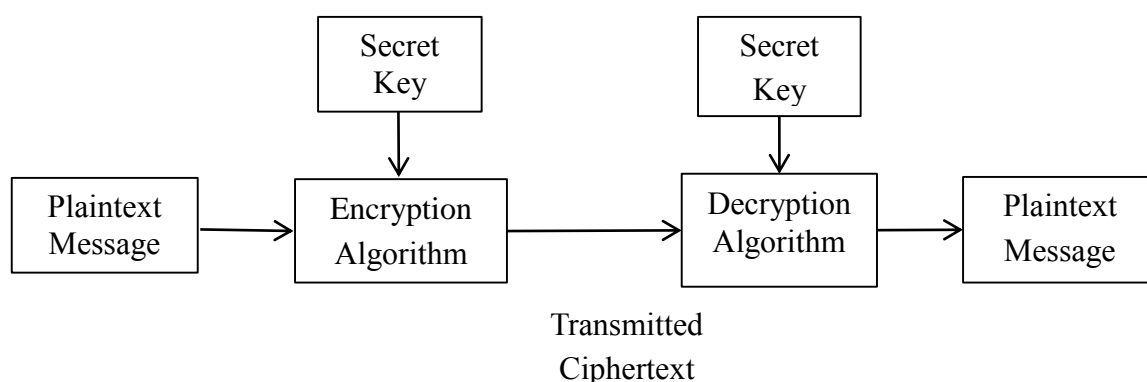
<sup>1</sup> Stallings W. Cryptography and Network security: Principles and Practice, 6<sup>th</sup>. edition, 20

công cũng có thể khai thác các thông tin quan trọng thông qua việc phân tích các tần số bằng nhau.

Để tăng độ an toàn của hệ mã này thì chuỗi khóa cần được tạo ra ngẫu nhiên (One-Time-Pad), trong trường hợp này thì khả năng phá vỡ hệ thống là không thể.

### 3.3.2. Mã hóa đối xứng

#### 3.3.2.1. Sơ đồ mã hóa đối xứng



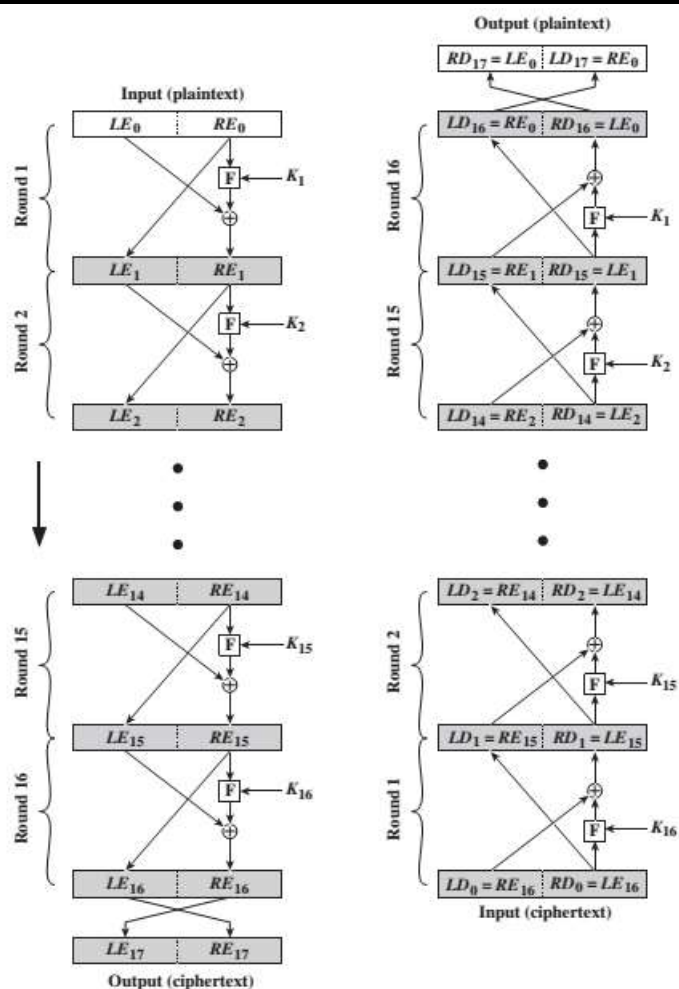
Hình 3.4. Sơ đồ mã hóa đối xứng

Mô hình mã hóa đối xứng (symmetric cipher model) gồm 5 thành phần: plaintext, encryption algorithm, secret key (khóa bí mật chung của bên gửi và bên nhận), ciphertext, decryption algorithm. Trong mô hình trên, khóa của bên gửi (sinh mã) và bên nhận (giải mã) giống nhau và được bảo mật tuyệt đối, do đó hệ thống trên còn có tên gọi khác là hệ mã hóa khóa bí mật.

Một số thuật toán mã hóa đối xứng: DES, AES, Blowfish, RC5, RC6 ...

#### 3.3.2.2. Cấu trúc Feistel<sup>2</sup>

Hầu hết các mã hóa khối sử dụng cấu trúc Feistel để giải quyết vấn đề mã và giải mã, như DES, 3DES, Lucifer, FREAL, Khufu, Khafre, LOKI, GOST, CAST, Blowfish, ... Dữ liệu đầu vào của thuật toán theo khối có với độ dài  $2w$  bit và khóa  $K$ . Khối dữ liệu gốc chia làm 2 phần:  $L_0$  và  $R_0$ , quá trình biến đổi dữ liệu qua  $n$  vòng, đầu ra mỗi vòng biến đổi có cấu trúc khối. Tại vòng  $i$  có dữ liệu đầu vào  $L_{i-1}$  và  $R_{i-1}$  của vòng trước, quá trình biến đổi sẽ sử dụng khóa  $K_i$  (được sinh ra từ khóa  $K$ ,  $K_i \neq K_j$ ,  $i \neq j$ ). Hình 1.4 mô tả quá trình biến đổi mã hóa và giải mã sử dụng cấu trúc Feistel.



Hình 3.5. Mã hóa và giải mã Feistel

Nguyên tắc thiết kế mã hóa Feistel: các yếu tố tỷ lệ thuận với sự an toàn trong mã hóa Feistel bao gồm: kích thước khối (block size), kích thước khóa (key size), số vòng biến đổi (number of rounds), số lượng khóa (subkey)  $K_i$  được sinh ra, sự phức tạp của hàm biến đổi  $F$ .

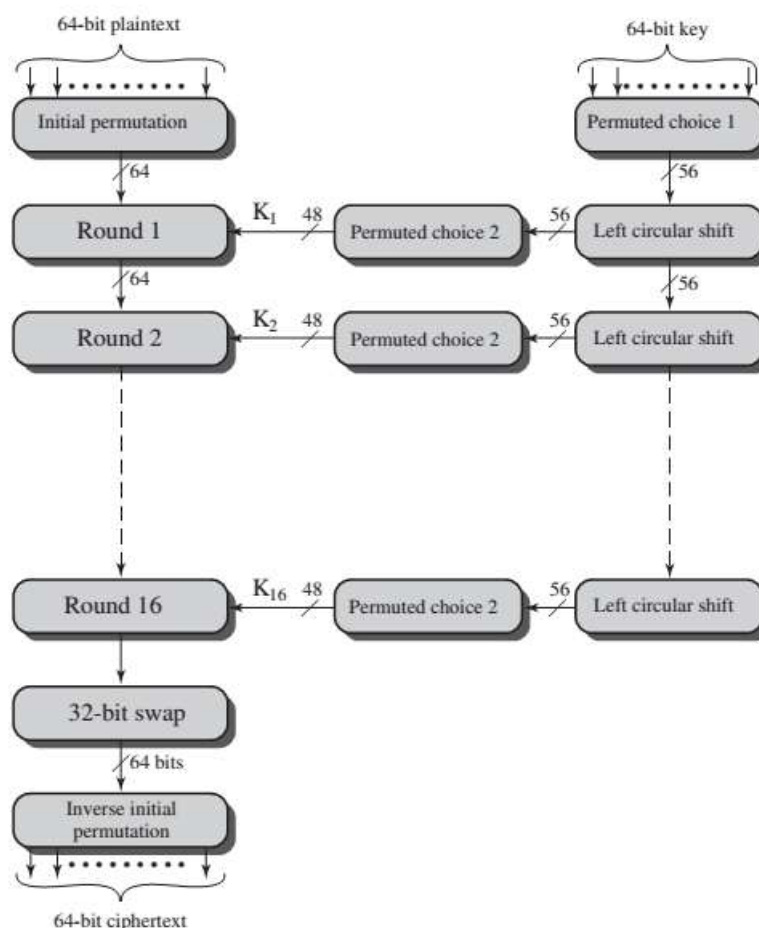
Quá trình giải mã Feistel tương tự như quá trình mã hóa. Quy tắc khái quát như sau: sử dụng bản mã làm đầu vào thuật toán nhưng sử dụng khóa  $K_n$  ở vòng thứ 1,  $K_{n-1}$  ở vòng thứ 2, ... và  $K_1$  ở vòng cuối cùng.

### 3.3.2.3. Tiêu chuẩn mã hóa dữ liệu

Tiêu chuẩn mã hóa dữ liệu (Data Encryption Standard – DES) được giới thiệu vào năm 1977 bởi NBS<sup>3</sup> (hiện nay là NIST<sup>4</sup>) và nhanh chóng được sử dụng phổ biến trên thế giới (như trong lĩnh vực tài chính, ...) trong việc mã hóa dữ liệu theo khối; quá trình mã hóa trên kích thước khối 64 bit, sử dụng khóa có độ dài 56 bit:

<sup>3</sup> National Bureau of Standards

<sup>4</sup> National Institute of Standards and Technology



Hình 3.6. Thuật toán mã hóa DES

Cơ sở của phép biến đổi DES bao gồm:

- Hoán vị dữ liệu ban đầu (initial permutation - IP)
- 16 vòng biến đổi với phép thay thế và hoán vị sử dụng khóa dài 48 bit độc lập nhau.
- Bản mã thu được là sự thay đổi của IP

Mô hình toán học của DES:

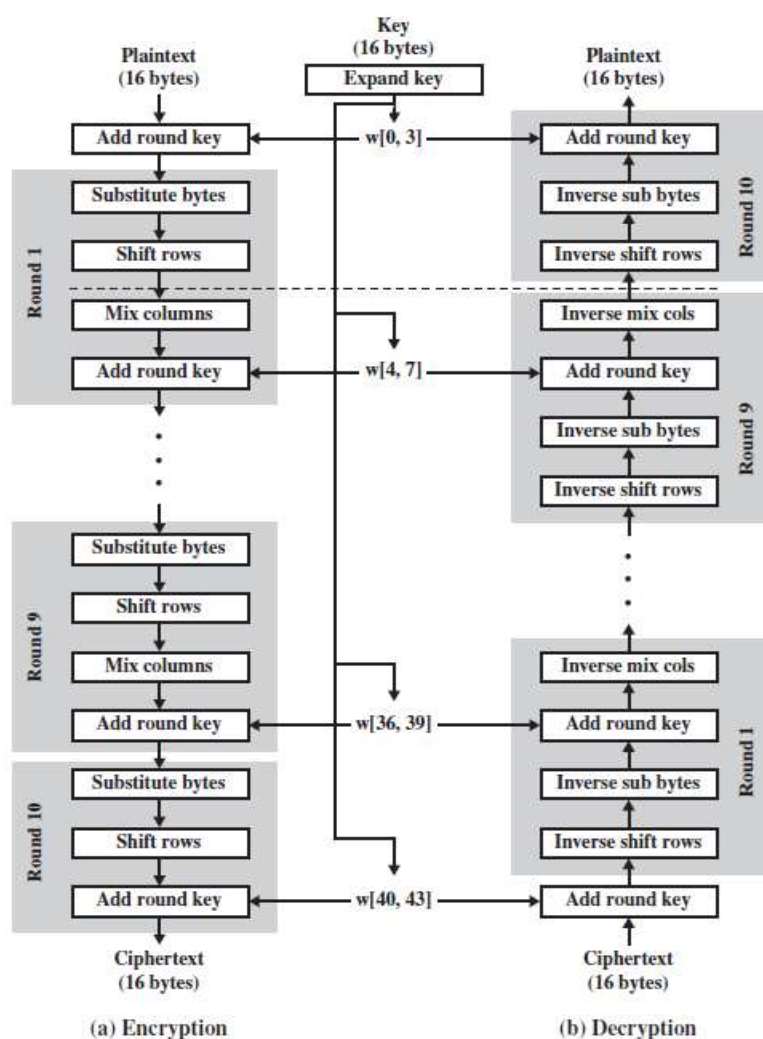
$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$$

Độ an toàn của DES: dựa chủ yếu vào độ dài của khóa, với độ dài khóa 56 bit tương đương với  $2^{56} = 7.2 \times 10^{16}$  giá trị khác nhau. Trong trường hợp này, việc tấn công DES bằng kỹ thuật vét cạn không khả thi với tốc độ của bộ vi xử lý hiện nay, tuy nhiên việc áp dụng phương pháp thống kê (statistical) như kỹ thuật phân tích vi sai (differential), tuyến tính (linear) và khóa liên quan (related key) mang lại hiệu quả rõ rệt.

### 3.3.2.4. Tiêu chuẩn mã hóa tiên tiến

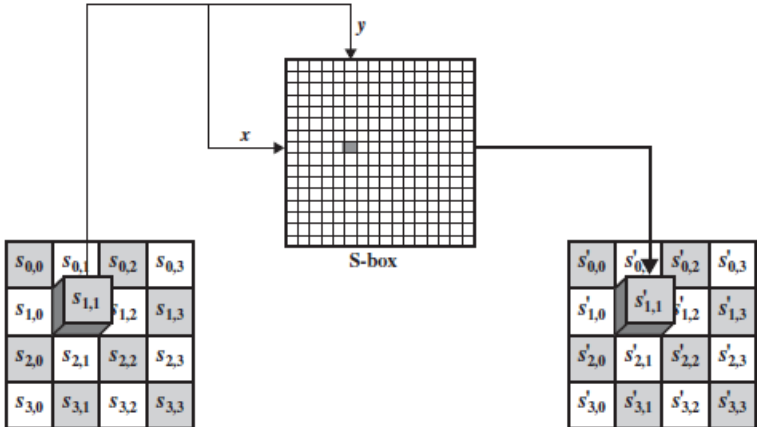
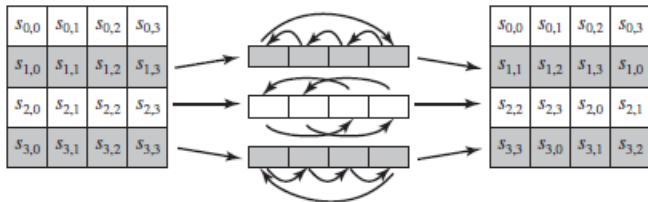
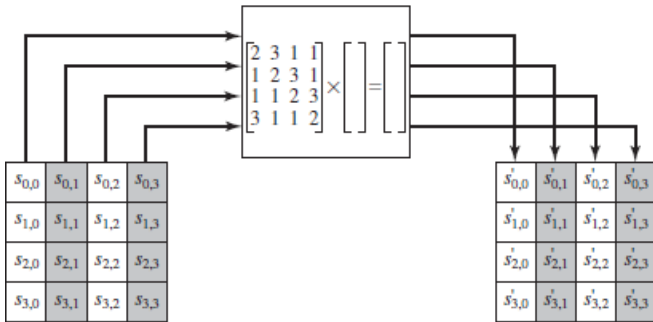
Tiêu chuẩn mã hóa tiên tiến (Advanced Encryption Standard - AES) được giới thiệu vào năm 2001 bởi NIST với mục đích thay thế DES có nhiều hạn chế, và được sử dụng rộng rãi trong các ứng dụng hiện nay, thuật toán được thiết kế bởi Joan Daemen và Vincent Rijmen với tên gọi ban đầu là Rijndael. Trong đó, thuật toán không sử dụng cấu trúc Feistel mà sử dụng mạng thay thế-hoán vị: trong mỗi vòng biến đổi được thay thế bằng bốn hàm chức năng: thay thế byte, hoán vị, phép toán số học trên trường hữu hạn và phép XOR với khóa. Khóa được sử dụng đa dạng với các kích thước khác nhau gồm: 128, 192, 256 bit trên dữ liệu khối 128 bit.



Hình 3.7. Mã hóa và giải mã AES

Mô tả vắn tắt quá trình mã hóa bao gồm 4 bước:

<p>AddRoundKey - mỗi byte của khối được kết hợp với khóa con, các khóa con này được tạo ra</p>	$  \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} \oplus \begin{bmatrix} w_i & w_{i+1} & w_{i+2} & w_{i+3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}  $
--	--

từ quá trình tạo khóa con Rijndael.	<i>Hình 1.7. Add round key transformation</i>
SubBytes - đây là quá trình thay thế trong đó mỗi byte sẽ được thay thế bằng một byte khác theo bảng tra.	 <p><i>Hình 1.8. Substitute byte transformation</i></p>
ShiftRows - đổi chỗ, các hàng trong khối được dịch vòng.	 <p><i>Hình 1.9. Shift row transformation</i></p>
MixColumns - quá trình trộn làm việc theo các cột trong khối theo một chuyển đổi tuyến tính.  Tại chu trình cuối thì MixColumns được thay thế bằng AddRoundKey	 <p><i>Hình 1.10. Mix column transformation</i></p>

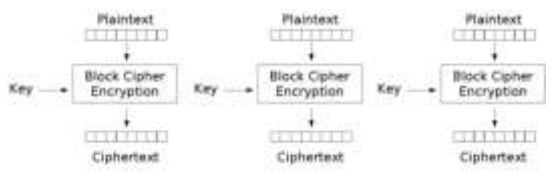
### 3.3.2.5. Các hệ thống mã hóa đối xứng khác

Tên hệ mã hóa	Năm giới thiệu	Đặc điểm
3DES	1999	3 lần biến đổi của DES (tương đương 48 vòng thay thế, hoán vị) với 3 khóa khác nhau, khóa của toàn bộ quá trình 168 bit (3*56 bit)

International Data Encryption Standard (IDEA)	1991	quá trình biến đổi qua 8 vòng, sử dụng khóa 128 bit trên khối dữ liệu 64 bit, được sử dụng trong PGP (Pretty Good Privacy)
Blowfish	1993	với mục đích thay thế DES, khóa sử dụng từ 128 bit đến 448 bit, biến đổi tối đa 16 vòng trên khối dữ liệu 64 bit, blowfish được ứng dụng nhiều trong các phần mềm thương mại
RC4	1987	mã hóa luồng với khóa từ 40 đến 2048 bit, được sử dụng trong truyền thông với SSL (Secure Sockets Layer) và WEP (Wired Equivalent Privacy).
RC5	1994	yêu cầu phần cứng thấp, kích thước khóa đến 2048 bit, xử lý dữ liệu có kích thước khối khác nhau gồm 32 bit, 64 bit, 128 bit với số vòng biến đổi từ 12 đến 255, được sử dụng trong nhiều ứng dụng trong đó có sản phẩm bảo mật dữ liệu RSA.
RC6	1998	mã hóa khối dữ liệu trên RC5 với kích thước khối dữ liệu 128 bit, sử dụng được 3 khóa với kích thước 128 bit, 192 bit, 256 bit. Nhanh và an toàn hơn RC5

### 3.3.2.6. Cơ chế hoạt động (Modes of Operation)

Do mã hóa khối thực hiện trên khối dữ liệu có kích thước cố định (ví dụ: DES với dữ liệu khối 64 bit, sử dụng khóa 56 bit) nhưng trong thực tế lại đòi hỏi phải xử lý dữ liệu có kích thước tùy ý theo bit hoặc theo byte. Bảng 1.x dưới đây mô tả vắn tắt một số cơ chế hoạt động của việc xử lý:

Cơ chế	Mô tả	Đặc điểm
<b>Electronic Codebook Book (ECB)</b> 	Dữ liệu được chia thành các khối độc lập, mỗi khối được xử lý và kết quả nhận được độc lập với nhau sau khi biến đổi. $C_i = E(P_i)$	Thích hợp cho các dữ liệu đầu vào khác nhau. Dữ liệu đầu vào giống nhau sẽ cho kết quả mã hóa giống nhau Một giải pháp giải quyết vấn



		đề trên là thêm khóa để biến đổi dữ liệu đầu vào để mỗi lần biến đổi sẽ có bản mã khác nhau (CBC).
<p>Cipher Block Chaining (CBC)</p>	<p>Mã hóa:</p> $C_0 = IV$ $C_i = E_k(P_i \oplus C_{i-1}),$ $i = 1, \dots, m$ $C = C_0   C_1   \dots   C_m$ <p>Giải mã:</p> $P_i = D_k(C_i) \oplus C_{i-1},$ $i = 1, \dots, m$ $M = P_1   P_2   \dots   P_m$	<p>Mỗi bản mã phụ thuộc vào các thông tin đầu vào khác, do đó khi có sự thay đổi ở 1 khối dữ liệu sẽ dẫn đến sự thay đổi của các bản mã ở sau đó.</p> <p>Giá trị ban đầu IV được gọi trước cho bên gửi và bên nhận</p>
<p>Cipher FeedBack (CFB)</p>	<p>Dữ liệu được xem như dòng bit, thêm vào đầu ra của thuật toán mã hóa khối và kết quả sẽ được đưa làm đầu vào cho mã hóa khối dữ liệu phía sau.</p> $C_i = P_i \oplus E(C_{i-1}),$ $C_{-1} = IV$ <p>Được sử dụng trong mã hóa luồng, chứng thực.</p>	<p>Phù hợp khi xử lý dữ liệu theo bit hoặc theo byte.</p> <p>Hạn chế là việc trì hoãn sau mỗi n bit.</p>

### 3.3.2.7. Ưu và nhược điểm của mã hóa đối xứng

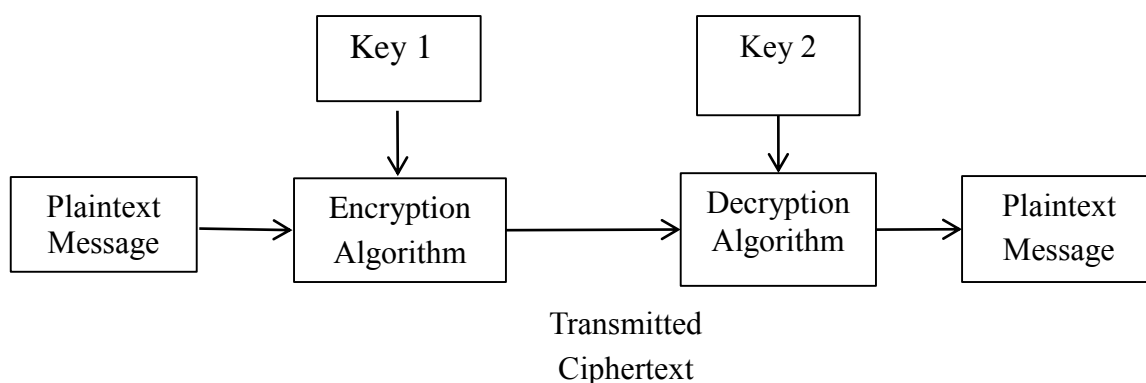
Ưu điểm: độ an toàn cao (phụ thuộc vào thuật toán và khóa), quá trình mã hóa và giải mã nhanh do đó mã hóa đối xứng được sử dụng phổ biến trong việc truyền dữ liệu.

Hạn chế: Một số vấn đề cần quan tâm của hệ thống mã hóa đối xứng liên quan đến khóa, bao gồm:

- Do quá trình mã hóa và giải mã sử dụng chung một khóa nên khóa (secret key) sử dụng cần phải được bảo quản an toàn tuyệt đối.
- Vấn đề phân phối khóa
- Vấn đề quản lý khóa (với hệ thống có  $n$  nút khác nhau thì số lượng khóa cần thiết cho hệ thống là:  $n(n+1)/2$ ).
- Không cung cấp tính chống thoái thác thông tin.

### 3.3.3. Mã hóa bất đối xứng

#### 3.3.3.1. Sơ đồ mã hóa bất đối xứng



Hình 3.8. Sơ đồ mã hóa bất đối xứng

Trong sơ đồ mã hóa bất đối xứng (asymmetric cipher model), khóa bên gửi (key 1) và bên nhận (key 2) khác nhau thuộc 1 bộ khóa, được sử dụng tương ứng cho việc mã hóa và giải mã. Trong đó, một khóa được công khai (public key), khóa còn lại được bảo quản bí mật gọi là khóa cá nhân (private key), do vậy hệ thống này còn có tên gọi khác là hệ mã hóa khóa công khai. Tùy vào mục đích sử dụng mà khóa cá nhân hoặc khóa công khai được sử dụng ở bên gửi hay bên nhận.

Có thể biểu diễn quá trình mã và giải mã ở hệ mã hóa bất đối xứng như sau:

- Mã hóa:  $C = E_{K_1}(P)$
- Giải mã:  $P = D_{K_2}(C)$

Trong đó  $P, C, E, D, K_1, K_2$  lần lượt là kí hiệu của văn bản gốc (Plaintext message), bản mã (Ciphertext), thuật toán mã hóa (Encryption algorithm), thuật toán giải mã (Decryption algorithm), bộ khóa (Key pair)  $K = \{K_1, K_2\}$ .

Hiện nay, mã hóa khóa công khai được ứng dụng rộng rãi trong nhiều lĩnh vực, trong đó bao gồm: trao đổi, phân phối khóa, chữ ký số, bảo mật dữ liệu.

Một số hệ mã hóa bất đối xứng: Merkle-Hellman, El-Gamal, RSA, ECC, Paillier

...

### 3.3.3.2. Hệ mã hóa Diffie-Hellman

Mã hóa Diffie-Hellman được giới thiệu vào năm 1976 bởi Diffie Whitfield và Martin Hellman. Ứng dụng của mã hóa Diffie-Hellman tập trung vào mục đích trao đổi khóa (khóa bí mật được trao đổi bằng giao thức Diffie-Hellman, sau đó được sử dụng trong các giao thức truyền tin sử dụng mã hóa đối xứng cho mục đích an toàn dữ liệu). Một số ứng dụng cụ thể: SSL, VPN, SSH.

Giao thức trao đổi khóa Diffie-Hellman giữa A, B gồm các bước sau:

- A, B thỏa thuận số nguyên tố  $q$  và phần tử sinh của  $q$ :  $\alpha$
- Tạo khóa của A, B: A (B) chọn giá trị bí mật  $X_A$ ,  $X_B$  ( $X_A, X_B < q$ ), tính toán giá trị công cộng  $Y_A = \alpha^{X_A} \bmod q$  ( $Y_B = \alpha^{X_B} \bmod q$ ). Giá trị  $Y_A$  ( $Y_B$ ) được gửi qua cho bên B (tương ứng A).
- Giá trị khóa bí mật chung của A, B được tính toán theo công thức tương ứng:

$$K = (Y_B)^{X_A} \bmod q,$$

$$K = (Y_A)^{X_B} \bmod q.$$

Giao thức trao đổi khóa sử dụng thuật toán Diffie-Hellman không cung cấp tính chứng thực thông tin giữa A, B. Do đó trong thực tiễn, giao thức này bị tấn công bởi kiểu tấn công chèn giữa (man-in-the-middle attack).

### 3.3.3.3. Hệ mã hóa ElGamal

Vào năm 1984, T. ElGamal giới thiệu hệ mã hóa khóa công khai trên cơ sở ý tưởng từ Diffie-Hellman. Hệ mã hóa ElGamal được sử dụng trong việc mã hóa dữ liệu, chữ ký số, trao đổi khóa.

Quá trình tạo khóa của A sử dụng hệ ElGamal gồm các bước chính sau:

- A, B thống nhất số nguyên tố  $q$  và phần tử sinh  $q$ :  $\alpha$
- Bên tạo khóa (A) chọn giá trị bí mật  $X_A$  ( $X_A < q-1$ ) và tính giá trị  $Y_A = \alpha^{X_A} \bmod q$ . Khi đó, bộ khóa  $K = \{PU, PR\}$  của A, với khóa công khai  $PU = \{q, \alpha, Y_A\}$  và khóa cá nhân  $PR = \{X_A\}$ .

Quá trình B sử dụng bộ khóa của A trong việc truyền dữ liệu  $M$  ( $M < q$ ):

- B chọn giá trị  $k$  ( $k < q$ ) và tính toán khóa  $K = (Y_A)^k \bmod q$ ,  $C_1 = \alpha^k \bmod q$ ,  $C_2 = KM \bmod q$ . Khi đó,  $(C_1, C_2)$  là bản mã được truyền đi.

Quá trình bên nhận (A) giải mã:

- Tính khóa  $K = (C_1)^{X_A} \bmod q$
- Tìm bản gốc theo công thức:  $M = (C_2 K^{-1}) \bmod q$ .

### 3.3.3.4. Hệ mã hóa RSA

Năm 1978, Ron Rivest, Adi Shamir và Len Adleman giới thiệu hệ mã hóa bất đối xứng với tên gọi RSA. Hệ mã hóa RSA được ứng dụng rộng rãi trong các ứng dụng hiện nay trong trao đổi khóa, mã hóa dữ liệu, chữ ký số.

Quá trình tạo bộ khóa  $K = \{PU, PR\}$  theo RSA theo các bước sau:

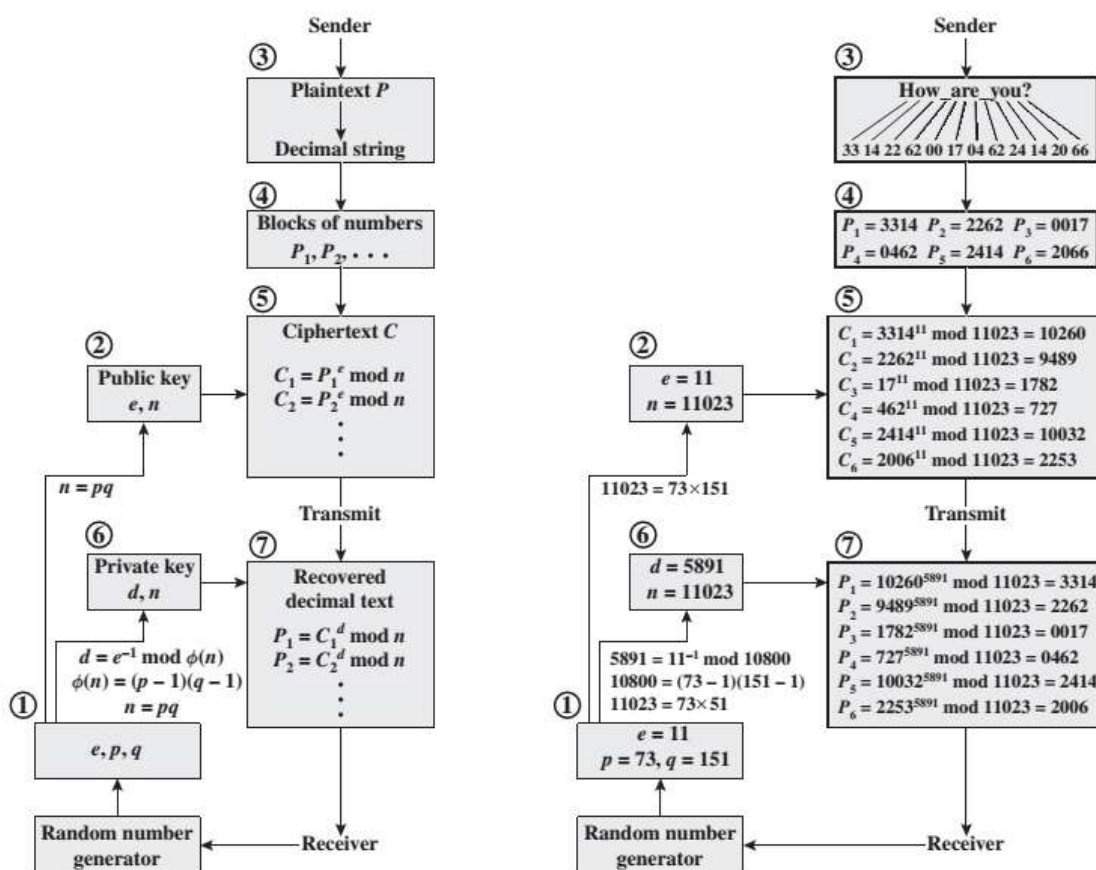
- Chọn số nguyên tố  $p, q$  khác nhau.
- Tính  $n = pq$ ,  $\phi(n) = (p-1)(q-1)$
- Chọn số nguyên  $e$  thỏa mãn:  $\text{UCLN}(e, \phi(n)) = 1$ ,  $1 < e < \phi(n)$
- Tìm  $d$ :  $d \equiv e^{-1} \bmod \phi(n)$
- Kết quả: khóa công khai  $PU = \{e, n\}$ , khóa cá nhân  $PR = \{d, n\}$ .

Quá trình mã hóa dữ liệu  $M$ , ( $M < n$ ) theo sử dụng khóa công khai theo công thức:

$$C = M^e \bmod n$$

Quá trình giải mã  $C$  sử dụng khóa cá nhân:

$$M = C^d \bmod n$$



Hình 3.9. Quá trình xử lý của RSA và ví dụ minh họa

Độ an toàn của RSA dựa vào độ phức tạp của bài toán phân tích một số nguyên dương cho trước  $n$  thành hai thừa số nguyên tố  $p, q$ . Do đó để tăng độ an toàn của RSA khi khởi tạo khóa người ta tăng kích thước các số nguyên tố  $p, q$ .

### 3.3.3.5. Ưu, nhược điểm của hệ mã hóa khóa công khai

**Ưu điểm:** Độ an toàn của mã hóa bất đối xứng cao, cung cấp được tính chứng thực, toàn vẹn dữ liệu.

**Hạn chế:** Do số phép tính toán trong quá trình biến đổi lớn nên tốc độ xử lý dữ liệu trong các hệ mã hóa chậm. Vì vậy, trong thực tiễn hệ mã hóa bất đối xứng ít được sử dụng cho việc truyền dữ liệu kích thước lớn.

## 3.4. Câu hỏi và bài tập

1. Trình bày ưu điểm, hạn chế của mã hóa đối xứng
2. Trình bày ưu điểm, hạn chế của mã hóa bất đối xứng
3. Giải thích tại sao mật mã luồng không bảo vệ được tính toàn vẹn dữ liệu
4. Nêu nguyên tắc của mã hóa khóa công khai? Tại sao khi truyền khóa sử dụng mã hóa khóa công khai không cần sử dụng kênh an toàn?
5. Điểm yếu của sơ đồ trao đổi khóa Diffie-Helman là gì? Giải thích?
6. Cho  $p = 23, q = 19, e = 283$ . Tìm số nguyên dương  $d$  sao cho  $d \cdot e = 1 \pmod{(p-1)(q-1)}$ .
7. Trình bày độ an toàn của hệ mã hóa khóa công khai RSA.
8. Tại sao số  $n = 2047$  không được chọn trong hệ mã hóa RSA?
9. Phân tích số  $n = 4386607$  thành các thừa số nguyên tố  $p, q$ , biết rằng  $\phi(n) = (p-1)(q-1) = 4382136$ .
10. Cho  $p = 11, q = 3, e = 3$ . Trình bày từng bước theo hệ mã hóa RSA thực hiện các yêu cầu sau
  - a. Tính bộ khóa
  - b. Alice muốn gửi message  $M = 13$  cho Bob, sử dụng bộ khóa của Bob thu được ở câu a để thực hiện việc mã hóa và giải mã.
  - c. Vẽ sơ đồ minh họa quá trình truyền dữ liệu với các tham số đã có ở câu b.
11. Cho bảng mã morse như sau

$A \mapsto \bullet - *$	$B \mapsto - \bullet \bullet \bullet *$	$C \mapsto - \bullet - \bullet *$
$d \mapsto - \bullet \bullet *$	$E \mapsto \bullet *$	$F \mapsto \bullet \bullet - \bullet *$
$O \mapsto - - - *$	$S \mapsto \bullet \bullet \bullet *$	$7 \mapsto - - \bullet \bullet \bullet *$

Thực hiện decode:  $- \bullet - \bullet * - - - * - \bullet \bullet \bullet *$

- 
12. Áp dụng thuật toán mật mã VIGENERE và mã Caesar để giải mã (Tìm X):  
VIGENERE (ROT2(X), IUH)=SSJXPCYXNKLAYCAKIVON
13. Cho bảng mô tả dưới đây gồm mục (a): thông tin gửi từ A đến B, Hãy xác định nội dung được mô tả ở mục (b) phù hợp và giải thích.  
(a) E(K, M), E(PU<sub>b</sub>, M), E(PR<sub>a</sub>, M), E(PU<sub>b</sub>, E(PR<sub>a</sub>, M))  
(b) Bảo mật, Chứng thực, Ký số.
14. Khái niệm độ phức tạp tính toán thay thế cho khái niệm thời gian khi đề cập đến an toàn của hệ thống mật mã. Giải thích độ phức tạp tính toán cung cấp như thế nào về cơ sở lý thuyết cho việc thiết kế các hệ thống mật mã hiện đại.
15. Mật mã Feistel được sử dụng trong thuật toán DES. Mô tả hoạt động của mật mã Feistel.
16. Mô tả ngắn gọn ba phương thức hoạt động của DES.
17. Độ phức tạp tính toán của một vấn đề được xem là số bước nguyên thủy của một số mô hình tính toán để giải quyết vấn đề đó. Xác định độ phức tạp của các vấn đề có giới hạn thời gian tính toán đa thức (P).
18. Cho một số ví dụ tình huống áp dụng mật mã để bảo vệ an toàn thông tin.
19. Alice muốn gửi thông tin cho Bob, Eve là kẻ tấn công. Alice và Bob đồng ý sử dụng mã hóa đối xứng. Quá trình trao đổi khóa bí mật K đã được diễn ra. Hãy phác thảo các bước chính mà Alice và Bob đã thực hiện trong quá trình mã hóa và giải mã.
20. Thuật toán thỏa thuận khóa Diffie-Hellman cho phép hai máy chủ tạo ra một bí mật chung.  
a) Giải thích hoạt động của giao thức trao đổi khóa Diffie-Hellman?  
b) Giải thích tại sao giao thức Diffie-Hellman cơ bản không cung cấp bất kỳ sự đảm bảo nào về phía bên kia mà giao thức được sử dụng?
21. Alice có khóa công khai RSA ( $n = 65; e = 5$ ).  
a) Hãy cho biết khóa cá nhân của Alice?  
b) Alice nhận được một thông tin mã hóa từ Bob có giá trị 19. Cho biết giá trị bản rõ ban đầu mà Bob muốn gửi với các tham số từ câu a).
22. Bob sử dụng hệ mã hóa El-Gamal với số nguyên tố  $p = 17$ ,  $\alpha = 3$ .  
a) Hãy tính khóa công khai của Bob trong trường hợp khóa cá nhân bằng 9.  
b) Alice sử dụng khóa công khai của Bob (câu a) và gửi cho Bob một bản mã (13;11). Hãy sử dụng khóa cá nhân của Bob để giải mã thông điệp trên.  
c) Chỉ ra một số ưu điểm và hạn chế của hệ mã hóa El-Gamal với RSA.
23. Các phát biểu sau là đúng hay sai? Giải thích.  
a) Về mặt lý thuyết, nếu khoá thực sự là ngẫu nhiên, không được sử dụng lại và được giữ bí mật thì DES và AES đều có thể được an toàn chống lại các cuộc tấn công bằng văn bản.  
b) Cấu trúc Feistel cho phép sử dụng cùng phần cứng hoặc phần mềm đã mã hóa và giải mã.
-

- c) Tất cả mật mã khối sử dụng S-box và hoán vị P-box.
- d) Trao đổi khóa Diffie-Hellman là sơ đồ mã hóa bất đối xứng có thể được sử dụng cho việc mã hóa và chữ ký số, tuy nhiên không hiệu quả như RSA.

24. Alice muốn gửi cho Bob một thông điệp bảo mật  $m$  sử dụng RSA nhưng không biết khóa công khai của Bob. Do đó, Bob gửi cho Alice khóa công khai  $(e, N)$  của mình qua đường email. Tuy nhiên, sự cố xảy ra trong quá trình truyền dẫn đến Alice nhận được khóa  $(e', N)$ , với  $e'$  khác  $e$  một bit. Alice mã hóa  $m$  sử dụng khóa nhận được và gửi cho Bob. Bob nhận được bản mã nhưng không thể giải mã, do vậy Bob gửi lại cho Alice khóa công khai và yêu cầu Alice thực hiện gửi lại bản mã một lần nữa. Toàn bộ quá trình trao đổi giữa Alice và Bob bị attacker nghe lén. Attacker có thể khôi phục lại bản rõ  $m$  hay không? Giải thích.

25. Alice sử dụng RSA để thực hiện việc mã hóa dữ liệu, do đó Alice thực hiện tạo bộ khóa với các số nguyên tố  $p, q$ . Kết quả bộ khóa của Alice thu được  $PU_A = \{e_A = 3, N\}$ ,  $PR_A = \{d_A, N\}$  với  $N = p \cdot q$ . Bob cũng muốn sử dụng hệ RSA cho việc mã hóa thông tin được an toàn, tuy nhiên Bob không biết cách tạo bộ khóa, do vậy Alice đã tạo giúp cho Bob bộ khóa  $PU_B = \{e_B = 5, N\}$ ,  $PR_B = \{d_B, N\}$ . Tom sử dụng khóa công khai của Bob và Alice để mã hóa văn bản  $m$  và gửi bản mã  $c_A, c_B$  đồng thời cho họ. Trong lúc gửi qua đường công cộng thì bản mã  $c_A, c_B$  bị kẻ tấn công nghe trộm. Hãy cho biết kẻ tấn công có thể khôi phục lại bản rõ  $m$  hay không, cho rằng  $UCLN(m, N) = 1$ .

## CHƯƠNG 4: HÀM BẮM VÀ ỨNG DỤNG

### 4.1. Định nghĩa

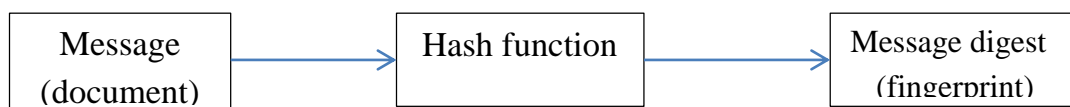
Hàm  $f$  là **hàm một chiều** (one-way function) nếu:

- cho  $x$  dễ dàng tính được  $f(x)$
- cho  $f(x)$  khó tìm được  $x$ .

Hàm  $f$  là **hàm của lật một chiều** (one-way trapdoor) nếu:

- cho  $x$  dễ dàng tính được  $f(x)$
- cho  $f(x)$  khó tìm được  $x$
- nếu có thêm thông tin (“trapdoor” information) thì dễ dàng tính được  $x$  từ  $f(x)$ .

Hàm băm (Hash function)  $H$  là thuật toán một chiều biến đổi dữ liệu đầu vào  $M$  tùy ý, đưa ra kết quả có kích thước cố định  $h = H(M)$ . Kết quả này được gọi là mã băm (hash code), kết quả băm (hash result) hoặc giá trị băm (hash value). Độ dài của giá trị băm phụ thuộc vào thuật toán sử dụng.



Data Format: Text string	Digits: cryptography
<input type="checkbox"/> HMAC	Key Format: Text string
<input checked="" type="checkbox"/> MD5	e0d00b9f337d357c80a2f8c0a4e60d
<input checked="" type="checkbox"/> MD4	ed6d864c848e61b8c9e853bc76a300a
<input checked="" type="checkbox"/> SHA1	48c910b6614c4a0a5851aa78571dd1e3c3e66ba
<input checked="" type="checkbox"/> SHA256	e06554818e902b4ba339066967c0000da3cda4d7eb4ef89c1e
<input checked="" type="checkbox"/> SHA384	e6026b9973d05353067070c57410ba5614773c4fed0e92d4712
<input checked="" type="checkbox"/> SHA512	cd700ec1a9830c273b5c40de34829a0a427294e41c3dfc24359
<input checked="" type="checkbox"/> RIPEMD160	e7892b45c7611640d356549d3d58c9ecb8d9e8c
<input checked="" type="checkbox"/> PANAMA	8999d30e83c0630e98bc0461326eb45abe792e34f3ad5001e58
<input checked="" type="checkbox"/> TIGER	c2c0eaa9028d98e0ed785be6d00e5423572498cd818c4e
<input checked="" type="checkbox"/> MD2	8ca6eb221b76c113e24411a0d8cf963
<input checked="" type="checkbox"/> ADLER32	21aa052d
<input checked="" type="checkbox"/> CRC32	e6390108

Data Format: File	Digits: C:\Users\Administrator\Desktop\playcn.txt
<input type="checkbox"/> HMAC	Key Format: Text string
<input checked="" type="checkbox"/> MD5	25733600ee6c5332ecd0c927ac016
<input checked="" type="checkbox"/> MD4	1d3c9e61c54318055dcd0d6d5b982e8c
<input checked="" type="checkbox"/> SHA1	6ec43990f61cd06403a198147a39925edc335fcb
<input checked="" type="checkbox"/> SHA256	3088d9c9acd2377202be8cd579df337d4ac2f01a867ef73e25e7
<input checked="" type="checkbox"/> SHA384	0d74b4489727d638bf4e4c445e242cb43950cf5340f71aeb807c
<input checked="" type="checkbox"/> SHA512	e60753e4e6b9efe645ec72a2e86c2ab6d86f2fd90638df51nde6e
<input checked="" type="checkbox"/> RIPEMD160	5745d3866336b1cc9d5d05fec7e96c5cd783c4b7
<input checked="" type="checkbox"/> PANAMA	8a1e344cbbc41f1bbec20ef670224fd45cb3cd83ac9a3ab9d3
<input checked="" type="checkbox"/> TIGER	763ebb706f17c365968b980ca0701e95b8b20bd10482c057
<input checked="" type="checkbox"/> MD2	feee53d8006e1bfc77e9c2231482191a
<input checked="" type="checkbox"/> ADLER32	411f12cd
<input checked="" type="checkbox"/> CRC32	b0cc3142

Một số yêu cầu đối với hàm băm:

- Với dữ liệu đầu vào là  $M$  thì kết quả băm  $h = H(M)$  là duy nhất;
- Tính một chiều (one-way property) cho trước giá trị băm  $h$  thì việc tìm  $M$  để  $h = H(M)$  là rất khó;



- Tính kháng đụng độ yếu (weak collision resistance): không thể tìm được dữ liệu  $M_2$  có cùng giá trị băm với dữ liệu  $M_1$  cho trước;
- Tính kháng đụng độ mạnh (strong collision resistance) Việc tìm 2 thông điệp  $M_1, M_2$  khác nhau nhưng có cùng giá trị băm rất khó xảy ra.

## 4.2. Một số hàm băm thông dụng

Hai hàm băm phổ biến hiện nay là MD-X và SHA-X.

**MD-X** (Message Digest gồm MD1, MD2, ..., MD5) trong đó MD5 được sử dụng rộng rãi trong các ứng dụng, xử lý dữ liệu theo từng khối dữ liệu đầu vào 512 bit cho kết quả 128 bit ở kết quả xử lý cuối cùng. Hiện nay, MD5 không còn được coi là an toàn

**SHA-X** (Secure hash algorithm: SHA0, SHA1, SHA256...) trong đó SHA1 (tên gọi khác SHA160) được chính phủ Mỹ sử dụng và chọn làm tiêu chuẩn quốc gia. Bảng băm SHA1 có kích thước 160 bit, quy trình xử lý tương tự như MD5 nhưng số vòng biến đổi nhiều hơn.

## 4.3. Ứng dụng của hàm băm

Vai trò cơ bản của hàm băm mật mã: giá trị băm đóng vai trò như đại diện của dữ liệu đầu vào, do đó hàm băm được sử dụng rộng rãi trong các ứng dụng:

### • Lưu trữ mật khẩu

Hầu hết các ứng dụng trên phần mềm hiện nay đều yêu cầu chứng thực khi làm việc, trong đó giao thức chứng thực bằng username/password được sử dụng phổ biến hơn cả. Do vậy, việc bảo vệ an toàn của mật khẩu luôn được xem trọng.

Mật khẩu bao gồm chuỗi các chữ cái (hoa, thường), chữ số và các ký tự đặc biệt (@, # ...).

Để đảm bảo an toàn, khi lưu trữ trong cơ sở dữ liệu mật khẩu sẽ được biến đổi để tránh vấn đề xem trộm, sao chép mật khẩu. Một trong những phương pháp hiệu quả được áp dụng là sử dụng hàm băm, khi đó thay vì mật khẩu được lưu trữ trong hệ thống thì được biến đổi và lưu trữ giá trị băm.

Username	Password
admin	@123!FitIuh
trandung	@123!Tran
<i>Lưu trữ không mã hóa mật khẩu</i>	

Username	Password
admin	69c919ee4881666e4c90d51d4a2ed505
trandung	168838fe639bd5d8b660d5b008978759
<i>Lưu trữ mã hóa mật khẩu với MD5</i>	

Do tính chất của hàm toán học một chiều, mật khẩu của tài khoản được bảo vệ ngay cả trong trường hợp file lưu trữ mật khẩu hệ thống bị sao chép.

### • Xác thực tính nguyên vẹn dữ liệu

Một trong những ứng dụng phổ biến khi truyền hoặc tải file từ internet, với dữ liệu tải lớn có thể xảy ra sai sót trên đường truyền (file ban đầu bị mất mát dữ liệu hoặc bị chèn bổ sung các mã độc ...), việc sử dụng hàm băm trong trường hợp này giúp người sử dụng có thể tự kiểm tra và phát hiện được các lỗi tương tự. Khi đó, bên gửi công bố

kết quả băm của dữ liệu gốc M cho bên nhận biết, bên nhận tự tính toán giá trị băm của M theo thuật toán thỏa thuận sẵn và so sánh kết quả băm so với bảng băm bên gửi đã công bố. Nếu dữ liệu trùng khớp thì M nhận được toàn vẹn và ngược lại.

**Download (only 249KB):**

[WinMD5 Freeware Download](#)

WinMD5Free.zip MD5: 73f48840b60ab6da68b03acd322445ee

WinMD5Free.exe MD5: 944a1e869969dd8a4b64ca5e6ebc209a

You may simply download it, then unzip and put the exe to any folder on your hard drive, and start to use.

*Hình 4.1. Minh họa thông tin khi tải file*

Trong nhiều trường hợp, người ta truyền thêm khóa bí mật K vào hàm băm, khi đó kết quả tính toán được gọi là HMAC:  $HMAC = H(M \parallel K)$ .

- Tham gia vào quá trình tạo chữ ký số (*chi tiết ở chương 5*).

#### 4.4. Tấn công hàm băm

Theo nguyên lý Dirichlet, không gian của giá trị băm nhỏ hơn không gian dữ liệu đầu vào nên chắc chắn tồn tại sự đụng độ (collision). Tấn công hàm băm bảo mật là tạo ra các tình huống đụng độ, được thực hiện bằng hai kỹ thuật: vét cạn (brute-force) và phân tích mã (cryptanalysis).

**Kỹ thuật tấn công vét cạn:** kẻ tấn công tạo ra một lượng lớn các văn bản và lần lượt tính toán, so sánh giá trị băm của chúng để tìm ra đụng độ. Trên thực tế, kích thước bảng băm theo các thuật toán thông dụng là 64 bit, 128 bit ..., do đó thời gian tính toán để tìm được đụng độ theo phương pháp là rất lớn. Chẳng hạn, MD5 cho bảng băm có kích thước 128bit thì để chắc chắn xảy ra đụng độ thì cần  $2^{128}$  dữ liệu đầu vào, dẫn đến số phép tính toán bảng băm và so sánh để tìm đụng độ.

Để giảm bớt số lượng phép tính (giảm thời gian tính toán), người ta thử với một số lượng văn bản đầu vào ít hơn, khi đó xuất hiện vấn đề xác suất thành công của việc tấn công có giảm nhiều không? Để trả lời câu hỏi đó, người ta dựa vào Nghịch lý ngày sinh nhật (Birthday Paradox). Phát biểu nghịch lý ngày sinh nhật như sau “*Cần bao nhiêu sinh viên để xác suất trùng ngày sinh nhật  $\approx 50\%$* ”<sup>6</sup>. Đáp án bài toán trên là 23 sinh viên dựa vào kỹ thuật tính toán xác suất. Bản chất của nghịch lý ngày sinh nhật đề khẳng định trong nhiều trường hợp xác suất để xảy ra đụng độ là lớn, điều đó dẫn đến mối đe dọa trong việc tấn công hàm băm mật mã khi thử 1 số lượng dữ liệu đầu vào là nhỏ hơn so với không gian băm. Tuy nhiên, việc nghiên cứu tấn công hàm băm dựa vào nghịch lý ngày sinh nhật cũng cho phép tính được kích thước tối thiểu bảng băm của một thuật toán để có thể chống lại các cuộc tấn công theo phương pháp tương tự một cách an toàn.

<sup>6</sup> Prof. Emil Simion, The Birthday paradox, 2012

**Kỹ thuật phân tích mã:**

Hạn chế của kỹ thuật tấn công vét cạn là số phép tính lớn, khi độ dài của bảng băm tương đối lớn ( $\geq 128$  bit) thì việc tìm và chạm mất rất nhiều thời gian. Hiện nay có nhiều phương pháp cho ta kết quả tốt hơn kỹ thuật vét cạn, có thể kể đến như:

- Tấn công theo kiểu gặp nhau ở giữa (meet – in – the – middle attack)
- Tấn công khác biệt giữa các module (The modular differential attack)
- Boomerang Attacks ...

**4.5.Câu hỏi và bài tập**

1. Định nghĩa hàm một chiều. Cho ví dụ minh họa.
2. Định nghĩa hàm băm một chiều. Cho ví dụ minh họa.
3. Định nghĩa hàm cửa lật một chiều. Cho ví dụ minh họa.
4. Để bảo đảm tính chứng thực dùng mã hóa đối xứng hay mã hóa khóa công khai, bản rõ phải có tính chất gì? Tại sao?
5. Nếu bản rõ là một dãy bit ngẫu nhiên, cần làm gì để bản rõ trở thành có cấu trúc?
6. Sử dụng MAC để chứng thực có ưu điểm gì so với chứng thực bằng mã hóa đối xứng?
7. Về mặt lý thuyết, giá trị Hash có thể trùng không? Vậy tại sao nói giá trị Hash có thể xem là “dấu vân tay của thông điệp”?
8. Tại sao để chứng thực một thông điệp  $M$ , người ta chỉ cần mã hóa khóa công khai giá trị Hash của  $M$  là đủ? Thực hiện như vậy có lợi ích gì hơn so với cách thức mã hóa toàn bộ  $M$ ?
9. Tìm hiểu phương pháp sử dụng hàm băm MD5 và SHA trong thư viện .NET, viết chương trình mã hóa password lưu trữ và kiểm tra password.
10. Giả sử Alice và Bob muốn tung đồng xu qua mạng (Alice tung và Bob đoán). Giao thức thực hiện như sau:

- i. Alice chọn giá trị  $X=0$  hay  $1$ .
- ii. Alice sinh một khóa  $K$  ngẫu nhiên gồm 256 bit
- iii. Dùng AES, Alice tính  $Y = E(X||R, K)$  trong đó  $R$  gồm 255 bit bất kỳ
- iv. Alice gửi  $Y$  cho Bob
- v. Bob đoán  $Z$  là  $0$  hay  $1$  và gửi  $Z$  cho Alice
- vi. Alice gửi khóa  $K$  cho Bob để Bob tính  $X||R = D(Y, K)$
- vii. Nếu  $X=Z$ , Bob đoán trúng. Nếu không Bob đoán sai.

Chúng tỏ rằng Alice có thể lừa Bob (chẳng hạn, Alice chọn  $X=1$ , thấy Bob đoán  $Z=1$  thì Alice sẽ lừa như thế nào để Bob giải mã  $Y$  thì có  $X=0$ ). Dùng hàm hash, hãy sửa đoạn giao thức trên để Alice không thể lừa được.

11. Nếu  $f(x)$ ,  $g(x)$  là hàm một chiều, thì  $f(x) \cdot g(x)$ ,  $f(x) + g(x)$  có phải là hàm một chiều không? Chứng minh.
12. Trình bày HMAC và giải thích mục đích sử dụng HMAC.
13. Giải thích cách hàm băm một chiều được sử dụng cho chứng thực tin nhắn.
14. Cho bảng băm của MD5: e10adc3949ba59abbe56e057f20f883e, trình bày một số kỹ thuật để tìm dữ liệu ban đầu.
15. Trong việc lưu trữ mật khẩu bằng MD5 của 1 ngân hàng sử dụng 6 số, và “salt” có độ dài 2 số. Attacker thu nhận được bảng băm: 8a1f5645e4bba257a46380e713ca1831 và salt=12. Cho biết mật khẩu đã được sử dụng.
16. Trình bày sự khác biệt giữa khả năng chống va chạm yếu và mạnh của hàm băm.
17. Kết quả băm của hàm  $h$  có độ dài 10 bit. Tính xác suất tồn tại có 2 bảng băm trùng nhau trong số 7 giá trị đầu vào khác nhau.
18. Cho rằng một hàm băm được sử dụng cho HMAC và các lỗ hổng đã được tìm thấy trong hàm băm, do đó HMAC không an toàn. Những thay đổi nào cần thiết trong HMAC để làm cho nó trở lại an toàn?
19. Cho giao thức trong đó người gửi thực hiện các hoạt động được mô tả:  
$$y = e_{k_1}(x \parallel H(k_2 \parallel x)),$$
trong đó lần lượt các ký hiệu:  $x$  : thông điệp,  $H$  : hàm băm,  $e$  : thuật toán mã hóa đối xứng,  $\parallel$ : phép nối các chuỗi,  $k_1, k_2$  : các khóa bí mật của bên gửi và bên nhận.
  - a) Hãy mô tả từng bước những hoạt động ở phía bên nhận cần thực hiện.
  - b) Xác định giao thức trên có cung cấp tính bảo mật, tính toàn vẹn và tính chống thoái thác hay không? Giải thích?
20. Cho hàm băm  $h(m_1, m_2) = m_1^e m_2^e \bmod(pq)$ , với  $p, q$  là các số nguyên tố trong hệ mật RSA,  $e$  có nghịch đảo theo modulo  $\phi(pq)$ . Phát biểu hàm băm đã cho là an toàn, chống xung đột là đúng hay sai? Giải thích.

## CHƯƠNG 5: MÃ CHỨNG THỰC THÔNG điệp

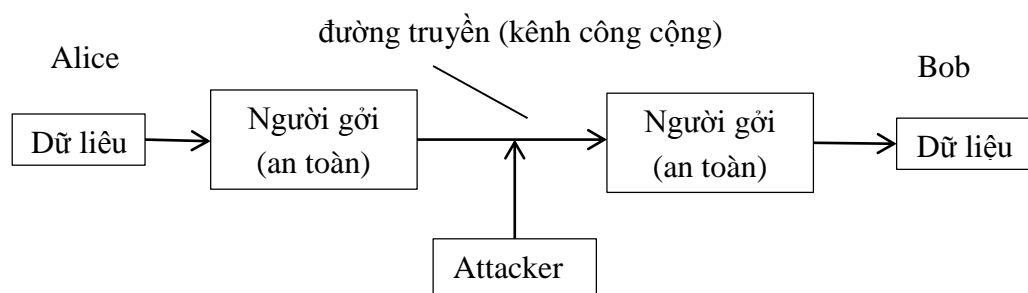
Trong chương này đề cập đến các khái niệm về toàn vẹn và xác thực thông điệp cùng các phương pháp thực hiện, trong đó tập trung vào nội dung của MAC: mô hình tổng quát, tính bảo mật, các đặc tính cũng như các yêu cầu và độ an toàn của MAC.

### 5.1.Toàn vẹn dữ liệu (Message Integrity)

Toàn vẹn dữ liệu là đảm bảo cho thông điệp không bị thay đổi.

Mục tiêu của toàn vẹn dữ liệu:

- Nội dung thông điệp chưa bị thay đổi
- Nguồn của thông điệp tin cậy
- Thông điệp chưa bị phát lại
- Thông điệp được xác minh đúng thời điểm
- Sự liên tục của thông điệp được duy trì



Hình 5.1. Sơ đồ truyền dữ liệu

### 5.2.Xác thực thông điệp (Message Authentication)

#### 5.2.1.Khái niệm

Xác thực thông điệp hay xác thực tính nguyên bản của dữ liệu (Data Origin Authentication) là một kiểu xác thực đảm bảo một thực thể được chứng thực là nguồn gốc thực sự tạo ra dữ liệu này ở một thời điểm nào đó.

Xác thực thông điệp bao hàm cả tính toàn vẹn dữ liệu, nhưng không đảm bảo tính duy nhất và sự phù hợp về thời gian của nó.

#### 5.2.2.Mục tiêu của xác thực thông điệp:

- Bảo vệ tính toàn vẹn của bản tin: bảo vệ bản tin không bị thay đổi hoặc có các biện pháp phát hiện nếu bản tin bị thay đổi trên đường truyền.
- Kiểm chứng danh tính và nguồn gốc: xem xét bản tin có đúng do người xưng tên gửi không hay do kẻ mạo danh gửi.

- Không chối từ bản gốc: trong trường hợp cần thiết, bản thân bản tin chứa các thông tin chứng tỏ chỉ có người xưng danh gửi, không một ai khác có thể làm điều đó. Như vậy người gửi không thể từ chối hành động gửi, thời gian gửi và nội dung của mẫu tin.

Các dạng tấn công điển hình vào tính xác thực: thay thế (Substitution), giả danh Masquerade), tấn công phát lại (Reply attack), phủ nhận (Repudiation).

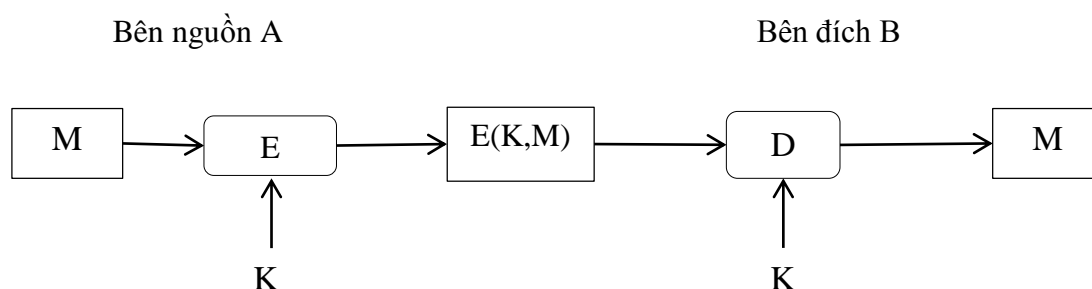
### 5.3. Các phương pháp chứng thực thông điệp

#### 5.3.1. Mã hóa thông điệp:

Sử dụng mã hóa khóa bí mật, mã hóa khóa công khai

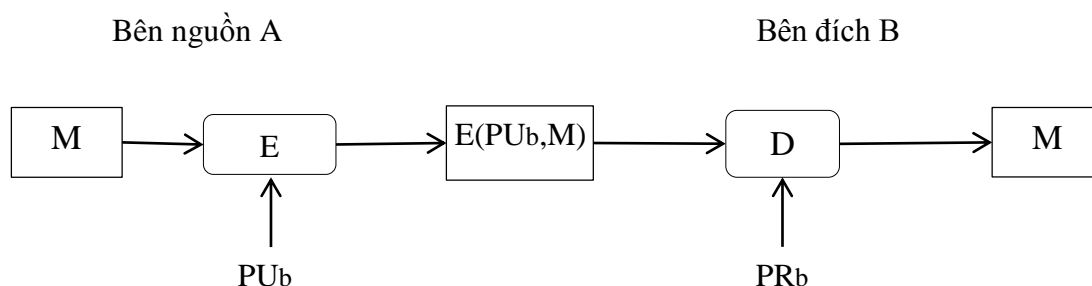
##### a. Mã hóa đối xứng

Mô hình mã hóa đối xứng được thể hiện ở **hình 5.2**. Một thông điệp  $M$  được truyền từ nguồn A đến đích B được mã hoá bằng khóa bí mật  $K$  chia sẻ bởi A và B. Nếu không bên nào biết khoá, thì sẽ có thông tin bí mật: Không bên nào khác có thể khôi phục bản rõ của thông báo. Theo đó, chỉ có B mới có khả năng khôi phục được bản mã.

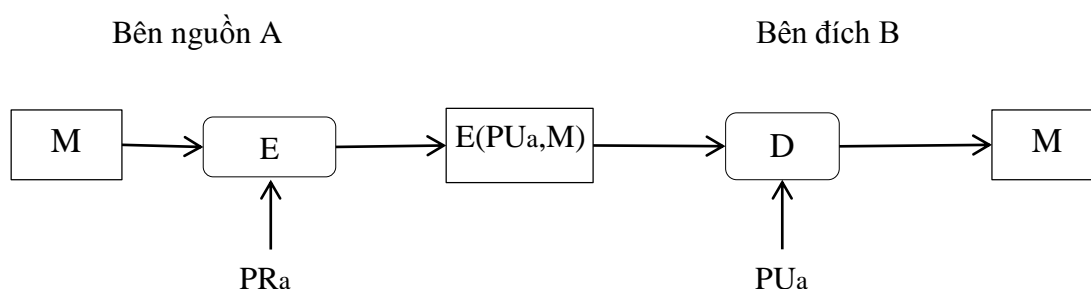


*Hình 5.2. Mã hóa đối xứng: bảo mật và chứng thực*

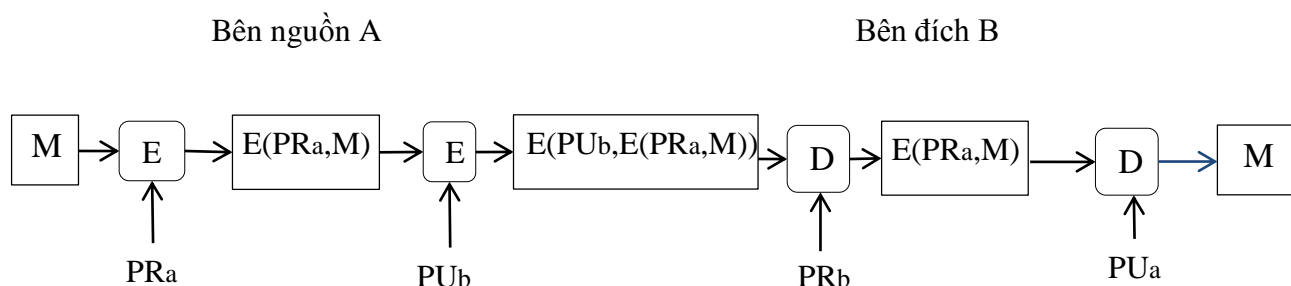
- Xác thực bằng mật mã khóa đối xứng nhằm:
  - Đảm bảo thông báo được gửi đúng nguồn do chỉ bên gửi biết khóa bí mật
  - Không thể bị thay đổi bởi bên thứ ba do không biết khóa bí mật
- Xác thực bằng mật mã khóa công khai
  - Đảm bảo việc xác thực, ngoài ra còn cung cấp chữ ký số



*Hình 5.3. Mã hóa khóa công khai: bảo mật*



*Hình 5.4. Mã hóa khóa công khai: chứng thực, ký số*



*Hình 5.5. Mã hóa khóa công khai: bảo mật, chứng thực, ký số*

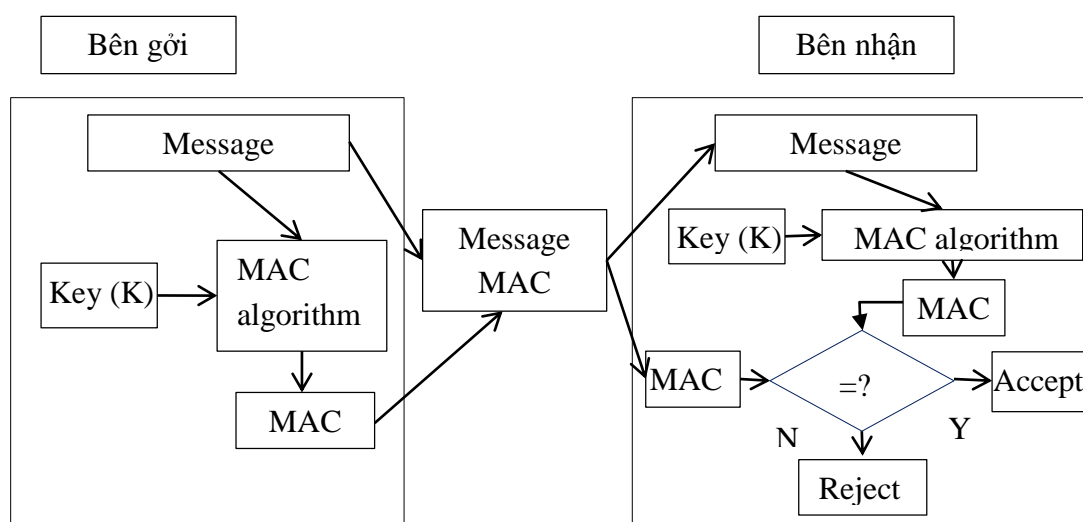
- Xác thực bằng mã hóa có một số nhược điểm:
  - Tốn thời gian để mã hóa cũng như giải mã toàn bộ thông báo
  - Trong nhiều trường hợp việc xác thực mà không cần bảo mật thông điệp (cho phép ai cũng có thể biết nội dung, chỉ cần không được sửa đổi)

### 5.3.2. Mã chứng thực thông điệp

Mã chứng thực thông điệp (Message authentication code - MAC) là một kỹ thuật chứng thực liên quan đến việc sử dụng một khoá bí mật để tạo ra một khối dữ liệu có kích thước nhỏ cố định và được đính kèm thông điệp.

Kỹ thuật này giả sử rằng 2 bên tham gia là A và B chia sẻ một khoá bí mật K. Khi A có một thông điệp gửi đến B, A sẽ tính toán MAC như là một hàm của thông điệp và khoá:  $MAC = C(K, M)$ , với

- M: thông điệp đầu vào có kích thước biến đổi
- C: hàm MAC một chiều biến đổi thông điệp có kích thước bất kỳ kết hợp với khoá K cho kết quả có độ dài cố định.
- K: khoá bí mật chia sẻ giữa người gửi và người nhận
- MAC: mã chứng thực thông điệp có chiều dài cố định



*Hình 5.6. Sơ đồ mã xác thực thông điệp*

Trong hình 5.6. , vì MAC có khóa  $K$  bí mật giữa người gửi và người nhận nên chỉ có người gửi và người nhận mới có thể tính được giá trị MAC tương ứng.

- Thông điệp kết hợp với MAC được truyền tới người nhận.
- Người nhận thực hiện các tính toán tương tự trên các thông điệp đã nhận sử dụng cùng một khóa bí mật, để tạo ra một MAC mới.
- MAC vừa tạo sẽ được so sánh với MAC nhận: nếu MAC nhận phù hợp với MAC vừa tính thì thông điệp không bị thay đổi trong quá trình truyền và chắc chắn được gửi tới từ người gửi đã biết. Ngược lại thông điệp đã bị thay đổi hoặc bị giả mạo bởi một bên thứ ba.
- Trong mô hình MAC: Bên nhận thực hiện cùng giải thuật của bên gửi trên thông báo và khóa bí mật và so sánh giá trị thu được với MAC trong thông báo. Chiều dài thông thường của MAC: 32...96 bit và của khóa  $K$ : 56...160 bit.

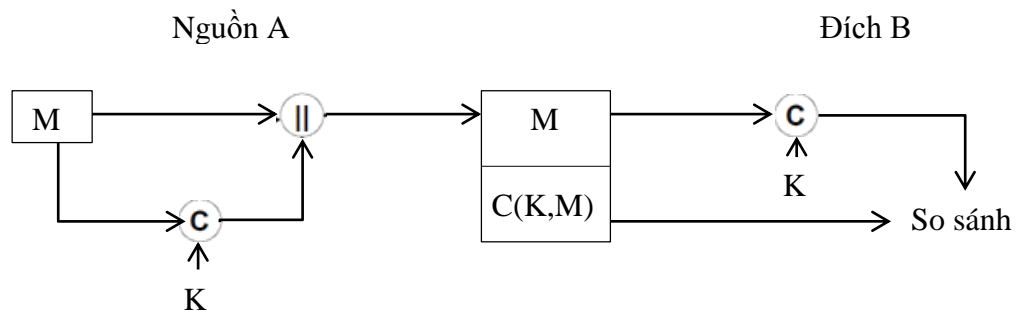
An toàn của MAC: Một số phương pháp tấn công MAC bao gồm:

- **Tấn công vét cạn (Brute-force attacks)**
- **Cryptanalytic Attacks:** Khai thác đặc tính của thuật toán MACs, phương pháp này có ưu thế về thời gian hơn tấn công phức tạp. Tuy nhiên, nếu so với việc tấn công hàm băm thì hiện có nhiều biến thể trong cấu trúc của MACs hơn nên tấn công MACs khó khăn, phức tạp hơn.
- **Tấn công phát lại (Reply attack):** kẻ tấn công phát lại bản tin  $M$  đã được chứng thực trong phiên truyền trước đó. Để chống các tấn công phát lại trong sử dụng MAC người ta bổ sung thêm:
  - Giá trị ngẫu nhiên
  - Tem thời gian

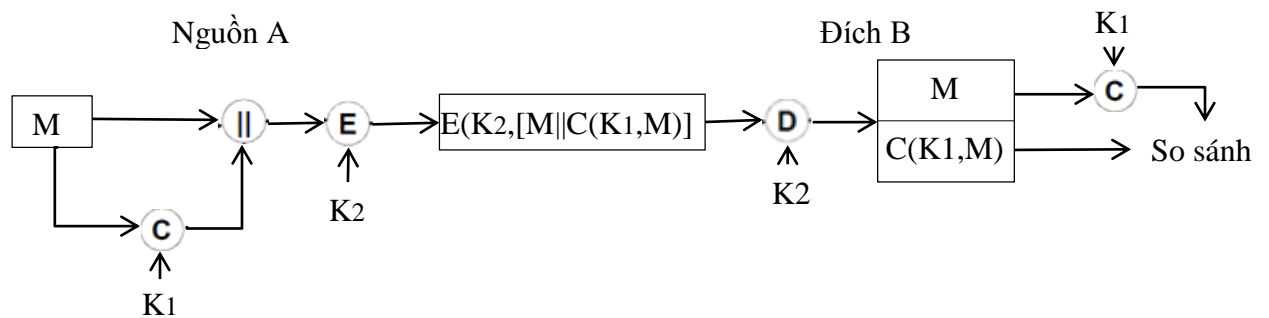
Ưu điểm và ứng dụng của MAC:

- a) Xác thực thông điệp

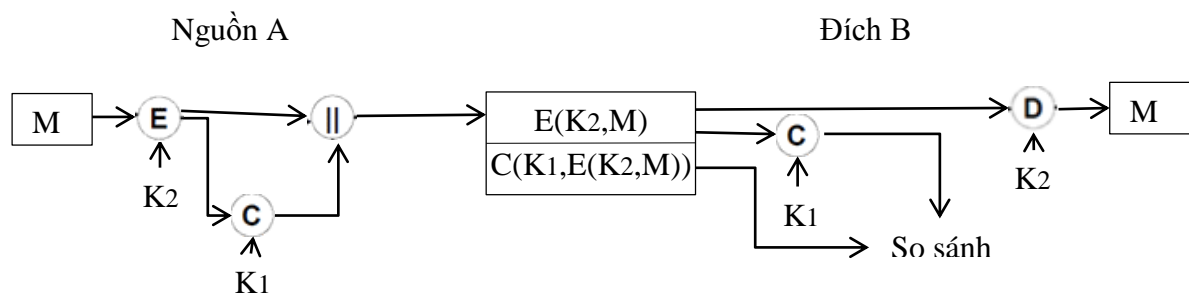




b) Chứng thực và bảo mật thông điệp, xác thực bản rõ



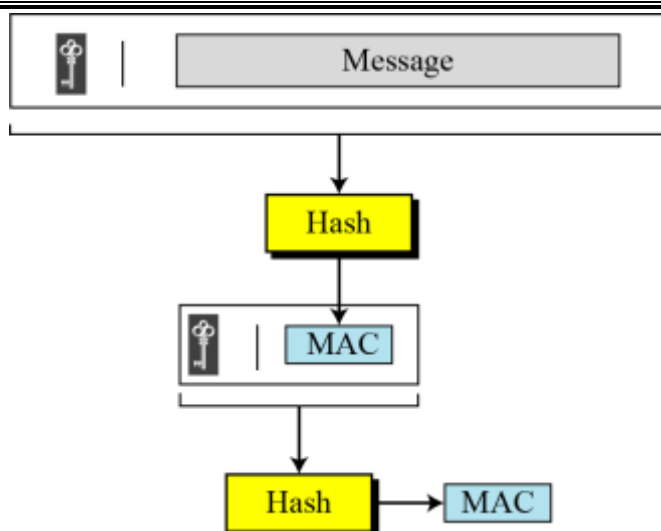
c) Chứng thực và bảo mật thông điệp, xác thực bản mã



### 5.3.3. Vài cơ chế của MAC

#### 5.3.3.1. Nested MAC

- Để tăng bảo mật của MAC
- Băm được áp dụng nhiều lần



*Hình 5.7. Sơ đồ Nested MAC*

### 5.3.3.2.Keyed Hash Functions as MACs

Với mục đích tạo ra MAC với độ an toàn cao, tốc độ nhanh và có khả năng áp dụng rộng rãi, người ra tạo ra HMAC (Hash-based Message Authentication Code – mã xác thực thông điệp trên cơ sở hàm băm). Theo đó, MAC được tạo ra bằng cách băm Khóa bí mật và thông điệp.

$$\text{KeyedHash} = \text{Hash}(\text{Key}|\text{Message})$$

Đặc điểm:

- Dùng hàm băm nguyên mẫu
- Khi có thuật toán băm khác tốt hơn (an toàn hơn, nhanh hơn) thì có thể thay thế vào sơ đồ một cách dễ dàng mà không ảnh hưởng đến các thành phần khác
- Quản lý khóa dễ dàng
- Độ an toàn của HMAC phụ thuộc vào độ an toàn của hàm băm

Trong một số trường hợp, để tăng độ an toàn cho MAC thì hàm băm được sử dụng nhiều lần

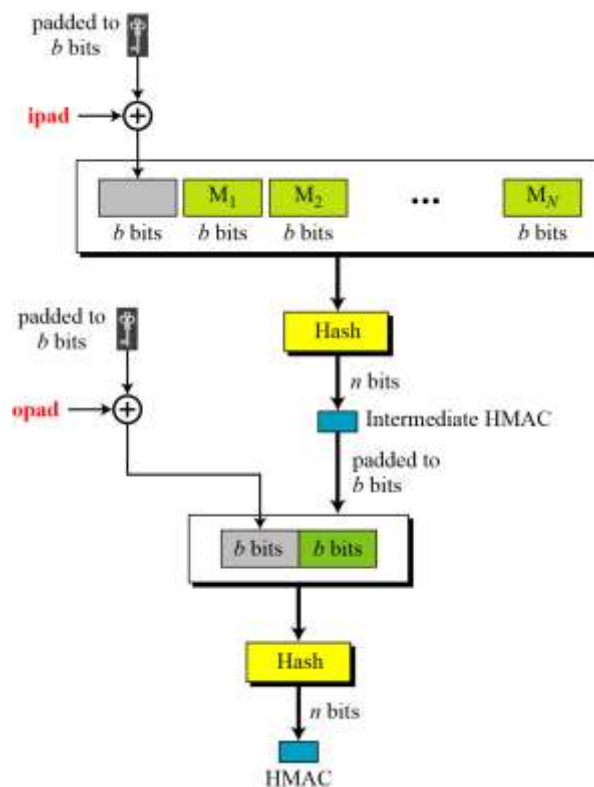
### 5.3.3.3.MAC dựa vào hàm băm (HMAC)

- Gọi là HMAC
- Đặc điểm
  - Dùng hàm băm nguyên mẫu (không chỉnh sửa)
  - Cho phép thay thế dễ dàng hàm băm được nhúng vào trong trường hợp các hàm băm nhanh hơn hoặc nhiều bảo mật được tìm ra hoặc yêu cầu
  - Duy trì hiệu năng ban đầu của hàm băm mà không mắc phải sự suy giảm nghiêm trọng

- Dùng và quản lý các khóa một cách dễ dàng.
- Có một sự phân tích mật mã hiệu được về sức mạnh của sự chứng thực

#### Bảo mật của HMAC

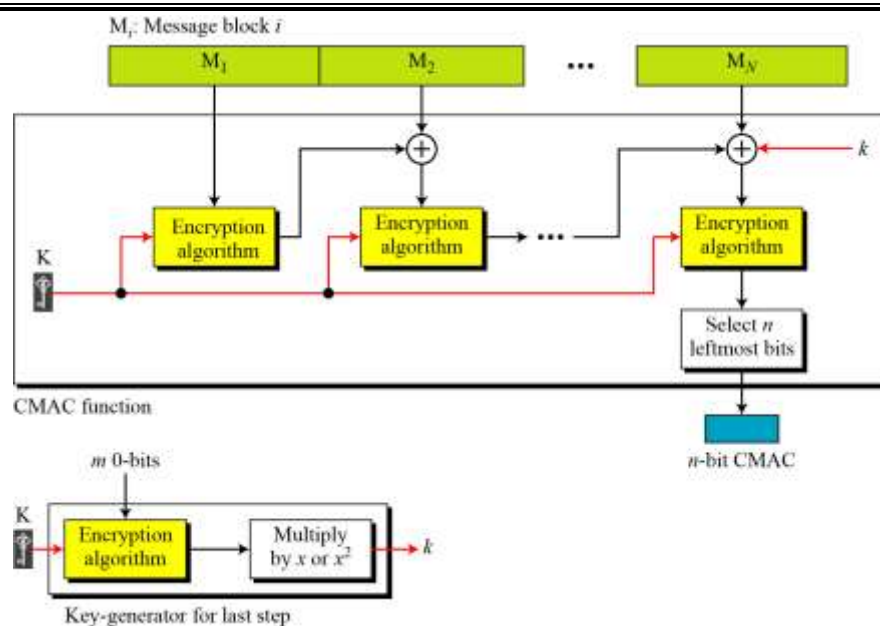
- Dựa lên bảo mật của hàm băm
- Tấn công HMAC:
  - Tấn công brute force trên khóa được dùng
  - Tấn công ngày sinh nhật
- Hàm băm được chọn sử dụng dựa trên ràng buộc về tốc độ và bảo mật



*Hình 5.8. Sơ đồ HMAC*

#### 5.3.3.4. MAC dựa vào mã hóa khối (CMAC)

- Có **DAA** (Data Authentication Algorithm), hiện nay đã lỗi thời
- **CMAC** (Cipher-based Message Authentication Code), được thiết kế để khắc phục những yếu kém của DAA
  - Được dùng rộng rãi trong chính phủ và doanh nghiệp
  - Có kích cỡ thông điệp giới hạn
  - Dùng 2 khóa và padding
  - Được thông qua bởi NIST SO800-38B



Hình 5.9. Sơ đồ Nested CMAC

#### 5.4. Câu hỏi và bài tập

1. Ý nghĩa thực sự của MAC là gì?
2. Trình bày sự khác nhau giữa MAC và hàm băm.
3. Trình bày hai lý do tại sao MAC có thể được ưa thích để xác thực bằng mã hóa đối xứng.
4. Cho khóa sử dụng cho MAC 128 bit và đầu ra của hàm MAC là 32 bit. Cần trung bình bao nhiêu cặp plaintext-MAC để thực hiện tấn công vét cạn. Cần bao nhiêu thời gian thực hiện công việc trên, cho rằng tấn công mỗi key cần 2 giây.
5. Bốn dịch vụ bảo mật: bảo mật dữ liệu, toàn vẹn dữ liệu, chứng thực dữ liệu, chống thoái thác nguồn gốc có liên quan đến các dịch vụ cung cấp trong thực tiễn. Hãy chỉ ra mối quan hệ giữa từng dịch vụ bảo mật với các dịch vụ cung cấp bằng cách lập bảng.
6. Alice muốn gửi tin nhắn cho Bob. Alice muốn Bob có thể xác minh rằng thông điệp không thay đổi khi truyền. Để làm điều này họ sử dụng một hàm MAC với một khóa bí mật  $K$  được chia sẻ để tạo và xác minh giá trị MAC. Hãy tóm tắt các bước mà Alice và Bob đã thực hiện để đảm bảo tính toàn vẹn của thông điệp bằng cách tạo và xác minh MAC.
7. Giải thích lý do tại sao xác thực tin nhắn không đủ để chứng minh nguồn gốc của tin nhắn nói chung và để giải quyết tranh chấp về việc liệu tin nhắn đã được gửi hay không.
8. Dịch vụ bảo mật được cung cấp bởi chữ ký số và giải thích dịch vụ này liên quan đến xác thực thư như thế nào.

---

## CHƯƠNG 6: CHỮ KÝ ĐIỆN TỬ

---

Sự xuất hiện và phát triển nhanh chóng của thương mại điện tử (mô hình B2B, B2C, G2C ...) và các loại hình trao đổi thông tin qua mạng đặt ra yêu cầu cho việc quản lý thông tin, trong đó việc xác thực thông tin một cách nhanh chóng, tin cậy, có tính pháp lý được quan tâm hàng đầu. Chữ ký điện tử được xem là một trong những giải pháp giải quyết các yêu cầu đó. Tại Việt Nam hiện nay chữ ký điện tử được công nhận, có giá trị pháp lý tương đương với văn bản giấy có chữ ký và đóng dấu. Trong chương này, tài liệu đề cập đến các khái niệm liên quan đến chữ ký điện tử cùng các ứng dụng trong thực tiễn, đồng thời tập trung làm rõ các khía cạnh của chữ ký số.

### 6.1. Khái niệm chữ ký điện tử

**Chữ ký điện tử (Electronic signature hoặc E- signature)** là một biểu tượng điện tử được gắn vào tài liệu dưới dạng điện tử và được sử dụng bởi người ký để ký tên. Ví dụ: một đoạn hình ảnh được chèn vào cuối email là một chữ ký điện tử.

Chữ ký điện tử là một giao thức hiệu quả tương tự như chữ ký thực được ứng dụng đảm bảo an toàn trong thương mại điện tử (e-Commerce) và quản trị điện tử (e-Governance).

Một số yêu cầu của chữ ký điện tử:

- Không thể giả mạo
- Xác thực
- Không thể thay đổi
- Không thể sử dụng lại

Phân loại chữ ký điện tử: gồm các nhóm chính

- Digital signature (chữ ký số) được tạo ra dựa trên lý thuyết mã hóa khóa công khai, sử dụng thông qua nhà cung cấp chính thức (Certificate Authority - CA)
- E-sign: là loại chữ ký không sử dụng PKI, chủ yếu dựa vào danh tính và nhận dạng Logs thích hợp cho các hệ thống đóng, tính bảo mật không cao.
- Biometric signatures: sử dụng những đặc điểm có tính cá nhân như là một kiểu chữ ký, ví dụ: dấu vân tay, trọng mắt. Loại này đòi hỏi công nghệ cao, tốn kém nhưng vẫn còn các lỗ hổng trong bảo mật.

Một số ứng dụng của chữ ký điện tử:

- Ứng dụng trong chính phủ điện tử: kê khai, nộp thuế trực tuyến, khai báo thông quan trực tuyến, quản lý nhà nước, bỏ phiếu điện tử ...
- Ứng dụng trong thương mại điện tử: mua bán, đặt hàng trực tuyến, thanh toán trực tuyến ...

- Ứng dụng trong các giao dịch trực tuyến: ví dụ giao dịch qua email ...
- Ứng dụng trong các hội nghị, thảo luận trực tuyến, làm việc từ xa, giáo dục điện tử ...

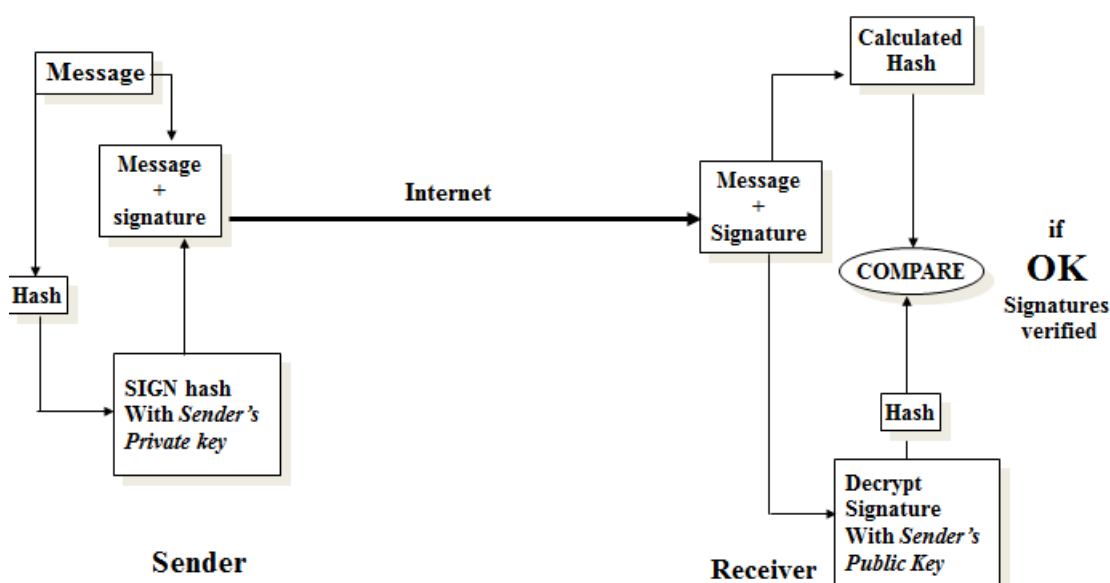
## 6.2. Chữ ký số

Chữ ký số là một dạng của chữ ký điện tử dựa trên công nghệ mã hóa khóa công khai nhằm cung cấp các dịch vụ bảo mật: xác thực, toàn vẹn và chống thoái thác. Người dùng có cặp khóa công khai – cá nhân để sử dụng ký các văn bản điện tử cũng như trao đổi thông tin mật. Khóa công khai được phân phối thông qua chứng thực khóa công khai.

So với chữ ký thông thường, chữ ký số thường không xuất hiện trong văn bản được ký, ngoài ra việc kiểm tra chữ ký sử dụng giải thuật rất khó giả mạo hơn so với chữ ký thường.

	Tài liệu giấy	Tài liệu điện tử
Chứng thực	Có thể bị sao chép	Không thể sao chép
Toàn vẹn	Chữ ký độc lập với tài liệu	Chữ ký phụ thuộc nội dung của tài liệu
Chống thoái thác	Cần chữ ký thành thạo, Có thể có lỗi trong quá trình hoạt động	Mọi máy tính người dùng để sử dụng được. Không phát sinh lỗi.

Sơ đồ chữ ký số dựa trên thuật toán ký số và bao gồm 3 giai đoạn: tạo bộ khóa, ký số và xác minh chữ ký.



Hình 6.1. Sơ đồ ký và xác minh chữ ký số

- Tạo bộ khóa (Key generation): tạo số ngẫu nhiên, tính bộ khóa (khóa cá nhân, khóa công khai).

- Ký số (Digital Signature): tính giá trị băm, mã hóa bằng băm sử dụng khóa cá nhân, đính chữ ký vừa tạo vào văn bản.
- Xác minh chữ ký (Verification of signatures): quá trình thực hiện kiểm tra chứng thực, toàn vẹn và không chối từ.

Như vậy, bản chất của chữ ký số là kết quả của việc mã hóa giá trị băm văn bản bằng hệ mã hóa khóa công khai sử dụng khóa cá nhân.

Tỷ lệ thành công của toàn bộ mô hình này phụ thuộc phần lớn vào 2 thuộc tính chính:

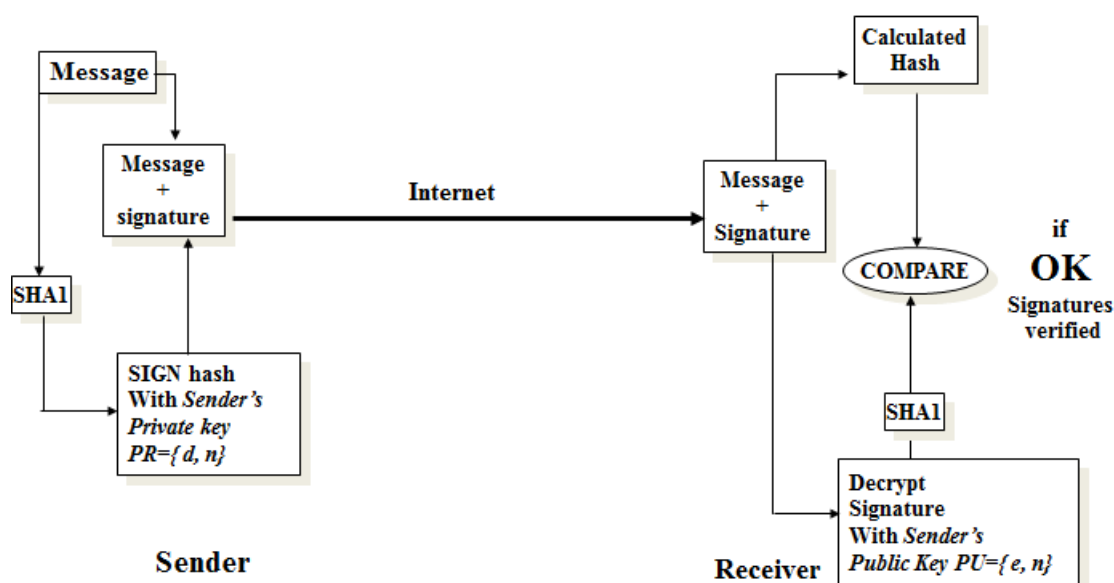
- Chữ ký được phát sinh từ một thông điệp cụ thể và khóa riêng tư có thể xác nhận được tính xác thực của thông điệp đó bằng cách sử dụng khóa công khai ứng với khóa riêng tư đó.
- Tiến trình phải tránh được việc một người không sở hữu khóa riêng tư lại phát sinh ra được chữ ký hợp lệ ứng với khóa riêng tư đó.

## 6.2. Một số loại chữ ký số

### 6.2.1. Chữ ký số RSA

Ý tưởng của chữ ký số RSA dựa trên hệ mã hóa RSA, trong đó vai trò của khóa cá nhân và khóa công khai thay đổi cho nhau. Bộ khóa (khóa cá nhân, khóa công khai) được sử dụng của người gửi, trong sơ đồ thì khóa cá nhân sử dụng để ký, người nhận sử dụng khóa công khai của người gửi để xác minh chữ ký.

Trong sơ đồ chữ ký số RSA: quá trình gửi và nhận văn bản dựa vào thuật toán băm (MD5, SHA-1) và thuật toán mã hóa khóa công khai RSA, gồm 3 giai đoạn:



Hình 6.2. Sơ đồ ký và xác minh chữ ký số RSA

- Giai đoạn 1: Tạo bộ khóa  $K=\{\mathbf{P}\mathbf{U}\mathbf{b}\mathbf{l}\mathbf{i}\mathbf{c}\ \mathbf{k}\mathbf{e}\mathbf{y}, \mathbf{P}\mathbf{R}\mathbf{i}\mathbf{v}\mathbf{a}\mathbf{t}\mathbf{e}\ \mathbf{k}\mathbf{e}\mathbf{y}\}$  theo giải thuật RSA ở mục ???, trong giai đoạn này server sử dụng thuật toán khởi tạo tham số bảo mật  $k$  để đưa ra các tham số chung của hệ thống  $p, q$ . Sau đó tính toán theo thuật toán để tìm được cặp khóa công khai/khóa cá nhân. Theo đó,  $PU = \{e, n\}$ ,  $PR = \{d, n\}$ .
- Giai đoạn 2: Thuật toán sinh chữ ký số: nhận đầu vào là giá trị băm của văn bản, sinh ra chữ ký số dựa vào khóa cá nhân sử dụng thuật toán RSA theo công thức:  $s = h(M)^d$ , trong đó  $M$ : ký hiệu văn bản cần ký số,  $h(M)$ : bảng băm SHA-1 của  $M$ .
- Giai đoạn 3: Thuật toán xác minh chữ ký số: đầu vào là văn bản kèm chữ ký số và khóa công khai của người sở hữu, văn bản đó. Đầu ra là kết quả kiểm tra “Đúng” hoặc “Sai”. Trong sơ đồ 6.2, việc giải mã  $s$  được tính theo công thức sau:  $h = s^e$ .

### 6.2.2.Chữ ký số El-Gamal

Mô hình ký số El-Gamal gồm 3 giai đoạn như sau:

- Tạo bộ khóa: Tạo bộ khóa theo giải thuật El-gamal (mục 3.4.3). Khi đó, bộ khóa  $K = \{PU, PR\}$  của A, với khóa công khai  $PU = \{q, \alpha, Y_A\}$  và khóa cá nhân  $PR = \{X_A\}$ .

Quá trình sử dụng bộ khóa được tạo ở giai đoạn 1 trong việc ký trên văn bản  $M$  và xác thực chữ ký như sau:

- Quá trình ký số văn bản  $M$  :
  - ✓ Tính  $m = H(M)$ ,  $0 \leq m \leq q-1$
  - ✓ Chọn số nguyên  $k$  sao cho:  $1 \leq k \leq q-1$  và  $UCLN(k, q-1) = 1$
  - ✓ Tính khóa  $S_1 = \alpha^k \bmod q$  và  $S_2 = k^{-1}(m - X_A S_1) \bmod (q-1)$ , với  $k^{-1}$  là số nghịch đảo của  $k \bmod (q-1)$
  - ✓ Chữ ký số thu được là:  $(S_1, S_2)$
- Quá trình xác minh chữ ký số:
  - ✓  $v_1 = \alpha^m \bmod q$
  - ✓  $v_2 = Y_A^{S_1} S_1^{S_2} \bmod q$
  - ✓ Nếu  $v_1 = v_2$  thì chữ ký được xác thực.

### 6.2.3.Chữ ký số DSS

Digital Signature Standard (DSS) là chuẩn chữ ký số của chính phủ Hoa Kỳ, được thiết kế bởi NIST và NSA vào đầu những năm 90, DSS được công bố vào năm 1991 (FIPS 186), được chấp nhận vào 1993, thay đổi chỉnh sửa vào 1996 (FIPS 186-1), mở rộng năm 2000 (FIPS 186-2).



Trong DSS tiêu chuẩn sử dụng giải thuật băm SHA và DSA. Trong phiên bản năm 2000 (FIPS 186-2) thay thế bằng giải thuật RSA và đường cong Elip.

Mô tả thuật toán ký số DSA:

- Quá trình tạo bộ khóa:
  - ✓ Công khai bộ  $(p, q, g)$ : với  $q$ : số nguyên tố độ dài 160 bit;  $p$  là số nguyên tố thỏa mãn điều kiện:  $2^{L-1} < p < 2^L$  ( $L \in [512; 1024]$  bit và là bội số của 64);  $g = h^{(p-1)/q}$  ( $1 < h < p-1$  và  $h^{(p-1)/q} \bmod p > 1$ )
  - ✓ Khóa cá nhân: chọn số tự nhiên  $x < q$
  - ✓ Khóa công cộng:  $y = g^x \bmod p$ .
- Quá trình ký số văn bản  $M$  của bên gửi:
  - ✓ Khởi tạo ngẫu nhiên khóa  $k$ , ( $k < q$ )
  - ✓ Tính toán chữ ký  $(r, s)$ , với  $r = (g^k \bmod p) \bmod q$ ,  
 $s = [k^{-1}(H(M) + xr)] \bmod q$
  - ✓ Chữ ký số  $(r, s)$  được gửi kèm với văn bản  $M$  đến bên nhận.
- Quá trình xác thực chữ ký:
  - ✓ Bên nhận có văn bản  $M$  và chữ ký  $(r, s)$  tiến hành xác minh theo các bước sau:
 
$$w = s^{-1} \bmod q$$

$$u_1 = [H(M)w] \bmod q$$

$$u_2 = (rw) \bmod q$$

$$v = [(g^{u_1} y^{u_2}) \bmod p] \bmod q$$
  - ✓ Nếu  $r = v$  thì chữ ký được xác thực.

### 6.3. Câu hỏi và bài tập

1. Cho biết các sự khác nhau cơ bản của chữ ký truyền thống và chữ ký điện tử?
2. Trong chữ ký số, bên nào sử dụng khóa cá nhân và khóa công cộng.
3. Lợi ích của chữ ký số là gì?
4. Mô tả ngắn gọn các bước trong mô hình ký số.
5. Vẽ sơ đồ chữ ký số RSA và cho ví dụ minh họa bằng số.
6. Tại sao trong chữ ký số có sự tham gia của hàm băm.
7. Alice muốn gửi văn bản  $M$  và chữ ký số  $\text{Sig}(M)$  đến Bob. Alice và Bob có một bản sao chính xác các khóa công khai của nhau, và đã đồng ý sử dụng một hàm băm cụ thể  $h$  và thuật toán bất đối xứng có thể hoạt động ở chế độ chữ ký  $S$  (tương ứng với chế độ giải mã  $D$ ) hoặc trong chế độ xác minh  $V$  (tương ứng với chế độ mã hóa  $E$ ). Hãy phác thảo các bước chính mà Alice

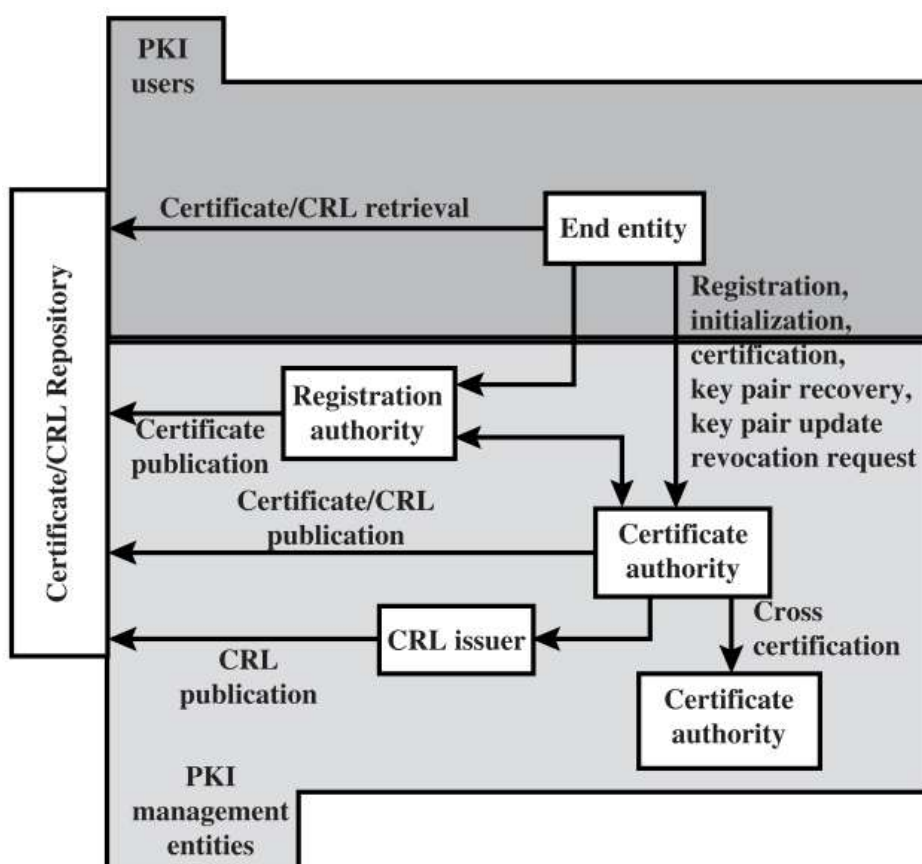
phải thực hiện khi ký M, và các bước mà người nhận Bob phải tuân theo để xác minh và xác nhận chữ ký Sig(M).

8. Hàm băm thường được dùng để kiểm tra tính toàn vẹn thông tin:
  - a. Liệt kê các yêu cầu cơ bản của hàm băm mật mã
  - b. Sự khác nhau của hai biến thể của chống va chạm
9. Dịch vụ bảo mật được cung cấp bởi chữ ký số như thế nào? Giải thích dịch vụ này liên quan đến xác thực?
10. Lý do chính đáng nào để một người gửi có thể từ chối để bác bỏ một thông điệp có chữ ký?
11. Bob có khóa công khai (77;13). Bob gửi cho Alice một hợp đồng giao dịch  $m$  và chữ ký số  $s$  thông qua internet. Alice nhận được ( $m = 3; s = 5$ ). Theo bạn, Alice có thể sử dụng các thông tin nhận được hay không? Giải thích?
12. Cho giao thức trong đó người gửi thực hiện các hoạt động được mô tả:
$$y = e_{k_1}(x \parallel \text{sign}_{k_{pr}}(H(x))),$$
trong đó lần lượt các ký hiệu:  $x$  : thông điệp,  $H$  : hàm băm,  $e$  : thuật toán mã hóa đối xứng,  $\parallel$ : phép nối các chuỗi,  $k$  : khóa bí mật của bên gửi và bên nhận,  $k_{pr}$  : khóa cá nhân người gửi.
  - a) Hãy mô tả từng bước những hoạt động ở phía bên nhận cần thực hiện.
  - b) Xác định giao thức trên có cung cấp tính bảo mật, tính toàn vẹn và tính chống thoái thác hay không? Giải thích?
13. Nếu một người sử dụng máy tính của người khác thì có mối đe dọa nào đến an toàn của chủ sở hữu chữ ký số hay không? Giải thích.
14. Theo sinh viên, có khả năng nào một người khác sử dụng chữ ký số của một người khác mà họ không biết?
15. Trong một tài liệu có thể đồng thời sử dụng nhiều chữ ký số hay không? Giải thích.

## CHƯƠNG 7: KIẾN TRÚC KHÓA CÔNG KHAI

RFC 2822 (Internet Security Glossary) định nghĩa kiến trúc khóa công khai (public-key infrastructure - PKI) là một tập hợp phần cứng, phần mềm, con người, chính sách và quy trình cần thiết để xây dựng, quản trị, lưu trữ, phân phối và hủy thẻ chứng thực số (digital certificate) dựa trên mã hóa bất đối xứng. Mục tiêu chính của việc phát triển một PKI là để bảo đảm tính bảo mật, tiện lợi và hiệu quả của khóa công khai. Nhóm làm việc về Public Key Infrastructure X.509 (PKIX) của Internet Engineering Task Force (IETF) đã thúc đẩy việc thiết lập một mô hình (chính là X.509) phù hợp cho việc triển khai một kiến trúc dựa trên certificate cho mạng Internet.

### 7.1.Mô hình kiến trúc khóa công khai



Hình 0.1 Mô hình kiến trúc khóa công khai

Hình 7.1 mô tả mối liên hệ giữa các thành phần chính của mô hình PKIX. Các thành phần này bao gồm:

- **Thực thể cuối (End entity):** Khái niệm mô tả người dùng cuối (end user), thiết bị (server, router, ...), hoặc một thực thể bất kỳ có thể được định danh trong trường subject của một thẻ chứng thực khóa công khai. Các thực thể cuối sẽ sử dụng hoặc hỗ trợ các dịch vụ liên quan đến PKI.
- **Certification Authority (CA):** Giúp cung cấp thẻ chứng thực và danh sách thẻ chứng thực bị hủy (Certificate Revocation List - CRL). CA cũng hỗ trợ các

chức năng quản trị, mặc dù những quyền này thường được phân cho một hoặc nhiều Registration Authority.

- Registration Authority (RA): Một thực thể tùy chọn giúp thực thi các chức năng quản trị của CA. RA thường gắn liền với quá trình đăng ký end entity nhưng nó cũng có thể hỗ trợ một số công việc khác.
- CRL issuer: Một thực thể tùy chọn. CA có thể phân quyền cho CRL issuer quảng bá các CRL.
- Repository: Khái niệm để mô tả một phương thức bất kỳ để lưu trữ certificate và CRL.

## **7.2.Các chức năng quản trị PKIX**

### **7.2.1.Đăng ký (Registration)**

Đây là tiến trình mà người dùng tiến hành đăng ký nhằm giúp CA (hoặc RA) nhận diện được mình trước khi CA đó cung cấp thẻ chứng thực cho người dùng đó. Việc đăng ký khởi động tiến trình gia nhập vào một kiến trúc khóa công khai (PKI).

Quy trình đăng ký thường bao gồm một số bước được thực hiện offline hoặc online để bảo đảm quy trình xác thực lẫn nhau (mutual authentication). Thông thường thì end entity sẽ được cấp một hoặc nhiều khóa bí mật để sử dụng cho việc xác thực diễn ra sau đó.

### **7.2.2.Khởi tạo (Initialization)**

Trước khi một hệ thống có thể vận hành một cách an toàn, cần phải cài đặt trên hệ thống đó các khóa có mối liên hệ phù hợp với một số khóa được lưu sẵn ở đâu đó trong PKI. Chẳng hạn, client cần được khởi tạo một cách an toàn với public key và các thông tin đã được bảo đảm khác về CA mà client này tin cậy. Những thông tin đó sẽ được sử dụng trong quá trình xác thực được dẫn của thẻ chứng thực.

### **7.2.3.Xác nhận (Certification)**

Đây là tiến trình mà một CA cung cấp một thẻ chứng thực (certificate) cho public key của một user, chuyển certificate đó đến hệ thống của client, và/hoặc đưa certificate đó vào repository.

### **7.2.4.Khôi phục cặp khóa (Key pair recovery)**

Các cặp khóa có thể được sử dụng để hỗ trợ việc tạo và xác nhận chữ ký số (digital signature) cũng như việc mã hóa và giải mã.

Khi một cặp khóa được sử dụng cho việc encryption/decryption, cần phải cung cấp một cơ chế để khôi phục các khóa dùng cho việc giải mã (decryption key) để phòng trường hợp việc truy cập vào các khóa sẵn có không thể thực hiện được. Nếu không, chúng ta có thể mất dữ liệu do dữ liệu đã được mã hóa.

Việc mất quyền truy cập vào decryption key có thể dẫn đến việc quên mật khẩu, mã PIN, hư hỏng đĩa lưu trữ, hư hại các token cứng, ... Key pair recovery cho phép các thực thể cuối khôi phục encryption/decryption key pair từ một nguồn dự phòng đã được xác thực, thường là CA đã cung cấp certificate cho thực thể cuối đó.

### 7.2.5. Cập nhật key pair

Các cặp khóa cần được cập nhật thường xuyên, nghĩa là thay thế chúng bằng một cặp khóa mới cũng như các certificate mới sẽ được cung cấp, việc cập nhật là bắt buộc khi thời gian tồn tại của certificate đã hết, dẫn đến việc certificate đó bị hủy đi.

### 7.2.6. Yêu cầu hủy (Revocation request)

Một người dùng đã được xác thực có thể báo với CA về một tình huống bất thường dẫn đến nhu cầu phải hủy certificate. Các lý do cho việc hủy này bao gồm private key bị tấn công thỏa hiệp, thay đổi tên, ...

### 7.2.7. Xác thực chéo (Cross certification)

Hai CA trao đổi thông tin được sử dụng trong việc thiết lập 1 thẻ xác thực chéo (cross-certificate). Một cross-certificate là một certificate được cấp bởi một CA cho một CA khác có chứa một CA signature key sử dụng cho việc cấp certificate.

## 7.3. Phân phối khóa (Key Distribution)

### 7.3.1. Phân phối khóa đối xứng sử dụng mã hóa đối xứng

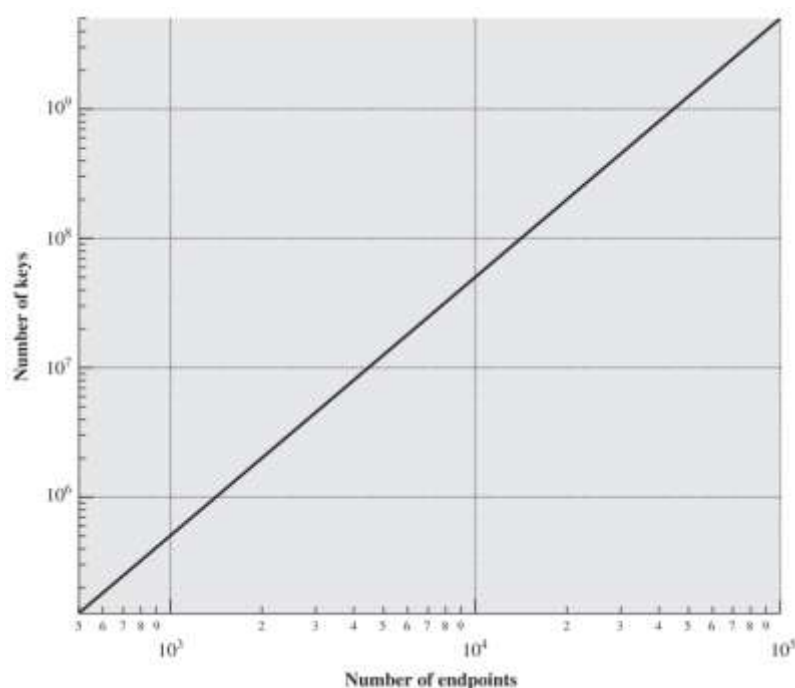
Để mã hóa đối xứng có thể hoạt động được, hai bên tham gia vào quá trình truyền thông phải dùng chung khóa, và khóa đó phải được giữ bí mật với những đối tượng khác. Thêm vào đó, sự thay đổi khóa thường xuyên lại là một nhu cầu cần phải bảo đảm để tránh việc tấn công theo dạng thỏa hiệp, khi người tấn công biết được khóa. Do đó, sức mạnh của hệ thống mã hóa lệ thuộc nhiều vào kỹ thuật phân phối khóa (key distribution technique), một thuật ngữ đề cập đến phương tiện để phân phối một khóa đến hai bên cần truyền dữ liệu mà không để những bên khác thấy được khóa.

Với 2 bên A và B, việc phân phối khóa có thể đạt được bằng một số cách, như trình bày dưới đây:

1. A chọn một key và chuyển nó đến B theo một phương thức vật lý.
2. Một bên thứ ba có thể chọn key và chuyển nó đến A và B theo một phương thức vật lý.
3. Nếu A và B có sử dụng khóa trước đó, một bên có thể truyền khóa mới đến bên còn lại, khóa mới được mã hóa bằng khóa cũ..
4. Nếu A và B cùng có kết nối mã hóa đến một bên thứ ba, C, thì C có thể phân phối một key trên các đường liên kết đã được mã hóa đó đến A và B.

Phương án 1 và 2 đòi hỏi chúng ta phải vận chuyển khóa bằng tay, không dùng kết nối mạng. Với các hệ thống phân tán, một máy bất kỳ có thể phải tham gia vào việc giao

địch với rất nhiều host khác theo thời gian. Do đó, mỗi thiết bị cần được cung cấp đồng nhiều khóa. Vấn đề trở nên đặc biệt khó khăn khi mạng phân tán có quy mô lớn. Phạm vi của vấn đề phụ thuộc vào số cặp host truyền thông cần được hỗ trợ. Nếu mã hóa đầu cuối-đến-đầu cuối (end-to-end encryption) được thực hiện ở tầng mạng thì sẽ cần một khóa cho mỗi cặp máy trên mạng muốn truyền nhận thông tin. Do đó, nếu có  $N$  host thì sẽ cần  $[N(N - 1)]/2$  khóa. Nếu việc mã hóa được thực hiện ở tầng ứng dụng, sẽ cần 1 khóa cho mỗi cặp người dùng hoặc tiến trình cần tham gia vào quá trình truyền thông. Cho nên, một mạng có thể có vài trăm thiết bị nhưng có thể có đến hàng nghìn người dùng và tiến trình. Hình 7.2 mô tả sự gia tăng tác vụ phân phối khóa với việc mã hóa end-to-end.



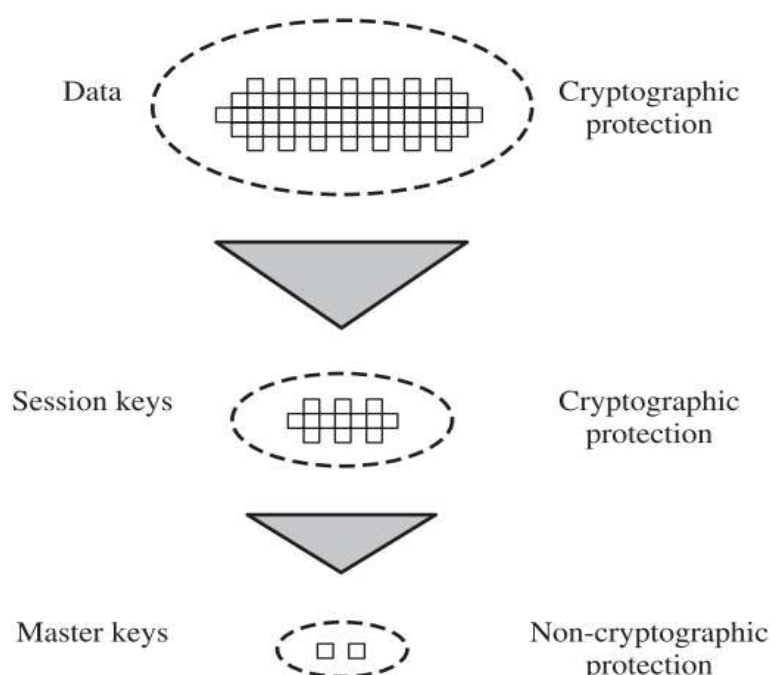
Hình 0.2 Số khóa cần thiết để hỗ trợ truyền thông ngẫu nhiên giữa các endpoint

Một mạng sử dụng mã hóa node-level bao gồm 1000 node có thể cần phân phối đến nửa triệu khóa. Nếu cùng mạng đó dùng 10.000 ứng dụng, sẽ cần đến 50 triệu khóa để hỗ trợ mã hóa ở mức ứng dụng.

Với lựa chọn thứ 3, chúng ta sẽ có sẵn đường kết nối mã hóa hoặc mã hóa end-to-end. Tuy nhiên, nếu một người tấn công thành công trong việc truy cập vào 1 khóa, tất cả những khóa tiếp theo sẽ bị lộ. Thêm vào đó, lượt phân phối khóa đầu tiên cũng bao gồm đến hàng triệu khóa.

Với mã hóa end-to-end, một vài biến thể của lựa chọn 4 đã được triển khai rộng rãi. Với mô hình này, một trung tâm phân phối khóa (Key Distribution Center – KDC) sẽ chịu trách nhiệm cho việc phân phối khóa đến các cặp người dùng (có thể là host, processe hoặc application) khi cần. Mỗi người dùng phải chia sẻ một khóa duy nhất với KDC để phục vụ cho việc phân phối khóa.

Hình 7.3 mô tả việc sử dụng KDC được dựa trên mô hình cây phân cấp khóa - Key Hierarchy.



Hình 0.3 Mô hình cây phân cấp khóa

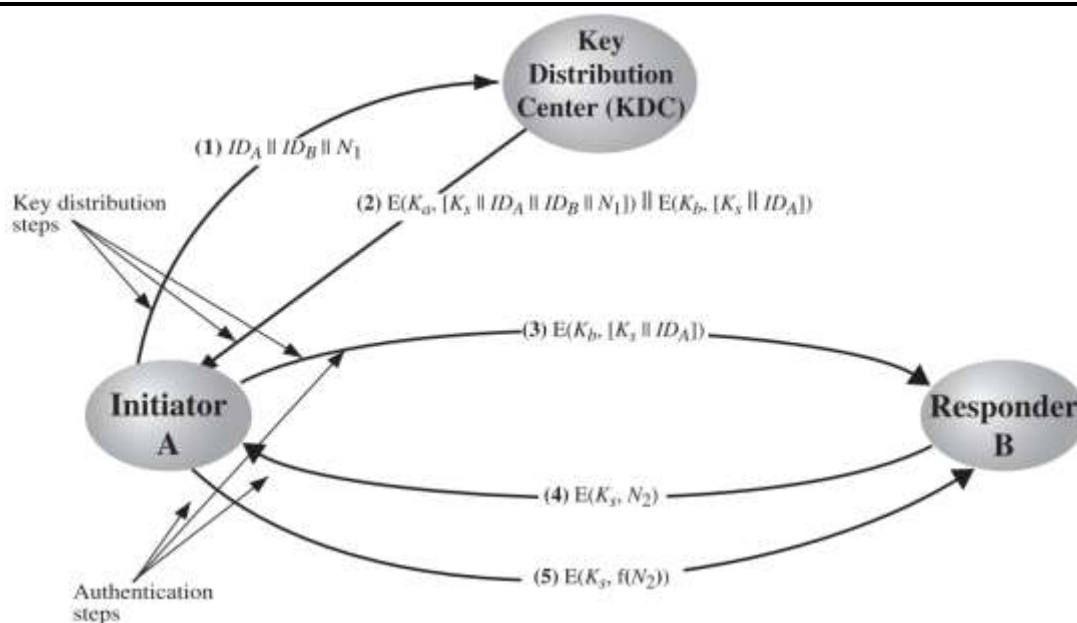
Quá trình truyền thông giữa các hệ thống đầu cuối được mã hóa bằng một khóa tạm, thường được gọi là khóa phiên (session key). Thông thường, session key được sử dụng trong quá trình tồn tại của kết nối logic, chẳng hạn như kết nối frame relay connection hoặc kết nối ở tầng giao vận, rồi sau đó bị hủy đi. Mỗi session key sẽ được lấy từ KDC thông qua cùng kết nối mạng với kết nối dành cho truyền thông end-user. Cụ thể, các session key sẽ được truyền dưới dạng mã hóa bằng cách sử dụng một khóa chính (master key) được chia sẻ và dùng chung bởi KDC và người dùng cuối.

Với mỗi người dùng cuối, có một master key duy nhất được chia sẻ với KDC. Lẽ dĩ nhiên, các master key phải được phân phối theo một cách nào đó. Tuy nhiên, quy mô của vấn đề đã giảm đi nhiều. Nếu có các thực thể muốn truyền thông theo cặp, như đã đề cập từ trước, thì sẽ có  $[N(N - 1)]/2$  session key cần có vào một thời điểm. Tuy nhiên, chỉ cần một master key cho mỗi thực thể. Do đó, các master key có thể được phân phối theo một cách nào đó mà không cần phải mã hóa, chẳng hạn như phân phối vật lý.

#### 7.3.1.1. Mô hình phân phối khóa

Vấn đề phân phối khóa có thể được giải quyết theo nhiều cách khác nhau. Mô hình 7.4 mô tả một ngữ cảnh tiêu chuẩn, với giả định rằng mỗi người dùng chia sẻ một master key duy nhất với KDC.





Hình 0.4 Ngữ cảnh phân phối khóa

Giả sử rằng người dùng A muốn thiết lập một kết nối logic với B và yêu cầu một khóa phiên dùng một lần (one-time session key) để bảo mật dữ liệu truyền đi trên kết nối này. A có một master key là  $K_a$ . Chỉ A và KDC biết về khóa này. Tương tự, B chia sẻ khóa chính  $K_b$  với KDC. Trình tự các bước diễn ra như sau:

1. A gửi đến KDC yêu cầu một session key để bảo vệ kết nối logic đến B. Thông điệp bao gồm định danh của A và B và một ID duy nhất cho giao tác này, gọi là *nonce*. Nonce có thể là time-stamp, biến đếm, hoặc một số ngẫu nhiên; yêu cầu tối thiểu là nó cần phải khác biệt với mỗi request. Thêm vào đó, để tránh tấn công giả mạo (masquerade), kẻ tấn công phải gặp khó khăn trong việc đoán giá trị của nonce. Do đó, giá trị của nonce nên là một số ngẫu nhiên.
2. KDC phản hồi với một thông điệp được mã hóa bằng  $K_a$ . Do đó, A sẽ là đối tượng duy nhất có thể đọc được thông điệp, và A cũng biết thông điệp này xuất phát từ KDC. Thông điệp bao gồm 2 thông tin từ A:
  - a. Session key dùng một lần,  $K_s$ , để dùng cho phiên làm việc.
  - b. Thông điệp request ban đầu, bao gồm cả nonce, để giúp A so khớp thông điệp phản hồi với request tương ứng.

Nhờ đó, A có thể xác định rằng thông điệp yêu cầu ban đầu có bị chỉnh sửa trước khi được nhận bởi KDC không, và, do có nonce, thông điệp này không phải là thông điệp bị lặp lại từ một yêu cầu nào khác trước đó.

Thêm vào đó, 2 thông tin sau cần được gửi cho B:

- a. Session key dùng một lần,  $K_s$ , để dùng cho phiên làm việc.
- b. Thông tin định danh của A (chẳng hạn, địa chỉ của A),  $ID_A$ .



Hai thông tin này được mã hóa bằng  $K_b$  (master key mà KDC chia sẻ với B), sau đó gửi cho B để thiết lập kết nối và xác nhận danh tính của A.

3. A lưu session key để sử dụng cho phiên làm việc sắp tới, chuyển tiếp cho B thông tin vốn được phát sinh ở KDC cho B, cụ thể là  $E(K_b, [K_s \parallel ID_A])$ . Vì thông tin này được mã hóa bằng  $K_b$ , nó được bảo vệ khỏi kiểu tấn công eavesdropping.

B giờ đây đã biết được session key  $K_s$ , biết rằng bên kia trong phiên truyền thông là A (nhờ vào  $ID_A$ ), và biết rằng thông tin xuất phát từ KDC (vì thông tin được mã hóa bằng  $K_b$ ).

Ở thời điểm này, một session key đã được truyền an toàn đến A và B, và 2 thực thể này có thể bắt đầu truyền thông một cách an toàn. Tuy nhiên, hai bước sau đây sẽ cần được thực hiện thêm:

4. B gửi nonce, được mã hóa bằng session key vừa có được, đến A.
5. Bên cạnh đó, bằng cách sử dụng  $K_s$ , A phản hồi với  $f(N_2)$ , với  $f$  là một hàm thực hiện một phép biến đổi trên  $N_2$ , chẳng hạn như cộng thêm 1.

Những bước này giúp B đảm bảo rằng thông tin ban đầu mà nó nhận được (ở bước 3) không phải là thông điệp lặp (replay message).

Như vậy, việc phân phối khóa chỉ diễn ra ở bước 1-3, còn bước 4,5 và bước 3 đóng góp vào quá trình xác thực.

### 7.3.1.2. Kiểm soát cây phân cấp khóa

Trên thực tế, không cần thiết phải giới hạn việc phân bổ chức năng phân phối khóa ở một KDC duy nhất. Trên thực tế, với những mạng rất lớn, sẽ không thực tế nếu làm điều đó. Thay vào đó, một cây KDC có thể được thiết lập. Chẳng hạn, có thể có các KDC cục bộ, mỗi KDC chịu trách nhiệm cho một domain nhỏ trên toàn mạng. Với các phiên truyền thông giữa các thực thể trong cùng domain cục bộ, KDC của domain đó sẽ chịu trách nhiệm phân phối khóa. Nếu hai thực thể ở các domain khác nhau muốn chia sẻ khóa, các KDC cục bộ của hai thực thể này có thể truyền thông với nhau thông qua KDC toàn cục. Trong trường hợp này, một KDC bất kỳ trong số ba KDC có liên quan có thể chọn khóa.

Mô hình phân cấp có thể được mở rộng thành ba cấp hoặc nhiều hơn nữa tùy thuộc vào số lượng người dùng và đặc tính địa lý của mạng.

### 7.3.1.3 Thời gian tồn tại của session key

Session key càng thay đổi nhanh thì càng an toàn, do người tấn công có ít ciphertext hơn để dò ra một session key cụ thể nào đó. Tuy nhiên, việc phân phối session key cũng làm trì hoãn việc bắt đầu trao đổi thông tin và làm lãng phí băng thông mạng. Người

quản trị an ninh phải cân bằng những vấn đề này khi xác định thời gian sống cho một session key cụ thể nào đó.

Với các giao thức hướng kết nối, một lựa chọn khả dĩ là sử dụng chung session key trong suốt thời gian kết nối được mở và sử dụng session key mới cho mỗi phiên làm việc mới. Nếu một kết nối logic có thời gian sống quá lâu, cần thiết phải thay đổi session key định kỳ, có thể là mỗi khi số thứ tự của PDU (protocol data unit) đi hết một chu kỳ.

Với các giao thức không dựa trên kết nối, không có việc khởi tạo hay ngắt kết nối. Do đó, không có thông tin rõ ràng về việc cần thay đổi session key theo chu kỳ như thế nào. Cách tiếp cận an toàn nhất là sử dụng một session key mới cho mỗi lượt truyền tải. Tuy nhiên, điều này lại đi ngược lại một trong những điểm mạnh chủ chốt của các giao thức không dựa trên kết nối, đó là overhead và delay được giữ ở mức tối thiểu với mỗi transaction. Một chiến lược tốt hơn có thể được áp dụng là mỗi session key có một khoảng thời gian sống xác định, hoặc chỉ tồn tại trong một số transaction nhất định.

### 7.3.2. Phân phối khóa sử dụng mã hóa bất đối xứng

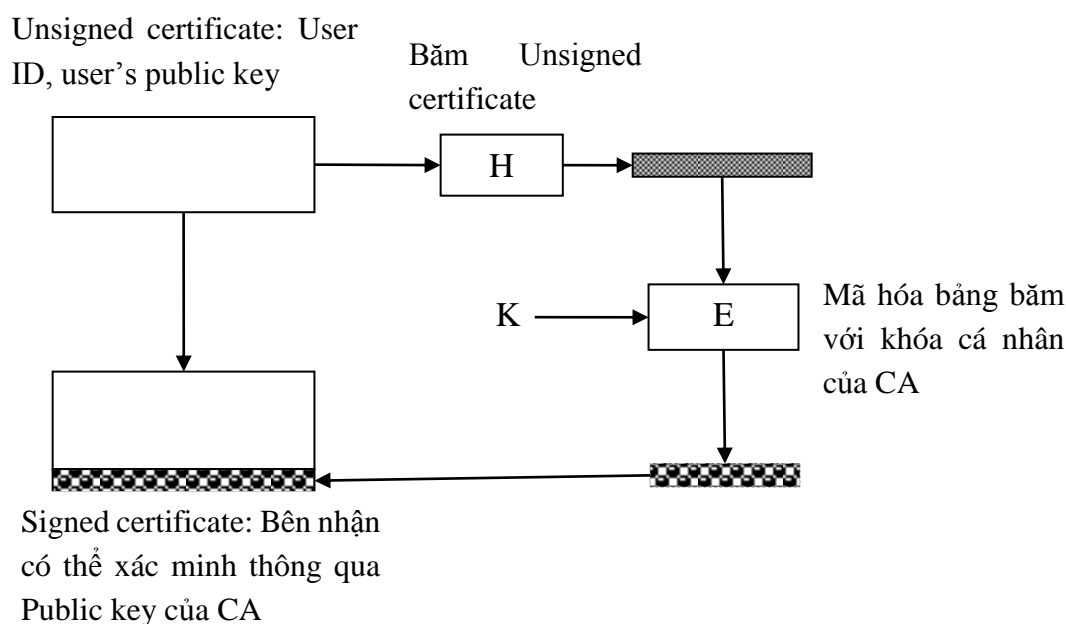
Một trong những chức năng chính của mã hóa khóa công khai là để giải quyết bài toán phân phối khóa. Trên thực tế có hai mặt tách biệt cần được xem xét khi sử dụng mã hóa bất đối xứng với mục đích này.

- Việc phân phối khóa công khai
- Việc sử dụng mã hóa khóa công khai để phân phối các khóa bí mật

#### 7.3.2.1. Chứng thực khóa công khai (Public-Key Certificate)

Do điểm cốt lõi của mã hóa khóa công khai là public key cần được công bố công khai, vì vậy nếu có một số giải thuật được chấp nhận rộng rãi, chẳng hạn RSA, một người bất kỳ có thể gửi public key của mình đến một người khác hoặc truyền broadcast khóa đó cho toàn bộ cộng đồng. Mặc dù cách tiếp cận này có vẻ tiện lợi nhưng nó có một điểm yếu lớn là bất kỳ ai cũng có thể gửi đi một thông điệp quảng bá như vậy. Điều đó có nghĩa ai đó có thể giả dạng là user A và gửi một public key đến một người khác, hoặc gửi quảng bá public key đó. Đến khi người dùng A phát hiện ra việc giả mạo đó, người giả danh đó đã có thể đọc tất cả thông điệp mã hóa dành cho A và dùng khóa giả mạo để xác thực.

Giải pháp cho vấn đề này là thẻ xác thực khóa công khai (public-key certificate). Về bản chất, một certificate bao gồm một public key cộng với user ID của người sở hữu khóa, và toàn bộ khối thông tin này được ký xác nhận bởi một bên thứ ba đáng tin cậy. Thông thường, bên thứ ba sẽ là một CA được tin cậy bởi cộng đồng người dùng. Một người dùng có thể trình public key của mình cho CA theo một cách thức an toàn và nhận một certificate. Sau đó, người dùng có thể quảng bá certificate của mình. Bất kỳ ai biết được public key của người dùng đó đều có thể lấy được certificate và xác nhận xem nó có hợp lệ không. Hình 7.5 mô tả quy trình trên.



Hình 0.5 Sử dụng Public Key Certificate

### 7.3.2.2. Phân phối khóa bí mật bằng public key

Với các cách thức mã hóa truyền thống, một yêu cầu cơ bản đối với hai thực thể tham gia truyền thông an toàn là chúng chia sẻ một khóa bí mật (secret key). Giả sử Bob muốn xây dựng một ứng dụng nhắn tin cho phép Bob gửi thư điện tử một cách an toàn đến bất kỳ ai có thể truy cập Internet hoặc vào một mạng mà cả hai đều kết nối vào được.

Giả định rằng Bob muốn hiện thực công việc này bằng mã hóa theo kiểu truyền thống. Với mã hóa truyền thống, Bob và đối tác của mình, chẳng hạn là Alice, phải có một cách thức để chia sẻ secret key mà không ai khác biết được. Làm thế nào để làm được chuyện đó nếu Alice và Bob ở xa nhau? Bob có thể mã hóa khóa này bằng một giải pháp mã hóa truyền thống và email nó cho Alice, nhưng điều này cũng có nghĩa là Bob và Alice phải đang có chung một secret key để mã hóa secret key mới này. Thêm vào đó, Bob và bất kỳ ai dùng email để truyền khác cũng sẽ phải đối mặt với một vấn đề: mỗi cặp thực thể tham gia truyền thông phải dùng chung một cặp khóa bí mật duy nhất.

Một giải pháp cho vấn đề này là truyền khóa sử dụng giải thuật Diffie-Hellman. Trên thực tế, cách tiếp cận này đang được sử dụng rộng rãi. Tuy nhiên, có một điểm yếu của giải pháp này: ở dạng đơn giản nhất của mình thì Diffie-Hellman không hỗ trợ xác thực hai bên tham gia truyền thông.

Một giải pháp khác tối ưu hơn là thẻ chứng thực khóa công khai. Khi Bob muốn truyền thông với Alice, Bob có thể làm những điều sau:

1. Chuẩn bị bản tin cần truyền.
2. Mã hóa bản tin đó bằng một phương pháp mã hóa truyền thống, sử dụng một session key dùng một lần.

3. Mã hóa session key bằng cách dùng mã hóa khóa công khai, sử dụng public key của Alice.
4. Đính kèm session key đã được mã hóa vào bản tin và gửi cho Alice.

Chỉ Alice mới có khả năng giải mã session key và sau đó khôi phục lại thông điệp ban đầu. Nếu Bob lấy được public key của Alice bằng cách sử dụng public-key certificate của Alice, thì Bob có thể bảo đảm rằng đó là một khóa hợp lệ.

#### **7.4. Thẻ chứng thực X.509**

X.509 là một phần của họ chuẩn X.500, bao gồm các chuẩn giúp định nghĩa dịch vụ thư mục (directory service). Thư mục là một server hoặc một tập hợp server phân tán lưu giữ một cơ sở dữ liệu thông tin người dùng. Các thông tin đó bao gồm cả phép ánh xạ từ tên người dùng thành địa chỉ mạng, cũng như các thuộc tính khác chứa thông tin về người dùng.

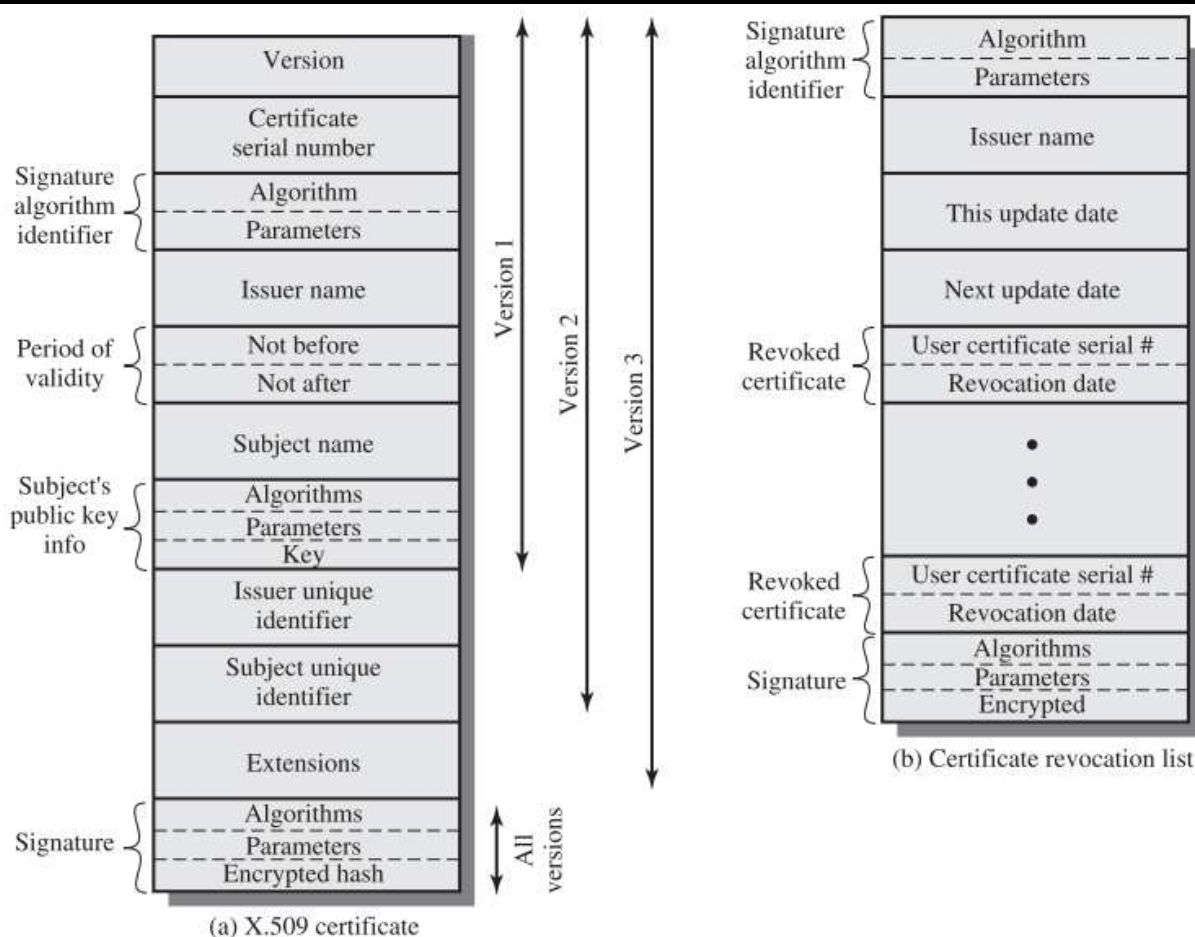
X.509 định nghĩa một framework làm nền tảng cho dịch vụ xác thực (authentication service) mà thư mục X.500 cung cấp cho người dùng. Thư mục có thể đóng vai trò là một repository chứa các thẻ chứng thực khóa công khai. Mỗi thẻ chứng thực chứa public key của một người dùng và được ký bằng private key của một CA tin cậy được. Thêm vào đó, X.509 định nghĩa các giao thức xác thực dự phòng khác dựa trên việc sử dụng public-key certificate.

X.509 là một chuẩn quan trọng vì cấu trúc certificate và các giao thức xác thực được định nghĩa trong X.509 được sử dụng trong nhiều ngữ cảnh khác nhau. Chẳng hạn, định dạng thẻ chức thực X.509 được dùng trong S/MIME, IP Security và SSL/TLS.

X.509 sử dụng mã hóa khóa công khai và chữ ký số. Chuẩn không chỉ định rõ phải sử dụng giải thuật gì nhưng đề xuất dùng RSA. Mô hình chữ ký số đòi hỏi việc sử dụng một hàm băm (hash function). Một lần nữa, chuẩn không bắt buộc phải sử dụng giải thuật băm nào. Phiên bản chuẩn năm 1988 có đề xuất một giải thuật băm; tuy nhiên, giải thuật này đã được chứng minh là không an toàn và bị bỏ đi trong phiên bản X.509 1993.

##### **7.4.1.Certificate**

Phần cốt lõi của mô hình X.509 là public-key certificate ứng với mỗi người dùng. Các thẻ chứng thực người dùng này được giả định rằng đã được tạo ra bởi một CA đáng tin cậy nào đó và được CA hoặc người dùng đặt trong directory. Bản thân máy chủ dịch vụ thư mục không chịu trách nhiệm về việc tạo public key hoặc tính năng chứng thực (certification) của hệ thống; nó chỉ cung cấp môi trường để người dùng nhận certificate.



Hình 0.6 Định dạng X.509

Hình 7.6 mô tả định dạng tổng quát của một certificate, bao gồm các thành phần sau:

- **Version:** Phiên bản định dạng certificate. Mặc nhiên là phiên bản 1. Nếu có Issuer Unique Identifier hoặc Subject Unique Identifier thì giá trị phiên bản phải là 2. Nếu có 1 hoặc nhiều extension, phiên bản sẽ là 3.
- **Serial number:** đây là 1 giá trị số nguyên duy nhất trong phạm vi quản lý bởi CA cấp certificate đang xét.
- **Signature algorithm identifier:** Giải thuật dùng để ký certificate cùng với các tham số nếu có. Vì thông tin này được lặp lại ở trường Signature ở cuối certificate, trường này ít có ý nghĩa thực tiễn.
- **Issuer name:** Định danh X.500 của CA tạo và ký certificate.
- **Period of validity:** Bao gồm hai ngày: ngày đầu và ngày cuối xác định khoảng thời gian hợp lệ của certificate.
- **Subject name:** Tên của user mà certificate đề cập đến. Cụ thể, certificate này xác nhận public key của chủ thể đang có private key tương ứng.
- **Subject's public-key information:** Public key của chủ thể, cộng với phần định danh của giải thuật mà khóa này sẽ được sử dụng, cùng với các tham số liên quan.



- Issuer unique identifier: Đây là một trường tùy chọn, chứa 1 chuỗi xác định CA cấp certificate trong trường hợp tên X.500 được dùng lại cho các thực thể khác.
- Subject unique identifier: Đây là một trường tùy chọn, chứa 1 chuỗi xác định chủ thể (subject) trong trường hợp tên X.500 được dùng lại cho các thực thể khác.
- Extensions: Một tập hợp gồm 1 hoặc nhiều trường mở rộng. Các extension được thêm vào ở phiên bản 3.
- Signature: Mô tả tất cả các trường khác của certificate; bao gồm cả hash code của các field khác được mã hóa bằng private key của CA. Trường này cũng chứa signature algorithm identifier.

Các trường định danh duy nhất được thêm vào ở phiên bản 2 để giải quyết vấn đề tái sử dụng tên subject và/hoặc tên issuer theo thời gian. Trên thực tế, các trường này ít được dùng đến.

Chuẩn sử dụng cấu trúc sau để định nghĩa một certificate:

$$CA \ll A \gg = CA \{V, SN, AI, CA, UCA, A, UA, A_p, T^A\}$$

với

- $Y \ll X \gg$  : certificate của user  $X$  được cấp bởi CA có tên  $Y$
- $Y \{I\}$  : phần chữ ký của  $I$  bởi  $Y$ ; bao gồm  $I$  và một hash code được mã hóa.
- $V$  : phiên bản của certificate
- $SN$  : serial number của certificate
- $AI$  : định dạng của giải thuật (algorithm identifier) được sử dụng để ký certificate
- $CA$  : tên của certificate authority
- $UCA$  : định danh duy nhất của CA (tùy chọn)
- $A$  : tên của user  $A$
- $UA$  : định danh duy nhất của user  $A$  (tùy chọn)
- $A_p$  : public key của user  $A$
- $T^A$  : thời gian hợp lệ của certificate

CA ký certificate bằng private key của mình. Nếu một user biết public key tương ứng với private key này thì user đó có thể xác nhận rằng một certificate được ký bởi CA đang đề cập là hợp lệ. Đây là cách tiếp cận điển hình khi vận hành chữ ký số đã được mô tả ở chương 6.

#### 7.4.2. Nhận certificate người dùng

Các thẻ chứng thực người dùng được phát sinh bởi CA có những thuộc tính sau:

- Bất kỳ người dùng nào có quyền truy xuất public key của CA đều có thể xác thực rằng public key của người dùng đã được xác thực.
- Không bên nào, ngoại trừ CA, có thể sửa đổi certificate mà không bị phát hiện.

Vì các certificate có tính không thể che giấu, chúng có thể được đặt trong một thư mục mà thư mục đó không cần phải làm gì đặc biệt để bảo vệ chúng.

Nếu tất cả người dùng cùng đăng ký vào một CA, câu chuyện sẽ dễ dàng vì tất cả user đó đều tin cậy CA. Tất cả user certificate có thể được lưu trong thư mục để truy xuất về sau (bởi tất cả user). Thêm vào đó, một user có thể truyền certificate của mình trực tiếp đến người khác. Trong mọi trường hợp, khi B đã có được certificate của A, B có thể tin tưởng rằng những thông điệp được mã hóa bằng public key của A sẽ được an toàn khỏi các loại hình tấn công eavesdropping và rằng thông điệp được ký bởi private key của A là không thể bị can thiệp.

Nếu số lượng người dùng lớn thì sẽ có khả năng không phải tất cả người dùng đều đăng ký vào chung một CA. Vì CA sẽ ký vào các certificate, mỗi user phải có một bản sao của public key to của CA đó để tiến hành xác thực các chữ ký đó. Khóa public này phải được cung cấp cho từng người dùng theo một cách thức tuyệt đối an toàn nào đó (có quan tâm đến tính toàn vẹn và tính xác thực), nhờ đó người dùng có thể tin cậy certificate liên quan. Do đó, với nhiều người dùng, có thể sẽ có nhiều CA và mỗi CA thì chịu trách nhiệm phân phối an toàn public key của mình đến một phần trong tổng số người dùng.

Giả sử A đã nhận được một certificate từ certification authority  $X_1$  và B nhận được một certificate từ CA  $X_2$ . Nếu A không biết được public key của  $X_2$  theo một cách thức an toàn thì certificate của B, vốn được cấp bởi  $X_2$ , hoàn toàn vô dụng đối với A. A có thể đọc certificate của B, nhưng A không thể xác thực chữ ký. Tuy nhiên, nếu hai CA đã trao đổi public key cho nhau một cách an toàn thì quy trình sau sẽ cho phép A nhận public key của B

1. A lấy certificate (từ directory) của  $X_2$ ; certificate này được ký bởi  $X_1$ . Vì A biết được public key của  $X_1$  một cách an toàn, A có thể lấy public key của  $X_2$  từ certificate của nó và xác thực bằng chữ ký của  $X_1$  trên certificate đó.
2. Sau đó, A trở lại thư mục và lấy certificate của B, vốn được ký bởi  $X_2$ . Vì bây giờ A đã có một bản sao đáng tin cậy của public key của  $X_2$ , A có thể xác thực chữ ký và nhận public key của B một cách an toàn.

A đã sử dụng một chuỗi (chain) certificate để có được public key của B. Theo X.509, chuỗi này được biểu diễn như sau:

$$X_1 \ll X_2 \gg X_2 \ll B \gg$$

Theo cách thức tương tự, B có thể lấy được public key của A với chuỗi đảo ngược:

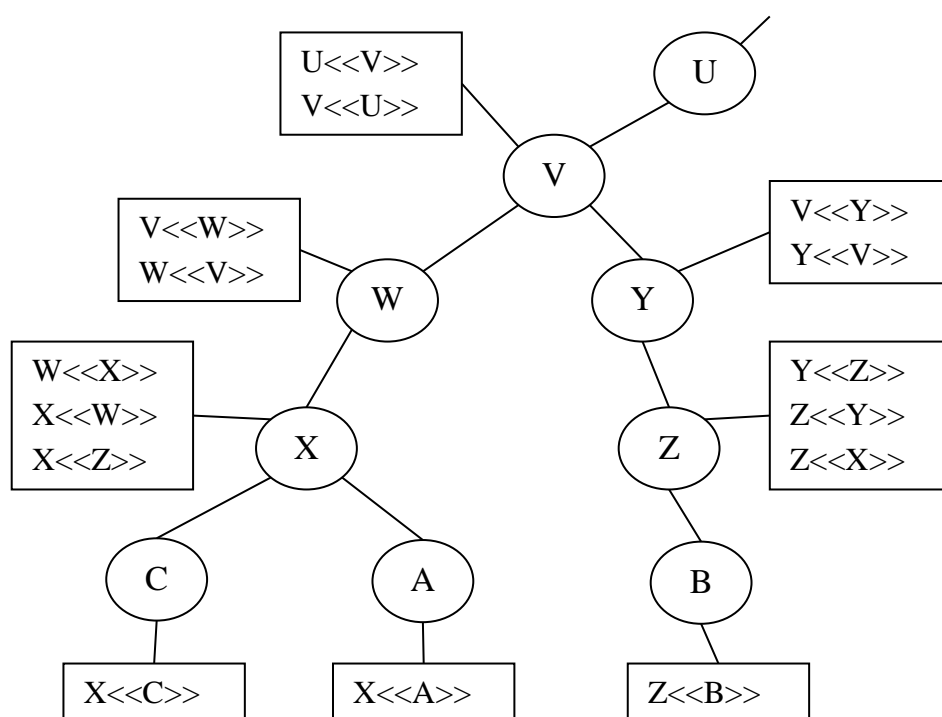
$$X_2 \ll X_1 \gg X_1 \ll A \gg$$

Mô hình này không giới hạn số certificate trong 1 chuỗi phải nằm ở con số 2. Một chuỗi gồm N thành phần được biểu diễn như sau:

$$X_1 \ll X_2 \gg X_2 \ll X_3 \gg \dots X_N \ll B \gg$$

Trong trường hợp này, mỗi cặp CA trong chuỗi  $(X_i, X_{i+1})$  phải tạo certificate cho nhau.

Tất cả certificate của CA được cấp bởi CA cần phải xuất hiện trong thư mục, và người dùng cần phải biết chúng được liên kết như thế nào trên đường dẫn đi đến public-key certificate của một người dùng khác. X.509 đề xuất rằng các CA có thể được sắp xếp theo dạng cây để việc duyệt qua các node trên cây được thuận lợi và thống nhất.



Hình 0.7 Ví dụ về cây X.509

Trong hình 7.8 minh họa các hình tròn được kết nối với nhau thể hiện mối liên hệ giữa các CA; các hộp thông tin liên quan chỉ ra các certificate được lưu trữ trong thư mục ứng với mỗi CA. Mục thông tin thư mục của mỗi CA bao gồm hai loại certificate:

- Forward certificate: Các certificate của X được phát sinh bởi các CA khác
- Reverse certificate: Các certificate dành cho các CA khác được phát sinh bởi X.

Trong ví dụ này, user A có thể yêu cầu các certificate sau từ directory để xây dựng đường xác thực đến B:

$$X \ll W \gg W \ll V \gg V \ll Y \gg Y \ll Z \gg Z \ll B \gg$$



### 7.4.3. Hủy certificate

Mỗi certificate có chứa thông tin về thời hạn có hiệu lực. Thông thường, một certificate mới sẽ được cấp ngay trước khi certificate cũ hết hạn. Thêm vào đó, có những nguyên nhân, được liệt kê dưới đây, sẽ dẫn đến chuyện hủy certificate trước khi nó hết hạn:

1. Private key của người dùng được xác định là đã bị chiếm đoạt.
2. Người dùng không còn được xác thực bởi CA, nguyên nhân có thể bao gồm việc đổi tên người dùng, certificate bị thay thế hoặc certificate không phù hợp với chính sách của CA
3. Certificate của CA được xác định là đã bị chiếm đoạt

Mỗi CA phải lưu giữ một danh sách tất cả các certificate đã bị xóa nhưng chưa hết hạn được cấp bởi CA đó, bao gồm cả các certificate cấp cho user và certificate cấp cho các CA khác. Danh sách này cần được thông báo trên directory.

Mỗi danh sách certificate bị xóa (Certificate Revocation List - CRL) được đưa lên directory đều được ký bởi thực thể tạo ra nó và bao gồm tên đối tượng tạo CRL (issuer name), ngày danh sách được tạo ra, ngày CRL kế tiếp dự định được tạo ra, và một mục cho mỗi certificate bị hủy. Mỗi mục bao gồm serial number của certificate và ngày xóa certificate đó. Vì các serial number là duy nhất trong mỗi CA, chỉ serial number là đủ để xác định certificate.

Khi một người dùng nhận một certificate trong một thông điệp nào đó, người dùng đó phải xác định xem certificate đó đã bị hủy hay chưa. Do đó, người dùng cần kiểm tra directory mỗi khi nhận được một. Để tránh thời gian trễ (và có thể là chi phí) khi tìm kiếm trên directory, mỗi user có thể lưu trữ một bản cache cục bộ chứa các certificate và danh sách các certificate bị hủy.

## 7.5. Kerberos

### 7.5.1. Giới thiệu

Kerberos là một dịch vụ phân phối khóa và xác thực người dùng được phát triển bởi Viện Công nghệ Massachusetts (Massachusetts Institute of Technology - MIT). Bài toán Kerberos giải quyết được phát biểu như sau:

Giả sử trong một môi trường phân tán mở, trong đó người dùng ở các máy trạm (workstation) muốn truy cập vào dịch vụ trên các máy chủ phân bố trên mạng. Chúng ta muốn các máy chủ có khả năng giới hạn truy cập với những người dùng đã được cấp quyền và có khả năng xác thực các yêu cầu dịch vụ. Trong môi trường này, một workstation không thể được xem là có thể tin cậy được trong việc xác định người dùng đang kết nối vào dịch vụ mạng. Cụ thể, có ba mối nguy hiểm sau tồn tại:

1. Một người dùng có thể có quyền truy cập vào một workstation cụ thể và giả dạng như người dùng sở hữu workstation đó.

2. Một người dùng có thể sửa đổi địa chỉ mạng của một workstation, nhờ thế request gửi từ địa chỉ bị can thiệp trông có vẻ đến từ một máy hợp lệ.
3. Một người dùng có thể nghe lén các thông tin trao đổi và sử dụng replay attack để thâm nhập vào server hoặc làm gián đoạn công việc.

Để giải quyết những vấn đề trên, thay vì xây dựng các giao thức xác thực phức tạp ở từng server, Kerberos cung cấp một server xác thực tập trung, có chức năng hỗ trợ các server xác thực người dùng cũng như hỗ trợ người dùng xác thực server.

Kerberos dựa hoàn toàn vào mã hóa đối xứng và không sử dụng mã hóa khóa công khai. Hiện nay phiên bản 5 (cập nhật từ phiên bản 4 một số vấn đề an ninh) của Kerberos được đề xuất làm một chuẩn Internet (RFC 4120).

### 7.5.2.Kerberos Version 4

#### 7.5.2.1.Mô hình truyền thông đơn giản

Trong một môi trường mạng không được bảo vệ, bất kỳ máy client nào cũng có thể đăng ký với một server bất kỳ để sử dụng dịch vụ của nó. Vấn đề rủi ro an ninh hiện hiện là sự mạo danh (impersonation). Một kẻ tấn công có thể giả dạng một người sử dụng khác và lấy được quyền truy cập vào các server. Để chống lại mối nguy cơ này, các server phải có khả năng xác nhận định danh của các client muốn yêu cầu dịch vụ. Mỗi server có thể thực hiện việc xác thực này với mỗi giao dịch client/server, tuy nhiên điều này sẽ là gánh nặng với server trong môi trường mở.

Một giải pháp thay thế là sử dụng máy chủ xác thực (Authentication Server - AS). Máy chủ này biết mật khẩu của tất cả người dùng và lưu trữ chúng trong một cơ sở dữ liệu tập trung. Thêm vào đó, AS chia sẻ một khóa bí mật duy nhất với mỗi server. Việc phân phối khóa này có thể được tiến hành theo kiểu phân phối vật lý hoặc theo một cách an toàn khác. Xem xét mô hình trao đổi sau:

$$\begin{aligned}
 (1) & C \rightarrow AS : ID_C \square P_C \square ID_V \\
 (2) & AS \rightarrow C : Ticket \\
 (3) & C \rightarrow V : ID_C \square Ticket \\
 Ticket &= E(K_v, [ID_C \square AD_C \square ID_V])
 \end{aligned}$$

Với:

C	:	máy client
AS	:	authentication server
V	:	server
ID <sub>C</sub>	:	định danh của user trên C
ID <sub>V</sub>	:	định danh của V
P <sub>C</sub>	:	password của user trên C

$AD_C$  : network address của C

$K_V$  : khóa mã hóa bí mật được chia sẻ bởi AS và V

Lưu ý trong mô hình trên, phần bên trái dấu “.” xác định bên gửi và bên nhận, phần bên phải thể hiện nội dung của thông điệp, và biểu tượng □ đại diện cho thao tác nối chuỗi.

Trong ngữ cảnh này, người dùng đăng nhập vào workstation và yêu cầu truy cập đến server V. Module người dùng C trên máy trạm yêu cầu người dùng nhập mật khẩu và sau đó gửi một thông điệp đến AS, thông điệp này bao gồm user ID, server ID, và mật khẩu người dùng. AS kiểm tra database của mình để xem mật khẩu do người dùng cung cấp có phù hợp với user ID không, cũng như người dùng đó có quyền truy cập vào server V hay không. Nếu cả hai phép thử này đều được vượt qua, AS chấp nhận rằng người dùng là xác thực và phải thuyết phục server tin vào điều đó. Để làm vậy, AS tạo ra một thẻ (ticket) chứa user ID cùng với địa chỉ mạng và server ID. Thẻ này được mã hóa bằng khóa bí mật được chia sẻ giữa AS và server này. Sau đó, thẻ này được gửi ngược lại C.

Vì ticket được mã hóa, nên ticket không thể bị can thiệp, sửa đổi bởi C hoặc bởi một kẻ tấn công nào đó. Với ticket này, giờ đây C có thể đăng ký với V để sử dụng dịch vụ. C gửi một thông điệp đến V; thông điệp này bao gồm ticket và ID của C. V giải mã ticket và xác nhận xem user ID trong ticket có giống như user ID (không mã hóa) trong thông điệp hay không. Nếu kết quả so sánh là khớp, server sẽ xem như người dùng đã được xác thực và tiến hành gán quyền truy cập dịch vụ.

Mỗi thành phần trong thông điệp (3) đều quan trọng. Ticket được mã hóa để tránh việc bị can thiệp hay ăn cắp. Server ID ( $ID_V$ ) được thêm vào để nhờ đó server có thể kiểm tra rằng mình đã giải mã ticket chính xác.  $ID_C$  được thêm vào ticket để chỉ ra rằng ticket đã được cấp dựa trên yêu cầu của C. Cuối cùng,  $AD_C$  có nhiệm vụ chống lại nguy cơ sau đây. Một người tấn công có thể bắt được ticket truyền trong thông điệp (2), sau đó dùng tên  $ID_C$ , và tạo một thông điệp có vẻ giống như (3) để truyền đi từ một workstation khác. Do đó, server có thể nhận được một ticket hợp lệ phù hợp với user ID và gán quyền truy cập cho người dùng trên máy khác với máy cần gán quyền. Để ngăn chặn dạng tấn công này, AS thêm vào ticket địa chỉ mạng mà từ đó thông điệp yêu cầu ban đầu thực sự được gửi đi. Do vậy, ticket chỉ hợp lệ khi nó được gửi đi từ cùng workstation với workstation đã yêu cầu ticket.

#### **7.5.2.2. Một ngữ cảnh an toàn hơn**

Mặc dù mô hình ở trên đã giải quyết được một số vấn đề xác thực trong môi trường mạng mở, các vấn đề đó vẫn tồn tại.

Đầu tiên, cần giảm thiểu số lần người dùng phải nhập mật khẩu, giả định rằng một ticket chỉ có thể được dùng một lần. Nếu người dùng C đăng nhập vào một máy trạm

vào buổi sáng và muốn kiểm tra mail tại máy chủ email, C phải cung cấp mật khẩu để nhận một ticket cho mail server. Nếu C muốn kiểm tra mail nhiều lần trong ngày, mỗi lần kiểm tra như vậy người dùng sẽ phải nhập mật khẩu một lần. Để cải thiện tình hình trong tình huống này bằng cách cho phép các ticket được tái sử dụng.

Với một phiên đăng nhập, workstation có thể lưu trữ ticket cho mail server sau khi nhận được nó và sử dụng cho việc truy cập nhiều lần vào mail server. Tuy nhiên, trong ngữ cảnh này, vẫn còn khả năng một người dùng cần ticket mới cho mỗi dịch vụ khác.

Vấn đề thứ hai là ngữ cảnh vừa xét có truyền tải mật khẩu dưới dạng văn bản thô [message (1)]. Một kẻ nghe lén có thể bắt được thông điệp chứa password.

Để giải quyết những vấn đề này, giải pháp được đưa ra là sử dụng một mô hình mới để tránh việc truyền mật khẩu plain-text và dùng một server mới, gọi là server gán ticket (ticket-granting server - TGS). Mô hình mới được mô tả dưới đây:

**Thực thi một lần cho mỗi phiên đăng nhập:**

$$(1) C \rightarrow AS : ID_C \sqcap ID_{tgs}$$

$$(2) AS \rightarrow C : E(K_C, Ticket_{tgs})$$

**Thực thi một lần cho mỗi loại dịch vụ:**

$$(3) C \rightarrow TGS : ID_C \sqcap ID_V \sqcap Ticket_{tgs}$$

$$(4) TGS \rightarrow C : Ticket_v$$

**Thực thi một lần cho mỗi phiên dịch vụ:**

$$(5) C \rightarrow V : ID_C \sqcap Ticket_v$$

$$Ticket_{tgs} = E(K_{tgs}, [ID_C \sqcap AD_C \sqcap ID_{tgs} \sqcap TS_1 \sqcap Lifetime_1])$$

$$Ticket_v = E(K_v, [ID_C \sqcap AD_C \sqcap ID_v \sqcap TS_2 \sqcap Lifetime_2])$$

Dịch vụ mới, TGS, cấp ticket cho người dùng đã được xác thực bởi AS. Do đó, đầu tiên người dùng phải yêu cầu một ticket-granting ticket ( $Ticket_{tgs}$ ) từ AS. Client module trên workstation người dùng lưu lại ticket này. Mỗi khi user muốn truy cập vào một dịch vụ mới, client đăng ký với TGS, sử dụng ticket để xác thực bản thân mình. Sau đó, TGS gán một ticket cho dịch vụ tương ứng. Client lưu mỗi service-granting ticket và sử dụng nó để xác thực người dùng của mình với server mỗi khi một dịch vụ cụ thể được yêu cầu.

Mô tả cụ thể các bước của mô hình trên như sau:

1. Client yêu cầu một ticket-granting ticket cho người dùng bằng việc gửi thông điệp chứa user ID cho AS, cùng với TGS ID để cho thấy đây là thông điệp yêu cầu dịch vụ của TGS.

2. AS phản hồi với một ticket được mã hóa bằng một khóa tạo ra từ mật khẩu người dùng ( $K_C$ ), khóa này vốn đã được lưu ở AS. Khi thông điệp phản hồi này đến được client, client thông báo cho user để yêu cầu user nhập mật khẩu, tạo ra khóa và tiến hành giải mã thông điệp vừa nhận được. Nếu mật khẩu vừa được cung cấp là đúng thì ticket sẽ được phục hồi thành công.
3. Client yêu cầu một thẻ gán dịch vụ (service-granting ticket) cho user. Để đạt được mục đích này, client gửi một thông điệp đến TGS có chứa user ID, ID của dịch vụ muốn truy cập, và thẻ gán ticket (ticket-granting ticket).
4. TGS giải mã ticket vừa đến bằng cách sử dụng một khóa chia sẻ duy nhất giữa AS và TGS ( $K_{TGS}$ ) và xác nhận việc giải mã thành công bằng việc tồn tại của ID của nó. TGS kiểm tra để bảo đảm rằng thời gian sống (lifetime) chưa hết. Sau đó, nó so sánh user ID và địa chỉ mạng với với thông tin nhận được để xác thực xem người dùng có đúng không. Nếu người dùng được phép truy cập vào server V, TGS cấp một ticket để gán quyền truy cập vào dịch vụ đang được yêu cầu.
5. Client yêu cầu truy cập vào một dịch vụ thay mặt cho người dùng. Để làm điều này, client truyền đi một thông điệp đến server chứa user ID và service-granting ticket. Server xác thực bằng cách sử dụng nội dung của ticket.

Mô hình mới này thoả hai yêu cầu: chỉ một lần truy vấn mật khẩu cho một phiên làm việc của người dùng và bảo vệ được mật khẩu người dùng.

#### 7.5.2.3. Mô hình xác thực của Kerberos version 4

Mặc dù mô hình ở phần trên đã có bước tiến triển về mặt bảo mật, còn hai vấn đề đang tồn tại.

Vấn đề đầu tiên là thời gian sống (lifetime) ứng với ticket-granting ticket. Nếu lifetime quá ngắn (chẳng hạn, bằng vài phút), thì user sẽ bị hỏi password một cách thường xuyên. Nếu lifetime quá dài (giả sử là vài tiếng đồng hồ), thì kẻ tấn công opponent sẽ có nhiều cơ hội để tấn công theo kiểu lặp (replay). Kẻ tấn công có thể lắng nghe các gói tin trên mạng và bắt một bản sao của ticket-granting ticket và sau đó chờ đến khi người dùng hợp lệ đăng xuất. Để rồi, người tấn công có thể giả dạng địa chỉ mạng của người dùng hợp lệ, và gửi thông điệp ở bước (3) đến TGS. Hệ quả là người tấn công có quyền truy cập vô hạn đến các tài nguyên cấp cho người dùng hợp lệ đó. Tương tự, nếu một kẻ tấn công bắt được service-granting ticket và sử dụng nó trước khi thẻ này hết hạn, kẻ tấn công sẽ có quyền truy cập đến dịch vụ tương ứng.

Vì thế, cần có yêu cầu bổ sung: một dịch vụ mạng (TGS hoặc một dịch vụ ứng dụng nào khác) phải có khả năng chứng minh rằng người đang sử dụng ticket cũng là người mà ticket đó được cấp cho.

Vấn đề thứ hai là có thể có yêu cầu đặt ra là các server phải có khả năng tự xác thực mình với người dùng. Không có dạng xác thực đó, một kẻ tấn công có thể phá hoại,

chỉnh sửa cấu hình để các thông điệp gửi đến một server cụ thể bị chuyển hướng đến một địa điểm khác. Server giả mạo sẽ có cơ hội hành động như một server thực thụ, có thể lấy được các thông tin của người dùng và từ chối cung cấp dịch vụ thực sự cho họ.

Chúng ta sẽ kiểm tra lần lượt các vấn đề này trong Bảng 7.1. Bảng này thể hiện giao thức Kerberos thực sự.

<p>(1) <math>C \rightarrow AS : ID_c \parallel ID_{tgs} \parallel TS_1</math></p> <p>(2) <math>AS \rightarrow C : E\left(K_{c,tgs}, \left[ K_{c,tgs} \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2 \parallel Ticket_{tgs} \right]\right)</math></p> $Ticket_{tgs} = E\left(K_{tgs}, \left[ K_{c,tgs} \parallel ID_c \parallel AD_c \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2 \right]\right)$ <p>(a) Authentication Service Exchange để nhận ticket-granting ticket</p>
<p>(3) <math>C \rightarrow TGS : ID_v \parallel Ticket_{tgs} \parallel Authenticator_c</math></p> <p>(4) <math>TGS \rightarrow C : E\left(K_{c,tgs}, \left[ K_{c,v} \parallel ID_v \parallel TS_4 \parallel Ticket_v \right]\right)</math></p> $Ticket_{tgs} = E\left(K_{tgs}, \left[ K_{c,tgs} \parallel ID_c \parallel AD_c \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2 \right]\right)$ $Ticket_v = E\left(K_v, \left[ K_{c,v} \parallel ID_c \parallel AD_c \parallel ID_v \parallel TS_4 \parallel Lifetime_4 \right]\right)$ $Authenticator_c = E\left(K_{c,tgs}, \left[ ID_c \parallel AD_c \parallel TS_3 \right]\right)$ <p>(b) Ticket-Granting Service Exchange để nhận service-granting ticket</p>
<p>(5) <math>C \rightarrow V : Ticket_v \parallel Authenticator_c</math></p> <p>(6) <math>V \rightarrow C : E\left(K_{c,v}, \left[ TS_5 + 1 \right]\right)</math></p> $Ticket_v = E\left(K_v, \left[ K_{c,v} \parallel ID_c \parallel AD_c \parallel ID_v \parallel TS_4 \parallel Lifetime_4 \right]\right)$ $Authenticator_c = E\left(K_{c,v}, \left[ ID_c \parallel AD_c \parallel TS_5 \right]\right)$ <p>(c) Trao đổi thông tin xác thực Client/Server để sử dụng dịch vụ</p>

*Bảng 0.1. Các thông điệp được trao đổi trong Kerberos Version 4*

Đầu tiên, vấn đề các thẻ ticket-granting bị bắt trên đường truyền và sự cần thiết phải xác định người trình ra ticket cũng là client mà ticket được cấp cho. Mỗi đe dọa là một kẻ tấn công sẽ cướp ticket và sử dụng trước khi ticket này hết hạn.

Để giải quyết vấn đề này, đầu tiên, AS sẽ cấp cho cả client và TGS một gói thông tin bí mật theo một cách thức an toàn. Sau đó, client có thể chứng minh định danh của mình với TGS bằng cách tiết lộ thông tin bí mật theo một cách thức an toàn. Một cách

hiệu quả để đạt được điều này là sử dụng một khóa mã hóa làm thông tin bí mật. Khóa này được gọi là khóa phiên (session key) trong Kerberos.

Bảng 7.1a thể hiện kỹ thuật phân phối session key. Như trước đây, client gửi một message đến AS để yêu cầu quyền truy cập TGS. AS phản hồi bằng một message, được mã hóa bằng khóa được tạo ra từ mật khẩu của user ( $K_c$ ), có chứa ticket. Thông điệp mã hóa này cũng chứa một bản sao của session key,  $K_{c,tgs}$ , với phần chỉ số dưới thể hiện đây là session key của C và TGS. Vì session key này nằm trong thông điệp được mã hóa bằng  $K_c$ , chỉ user client có thể đọc nó. Session key này cũng được đóng gói trong ticket, vốn chỉ có thể được đọc bởi TGS. Do đó, session key đã được truyền một cách an toàn đến cả C và TGS.

Lưu ý rằng nhiều thông tin cộng thêm cũng được thêm vào trong giai đoạn trao đổi đầu tiên này. Message (1) có chứa một timestamp, nhờ đó AS biết rằng message hợp lệ về mặt thời gian. Message (2) chứa nhiều thành phần của ticket dưới dạng mà C có thể truy cập được. Điều này cho phép C xác thực rằng ticket là dành cho TGS và biết được thời hiệu của nó.

Được trang bị ticket và session key, C đã sẵn sàng để tiếp cận TGS. C gửi đến TGS một thông điệp có chứa ticket cộng với ID của dịch vụ được yêu cầu (message (3) trong Bảng 7.1b). Thêm vào đó, C gửi một thông điệp xác thực (authenticator) có chứa ID và địa chỉ của người dùng trên C và một timestamp. Không giống như ticket, vốn có thể tái sử dụng, authenticator được thiết kế để chỉ có thể sử dụng và có thời gian sống rất ngắn. TGS có thể giải mã ticket bằng khóa mà nó chia sẻ với AS. Ticket này chỉ ra rằng người dùng C đã được cung cấp session key  $K_{c,tgs}$ . Về mặt thông điệp, ticket sẽ nói rằng, “Bất kỳ ai sử dụng  $K_{c,tgs}$  thì phải là C”. TGS sử dụng session key để giải mã authenticator. Sau đó, TGS có thể kiểm tra tên và địa chỉ lấy từ authenticator với thông tin tương ứng trên ticket và với địa chỉ mạng của thông điệp đến. Nếu tất cả đều trùng khớp, TGS coi như chắc chắn rằng phía gửi ticket thực sự là người sở hữu thực sự của ticket đó. Về mặt thông điệp, authenticator sẽ nói rằng, “Ở thời điểm  $TS_3$ , tôi sử dụng  $K_{c,tgs}$ ”. Lưu ý rằng ticket không chứng minh về định danh của bất cứ ai nhưng là một cách để truyền khóa một cách an toàn. Nhiệm vụ chứng minh định danh là của authenticator. Bởi vì authenticator chỉ có thể được dùng một lần và có thời gian sống ngắn, nguy cơ có một kẻ tấn công nào đó lấy được cả ticket và authenticator và sử dụng về sau được giải quyết.

Thông điệp phản hồi từ TGS trong message (4) có khuôn dạng giống như message (2). Message được mã hóa bằng session key được chia sẻ bởi TGS và C và bao gồm một session key được chia sẻ bởi C và server V, ID của V, và timestamp của ticket. Bản thân ticket cũng bao gồm một session key giống như vậy.



Giờ đây C có một service-granting ticket tái sử dụng được cho V. Khi C trình ra ticket này, ở message (5), nó cũng gửi đi một thông điệp xác thực (authenticator). Server có thể giải mã ticket, khôi phục session key, và giải mã authenticator.

Nếu cần thiết phải có xác thực lẫn nhau (mutual authentication), server có thể phản hồi theo message (6) ở Bảng 7.1. Server trả về giá trị của timestamp lấy từ authenticator, cộng thêm 1, và mã hóa trong session key. C có thể giải mã thông điệp này để lấy ra timestamp đã được cộng thêm. Vì message được mã hóa bằng session key, C được bảo đảm rằng nó chỉ có thể được tạo ra bởi V. Nội dung của message cam đoan với C rằng đây không phải là thông điệp replay của một thông điệp phản hồi trước đó.

Cuối cùng, để kết luận cho quy trình này, client và server chia sẻ một secret key. Khóa này được sử dụng để mã hóa các thông điệp trong tương lai giữa hai bên hoặc để trao đổi một session key ngẫu nhiên mới cho mục đích đó.

Bảng 7.2. tóm tắt các mô tả về mỗi thành phần trong giao thức Kerberos.

<b>Message (1)</b>	Client yêu cầu ticket-granting ticket.
$ID_C$	Báo cho AS biết định danh user trên client.
$ID_{tgs}$	Báo cho AS biết user yêu cầu truy cập đến TGS.
$TS_1$	Cho phép AS xác định xem đồng hồ ở client clock có đồng bộ với đồng hồ ở AS không.
<b>Message (2)</b>	AS trả về ticket-granting ticket.
$K_c$	Mã hóa dựa trên user password, cho phép AS và client xác nhận mật khẩu, và bảo vệ nội dung của message (2).
$K_{c,tgs}$	Bản sao session key mà client có thể truy cập được, được tạo bởi AS để cho phép truyền thông an toàn giữa client và TGS mà không đòi hỏi chúng phải chia sẻ một khóa vĩnh viễn.
$ID_{tgs}$	Xác nhận rằng ticket này là dành cho TGS.
$TS_2$	Thông báo cho client thời điểm ticket này được cấp.
$Lifetime_2$	Thông báo cho client thời hiệu của ticket.
$Ticket_{tgs}$	Ticket được sử dụng bởi client để truy cập TGS.

(a) Authentication Service Exchange



<b>Message (3)</b>	Client yêu cầu service-granting ticket.
$ID_V$	Báo cho TGS biết rằng user yêu cầu truy cập server V.
$Ticket_{tgs}$	Đảm bảo với TGS rằng user này đã được xác thực bởi AS.
$Authenticator_c$	Được phát sinh bởi client để xác thực ticket.
<b>Message (4)</b>	TGS trả về service-granting ticket.
$K_{c,tgs}$	Khóa chia sẻ giữa C và TGS để bảo vệ nội dung của message (4).
$K_{c,v}$	Bản sao session key mà client có thể truy cập được, được tạo bởi TGS để cho phép việc truyền thông an toàn giữa client và server mà không cần client và server phải chia sẻ một khóa vĩnh viễn.
$ID_V$	Xác nhận rằng ticket này là cho server V.
$TS_4$	Thông báo cho client biết thời điểm ticket này được cấp.
$Ticket_V$	Ticket được sử dụng bởi client để truy cập server V.
$Ticket_{tgs}$	Có thể tái sử dụng, nhờ đó user không phải nhập lại password.
$K_{tgs}$	Ticket được mã hóa bằng một key vốn chỉ được biết đến bởi AS và TGS.
$K_{c,tgs}$	Bản sao session key mà TGS có thể truy cập được, được sử dụng để giải mã authenticator, nhờ đó xác thực ticket.
$ID_C$	Xác định người sở hữu ticket.
$AD_C$	Ngăn ngừa việc sử dụng ticket từ một máy trạm khác với máy trạm đã yêu cầu ticket.
$ID_{tgs}$	Giúp server bảo đảm rằng nó đã giải mã ticket một cách đúng đắn.
$TS_2$	Thông báo cho TGS thời điểm ticket được cấp.
$Lifetime_2$	Ngăn ngừa tấn công replay khi ticket đã hết hạn.
$Authenticator_c$	Bảo đảm với TGS rằng client trình ticket cũng là client được cấp ticket. Có thời gian sống rất ngắn để tránh replay.

$K_{c,tgs}$	Authenticator được mã hóa bằng khóa bí mật, chỉ được biết bởi client và TGS.
$ID_C$	Phải khớp với ID trong ticket; dùng để xác thực ticket.
$AD_C$	Phải khớp với địa chỉ trong ticket; dùng để xác thực ticket.
$TS_3$	Thông báo cho TGS thời điểm authenticator được phát sinh.

## (b) Ticket-Granting Service Exchange

<b>Message (5)</b>	Client yêu cầu dịch vụ.
$Ticket_V$	Giúp server bảo đảm user này đã được xác thực bởi AS.
$Authenticator_c$	Được phát sinh bởi client để kiểm tra ticket.
<b>Message (6)</b>	Xác thực server với client (tùy chọn).
$K_{c,v}$	Giúp C bảo đảm rằng message này đến từ V.
$TS_5 + 1$	Giúp C bảo đảm rằng đây không phải là một thông điệp replay của một thông điệp phản hồi trước đó.
$Ticket_V$	Có thể tái sử dụng, nhờ đó client không cần phải yêu cầu một ticket mới từ TGS cho mỗi lần truy cập vào cùng server.
$K_v$	Ticket được mã hóa bằng một khóa mà chỉ TGS và server biết.
$K_{c,v}$	Bản sao session key mà client có thể truy cập được; được sử dụng để giải mã authenticator, nhờ đó xác thực ticket.
$ID_C$	Xác định người sở hữu ticket.
$AD_C$	Ngăn ngừa việc sử dụng ticket từ một máy trạm khác với máy trạm đã yêu cầu ticket.
$ID_V$	Giúp server bảo đảm rằng nó đã giải mã ticket một cách đúng đắn..
$TS_4$	Thông báo cho server thời điểm ticket được cấp.
$Lifetime_4$	Ngăn ngừa tấn công replay khi ticket đã hết hạn.

$Authenticator_c$	Bảo đảm với server rằng client trình ticket cũng là client được cấp ticket. Có thời gian sống rất ngắn để tránh replay.
$K_{c,v}$	Authenticator được mã hóa bằng khóa bí mật, chỉ được biết bởi client và server.
$ID_c$	Phải khớp với ID trong ticket; dùng để xác thực ticket.
$AD_c$	Phải khớp với địa chỉ trong ticket; dùng để xác thực ticket.
$TS_s$	Thông báo cho server thời điểm authenticator được phát sinh.

## (c) Client/Server Authentication Exchange

*Bảng 0.1 Mô tả các thành phần trong giao thức Kerberos***7.5.2.4.Kerberos Realm**

Kerberos realm là một môi trường cung cấp dịch vụ Kerberos gồm một máy chủ Kerberos, một số client, và một số máy chủ ứng dụng đáp ứng các điều sau:

1. Kerberos server phải có user ID và mật khẩu đã được băm (hashed password) của tất cả người dùng trong cơ sở dữ liệu của nó. Mọi người dùng phải được đăng ký với Kerberos server.
2. Kerberos server phải chia sẻ một secret key với mỗi server. Tất cả server đều phải được đăng ký với Kerberos server.

Khái niệm realm có thể được giải thích như sau: Một **Kerberos realm** là một tập hợp các node được quản lý và dùng chung Kerberos database. Kerberos database nằm trên máy tính Kerberos master, vốn được bảo mật cẩn thận. Một bản sao chỉ đọc của Kerberos database có thể tồn tại trên các máy tính Kerberos khác. Tuy nhiên, tất cả thay đổi trên database phải được thực hiện trên máy tính master. Việc thay đổi hoặc truy cập vào nội dung của Kerberos database yêu cầu người dùng phải cung cấp mật khẩu Kerberos master.

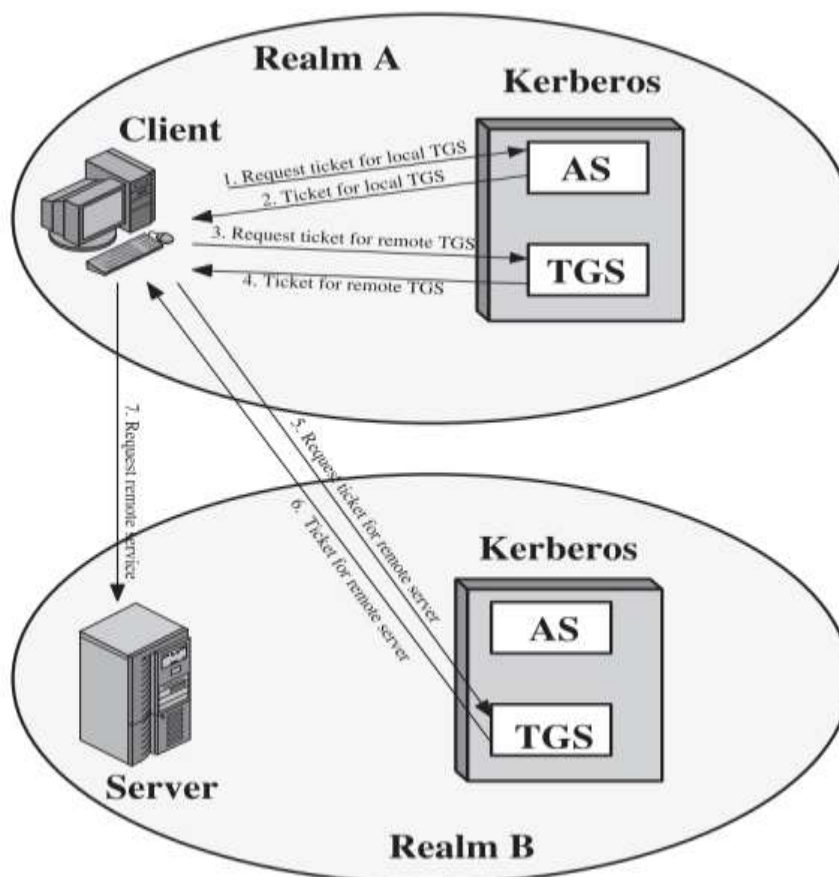
Một khái niệm có liên quan khác là **Kerberos principal**. Đây là một dịch vụ hoặc người dùng mà hệ thống Kerberos đã nhận biết. Mỗi Kerberos principal được nhận diện bằng tên của nó (principal name). Principal name bao gồm ba phần

- Tên dịch vụ hoặc người dùng
- Tên instance
- Realm name

Bên cạnh đó, trên thực tế, các người dùng và server có thể thuộc về các realm khác nhau. Người dùng ở realm này có thể cần truy cập để server ở những realm khác, cũng

như các server có thể sẵn sàng cung cấp dịch vụ cho những người dùng không thuộc về trong realm của mình, miễn là có cơ chế để xác thực chuyện này.

Cơ chế truy cập đó được mô tả ở Hình 7.9.



Hình 0.8 Yêu cầu dịch vụ ở một realm khác

## 7.6. Câu hỏi và bài tập

1. Liệt kê các cách thức khóa bí mật được phân phối giữa hai thực thể truyền thông.
2. Nêu sự khác biệt giữa session key và master key?
3. Tại sao việc quản lý khóa mật mã được xem như một nhiệm vụ quan trọng?
4. Liệt kê một số thành phần của PKI.
5. Mô tả các cơ chế hoặc phương pháp có thể sử dụng để bảo vệ các dịch vụ, bảo vệ khóa.
6. Liệt kê các vấn đề chính của một hệ thống quản lý khóa cần giải quyết.
7. Nêu lý do tại sao hệ thống mã hóa khóa công khai lại áp dụng tốt cho các dịch vụ trực tuyến? Tại sao mã hóa đối xứng lại không phù hợp cho các dịch vụ trên.
8. Chứng chỉ số là gì?
9. Nêu các chức năng được cung cấp bởi PKI.
10. Certifying Authority (CA) là gì? Cho biết vai trò của CA trong PKI.
11. Trong trường hợp CA không còn hoạt động, và sau đó yêu cầu các đối tác chuyển sang một CA khác để có chứng chỉ số mới, thì điều gì sẽ xảy ra với các giao dịch trước đó (bao gồm vấn đề pháp lý và tài chính)?

12. Key distribution center (KDC) là gì? Trình bày chức năng của KDC.
13. Trong môi trường Kerberos, realm là gì?
14. Liệt kê các điểm chính của quản lý khóa.
15. Xem xét một kỹ thuật xác thực một chiều dựa trên mã hóa bất đối xứng dưới đây:

$$\begin{aligned} A &\rightarrow B: ID_A \\ B &\rightarrow A: R_1 \\ A &\rightarrow B: E(PR_a, R_1) \end{aligned}$$

- a) Hãy giải thích hoạt động của giao thức trên.
- b) Giao thức này dễ bị tổn thương bởi loại hình tấn công nào? Lý giải nguyên nhân.
16. Xem xét một kỹ thuật xác thực một chiều dựa trên mã hóa bất đối xứng dưới đây:

$$\begin{aligned} A &\rightarrow B: ID_A \\ B &\rightarrow A: E(PU_a, R_2) \\ A &\rightarrow B: R_2 \end{aligned}$$

- a) Hãy giải thích hoạt động của giao thức trên.
- b) Giao thức này dễ bị tổn thương bởi loại hình tấn công nào? Lý giải nguyên nhân.
17. Xem xét giao thức dưới đây:

$$\begin{aligned} A &\rightarrow KDC: ID_A \parallel ID_B \parallel N_1 \\ KDC &\rightarrow A: E\left(K_a, \left[K_s \parallel ID_B \parallel N_1 \parallel E\left(K_b, [K_s \parallel ID_A]\right)\right]\right) \\ A &\rightarrow B: E\left(K_b, [K_s \parallel ID_A]\right) \\ B &\rightarrow A: E\left(K_s, N_2\right) \\ A &\rightarrow B: E\left(K_s, f\left(N_2\right)\right) \end{aligned}$$

- a) Hãy giải thích hoạt động của giao thức trên.
- b) Hãy tìm một phương án tấn công khả thi vào giao thức này. Giải thích cách thức thực hiện kiểu tấn công đó.
- c) Hãy giải thích cách thức để chống lại kiểu tấn công đó. Không cần mô tả chi tiết mà chỉ cần nêu ra các ý tưởng cơ bản.
18. Chứng chỉ số hoạt động thế nào trong thư điện tử?
19. Chứng chỉ số hoạt động thế nào trong một website?
20. Nêu sự khác nhau giữa PKI và mật mã khóa công khai.

---

## CHƯƠNG 8: CHỨNG THỰC THỰC THỂ

---

Chương này đề cập đến chứng thực thực thể: phân biệt giữa chứng thực thông điệp và chứng thực thực thể; các phương pháp trong chứng thực thực thể, nhận dạng cùng một số giao thức thông dụng.

### 8.1. Khái niệm

Chứng thực thực thể (Entity authentication) là một kỹ thuật được thiết kế để một bên (party) chứng minh nhận dạng (identity) của một bên khác. Một thực thể có thể là một người, một tiến trình, một client, hoặc một server. Thực thể mà nhận dạng cần được chứng minh được gọi là *bên yêu cầu* (claimant); bên xác thực nhận dạng của claimant thì được gọi là *bên thẩm tra* (verifier).

Sự khác nhau giữa Chứng thực thực thể và chứng thực thông điệp là chứng thực thông điệp không cần thực hiện theo thời gian thực, ví dụ, Alice gửi một thông điệp cho Bob, khi Bob chứng thực thông điệp thì Alice có thể không cần có mặt ngay trong tiến trình giao tiếp. Chứng thực thông điệp chứng thực một cách đơn giản cho từng thông điệp một vì vậy tiến trình thực hiện chứng thực cần được lặp đi lặp lại cho mỗi thông điệp khác nhau. Trong khi đó, chứng thực thực thể phải được thực hiện theo thời gian thực và được chứng thực trong suốt phiên giao dịch.

Trong chứng thực thực thể, claimant có thể trình ra đặc tính nhận dạng cho verifier, gồm ba loại:

- **Những gì được biết:** Là một bí mật chỉ được biết bởi claimant mà có thể được kiểm tra bởi verifier. Ví dụ: Password, Pin, secret key, private key.
- **Những gì có được:** là một thứ mà có thể chứng minh nhận dạng của claimant. Ví dụ: passport, bằng lái xe, chứng minh nhân dân, credit card, smart card
- **Đặc tính vốn sẵn có** (inherent) của Claimant. Ví dụ: Chữ ký thông thường, dấu vân tay, giọng nói, đặc tính khuôn mặt, võng mạc và chữ viết tay ...

### 8.2. Mật khẩu (Password)

Chứng thực bằng mật khẩu (Password-based Authentication) là một phương pháp xác thực đơn giản và được sử dụng phổ biến nhất, trong đó password là một thứ mà claimant nắm giữ. Một password được dùng khi một người dùng cần truy xuất một hệ thống để sử dụng nguồn tài nguyên của hệ thống. Mỗi người dùng có một định danh và một password bí mật. Có thể chia mật khẩu thành hai loại: mật khẩu cố định (Fixed password) và mật khẩu một lần (one-time password).

**8.2.1.Fixed password:**

Là một password được dùng lặp đi lặp lại mỗi lần truy xuất. Có 3 cơ chế được xây dựng theo hướng này đó là User ID và Password File; Hashing the password; Salting the password.

- **User ID và Password file:**

- Đây là loại chứng thực này khá thô sơ, User ID và Password được lưu trong một file ở dạng bản rõ. Để truy xuất tài nguyên, người dùng gửi bản rõ của User ID và Password đến hệ thống. Hệ thống dùng User ID để tìm password trong tập tin. Nếu Password trùng khớp với Password trong hệ thống, thì quyền truy xuất được gán ngược lại từ chối.
- Để tấn công loại chứng thực trên, người ta sử dụng được nhiều phương pháp khác nhau, trong đó có thể kể đến:
  - ✓ Tấn công nghe trộm: Kẻ tấn công theo dõi đường truyền và bắt lấy thông tin username, password (không mã hóa) khi người dùng truyền qua mạng.
  - ✓ Truy xuất tập tin password: Kẻ tấn công cố gắng truy xuất vào tập tin chứa ID/password của hệ thống và đánh cắp tập tin chứa thông tin về tài khoản người dùng.
- Nhằm tăng cường độ an toàn của mật khẩu, người ta sử dụng kỹ thuật băm trong việc lưu trữ.

- **Hashing the password (băm mật khẩu)**

- Password được băm trước khi lưu trữ (thay cho việc lưu trữ bản rõ của password) vào tập tin password. Trong trường hợp thông tin chứa mật khẩu bị đánh cắp hoặc có người không hợp pháp tiếp cận thì cũng khó có thể đoán được mật khẩu ban đầu, việc khôi phục lại mật khẩu gốc từ bảng băm cũng rất khó khăn và tốn nhiều chi phí.
- Khi password được tạo, thì hệ thống băm password và lưu giá trị băm (bảng băm) vào tập tin mật khẩu.
- Khi người dùng nhập ID và password, hệ thống tính toán giá trị băm của mật khẩu được nhập và so sánh với giá trị băm đã được lưu trong tập tin. Nếu trùng khớp, thì người dùng được gán quyền truy xuất, ngược lại, quyền truy xuất sẽ bị từ chối.

- **Salting the password**

- Việc băm mật khẩu khi lưu trữ giúp tăng cường độ an toàn của việc bảo vệ các thông tin về tài khoản, tuy nhiên để tiếp tục cải thiện độ an toàn của mật khẩu trước các phương pháp tấn công, khi chuỗi password được tạo, một chuỗi ngẫu nhiên (gọi là salt) được ghép thêm vào password đó. Sau đó, password mới hình thành được băm (bằng giải thuật băm), việc lưu trữ thông tin chứa tài khoản vào tập tin bao gồm: ID, salt, giá trị băm. Khi người dùng yêu cầu truy

xuất, hệ thống sẽ trích chuỗi salt, ghép nó với password nhận được, thực hiện băm, so sánh kết quả với giá trị đã lưu trữ. Nếu trùng khớp thì quyền truy xuất được gán, ngược lại sẽ bị từ chối.

- Việc dùng salt làm cho các kỹ thuật tấn công mật khẩu trở nên khó khăn hơn (trong đó hạn chế rất nhiều việc tấn công từ điển (dictionary attack)). Ví dụ: nếu mật khẩu gốc có 6 ký số và salt có 4 ký số, thì việc băm sẽ được thực hiện trên giá trị có 10 ký số. Điều này làm cho kẻ tấn công phải xử lý, dò tìm trên bộ dữ liệu gồm 10 số, dẫn đến tốn kém về thời gian và chi phí. Việc thực hiện salt rất hiệu quả nếu chuỗi salt là một số ngẫu nhiên dài.

- **Kỹ thuật nhận dạng kết hợp nhiều yếu tố.**

Nhằm tăng độ an toàn cho hệ thống, người ta kết hợp các yếu tố khác nhau cho việc nhận dạng, xác thực, trong đó có thể kể đến việc xác thực cho thẻ ATM với mã PIN, trong trường hợp này: “something possessed” là smart card, và “something known” là Pincode, pincode giúp tăng cường tính bảo mật cho smart card trong trường hợp bị mất, đánh cắp.

### 8.2.2. One-time Password

Là password mà được sử dụng chỉ một lần. Loại password này làm cho các tấn công nghe lén và salting vô tác dụng. Một số phương thức thực hiện password một lần:

- User và System thỏa thuận một danh sách các Password
  - Mỗi Password trong danh sách được dùng một lần
  - System và User phải giữ gìn danh sách Password
  - Nếu User không dùng các password một cách tuần tự thì System phải thực hiện tìm kiếm và so khớp
  - Password chỉ hợp lệ một lần và không sử dụng lại
- User và System thỏa thuận cập nhật một cách tuần tự Password
  - User và System thỏa thuận một Password gốc,  $P_1$  mà Password này chỉ hợp lệ cho lần đầu tiên truy suất.
  - Trong quá trình truy xuất lần đầu tiên này, user phát sinh một password mới  $P_2$ , và mã hóa password này với khóa là  $P_1$ ,  $P_2$  là password cho lần truy xuất kế tiếp và cứ thế tiếp tục phát sinh  $P_{i+1}$  từ  $P_i$  cho lần truy xuất thứ  $P_{i+1}$
  - Nếu đối phương đoán ra được  $P_1$  thì có thể tìm được tất cả các Password khác
- User và System tạo ra một Password được cập nhật một cách tuần tự sử dụng hàm băm.
  - Hướng này được đề xuất bởi Leslie Lamport (*Lamport one-time password*)



- User và System thỏa thuận một Password gốc  $P_0$  và số đếm  $n$ . System tính  $h^n(P_0)$  với  $h^n$  là hàm băm lần thứ  $n$  theo công thức sau:

$$h^n(x) = h(h^{n-1}(x)), \quad h^1(x) = h(x).$$

- System lưu nhận dạng của user thông qua giá trị  $n$  và giá trị  $h^n(P_0)$ .

### 8.3.Challenge-Response

Trong chứng thực password, các claimant chứng minh nhận dạng của họ bằng cách chứng minh rằng họ biết password (mật khẩu bí mật). Tuy nhiên, bởi vì claimant cung cấp thông tin này qua đường công cộng nên có thể bị kẻ tấn công chặn bắt, nghe lén. Trong chứng thực Challenge-Response (thử thách – phản hồi) thì claimant chứng minh rằng mình biết bí mật mà không cần phải gửi thông tin này cho ai. Nói cách khác, claimant không gửi bí mật cho bên thẩm tra, bên thẩm tra có nó hoặc tự tìm ra nó.

Các hướng chính để tạo nên Challenge-reponse:

#### 8.3.1.Dùng Sysmetric-Key Cipher

Bí mật (secret) ở đây là **khóa bí mật được chia sẻ** giữa Claimant và Verifier. Sử dụng **hàm mã hóa** áp dụng cho challenge này. Một số cách thức theo cơ chế này:

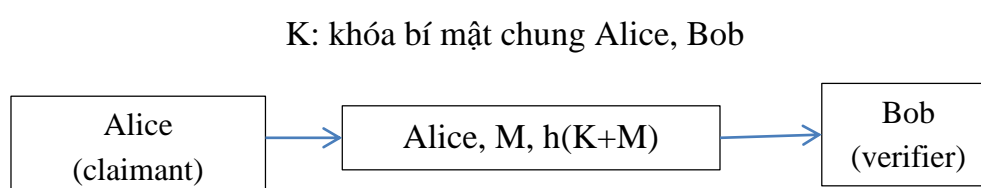
- **Nonce challenge.** Verifier gửi một nonce (thử thách, text ngẫu nhiên được dùng chỉ một lần) đến claimant. Một nonce phải được thay đổi theo thời gian (time-varying). Claimant phản hồi challenge bằng cách dùng khóa bí mật được chia sẻ giữa claimant và verifier. Mô hình trên gồm các bước sau:
  - ✓ Claimant và Verifier thỏa thuận khóa bí mật (secret key) để sử dụng chung
  - ✓ Claimant gửi yêu cầu đến verifier
  - ✓ Verifier nhận được yêu cầu, gửi cho Claimant thử thách (thông thường là text)
  - ✓ Claimant mã hóa thử thách, sử dụng secret key và gửi kết quả cho Verifier
  - ✓ Verifier thực hiện việc xác minh dữ liệu nhận được, nếu chính xác thì gán quyền truy xuất.
- **Timestamp challenge:** giá trị time-varying là một timestamp thay đổi một cách ngẫu nhiên cho mỗi lần sử dụng. Theo đó thông điệp challenge là thời gian hiện hành được gửi từ verifier đến claimant. Tuy nhiên, giả sử rằng đồng hồ của client và server là đồng bộ; claimant biết được thời gian hiện hành. Điều này có nghĩa rằng không cần đổi với thông điệp challenge/ Thông điệp thứ nhất và thứ hai có thể được kết hợp. Kết quả là việc chứng thực đó có thể được thực hiện bằng cách dùng một thông điệp, response đối với một challenge ẩn là thời gian hiện hành.
- Chứng thực hai chiều (Bidirectional authentication) Hai phương thức trên là đối với chứng thực một chiều. Alice được chứng thực đối với Bob, nhưng không có chiều

ngược lại. Nếu Alice cũng cần đảm bảo về nhận dạng của Bob, thì cần chứng thực hai chiều.

### 8.3.2.Using Keyed-Has Functions

Thay vì dùng mã hóa/giải mã cho chứng thực thực thể, có thể dùng một Keyed-has function (MAC). Một trong những ưu điểm của cơ chế này là nó bảo toàn tính toàn vẹn của thông điệp challenge và response và cùng thời điểm dùng bí mật, khóa.

Hình 8.1 mô tả việc sử dụng hàm băm khóa (Keyed-hash function) để tạo ra một challenge/response với một timestamp. Trong trường hợp này, timestamp được gửi cả bản rõ  $M$  và bằng băm  $h(K + M)$ , trong đó  $K$  là khóa bí mật của Alice và Bob. Khi nhận được thông điệp, Bob lấy bản rõ  $T$ , áp dụng hàm keyed-hash, và sau đó thực hiện việc chứng thực thông tin đã nhận được từ Alice.

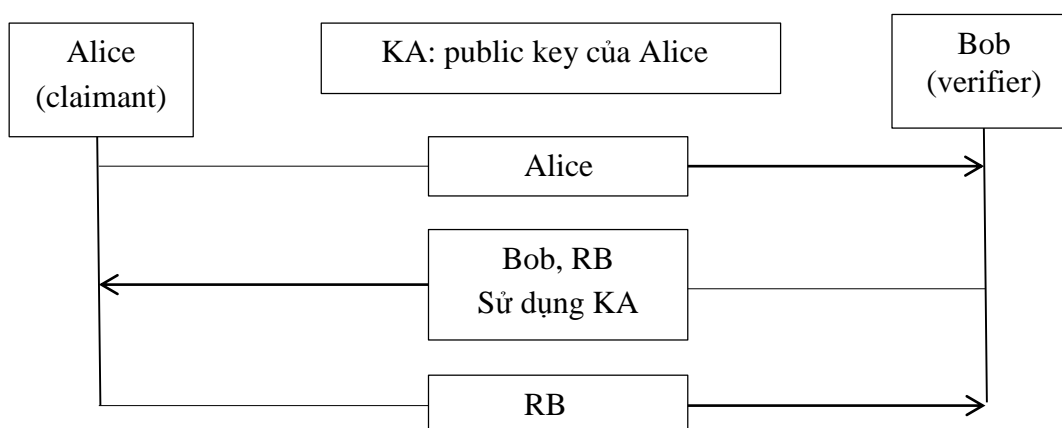


Hình 8.1. Mô phỏng challenge/response sử dụng hàm băm

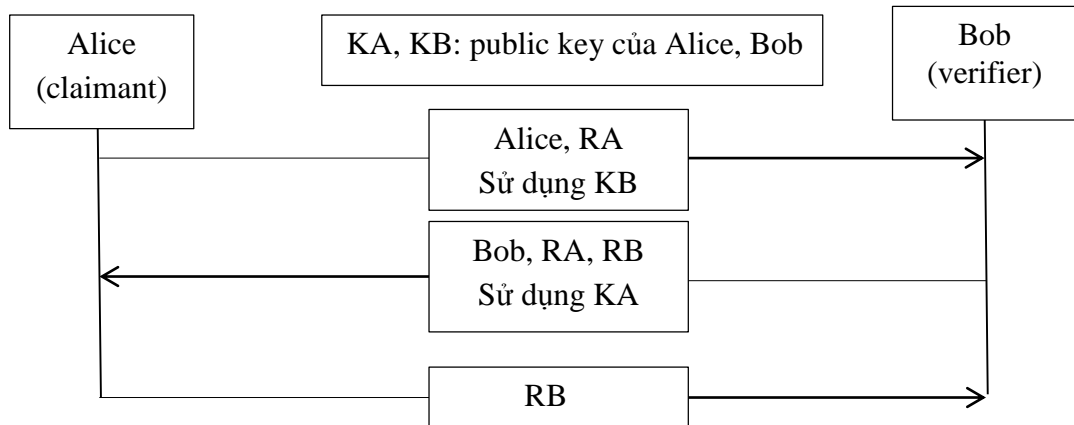
### 8.3.3.Sử dụng mã hóa khóa đối xứng (Asymmetric-Key Cipher)

Sử dụng mã hóa khóa bất đối xứng, Secret là khóa cá nhân (private key) của claimant. Claimant phải chỉ ra rằng private của cô ta có liên quan đến Public key bằng cách Verifier mã hóa challenge (challenge được tạo ra bằng cách dùng Public key của claimant), sau đó claimant giải mã bằng private key: Theo cách này ta có:

- Unidirectional, asymmetric-key authentication: Bob mã hóa challenge bằng cách dùng khóa công khai của Alice. Alice giải mã thông điệp bằng khóa bí mật của cô ta và gửi nonce cho Bob.



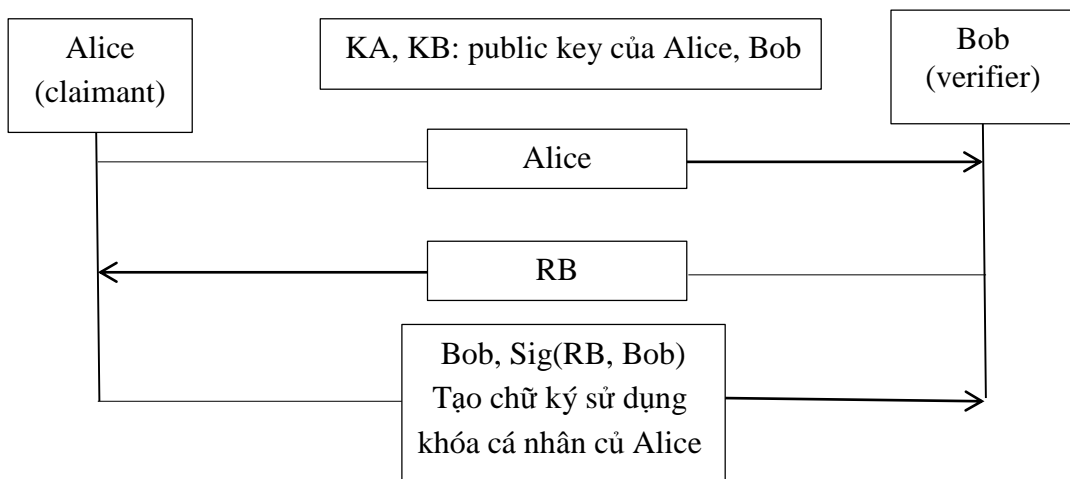
- Bidirectional, asymmetric-key: hai khóa công khai được dùng mỗi khóa cho mỗi hướng. Alice gửi nhận dạng của cô ta và nonce được mã hóa bằng khóa công khai của Bob. Bob đáp ứng lại bằng nonce của anh ta được mã hóa bằng khóa công khai của Alice.



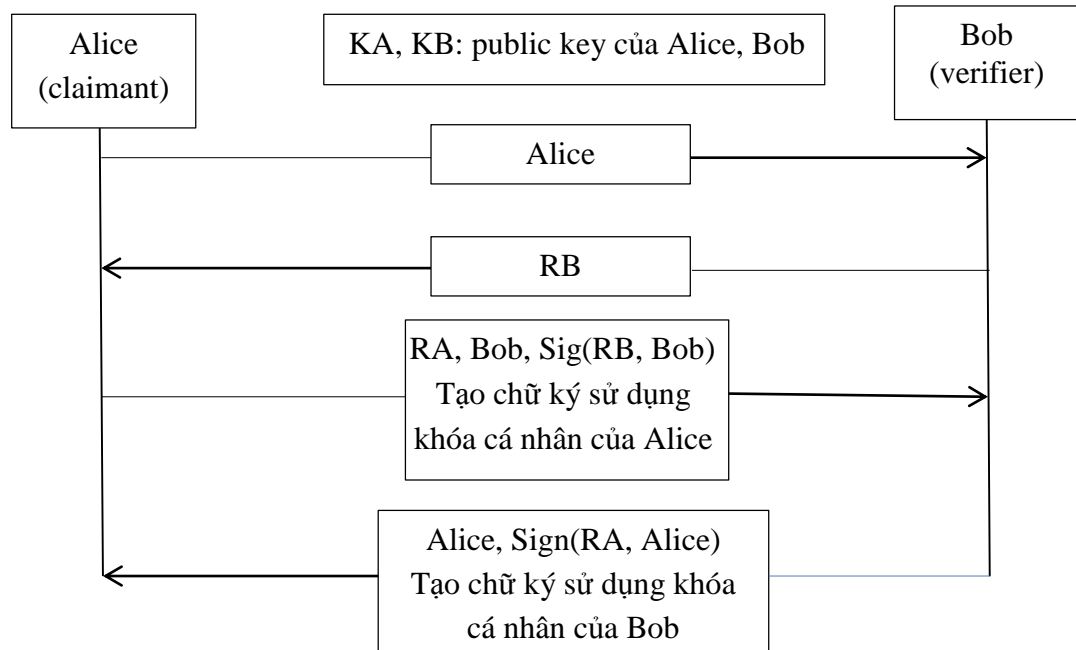
#### 8.3.4. Sử dụng chữ ký số (Digital Signature)

Digital Signature được dùng để chứng thực thực thể. Claimant dùng khóa cá nhân của mình để tạo chữ ký.

- Digital signature, unidirectional: Bob dùng bản rõ để thử thách và Alice thực hiện ký.



- Digital signature, bidirectional authentication: Alice và Bob chứng thực lẫn nhau



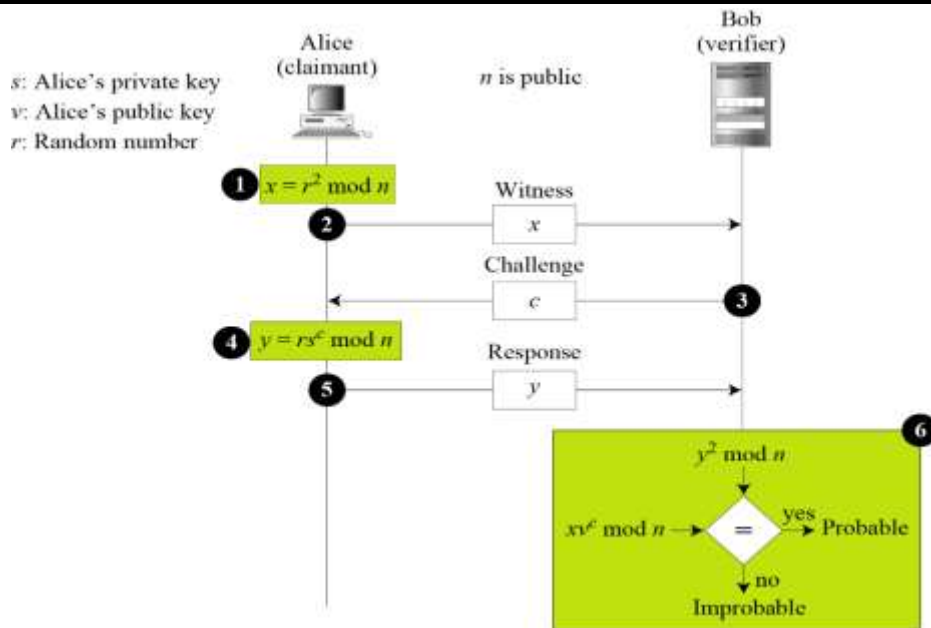
## 8.4. ZERO-KNOWLEDGE

Trong chứng thực Challenge-response, bí mật của claimant không được gửi đến verifier, mà claimant áp dụng một hàm trên challenge gửi bởi Verifier mà bao gồm cả bí mật của cô ta. Trong một số phương pháp challenge-response, verifier biết bí mật của claimant, mà có thể bị lạm dụng bởi những verifier không tin cậy. Khi đó, verifier có thể trích lọc ra được những thông tin về bí mật từ claimant bằng cách chọn trước một tập các challenge.

Trong chứng thực Zero-knowledge, claimant không tiết lộ bất kỳ cái gì mà có thể gây nguy hại đến tính bảo mật của bí mật. **Claimant chứng minh với verifier rằng mình biết một bí mật mà không hề tiết lộ nó.**

### 8.4.1. Giao thức Fiat-Shamir

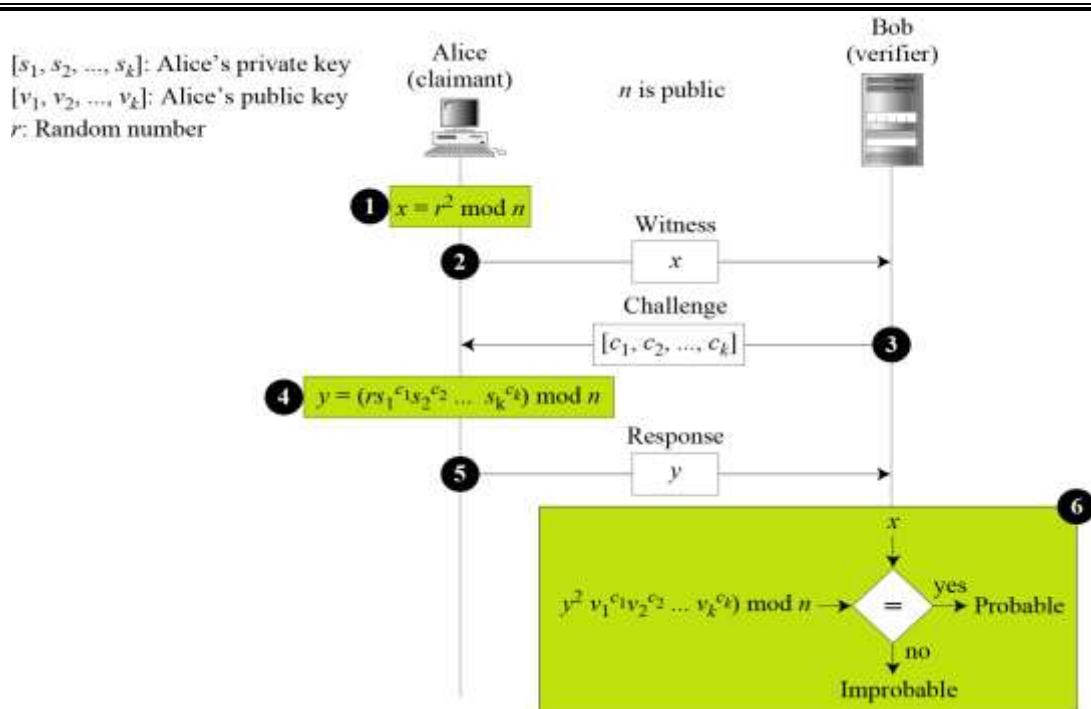
Trong giao thức này, một bên thứ ba tin cậy chọn hai số nguyên tố lớn  $p$  và  $q$  khác nhau. Giá trị  $n = p \cdot q$  được công khai;  $p$  và  $q$  được giữ bí mật. Alice (claimant) chọn một giá trị bí mật  $s$  ( $1 < s < n - 1$ ) và công khai  $v = s^2 \bmod n$  với bên thứ ba. Việc thẩm tra của Alice bởi Bob có thể được thực hiện qua các bước như hình dưới.



- Alice (claimant) chọn một số ngẫu nhiên  $r$ ,  $0 < r < n-1$  và tính giá trị  $x = r^2 \bmod n$  gửi cho Bob,  $x$  được gọi witness.
- Bob (verifier) cho Alice thử thách  $c$  ( $c$  nhận giá trị 0 hoặc 1).
- Alice tính  $y = rs^c \bmod n$  và gửi cho Bob.
- Bob tính  $y^2 \bmod n$  và so sánh với  $xv^c \bmod n$ . Việc xác minh trên được lặp đi lặp lại nhiều lần. Claimant phải qua kiểm tra ở mỗi vòng để được thẩm định. Nếu không qua được ở một vòng thì tiến trình được bỏ qua và Alice không được chứng thực.

#### 8.4.2. Giao thức Feige-Fiat-Shamir

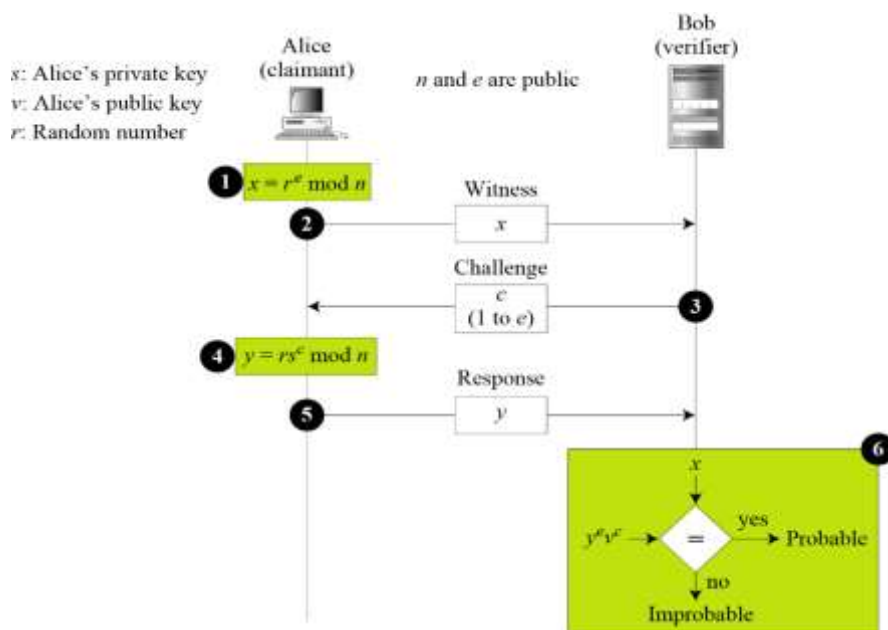
Giao thức Feige-Fiat-Shamir tương tự như giao thức Fiat-Shamir tuy nhiên sử dụng vector  $[s_1, s_2, \dots, s_k]$  cho khóa cá nhân, vector  $[v_1, v_2, \dots, v_k]$  ( $v_i = s_i^2 \bmod n$ ) cho khóa công khai và vector thử thách  $[c_1, c_2, \dots, c_k]$ . Các khóa riêng được chọn một cách ngẫu nhiên nhưng chúng phải nguyên tố cùng nhau với  $n$ . Các khóa công khai được chọn sao cho  $v_i = (s_i^2)^{-1} \bmod n$ . Ba bước trong tiến trình được minh họa trong hình.



### 8.4.3. Giao thức Guillou-Quisquater

Giao thức này là một sự mở rộng của giao thức Fiat-Shamir mà trong đó ít vòng hơn có thể được dùng để chứng minh nhận dạng của claimant. Một bên thứ ba tin cậy chọn hai số nguyên tố  $p, q$ ,  $v$  công khai,  $s$  bí mật sao cho  $s^e \cdot v = 1 \bmod n$ .

Việc thẩm tra của Alice bởi Bob có thể được thực hiện qua các bước như hình sau:



Việc thẩm định được lặp đi lặp lại vài lần với giá trị của  $c$  từ 1 đến  $e$ . Claimant phải qua kiểm tra ở mỗi vòng để được thẩm định. Nếu không qua được ở một vòng thì tiến trình được bỏ qua và không được chứng thực.

### 8.5. Biometrics

Sinh trắc học (Biometric) là phép đo lường về các đặc tính sinh lý học hoặc hành vi học mà nhận dạng một con người. Sinh trắc học đo lường các đặc tính mà không thể đoán, đánh cắp hoặc chia sẻ. Trong thực tiễn, sinh trắc học đã được áp dụng trong nhiều lĩnh vực khác nhau, ví dụ: truy xuất các thiết bị, các hệ thống thông tin, giao dịch ở các điểm bán (trả tiền), phục vụ công tác điều tra bằng cách phân tích ADN hoặc vân tay, kiểm soát nhập cư cũng sử dụng một số kỹ thuật sinh trắc học ...

Các thành phần (components) của sinh trắc học bao gồm các thiết bị thu nhận đặc tính của sinh trắc học, xử lý các đặc tính sinh trắc học và thiết bị lưu trữ.

Chứng thực (Authentication) được thực hiện bởi sự thẩm tra (verification) hoặc nhận dạng (identification), trong đó:

- Verification: Đặc tính của một người được so khớp với một mẫu tin đơn trong cơ sở dữ liệu để xác định.
- Identification: Đặc tính của một người được so khớp với tất cả các mẫu tin có trong cơ sở dữ liệu để xác định.

Về mặt kỹ thuật sinh trắc học có thể được chia thành: sinh lý học (physiological) và dáng điệu học (behaviorial), trong đó:

- Physiological: vân tay (fingerprint), con ngươi (iris), võng mạc (retina), face, hands, voice, DNA ...
- Behaviorial: signature, keystroke ...

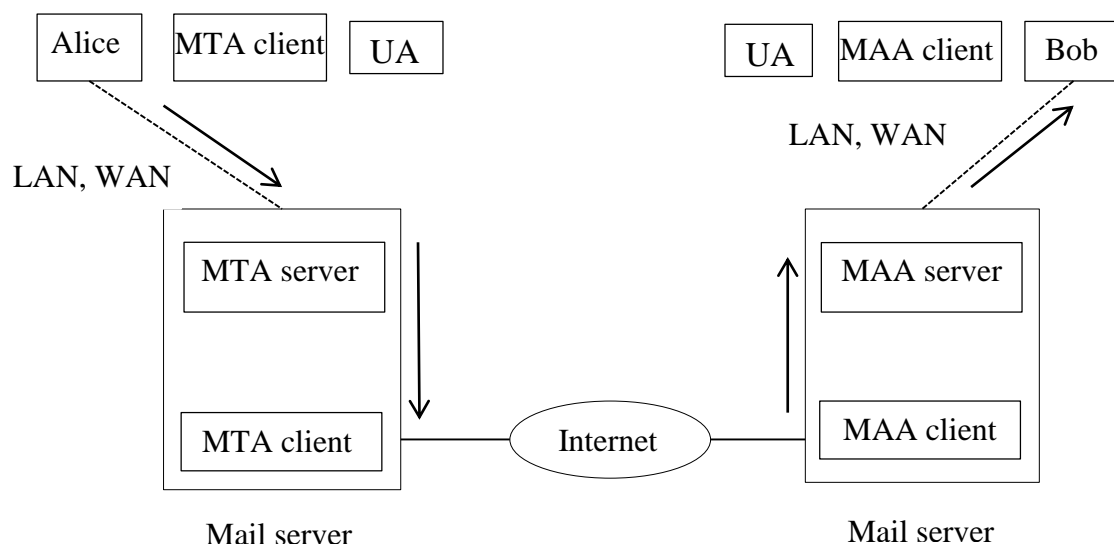
### 8.6. Câu hỏi và bài tập

1. Phân biệt giữa chứng thực nguồn gốc dữ liệu và chứng thực thực thể.
2. Liệt kê và định nghĩa các loại vật chứng nhận dạng trong chứng thực thực thể.
3. Giải thích vai trò logic của chứng thực.
4. Phân biệt giữa fixed passwords và one-time password.
5. Những ưu điểm và nhược điểm khi dùng password dài.
6. Nonce là gì? Nonce được sử dụng cho mục đích nào?
7. Phân biệt chứng thực thực thể challenge-response và zero-knowledge.
8. Chứng thực thực thể bằng sinh trắc học có ưu điểm và khuyết điểm gì so với những loại chứng thực khác.
9. Liệt kê các 10 ứng dụng thực tế mà bạn biết có sử dụng các phương pháp chứng thực thực thể.
10. Giải thích tại sao hệ thống nhận dạng thách thức được sử dụng trong thực tiễn.
11. Giải thích cách sử dụng mã hóa khóa công khai trong nhận dạng.
12. Sử dụng ngôn ngữ lập trình để viết chương trình có chức năng như sau:
  - Cho 10 người dùng, mỗi người dùng tạo một UserID ứng với một Password và lưu chúng vào tập tin

- Cho người dùng nhập vào một UserID và Password, Kiểm tra xem UserID và Password có phải là một cặp được lưu trữ trong tập tin Password hay không.

13. Tương tự như bài 8 nhưng lưu trữ giá trị băm của password.
14. Cho  $p=569$ ,  $q=683$ , và  $s=157$ , trình bày ba vòng của giao thức Fiat-Shamir bằng cách tính toán các giá trị và liệt kê ra
15. Cho  $p=683$ ,  $q=811$ ,  $s_1=157$  và  $s_2=43215$ , chạy minh họa giao thức Feige-Fiat-Shamir
16. Cho  $p=683$ ,  $q=811$ ,  $v=157$  chạy minh họa giao thức Guillou-Quisquater
17. Mô tả các mục tiêu mà sơ đồ chứng thực password lý tưởng cần đạt được. Trình bày 5 kiểu tấn công mật khẩu.
18. Giải thích các nguyên nhân chính của việc sử dụng mật khẩu trong chứng thực. Trình bày một kỹ thuật tấn công social engineering vào mật khẩu.



**CHƯƠNG 9: MỘT SỐ GIAO THỨC BẢO MẬT THÔNG DỤNG****9.1. Các giao thức bảo mật E-mail****9.1.1. Kiến trúc e-mail***Hình 9.1. Kiến trúc e-mail*

Hình 9.1 minh họa quá trình trao đổi e-mail một chiều từ Alice sang Bob. Giả sử Alice và Bob làm việc cho doanh nghiệp có e-mail server và mọi nhân viên trong công ty kết nối đến e-mail server thông qua mạng nội bộ (LAN). Mặt khác, Alice và Bob cũng có thể kết nối đến e-mail server của một nhà cung cấp dịch vụ internet (Internet Service Provide - ISP) thông qua WAN (wide area network) như đường điện thoại hoặc cáp.

Quản trị viên e-mail server bên phía Alice tạo hệ thống cho phép gửi đến internet lần lượt theo từng e-mail. Quản trị viên e-mail server bên phía Bob tạo hộp thư (mailbox) cho mọi người dùng kết nối đến server. Mailbox sẽ lưu trữ e-mail cho đến khi nó được gửi đến người nhận.

Khi Alice cần gửi thông tin đến Bob, Alice sử dụng chương trình để soạn thư (User agent - UA). Sau đó, sử dụng chương trình dịch vụ thư điện tử (Message transfer agent - MTA) để gửi đến mail server tại địa điểm Alice. Mail server nhận được thư của Alice và các người khác, nó sẽ chuyển đến các đích tương ứng. Trong trường hợp này, e-mail của Alice sẽ được gửi đến mail server bên phía Bob. Client và server được đáp ứng cho việc truyền e-mail giữa hai server. Khi một e-mail được gửi đến server đích, nó được lưu trữ trong hộp thư của Bob, e-mail này sẽ được giữ lại cho đến khi Bob nhận được.

Khi Bob muốn nhận thư (ví dụ như từ Alice), anh ta sử dụng chương trình (Message access agent - MAA). MAA được thiết kế như chương trình cài đặt trên máy tính của Bob và trên mail server.

Một số điểm lưu ý trong kiến trúc e-mail:

- E-mail gửi từ Alice sang Bob được lưu trữ. Alice có thể gửi e-mail hôm nay nhưng Bob có thể đọc vài ngày sau đó. Trong thời gian đó, e-mail được lưu trữ trong hộp thư.
- Việc truyền giữa Alice và Bob thông qua hai chương trình ứng dụng chính: MTA client trên máy tính Alice và MAA client trên máy tính Bob.
- MTA client là chương trình “push”, dùng để gửi thư khi Alice muốn gửi nó. MAA client là chương trình “pull”, sử dụng khi Bob đọc e-mail.
- Alice và Bob không thể truyền trực tiếp bằng cách sử dụng MTA client bên địa điểm gửi và MTA server bên phía nhận. MTA server cần phải hoạt động liên tục, vì Bob không biết khi nào thư được gửi đến.

### 9.1.2. Bảo mật e-mail

Trong kiến trúc e-mail, Alice và Bob không tạo kết nối phiên. Alice gửi thông tin cho Bob và sau đó một thời gian Bob đọc email và phản hồi (nếu có). Trong phần này giới thiệu 1 số phương pháp bảo mật e-mail:

- Cryptographic algorithms: người gửi cần phải thông tin về các thuật toán được sử dụng kèm trong e-mail. Ví dụ: Alice chọn AES cho việc mã hóa/giải mã dữ liệu và SHA160 cho tính toán giá trị băm. Khi gửi thư cho Bob, Alice đính kèm thông tin AES và SHA160 trong đó. Bob nhận được sẽ biết được thuật toán mã hóa và hàm băm đã được sử dụng.
- Cryptographic secrets: sử dụng mã hóa đối xứng cho việc mã hóa và giải mã sử dụng khóa bí mật (secret key). Khóa bí mật được gửi cho phía bên nhận thông qua việc mã hóa nó bằng khóa công khai của bên nhận bằng hệ thống mã hóa khóa công khai. Khi đó, bên nhận sử dụng khóa cá nhân để giải mã, nhận được secret key, sau đó sử dụng secret key để giải mã các thông tin trong e-mail.
- Certificates: Trong việc truyền e-mail nhiều trường hợp cần sử dụng thuật toán khóa công khai. Ví dụ, Alice và Bob cần mã hóa secret key hoặc ký văn bản. Để mã hóa secret key, Alice cần public key của Bob; và để xác thực văn bản, Bob cần public key của Alice. Vì vậy, để chứng thực và bảo mật thông tin kích thước nhỏ cần thiết sử dụng đến public key.

Một số giao thức bảo mật được ứng dụng trong thực tiễn: PGP, S/MIME...

## 9.2. Các giao thức bảo mật mạng

Nhắc lại một số khó khăn đã được nêu ở **chương** trong việc đảm bảo an toàn mạng máy tính: Hệ thống mở, tài nguyên phát tán, người dùng ẩn danh, TCP/IP không được thiết kế cho việc xác thực các bên và mã hóa gói tin. Do đó, dữ liệu truyền đi trong mạng đối mặt với nhiều nguy cơ thất thoát, đánh cắp, thay đổi thông tin ... Trong phần này đề

cập đến việc bảo mật tầng giao vận (điều khiển quá trình truyền dữ liệu giữa các tiến trình, đôn kênh, phân kênh, TCP/UDP) trong kiến trúc phân tầng.

Bảo mật tầng giao vận (Transport layer): dịch vụ cung cấp bảo mật end-to-end cho các ứng dụng an toàn như TCP. Ý tưởng cơ bản là cung cấp dịch bảo mật cho việc trao đổi trên internet, như trong việc giao dịch online. Một số yêu cầu cho việc bảo mật:

- Người dùng cần biết chắc chắn server truy cập là của bên cung cấp (như cửa hàng, siêu thị ...), bảo vệ việc rò rỉ thông tin về tài khoản, cá nhân khỏi những kẻ mạo danh (chứng thực thực thể).
- Các bên giao dịch cần chắc chắn nội dung thông tin không bị chỉnh sửa khi truyền (toàn vẹn thông tin).
- Các bên giao dịch cần đảm bảo không có trung gian làm gián đoạn giao dịch hoặc đánh cắp thông tin như số tài khoản.

Các giao thức đang sử dụng phổ biến hiện nay cung cấp tính bảo mật của tầng vận chuyển là Secure sockets layer (SSL) và Transport layer security (TLS). Mục đích chính của các giao thức trên là cung cấp cho server và client việc chứng thực, bảo mật và toàn vẹn dữ liệu. Ví dụ để tăng cường an toàn cho HTTP, người ta triển khai SSL (hoặc TLS) trên server và client, lúc này người dùng sử dụng URL https:// cho phép đóng gói thông tin truyền đi trong HTTP bằng SSL (hoặc TLS) packet. Khi đó các thông tin về tài khoản được bảo vệ an toàn trong giao dịch online.

### 9.3.Các giao thức thanh toán điện tử

Hiện nay, Thương mại điện tử đã trở thành xu hướng phát triển mới trên toàn thế giới bằng cách sử dụng Internet. Các doanh nghiệp đã thay đổi hình thức kinh doanh từ cách tiếp cận truyền thống bằng thương mại điện tử. Khi các cá nhân và doanh nghiệp tăng cường chia sẻ thông tin, thì một trong những mối quan tâm lớn nhất là giao dịch một cách an toàn và thuận tiện. Trong phần này, tác giả giới thiệu một số giao thức an toàn trong thanh toán điện tử cũng như một số cơ chế giúp cho việc giao dịch được an toàn hơn trong môi trường internet.

Giao dịch điện tử (Electronic transaction) là hoạt động chính trong thương mại điện tử, các giao dịch điện tử là các trao đổi thông tin giữa các bên liên quan đến giao dịch. Thông tin bao gồm: các đơn đặt hàng, thông tin xác nhận, thẻ tín dụng, tài liệu ... Khi các giao dịch được tiến hành thì điều quan trọng nhất là đảm bảo cho chúng khỏi bất kỳ các mối đe dọa nào. Yếu tố quan trọng nhất trong giao dịch điện tử là bảo mật giao dịch (transaction security)

Internet không cung cấp tính bảo mật cho các giao dịch điện tử. Thông tin đi qua internet được chuyển qua nhiều hệ thống máy tính, thiết bị trước khi đến được server tin cậy. Do đó, ở bất kỳ thời điểm nào, thông tin cũng có thể bị đe dọa (thay đổi, chỉnh sửa, gián đoạn ...). Để tăng cường an ninh trong giao dịch điện tử, có nhiều giải pháp được đưa ra. Trong phần này, tài liệu cung cấp hai giải pháp thường được áp dụng trong

thương mại điện tử dựa trên nền tảng internet: Secure Socket Layer và Secure Electronic Transaction.

### 9.3.1. Secure Socket Layer

Secure Socket Layer (SSL) là một giao thức bảo mật được phát triển bởi Netscape nhằm bảo vệ đường truyền thông qua Internet. SSL hỗ trợ nhiều ứng dụng như FTP, Gopher, Network News Transfer Protocol (NNTP), HTTP. Khi SSL được sử dụng để bảo vệ HTTP, nó giúp người dùng Web tin rằng họ giao tiếp với Web server gửi hoặc nhận thông tin một cách an toàn. SSL sử dụng hệ thống mã hóa RSA để xác thực và mã hóa. SSL hoạt động để bảo vệ truyền thông Internet bằng các tính năng sau:

- ✓ Xác thực máy chủ
- ✓ Mã hóa
- ✓ Toàn vẹn dữ liệu

Để bắt đầu một kết nối TCP/ IP giữa một trình duyệt web (phía client) và một web server an toàn, SSL tham gia một thỏa thuận an ninh (handshake) cho phép máy khách và máy chủ thỏa thuận về mức độ an toàn sẽ được sử dụng. Thỏa thuận trên bao gồm kiểm tra chứng chỉ số (digital certificates) của máy chủ. Chứng chỉ số được cấp bởi các bên thứ ba tin cậy là các file điện tử (chứa tên người dùng, khóa công khai của người dùng và tên của Tổ chức cấp chứng chỉ và các thuộc tính khác). Do đó, một chứng chỉ số kiểm tra kết nối giữa khóa công khai và ID (Identification) của server. Nếu quá trình khởi tạo này được thực hiện thành công, tất cả việc truyền dữ liệu giữa client và server sẽ được mã hóa bởi hệ mật mã RSA.

SSL sử dụng khóa có chiều dài 64 bit hoặc 128 bit, khóa này gọi là khóa phiên (session key). Sử dụng khóa có chiều dài càng lớn thì độ an toàn càng cao. Trong các phiên bản trình duyệt web hỗ trợ SSL 4.0 cho phép người dùng sử dụng key với độ dài 128 bit mã hóa các giao dịch.

### 9.3.2. Secure Electronic Transaction

Secure Electronic Transaction (Giao dịch điện tử an toàn - SET) do Visa và MasterCard phối hợp phát triển và được hỗ trợ bởi GTE, IBM, Microsoft, Netscape, RSA, SAIC, Terisa và VeriSign. Mục tiêu chính của SET là cung cấp thanh toán an toàn bằng thẻ tín dụng qua mạng mở như Internet .

SET về cơ bản tương tự như SSL, bảo vệ thông tin thanh toán dựa trên việc chứng thực (chứng thực bên bán và chủ thẻ) và mã hóa thông tin thanh toán. SET bảo vệ thanh toán bằng thẻ tín dụng bằng các yếu tố sau:

- ✓ Nó cho phép chủ thẻ xác thực rằng bên bán được ủy quyền chấp nhận thanh toán bằng thẻ an toàn sử dụng công nghệ SET
- ✓ Nó cho phép bên bán sử dụng công nghệ SET để xác thực thẻ thanh toán trong giao dịch

- ✓ Công nghệ SET sử dụng hệ thống mã hóa tiên tiến để bảo vệ thông tin thanh toán cá nhân trong quá trình truyền qua mạng
- ✓ Công nghệ SET chỉ bảo đảm rằng người nhận chỉ đọc thông tin thanh toán. Thông tin chỉ có thể được giải mã bởi bên bán và một tổ chức tài chính mà cả hai đều sử dụng công nghệ SET hợp lệ

Dựa theo đặc điểm kỹ thuật, SET sử dụng cả mã hóa đối xứng và bất đối xứng để bảo vệ giao dịch. Một giao dịch an toàn điển hình sử dụng SET trong thương mại điện tử B2C như mua sắm trực tuyến có thể được mô tả như sau:

Cả hai bên (bên mua và bên bán) trao đổi khoá công cộng của họ thông qua Chứng chỉ số (digital certificate) được cấp bởi nhà cung cấp chứng chỉ số (Certificate Authority) trong quá trình khởi tạo.

Tại nơi chủ thẻ:

Bước 1. Chủ thẻ mã hóa thông tin thanh toán bằng cách sử dụng khóa cá nhân (private key), đồng nghĩa với việc chủ thẻ ký số thanh toán.

Bước 2. Chủ thẻ tiếp tục mã hóa thông tin thanh toán đã được mã hóa ở trên bằng mã hoá đối xứng sử dụng khóa ngẫu nhiên để chắc chắn thông tin bảo mật.

Bước 3. Cuối cùng chủ thẻ mã hóa thông tin (từ bước 2) bằng khóa công khai của bên bán, tạo ra một "phong bì số" (digital envelope) an toàn và gửi nó tới bên bán.

Bên phía bán:

Bước 1. Bên bán mở "phong bì số" bằng cách sử dụng khóa cá nhân. Trong trường hợp này chỉ có bên bán mới có thể mở được.

Bước 2. Bên bán kiểm tra chữ ký số của chủ thẻ bằng cách sử dụng khóa công khai của chủ thẻ.

Bước 3. Bên bán giải mã thông tin thanh toán bằng cách sử dụng khóa đối xứng được gửi bởi chủ thẻ.

Mô hình trên cho phép bên bán truy cập đến số tài khoản của chủ thẻ. Để ẩn số thẻ thanh toán đối với bên bán, giao thức SET sử dụng cổng thanh toán (nằm trong bộ xử lý thẻ tín dụng). Lúc này, thông tin đặt hàng của chủ thẻ sẽ được mã hóa bằng khóa công khai của người bán và chi tiết của thẻ tín dụng được mã hóa bằng khóa công khai của cổng thanh toán và gửi cho bên bán. Khi đó, bên bán chỉ có thể xem thông tin đặt hàng và chuyển chi tiết về thẻ tín dụng sang cổng thanh toán để xác minh.

#### 9.4. Câu hỏi và bài tập

1. Giao thức nào được dùng để mã hóa giữa Web server và web client? Giải thích sự hoạt động của giao thức đó.
2. Mô tả tấn công chèn giữa (Main-in-the-middle) đối với Wifi và cho biết hậu quả của tấn công kiểu này và biện pháp phòng chống.

3. Nếu ở nơi công cộng sử dụng https trong trình duyệt qua WiFi thì nguy cơ mất an toàn đối với thông tin thẻ tín dụng khi thực hiện mua hàng trực tuyến có khả năng xảy ra không? Giải thích lí do?
4. Giải thích lí do tại sao giải pháp chia sẻ khóa bí mật lại không phù hợp để sử dụng trên internet.
5. Nêu thuật toán mã hóa và công nghệ bảo mật trong SET.

### **TÀI LIỆU THAM KHẢO**

- [1] Stallings W. *Cryptography and Network Security : Principles and Practice*. Prentice Hall. 2011.
- [2] Luật sở hữu trí tuệ năm 2005
- [3] Nghị định số 100/2006/NĐ-CP ngày 21 tháng 9 năm 2006 quy định chi tiết và hướng dẫn thi hành một số điều của Bộ luật Dân sự, Luật Sở hữu trí tuệ về quyền tác giả và quyền liên quan quyền tác giả
- [4] Nghị định số 85/2011/NĐ-CP ngày 20/9/2011 về sửa đổi, bổ sung một số điều của Nghị định số 100/2006/NĐ-CP
- [5] P. Szor. *The Art of Computer Virus Research and Defense*. Addison-Wesley Professional, 2005.
- [6] Michael Sikorski, Andrew Honig. *Practical Malware Analysis*. Starch Press, 2012.





# MỤC LỤC

# MỤC LỤC

<b>CHƯƠNG 1: TỔNG QUAN VỀ AN TOÀN THÔNG TIN .....</b>	<b>1</b>
1.1.Các khái niệm cơ bản.....	1
1.2.Các nguyên tắc nền tảng của an toàn thông tin .....	2
1.2.1.Tính bí mật ( <i>Confidentiality</i> ) .....	2
1.2.2.Tính toàn vẹn ( <i>Integrity</i> ).....	2
1.2.3.Tính sẵn sàng ( <i>Availability</i> ).....	3
1.2.4.Tính chống thoái thác ( <i>Non-repudiation</i> ).....	3
1.3.Các loại hình tấn công và nguy cơ mất ATTT .....	3
1.3.1.Các khái niệm tấn công.....	3
1.3.2.Một số kỹ thuật tấn công mạng.....	4
1.3.4.Các nguy cơ mất ATTT .....	7
1.4.Giải pháp đảm bảo an toàn thông tin .....	7
1.5.Các bài toán an toàn thông tin cơ bản.....	8
1.5.1.Ví dụ: .....	8
1.5.2.1.Bài toán bảo mật: mã hóa và phong bì số.....	10
1.5.3.Bài toán chứng thực và toàn vẹn: chữ ký số và mã chứng thực .....	12
1.6.Pháp luật về an toàn thông tin.....	13
1.6.1.Tin tặc, tội phạm kỹ thuật .....	13
1.6.1. Một số tội phạm tin học liên quan đến Internet .....	14
1.6.2.Vấn đề sở hữu trí tuệ và bản quyền .....	14
1.6.3.Luật về tội phạm tin học ở Việt Nam .....	15
1.7.Các phần mềm độc hại.....	15
1.8.Câu hỏi và bài tập .....	15
<b>CHƯƠNG 2: MÃ ĐỘC .....</b>	<b>17</b>
2.1.Giới thiệu Malware .....	17
2.2.Phân loại Malware .....	17
2.2.1.Virus.....	17
2.2.2.Worm .....	19
2.2.3.Trojan.....	20

2.2.4.Backdoor .....	20
2.2.5.Keylogger.....	21
2.2.6.Rootkit .....	21
2.2.7.Adware/Spyware.....	21
2.2.8.Attacker Tool .....	22
2.3.Các kĩ thuật lây nhiễm và phá hoại trong Malware .....	22
2.3.1.Kĩ thuật lây nhiễm .....	22
2.3.2.Kĩ thuật phá hoại.....	22
2.4.Tổng quan các kỹ thuật phát hiện Malware.....	23
2.5.Câu hỏi và bài tập .....	23
<b>CHƯƠNG 3: MÃ HÓA .....</b>	<b>26</b>
3.1.Khái niệm về mã hóa .....	26
3.2.Sơ đồ mã hóa .....	27
3.3.Phân loại mã hóa.....	27
3.3.1.Mã hóa cổ điển.....	28
3.3.2.Mã hóa đối xứng .....	31
3.3.3.Mã hóa bất đối xứng .....	38
3.4.Câu hỏi và bài tập .....	41
<b>CHƯƠNG 4: HÀM BẮM VÀ ỨNG DỤNG .....</b>	<b>44</b>
4.1.Định nghĩa.....	44
4.2.Một số hàm băm thông dụng .....	45
4.3.Ứng dụng của hàm băm.....	45
4.4.Tấn công hàm băm.....	46
4.5.Câu hỏi và bài tập .....	47
<b>CHƯƠNG 5: MÃ CHỨNG THỰC THÔNG ĐIỆP .....</b>	<b>49</b>
5.1.Toàn vẹn dữ liệu (Message Integrity).....	49
5.2.Xác thực thông điệp (Message Authentication) .....	49
5.2.1.Khái niệm.....	49
5.2.2.Mục tiêu của xác thực thông điệp:.....	49
5.3.Các phương pháp chứng thực thông điệp .....	50
5.3.1.Mã hóa thông điệp: .....	50
5.3.2.Mã chứng thực thông điệp .....	51

5.3.3. Vài cơ chế của MAC .....	53
5.4. Câu hỏi và bài tập .....	56
<b>CHƯƠNG 6: CHỮ KÝ ĐIỆN TỬ.....</b>	<b>57</b>
6.1. Khái niệm chữ ký điện tử .....	57
6.2. Chữ ký số .....	58
6.2. Một số loại chữ ký số.....	59
6.2.1. Chữ ký số RSA .....	59
6.2.2. Chữ ký số El-Gamal .....	60
6.2.3. Chữ ký số DSS.....	60
6.3. Câu hỏi và bài tập .....	61
<b>CHƯƠNG 7: KIẾN TRÚC KHÓA CÔNG KHAI .....</b>	<b>63</b>
7.1. Mô hình kiến trúc khóa công khai .....	63
7.2. Các chức năng quản trị PKIX .....	64
7.2.1. Đăng ký (Registration) .....	64
7.2.2. Khởi tạo (Initialization) .....	64
7.2.3. Xác nhận (Certification) .....	64
7.2.4. Khôi phục cặp khóa (Key pair recovery).....	64
7.2.5. Cập nhật key pair .....	65
7.2.6. Yêu cầu hủy (Revocation request).....	65
7.2.7. Xác thực chéo (Cross certification) .....	65
7.3. Phân phối khóa (Key Distribution).....	65
7.3.1. Phân phối khóa đối xứng sử dụng mã hóa đối xứng .....	65
7.3.2. Phân phối khóa sử dụng mã hóa bất đối xứng.....	70
7.4. Thẻ chứng thực X.509 .....	72
7.4.1. Certificate.....	72
7.4.2. Nhận certificate người dùng .....	74
7.4.3. Hủy certificate.....	77
7.5. Kerberos.....	77
7.5.1. Giới thiệu .....	77
7.5.2. Kerberos Version 4 .....	78
7.6. Câu hỏi và bài tập .....	88
<b>CHƯƠNG 8: CHỨNG THỰC THỰC THỂ .....</b>	<b>90</b>

---

8.1.Khái niệm.....	90
8.2.Mật khẩu (Password) .....	90
8.2.1.Fixed password: .....	91
8.2.2.One-time Password .....	92
8.3.Challenge-Response .....	93
8.3.1.Dùng Sysmetric-Key Cipher.....	93
8.3.2.Using Keyed-Has Functions .....	94
8.3.3.Sử dụng mã hóa khóa đối xứng (Asymmetric-Key Cipher).....	94
8.3.4.Sử dụng chữ ký số (Digital Signature) .....	95
8.4. ZERO-KNOWLEDGE .....	96
8.4.1.Giao thức Fiat_Shamir.....	96
8.4.2.Giao thức Feige-Fiat-Shamir .....	97
8.4.3.Giao thức Guillou-Quisquater .....	98
8.5.Biometrics .....	99
8.6.Câu hỏi và bài tập .....	99
<b>CHƯƠNG 9: MỘT SỐ GIAO THỨC BẢO MẬT THÔNG DỤNG .....</b>	<b>101</b>
9.1.Các giao thức bảo mật E-mail.....	101
9.1.1.Kiến trúc e-mail .....	101
9.1.2.Bảo mật e-mail.....	102
9.2.Các giao thức bảo mật mạng.....	102
9.3.Các giao thức thanh toán điện tử .....	103
9.3.1.Secure Socket Layer .....	104
9.3.2. Secure Electronic Transaction .....	104
9.4. Câu hỏi và bài tập .....	105