**CRYPTOGRAPHY – MẬT MÃ HỌC**

# **1.Những kiến thức cơ bản.**

## a.Một số khái niệm cơ bản

* **Mật mã học** (*Cryptography*) là ngành khoa học nghiên cứu về việc đảm bảo an toàn thông tin. Mật mã học gắn liền với quá trình mã hóa nghĩa là chuyển đổi thông tin từ dạng "có thể hiểu được" thành dạng "không thể hiểu được" và ngược lại là quá trình giải mã. Cryptography giúp đảm bảo những **tính chất** sau cho thông tin:
* *Tính bí mật (confidentiality):* thông tin chỉ được tiết lộ cho những ai được phép
* *Tính toàn vẹn (integrity):* thông tin không thể bị thay đổi mà không bị phát hiện.
* *Tính xác thực (authentication):* người gửi (hoặc người nhận) có thể chứng minh đúng họ.
* *Tính chống chối bỏ (non-repudiation):* người gửi hoặc nhận sau này không thể chối bỏ việc đã gửi hoặc nhận thông tin.
* **Thám mã** (*cryptanalysis*): nghiên cứu cách phá các hệ mật nhằm phục hồi bản rõ ban đầu từ bản mã, nghiên cứu các nguyên lí và phương pháp giải mã mà không biết khóa. Có 3 phương pháp tấn công cơ bản của thám mã:
* Tìm khóa vét cạn
* Phân tích thống kê
* Phân tích toán
* **Bản rõ** (*Plaintext*): Dạng ban đầu của thông báo **Bản mã** (Ciphertext): Dạng mã của bản rõ ban đầu **Khóa (Key)**: thông tin tham số dùng để mã hóa **Mã hóa** (Encryption): Quá trình biến đổi thông tin từ dạng bản rõ sang bản mã bằng khóa hoặc không cần khóa **Giải mã** (Decryption): Quá trình ngược lại biến đổi thông tin từ dạng bản mã sang bản rõ

# b.Phân loại thuật toán mã hóa

Có nhiều cách phân loại thuật toán mã hóa nhưng cách phân loại theo khóa được sử dụng phổ biến hơn

## Thuật toán mã hóa khóa đối xứng

* Mã hóa đối xứng còn có một số tên gọi khác như Secret Key Cryptography (hay Private Key Cryptography), sử dụng cùng một khóa cho cả hai quá trình mã hóa và giải mã.
* Quá trình thực hiện như sau :Trong hệ thống mã hóa đối xứng, trước khi truyền dữ liệu 2 bên gửi và nhận phải thỏa thuận về khóa dùng chung cho qúa trình mã hóa và giải mã, khóa được sinh ra bởi bộ sinh khóa. Sau đó, bên gửi sẽ mã hóa Plaintext bằng cách sử dụng khóa bí mật này và gửi thông điệp đã mã hóa cho bên nhận. Bên nhận sau khi nhận được thông điệp đã mã hóa sẽ sử dụng chính khóa bí mật mà hai bên thỏa thuận để giải mã và lấy ra Plaintext.
* Những gì bạn nhìn thấy hình trên chính là quá trình tiến hành trao đổi thông tin giữa bên gửi và bên nhận thông qua việc sử dụng phương pháp mã hóa đối xứng.Trong quá trình này thì thành phần quan trọng nhất cần phải được giữ kín chính là khóa. Việc trao đổi, thỏa thuận về thuật toán được sử dụng trong việc mã hóa có thể tiến hành một cách công khai, nhưng bước thỏa thuận về khóa trong việc mã hóa và giải mã phải tiến hành bí mật. Phải có bước thỏa thuận về khóa, vì hai lý do cơ bản sau đây :
* Nếu như hai bên đã thống nhất thuật toán mã hóa, nhưng sau đó nơi gửi gửi thông điệp đã mã hóa tới cho nơi nhận mà không cho biết khóa đã sử dụng trong quá trình mã hóa. Nơi nhận không có khóa để giải mã cho nên sẽ chẳng hiểu được nội dung trong tài liệu muốn nói gì. Vì thế bắt buộc ngoài việc trao đổi về thuật toán thì cần phải trao đổi về khóa
* Khóa phải được trao đổi theo một kênh bí mật nào đó.Ví dụ trực tiếp trao đổi (mặt đối mặt) hay gián tiếp trao đổi (thông qua điện thoại, email, tin nhắn). Phải thực hiện qua kênh truyền bí mật vì rất có thể sẽ có một bên thứ ba nghe lén cuộc trao đổi giữa hai bên và có được khóa, như vậy thông tin trao đổi sẽ bị kẻ khác biết được. Thêm vào đó hai bên buộc phải tin cậy lẫn nhau, không thể nhờ một người khác gửi hộ khóa được vì rất có thể người này sẽ dùng khóa đó để ăn cắp thông tin Thuật toán mã hóa loại này có ưu điểm là nhanh, độ an toàn gần như tuyệt đối nếu khô để lộ khóa bí mật. Chính vì vậy mấu chốt của việc sử dụng loại thuật toán mã hóa này là việc truyền khóa bí mật cho bên nhận một cách an toàn.
* Một số thuật toán mã hóa khóa công khai phổ biến
* DES viết tắt của Data Encryption Standard. Với DES, Plaintext được mã hóa theo từng khối 64 bits và sử dụng một khóa là 64 bits, nhưng thực tế thì chỉ có 56 bits là thực sự được sự dụng để tạo khóa. Là một thuật toán được sử dụng rộng rãi nhất trên thế giới với bề dày lịch sử hơn 20 năm, được phát triển bởi IBM Team vào những năm 1970, sau đó được phát triển bởi NIST (National Institute of Standard and Technology) cho các ứng dụng thương mại. Hiện tại DES không còn được đánh giá cao do kích thước của khóa là quá nhỏ 56 bit và nó dễ dàng bị phá vỡ chỉ trong vòng 24 tiếng đồng hồ.
* Triple DES (3DES) cải thiện độ mạnh của DES bằng việc sử dụng một quá trình mã hóa-giải mã-mã hóa mà sử dụng 3 khóa.Khối 64-bit Plaintext đầu tiên được mã hóa sử dụng khóa thứ nhất. Sau đó, dữ liệu bị mã hóa được giải mã bằng việc sử dụng một khóa thứ hai. Cuối cùng, sử dụng khóa thứ ba và kết quả của quá trình giải mã trên để mã hóa.
* AES : viết tắt của Advanced Encryption Standard. Xuất hiện theo lời kêu gọi của NIST, cần phải phát triển một thuật toán mới thay thế cho DES. AES chính thức thay thế cho DES vào tháng 11 năm 2001. Nó hỗ trợ độ lớn nhỏ nhất của khóa là 128, 192 và 256 bits. Và hiện giờ nó đang được sử dụng phổ

## Thuật toán mã hóa bất đối xứng

* Hay còn được gọi với một cái tên khác là mã hóa khóa công khai (Public Key Cryptography), nó được thiết kế sao cho khóa sử dụng trong quá trình mã hóa khác biệt với khóa được sử dụng trong quá trình giải mã. Hơn thế nữa, khóa dùng trong quá trình giải mã không thể được tính toán hay suy luận từ khóa dùng để mã hóa và ngược lại, tức là hai khóa này có quan hệ với nhau về mặt toán học nhưng không thể suy diễn được ra nhau. Thuật toán này được gọi là Public-Key bởi vì khóa dùng cho việc mã hóa được công khai cho tất cả mọi người. Một người hòan toàn xa lạ có thể dùng khóa này để mã hóa dữ liệu nhưng chỉ duy nhất người mà có khóa giải mã tương ứng mới có thể đọc được dữ liệu mà thôi. Do đó trong thuật mã hóa này Encryption key được gọi là Public key còn Decryption Key được gọi là Private key.Tại sao lại có sự xuất hiện của Mã hóa khóa công khai? Như đã trình bày với các bạn ở trên, thuật toán mã hóa khóa riêng tuy có tốc độ thực hiện rất nhanh nhưng nó có một số nhược điểm như sau:

### Khóa phải được trao đổi theo một kênh bí mật.

### Nếu mất khóa thì thông tin hoàn toàn có thể bị lấy cắp hoặc giả mạo.

### Khóa cần phải thay đổi một cách định kì

### Khi số lượng người dùng tăng lên thí số lượng khóa được sử dụng cũng tăng lên.

* Vậy là điểm yếu của thuật toán mã hóa khóa riêng nằm ở khâu quản lý khóa thế nào cho hợp lý. Mã hóa khóa công khai ra đời đã giải quyết được vấn đề này. Hình minh họa ở trên cho chúng ta thấy được quá trình truyền tin an toàn dựa vào hệ thống mã hóa khóa công khai. Như các bạn thấy, trong hệ thống mã hóa này thì mỗi một người sử dụng khi tham gia vào đều được cấp 2 khóa : Một khóa dùng cho việc mã hóa dữ liệu (Public key) và một khóa dùng cho việc giải mã dữ liệu (Private key), trong đó Public key được đưa ra cho tất cả mọi người cùng biết, còn Private key phải được giữ kín một cách tuyệt đối.Giả sử hai phía muốn truyền tin cho nhau thì quá trình truyền sử dụng mã hóa khóa công khai được thực hiện như sau :

### Sender yêu cầu cung cấp hoặc tự tìm khoá công khai của Receiver trên một Server chịu trách nhiệm quản lý khoá công khai.

### Sau đó hai phía thống nhất thuật toán dùng để mã hóa dữ liệu, Sender sử dụng khóa công khai của Receiver cùng với thuật toán đã thống nhất để mã hóa thông tin bí mật.

### Thông tin sau khi mã hóa được gửi tới Receiver, lúc này chính Sender cũng không thể nào giải mã được thông tin mà anh ta đã mã hóa (khác với mã hóa khóa riêng).

### Khi nhận được thông tin đã mã hóa, Receiver sẽ sử dụng khóa bí mật của mình để giải mã và lấy ra thông tin ban đầu.

* Vậy là với sự ra đời của Mã hóa khóa công khai thì khóa được quản lý một cách linh hoạt và hiệu quả hơn.Người sử dụng chỉ cần bảo vệ khóa Private key. Hệ thống này an toàn hơn nhiều so với mã hóa khóa riêng, người mã hóa không thể giải mã được dữ liệu đã mã hóa bằng khóa công khai của người khác.Tuy nhiên nhược điểm của Mã hóa khóa công khai nằm ở tốc độ thực hiện, nó chậm hơn mã hóa khóa riêng cỡ ~1000 lần. Do đó người ta thường kết hợp hai hệ thống mã hóa khóa riêng và công khai lại với nhau và được gọi là Hybrid Cryptosystems (Hệ thống mã hóa lai). Một số thuật toán mã hóa công khai phổ biến :

### RSA : Cái tên RSA là ba chữ cái bắt đầu từ ba cái tên của ba tác giả: Ron Rivest, Adi Shamir ,Len Adleman. Sử dụng đồng thời cho mã hóa khóa công khai và chữ kí điện tử. Độ an toàn của thuật toán mã hóa RSA dựa trên việc phân tích một số nguyên tố rất lớn thành hai số nguyên tố.

### Mã hóa một chiều

* Đôi khi ta chỉ cần mã hóa thông tin chứ không cần giải mã thông tin, khi đó ta sẽ dùng đến phương pháp mã hóa một chiều (Chỉ có thể mã hóa chứ không thể giải mã). Thông thường phương pháp mã hóa một chiều sử dụng một hàm băm (hash function) để biến một chuỗi thông tin thành một chuỗi hash có độ dài nhất định. Ta không có bất kì cách nào để khôi phục (hay giải mã) chuỗi hash về lại chuỗi thông tin ban đầu Hàm băm (Hash function) là một hàm mà nó nhận vào một chuỗi có độ dài bất kì, và sinh ra một chuỗi kết quả có độ dài cố định (Gọi là chuỗi hash), dù hai chuỗi dữ liệu đầu vào, được cho qua hàm băm thì cũng sinh ra hai chuỗi hash kết quả khác nhau rất nhiều. Ví dụ như đối với kiểu dữ liệu Hash-table, ta có thể coi đây là một dạng kiểu dữ liệu mảng đặc biệt mà index nó nhận vào là một chuỗi, nó được định nghĩa bằng cách bên trong nó chứa một mảng thông thường, mỗi khi truyền vào index là một chuỗi, thì chuỗi này sẽ đi qua hàm băm và ra một giá trị hash, giá trị này sẽ tương ứng với index thật của phần tử đó trong mảng bên dưới. Đặc điểm của hash function là khi thực hiên băm hai chuỗi dữ liệu như nhau, dù trong hoàn cảnh nào thì nó cũng cùng cho ra một chuỗi hash duy nhất có độ dài nhất định và thường nhỏ hơn rất nhiều so với chuỗi gốc, và hai chuỗi thông tin bất kì dù khác nhau rất ít cũng sẽ cho ra chuỗi hash khác nhau rất nhiều. Do đó hash function thường được sử dụng để kiểm tra tính toàn vẹn của dữ liệu.
* Giả sử bạn có một file dữ liệu định up lên mạng, và bạn muốn người dùng có thể kiểm tra xem dữ liệu họ down về có chính sát dữ liệu mình up lên hay không. Thì bạn sẽ dùng một hash function để băm dữ liệu của file đó ra một chuỗi hash, và gửi kèm cho người dùng chuỗi hash này. Khi đó, người dùng chỉ việc dùng đúng hash function đó để tìm chuỗi hash hiện tại của file down về, rồi so sánh với chuỗi hash ban đầu, nếu hai chuỗi này giống nhau thì dữ liệu down về vẫn toàn vẹn.
* Ngoài ra có một ứng dụng mà có thể bạn thường thấy, đó là để lưu giữ mật khẩu. Vì mật khẩu là một thứ cực kì quan trọng, do đó ta không nên lưu mật khẩu của người dùng dưới dạng rõ, vì như vậy nếu bị hacker tấn công, lấy được CSDL thì hacker có thể biết được mật khẩu của người dùng. Do đó, mật khẩu của người dùng nên được lưu dưới dạng chuỗi hash, và đối với server thì chuỗi hash đó chỉnh là “mật khẩu” đăng nhập (lúc đăng nhập thì mật khẩu mà người dùng nhập cũng được mã hóa thành chuỗi hash và so sánh với chuỗi hash trong CSDL của server). Dù hacker có lấy được CSDL thì cũng không tài nào có thể giải mã được chuỗi hash để tìm ra mật khẩu của người dùng.
* Thuật toán mã hóa một chiều (hàm băm) mà ta thường gặp nhất là MD5 và SHA MD5 đã được chứng minh là không còn an toàn nhưng nó vẫn được sử dụng rất phổ biến trong việc lưu mât khẩu SHA ra đời sau MD5 và đã có phiên bản SHA-256, SHA-512 có độ an toàn cao hơn

c.Triển khai :

Mã hóa đối xứng(bí mật)

* DES:
* AES:

Mã hóa bất đối xứng:

RSA

Mã hóa 1 chiều(Hàm băm) – MD5 – SHA MD5