### Chương 1

### CÁC KHÁI NIỆM CƠ SỞ & HỆ MÃ CỔ ĐIỂN

#### 1.Mật mã là gì?

Mật mã là một lĩnh vực khoa học chuyên nghiên cứu về các phương pháp và kỹ thuật đảm bảo an toàn và bảo mật trong truyền tin liên lạc với giả thiết sự tồn tại của các thế lực thù địch, những kẻ muốn ăn cắp thông tin để lợi dụng và phá hoại.

Cụ thể hơn, các nhà nghiên cứu lĩnh vực này quan tâm xây dựng hoặc phân tích (để chỉ ra điểm yếu) các giao thức mật mã (cryptographic protocols), tức là các phương thức giao dịch có đảm bảo mục tiêu an toàn cho các bên tham gia (với giả thiết môi trường có kẻ đối địch, phá hoại).

Ngành Mật mã (cryptology) thường được quan niệm như sự kết hợp của 2 lĩnh vực con:

1. Sinh, chế mã mật (cryptography): nghiên cứu các kỹ thuật toán học nhằm cung cấp các công cụ hay dịch vụ đảm bảo an toàn thông tin
2. Phá giải mã (cryptanalysis): nghiên cứu các kỹ thuật toán học phục vụ phân tích phá mật mã và/hoặc tạo ra các đoạn mã giản nhằm đánh lừa bên nhận tin.

Hai lĩnh vực con này tồn tại như hai mặt đối lập, “đấu tranh để cùng phát triển” của một thể thống nhất là ngành khoa học mật mã (cryptology). Tuy nhiên, do lĩnh vực thứ hai (cryptanalysis) ít được phổ biến quảng đại nên dần dần, cách hiểu chung hiện nay là đánh đồng hai thuật ngữ cryptography và cryptology. Theo thói quen chung này, hai thuật ngữ này có thể dùng thay thế nhau. Thậm chí cryptography là thuật ngữ ưa dùng, phổ biến trong mọi sách vở phổ biến khoa học, còn cryptology thì xuất hiện trong một phạm vi hẹp của các nhà nghiên cứu học thuật thuần túy.

#### 2. Các ứng dụng của mật mã là gì?

* Với các chính phủ: bảo vệ truyền tin mật trong quân sự và ngoại giao, bảo vệ thông tin các lĩnh vực tầm cỡ lợi ích quốc gia.
* Trong các hoạt động kinh tế: bảo vệ các thông tin nhạy cảm trong giao dịch như hồ sơ pháp lý hay y tế, các giao dịch tài chính hay các đánh giá tín dụng …
* Với các cá nhân: bảo vệ các thông tin nhạy cảm, riêng tư trong liên lạc với thế giới qua các giao dịch sử dụng máy tính và/hoặc kết nối mạng.

#### 3. Những kỷ nguyên quan trọng trong ngành mật mã ?

*Thời kỳ tiền khoa học*: Tính từ thượng cổ cho đến 1949. Trong thời kỳ này, khoa mật mã học được coi là một ngành mang nhiều tính thủ công, nghệ thuật hơn là tính khoa học. Các hệ mật mã được phát minh và sử dụng trong thời kỳ này được gọi là các hệ mật mã cổ điển

*Kỷ nguyên mật mã được coi là ngành khoa học*: được đánh dấu bởi bài báo nổi tiếng của Claude Shannon “Commication theory of secretcy systems” , được công bố năm 1949. Công trình này dựa trên một bài báo trước đó của ông mà trong đó ông cũng đã khai sáng ra ngành khoa học quan trọng khác, *lý thuyết thông tin* (inforrmation theory).

#### 4. Mô hình truyền tin mật cơ bản như thế nào?



Sender S



Receiv

er

R



Ene

my E



K

ey Z



Y=E

Z

X

(

)



Ke

y Z

‟



X=D

Z

)

(

Y

Người phát S (sender) muốn gửi một thông điệp (message) X tới người nhận R (receiver) qua một kênh truyền tin (communication channel). Kẻ thù E (enenmy) lấy/nghe trộm thông tin X. Thông tin X là ở dạng đọc được, còn gọi là bản rõ (plaintext). Để bảo mật, S sử dụng một phép biến đổi mã hoá (encryption), tác động lên X, để chế biến ra một bản mã Y (cryptogram, hay ciphertext), không thể đọc được. Ta nói bản mã Y đã che giấu nội dung của bản rõ X bản đầu. Giải mã (decryption) là quá trình ngược lại cho phép người nhận thu được bản rõ X từ bản mã Y.

Để bảo mật, các khối biến đối sinh và giải mã là các hàm toán học với tham số khoá (key). Khóa là thông số điều khiển mà sở hữu kiến thức về nó thông thường là hạn chế. Thông thường khoá (Z) chỉ được biết đến bởi các bên tham gia truyền tin S và R.

#### 5. Hệ thống mật mã đối xứng (Symmetric Key Cryptosystem - SKC) là gì?

Trong mô hình của hệ thống này, khóa của hai thuật toán sinh mã và giải mã là giống nhau và bí mật đối với tất cả những người khác; nói cách khác, hai bên gửi và nhận tin chia sẻ chung một khóa bí mật duy nhật. Vai trò của hai phía tham gia là giống nhau và có thể đánh đổi vai trò, gửi và nhận tin, cho nên hệ thống được gọi là “mã hóa đối xứng”. Chúng ta sẽ sử dụng ký hiệu viết tắt theo tiếng Anh là SKC.

#### 6. Nhược điểm của hệ thống mật mã đối xứng?

Hệ thống mật mã khóa bí mật đối xứng có những nhược điểm lớn trên phương diện quản lý và lưu trữ, đặc biệt bộc lộ rõ trong thế giới hiện đại khi liên lạc qua Internet đã rất phát triển. Nếu như trong thế giới trước kia liên lạc mật mã chỉ hạn chế trong lĩnh vực quân sự hoặc ngoại giao thì ngày nay các đối tác doanh nghiệp khi giao dịch qua Internet đều mong muốn bảo mật các thông tin quan trọng. Với hệ thống khóa bí mật, số lượng khóa bí mật mà mỗi công ty hay cá nhân cần thiết lập với các đối tác khác có thể khá lớn và do đó rất khó quản lý lưu trữ an toàn các thông tin khóa riêng biệt này.

Một khó khăn đặc thù khác nữa là vấn đề xác lập và phân phối khóa bí mật này giữa hai bên, thường là đang ở xa nhau và chỉ có thể liên lạc với nhau qua một kênh truyền tin thông thường, không đảm bảo tránh được nghe trộm.

#### 7. Thành phần của hệ thống mật mã khóa công khai hay phi đối xứng (Public Key Cryptosystem – PKC).

* Thành phần khóa công khai, có thể đăng ký phổ biến rộng khắp, dùng để sinh mã hoặc để xác thực chữ ký điện tử
* Thành phần khóa bí mật, chỉ dành riêng cho bản thân, dùng để giải mã hoặc tạo ra chữ ký điện tử.

Chỉ với cặp khóa này, thực thể chủ có thể giao dịch bảo mật với quảng đại xã hội, trong đó việc quản lý và lưu trữ có thể được tổ chức chặt chẽ mà việc phải tự nhớ thông tin mật là tối thiểu (giống như việc chỉ nhớ 1 mật khẩu hay một số PIN tài khoản ngân hàng).

#### 8. Cách đánh giá tính bảo mật của các hệ mật mã?

Ta có thể kết luận một hệ mã mật là *không an toàn* (insecure), bằng việc chỉ ra cách phá nó trong một mô hình tấn công (khái niệm sẽ giới thiệu sau đây) phổ biến, trong đó ta chỉ rõ được các mục tiêu về ATBM (security) không được đảm bảo đúng. Tuy nhiên để kết luận rằng một hệ mã là an toàn cao thì công việc phức tạp hơn nhiều. Thông thường, người ta phải đánh giá hệ mật mã này trong nhiều mô hình tấn công khác nhau, với tính thách thức tăng dần. Để có thể khẳng định tính an toàn cao, cách làm lý tưởng là đưa ra một chứng mình hình thức (formal proof), trong đó người ta chứng minh bằng công cụ toán học là tính ATBM của hệ mã đang xét là tương đương với một hệ mã kinh điển, mà tính an toàn của nó đã khẳng định rộng rãi từ lâu.

Như đã nói trên, người ta phủ định tính an toàn của một hệ mã mật thông qua việc chỉ ra cách phá cụ thể hệ mã này trên một mô hình tấn công (attack model) cụ thể. Mỗi mô hình tấn công sẽ định nghĩa rõ năng lực của kẻ tấn công, bao gồm năng lực tài nguyên tính toán, loại thông tin mà nó có khả năng tiếp cận để khai thác và khả năng tiếp xúc với *máy mật mã* (thiết bị phần cứng có cài đặt thuật toán sinh và giải mã). Các mô hình tấn công thường được sắp xếp theo thứ tự mạnh dần của năng lực kẻ tấn công. Nếu một hệ mật mã bị phá vỡ trong một mô hình tấn công căn bản (năng lực kẻ tấn công là bình thường) thì sẽ bị đánh giá là hoàn toàn không an toàn. Sau đây là một số mô hình tấn công phổ biến.

#### 9. Kể tên các mô hình tấn công phổ biến và cách hoạt động?

*Tấn công chỉ-biết-bản-mã* (*ciphertext-only attack*). Ở đây kẻ địch E chỉ là một kẻ hoàn toàn bên ngoài, tìm cách nghe trộm trên đường truyền để lấy được các giá trị Y, bản mã của thông tin gửi đi. Mặc dù kẻ địch E chỉ biết các bản rõ Y, nhưng mục tiêu nó hướng tới là khám phá nội dung một/nhiều bản rõ X hoặc lấy được khóa mật Z (trường hợp phá giải hoàn toàn). Đây là mô hình tấn công căn bản nhất trong đó kẻ địch không có năng lực quan hệ đặc biệt (như một số hình thức tấn công sau), diện thông tin tiếp xúc chỉ là các bản mã. Rõ ràng nếu một hệ mã mà không đứng vững được trong mô hình này thì phải đánh giá là không đáng tin cậy.

*Tấn công biết-bản-rõ* (*known-plaintext attack*). Mặc dù tên gọi hơi dễ hiểu nhầm, thực chất trong mô hình này ta chỉ giả thiết là E có thể biết một số cặp X-Y (bản rõ và bản mật tương ứng) nào đó. Nguyên nhân E thu được có thể hoàn toàn tình cờ hoặc nhờ một vài tay trong là nhân viên thấp cấp trong hệ thống. Tất nhiên mục tiêu của E là khám phá nội dung các bản rõ quan trọng khác và/hoặc lấy được khóa mật. Rõ ràng mô hình tấn công này làm mạnh hơn so với *tấn công chỉ qua bản mã*: Việc biết một số cặp X-Y sẽ làm bổ sung thêm đầu mối phân tích; đặc biệt từ bây giờ E có thể dùng phép thử loại trừ để vét cạn không gian khóa (*exshautive key search*) và tìm ra khóa đúng tức là sao cho Enc (K,X)=Y.

*Tấn công bản-rõ-chọn*-*sẵn* (*chosen-plaintext attack*). Trong mô hình này, không những E thu nhặt được một số cặp X-Y mà một số bản rõ X do bản thân E soạn ra (chosen plaintext). Điều này thoạt nghe có vẻ không khả thi thực tế, tuy nhiên ta có thể tưởng tượng là E có tay trong là một thư ký văn phòng của công ty bị tấn công, ngoài ra do một qui định máy móc nào đó tất cả các văn bản dù quan trọng hay không đều được truyền gửi mật mã khi phân phát giữa các chi nhánh của công ty này. Có thể nhận xét thấy rằng, việc được tự chọn giá trị của một số bản rõ X sẽ thêm nhiều lợi ích cho E trong phân tích quan hệ giữa bản mã và bản rõ để từ đó lần tìm giá trị khóa.

Một cách tương tự, người ta cũng sử dụng mô hình *tấn công bản-mã-chọn-sẵn* (*chosen-ciphertext attack*) trong đó kẻ địch có thể thu nhặt được một số cặp X-Y mà Y là giá trị được thiết kế sẵn. Trong thực tế điều này có thể xảy ra nếu như kẻ địch có thể truy nhập được vào máy mật mã 2 chiều (có thể sử dụng với cả 2 chức năng là sinh mã và giải mã). Tất nhiên cả hai dạng tấn công rất mạnh nói trên kẻ thù đều có thể khôn ngoan sử dụng một chiến thuật thiết kế bản rõ (hay bản mã) chọn sẵn theo kiểu thích nghi (adaptive), tức là các bản rõ chọn sau có thể thiết kế dựa vào kiến thức phân tích dựa vào các cặp X-Y đã thu nhặt từ trước.

#### 10. Các mô hình đánh giá với các mức độ mạnh đến yếu

*Bảo mật vô điều kiện* (*unconditional security*): Đây là mô hình đánh giá ATBM mức cao nhất, trong đó “vô điều kiện” được hiểu theo ý nghĩa của lý thuyết thông tin (information theory), trong đó các ý niệm về “lượng tin” được hình thức hóa thông qua các phép toán xác suất. Trong mô hình này, kẻ địch được coi là không bị hạn chế về năng lực tính toán, tức là có thể thực hiện bất kỳ khối lượng tính toán cực lớn nào đặt ra trong khoảng thời gian ngắn bất kỳ. Mặc dù có năng lực tính toán siêu nhiên như vậy, mô hình này chỉ giả thiết kẻ tấn công là người ngoài hoàn toàn (tức là ứng với mô hình tấn công chỉ-biết-bản-mã). Một hệ mật mã đạt được mức an toàn vô điều kiện, tức là có thể đứng vững trước sức mạnh của một kẻ địch bên ngoài (chỉ biết bản mã) có khả năng không hạn chế tính toán, được gọi là đạt đến *bí mật tuyệt đối* (*perfect secretcy*).

Một cách khái quát, việc nghe trộm được bản mã đơn giản là chỉ cung cấp một lượng kiến thức zero tuyệt đối, không giúp gì cho việc phá giải mã của kẻ địch. Việc biết bản mã không đem lại chút đầu mối gì cho khả năng lần tìm ra khóa của hệ mã.

*Bảo mật chứng minh được* (*provable security*): Đây cũng là một mô hình đánh giá mức rất cao, lý tưởng trong hầu hết các trường hợp. Một hệ mật mã đạt được mức đánh giá này đối với một mo hình tấn công cụ thể nào đó, nếu ta có thể chứng mình bằng toán học rằng tính an toàn của hệ mật là được qui về tính NP-khó của một bài toán nào đó đã được biết từ lâu (ví dụ bài toán phân tích ra thừa số nguyên tố, bài toán cái túi, bài toán tính logarit rời rạc ...). Nói một cách khác ta phải chứng minh được là kẻ thù muốn phá được hệ mã thì phải thực hiện một khối lượng tính toán tương đương hoặc hơn với việc giải quyết một bài toán NP-khó đã biết.

*Bảo mật tính toán được*, hay *bảo mật thực tiễn* (*computational security* hay *practical security*): Đây là một trong những mức đánh giá thường được áp dụng nhất trong thực tế (khi những mức bảo mật cao hơn được cho là không thể đạt tới). Khi đánh giá ở mức này với một hệ mã cụ thể, người ta lượng hóa khối lượng tính toán đặt ra để có thể phá hệ mã này, sử dụng kiểu tấn công mạnh nhất đã biết (thường kèm theo đó là mô hình tấn công phổ biến mạnh nhất). Từ việc đánh giá được khối lượng tính toán này cùng thời gian thực hiện (với năng lực kẻ địch mạnh nhất có thể trên thực tế), và so sánh với thời gian đòi hỏi đảm bảo tính mật trên thực tế, ta có thể đánh giá hệ mã có đạt an toàn thực tiễn cao hay không. Đôi khi, cơ sở đánh giá cũng dựa vào một bài toán khó nào đó mặc dù không đưa ra được một chứng minh tương đương thực sự

*Bảo mật tự tác* (*ad hoc security*): Một số hệ mật mã riêng được một số công ty hoặc cá nhân tự chế để phục vụ mục đích đặc biệt dùng nội bộ. Tác giả loại hệ mật mã có thể sử dụng những lập luận đánh giá hợp lý nhất định dựa trên việc ước đoán khối lượng tính toán của kẻ địch khi sử dụng những tấn công mạnh nhấn đã biết và lập luận về tính bất khả thi thực tiễn để thực hiện. Mặc dù vậy hệ mật mã này vẫn có thể bị phá bởi những tấn công có thể tồn tại mà chưa được biết tới đến thời điểm đó; vì vây, thực tế bảo mật ở mức này hàm nghĩa không có một chứng minh đảm bảo thực sự, nên không thể coi là tin cậy với đại chúng.

#### 11. Các loại mật mã

*Mật mã cộng (Additive cipher) - Mật mã Xeda (Ceasar)*

Mật mã cộng (Additive cipher) là một mật mã một bảng thế đặc biệt trong đó, phép biến đổi mã được biểu diễn thông qua phép cộng đồng dư như sau. Giả sử ta gán các giá trị từ A-Z với các số 1-25,0. Thế thì một chữ plaintext X có thể mã thành ciphertext Y theo công thức:

*Y = X*  *Z,*

trong đó Z là giá trị của khoá,  là ký hiệu phép cộng đồng dư modulo 26.

Mật mã nhân tính (multiplicative cipher)

Bảng thế cũng có thể được xây dựng từ phép nhân đồng dư của chữ cái trong bảng gốc với giá trị của khóa:

*Y=X**Z*

Khoảng đầu thiên nhiên kỷ thứ hai, mật mã một bảng thế đã bị phá và các nhà khoa học đã dần nghĩ đến các nguyên tắc thiết kế mã tốt hơn, nhằm tránh bảo tồn các qui luật thống kê từ TIN sang MÃ (bản rõ sang bản mã). Ta sẽ xem xét một số mã như vậy sau đây.

*Mã với bảng thế đồng âm (homophonic substitution ciphers)*

Trong các cipher loại này, ánh xạ chữ cái TIN- MÃ không còn là 1-1 nữa mà là một-nhiều. Tức là mỗi chữ của bảng chữ cái tin sẽ được mã hoá thành 1 chữ trong 1 tập con các chữ mã nào đó. Mỗi chữ mã trong tập con này được gọi là homophone, tạm dịch là đồng âm.

Thông thường người ta bố trí số lượng đồng âm ứng với mỗi chữ tin tỷ lệ với tần xuất xuất hiện của chữ đó trong ngôn ngữ tự nhiên. Vì vậy đồ thị tần xuất của các chữ cái trong bản mã sẽ trở nên bằng phẳng. Mặc dù các cipher loại này là khó phá hơn nhưng chúng lại bị tăng thêm độ dư thừa so với tin gốc.

*Sử dụng nhiều bảng thế (mã đa bảng thế)*

#### 12. Phương pháp giải Vigenere cipher ?

Trong Vigenere Cipher, người ta dùng tất cả 26 bảng thế là sự thuđược từ bảng gốc chữ cái tiếng Anh mà dịch đi từ 0-25 vị trí. Sự hoà trộn này có quy luật hoàn toàn xác định bởi khoá. Mỗi chữ của khoá sẽ xác định mỗi bảng thế được dùng.

*Phương pháp giải mã Vigenere.*

Ý tưởng của phương pháp này gồm 3 bước như sau:

1. Đi tìm chu kỳ p (độ dài khoá)
2. Chia tách MÃ thành p đoạn phân mã, mỗi đoạn bao gồm các chữ ở vị trí kp+i (k=1,2,3 ... ; i=0,p-1), tức là được mã hoá theo bảng thế với chữ khoá chỉ số *i*.
3. Dùng phương pháp một bảng thế đã biết để giải từng đoạn phân mã (cụ thể là với mã Vigenere chỉ cần một phép dịch đúng)

Người ta sử dụng khái niệm IC (Index of Coincidence) để tính chu kỳ p.

Theo định nghĩa, IC xác định qua công thức:

25 *fi* (*fi -1*)

 i=0

IC = ----------------- n(n-1)

Trong đó *f* là xác xuất của phép thử - nhặt ra 2 con chữ ngẫu nhiên bất kỳ từ trong một đoạn văn bản - để thu được cùng một chữ  cho trước.

Phương pháp thực hành

1. Đặt k=1
2. Kiểm tra xem p có phải nhận giá trị k hay không.
   1. Chia Mã thành k phân mã và tính IC của các phân mã.
   2. Nếu như chúng đều xấp xỉ nhau và đều xấp xỉ 0.068 thì p=k Nếu chúng khác nhau nhiều và nhỏ hơn nhiều so với 0.068 thì p>k
3. Tăng k lên một đơn vị và lặp lại bước 2.

#### 13 One-time-pad (Vernam cipher) là gì?

Mật mã One-time-pad được đề xuất bởi G. Vernam (1917); sau đó đã được chứng minh là đảm bảo bí mật tuyệt đối (perfect secretcy - 1949). Như tên gọi của nó, trong One-time-pad khóa được viết trên 1 băng (tape) dài, và sử dụng đúng 1 lần. Đồng thời chuỗi khóa là chuỗi văn bản sinh ngẫu nhiên, có độ dài bằng văn bản sử dụng hoặc hơn. Thao tác mã hóa đơn giản là phép dịch theo bảng thế ứng với chữ khóa tương ứng hoặc XOR nếu xử lý theo chuỗi nhị phân.

Sinh mã: Y = X + Z (mod 26)

Giải mã : X = Y - Z (mod 26)

Vì vậy, One-time-pad có thể coi là mã Vigenere với khóa là một chuỗi ngẫu nhiên có độ dài đúng bằng văn bản, như ví dụ sau sẽ cho thấy

Chú ý rằng khóa chỉ được dùng đúng một lần, tức là vứt bỏ sau khi dùng. Nếu dùng lại thì không còn đảm bảo an toàn nữa.

#### 14. Khái niệm bí mật tuyệt đối

Trong hệ thống *đảm bảo bí mật tuyệt đối*, bản mã bị tiết lộ cho kẻ thù không hề đem lại một ý nghĩa nào cho phân tích tìm khóa phá mã. Sự kiện nghe trộm bản mã (có độ dài bất kỳ) sẽ không làm thay đổi phân phối xác xuất ban đầu của plaintext.

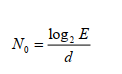
*Hay là, một hệ thống là có bí mật tuyệt đối nếu:*

*P(X) = P(X/Y)*  *TIN X VÀ MÃ Y*

**Định lý Shannon.** Trong hệ thống có BMTĐ, số lượng khoá có thể (độ lớn không gian khoá) phải lớn hơn hoặc bằng số lượng thông báo có thể (độ lớn không gian TIN).

Điều này cho thấy để đạt được BMTĐ thì khoá phải rất dài, do đó việc trao chuyển khoa giữa hai bên truyền tin sẽ làm cho hệ thống trở nên phi thực tế. Như vậy, nhìn chung chúng ta không thể đạt được bí mật tuyệt đối mà chỉ có thể có được các hệ thống với mức an toàn thực tế (Practical security) được cài đặt tuỳ theo giá trị của thông tin cần bảo vệ và thời gian sống của nó.

#### 15. Đánh giá mức độ bảo mật của một cipher?

Shannon đưa ra một khái niệm, *unicity distance*, để “đo” mức an toàn của một hệ mã: Unicity distance, ký hiệu N0, là độ dài tối thiểu của bản mã nghe trộm được để có thể xác định được khóa đúng duy nhất. Unicity distance có thể được tính theo công thức:

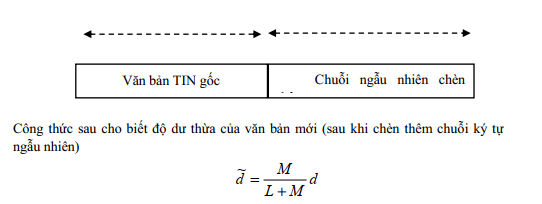
Trong đó d là *độ dư thừa* của ngôn ngữ sử dụng của TIN.

#### 16. Ta có thể “tăng” tính mật của một hệ mã cho trước hay không?

* Tăng độ lớn không gian khóa
* Giảm tính dư thừa của ngôn ngữ văn bản TIN: tiền xử lý qua 1 bước thuật toán nén

Chú ý: một thuật toán nén lý tưởng có thể đem lại độ dư thừa 0, do đó N0  0

* Có thể chèn thêm một đoạn văn bản ngẫu nhiên để “phẳng hóa“ độ thị tần xuất của văn bản TIN. Ta sẽ xét cụ thể biện pháp này dưới đây



### Chương II

### MẬT MÃ KHỐI VÀ MẬT MÃ KHÓA ĐỐI XỨNG

#### 1. Các điều kiện cần cho mật mã khối an toàn là:

* Kích thước khối phải đủ lớn để chống lại các loại tấn công phá hoại bằng phương pháp thống kê. Tuy nhiên cần lưu ý rằng kích thước khối lớn sẽ làm thời gian trễ lớn.
* Không gian khóa phải đủ lớn (tức là chiều dài khóa phải đủ lớn) để chống lại tìm kiếm vét cạn.Tuy nhiên mặt khác, khóa cần phải đủ ngắn để việc làm khóa, phân phối và lưu trữ được hiệu quả.

Về các nguyên lý thiết kế mật mã khối, người ta đã ghi nhận 2 nguyên tắc cơ sở sau để có bảo mật cao, đó là việc tạo ra confusion (tính hỗn loạn, rắc rối) và diffusion (tính khuếch tán).

*Confusion.* (Hỗn loạn, rắc rối) Sự phụ thuộc của bản mã đối với bản rõ phải thực phức tạp để gây rắc rối, cảm giác hỗn loạn đối với kẻ thù có ý định phân tích tìm qui luật để phá mã. Quan hệ hàm số của mã-tin là phi tuyến (non-linear).

*Diffusion.* (Khuếch tán) Làm khuếch tán những mẫu văn bản mang đặc tính thống kê (gây ra do dư thừa của ngôn ngữ) lẫn vào toàn bộ văn bản. Nhờ đó tạo ra khó khăn cho kẻ thù trong việc dò phá mã trên cơ sở thống kê các mẫu lặp lại cao. Sự thay đổi của một bit trong một khối bản rõ phải dẫn tới sự thay đối hoàn toàn trong khối mã tạo ra.

Một cách đơn giản nhất, *confusion* có thể được thực hiện bằng phép thay thế (substitution) trong khi *diffusion* được tạo ra bằng các phép chuyển đổi chỗ (transposition/permutation) hay hoán vị. Toàn bộ sơ đồ biến đổi mật mã sẽ là một lưới các biến đổi thay thế-hoán vị (substitution-permutation network).

Bên cạnh các nguyên tắc tạo tính bảo mật nói trên, việc thiết kế mật mã khối cũng đề cao các nguyên tắc cài đặt hiệu quả.:

Cài đặt cho phần mềm cần đảm bảo tính mềm dẻo và giá thành thấp.

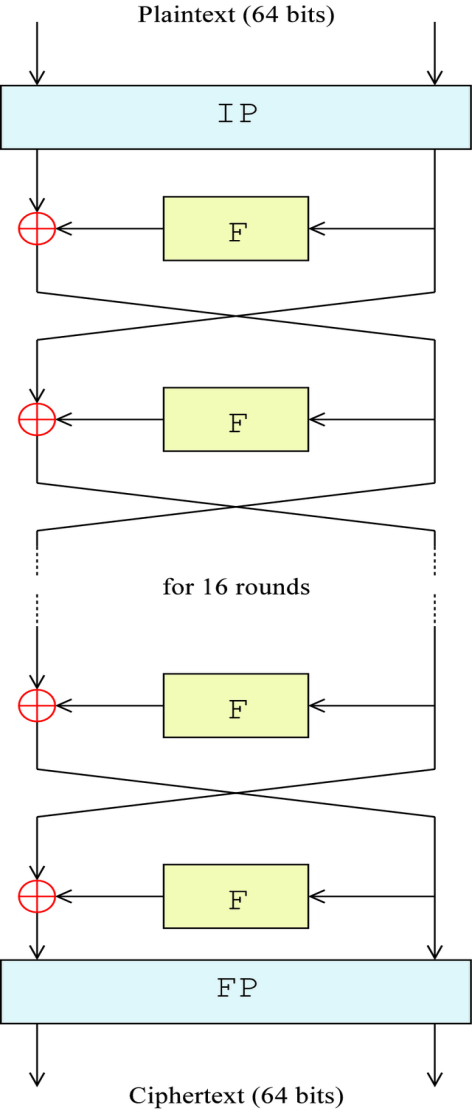
Cài đặt cho phần cứng cần đảm bảo tốc độ cao và tính kinh tế.

Để đáp ứng tốt các nguyên lý thiết kế đã nêu trên, các thuật toán mật mã khối thường được tổ chức như một cấu trúc nhiều vòng lặp.

#### 2. Khái niệm vòng lặp

Một cách phổ biến, các hệ mã khối thường được thiết kế theo cấu trúc nhiều vòng lặp với mỗi vòng lặp lại gọi thực hiện một hàm *f* cơ sở (nhưng với các tham số khác nhau). Theo đó, đầu vào của một vòng lặp là đầu ra của vòng lặp trước và một khóa con phát sinh từ khóa đầy đủ dựa trên một thuật toán lập lịch khóa (key scheduler), hay cũng gọi là thuật toán sinh khóa con.

Giải mã sẽ là một quá trình ngược, trong đó các khóa con sử dụng tại mỗi vòng lặp sẽ được lập lịch để sử dụng theo thứ tự ngược.



**Hình 2.1** *Sơ đồ minh họa một cấu trúc 16 vòng lặp, với đầu vào và ra đều có kích thức 64 bits (Nguồn: Wikipedia). Có hai khối hoán vị đầu và cuối (IP và FP). Hàm F cơ sở chỉ nhận đầu vào 32 bits, nhưng tác động của nó sẽ rộng khắp qua chỉ 2 vòng nhờ sự hoán vị 2 nửa trái và phải.*

Thông thường, hàm cơ sở vòng lặp *f* được thiết kế có một tính chất đặc biệt là tính đối hợp hàm (involution), tức là nó bằng hàm ngược của nó: f = f-1 hay là f(f(x)) = x

Chúng ta sẽ tìm hiểu chi tiết một hệ mã khối điển hình, đó là chuẩn mật mã DES (Data Encryption Standard); chuẩn này ra đời vào năm 1977 và đã thống trị ứng dụng mật mã suốt 2 thập kỷ sau đó. Tuy nhiên chuẩn mật mã này đã trở nên lạc hâu, kém an toàn và được thay thế bởi chuẩn mới AES (Advanced Encryption Standard).

#### 3. Lịch sử của DES

Vào những năm đầu thập kỷ 70, nhu cầu có một chuẩn chung về thuật toán mật mã đã trở nên rõ ràng. Các lý do chính là:

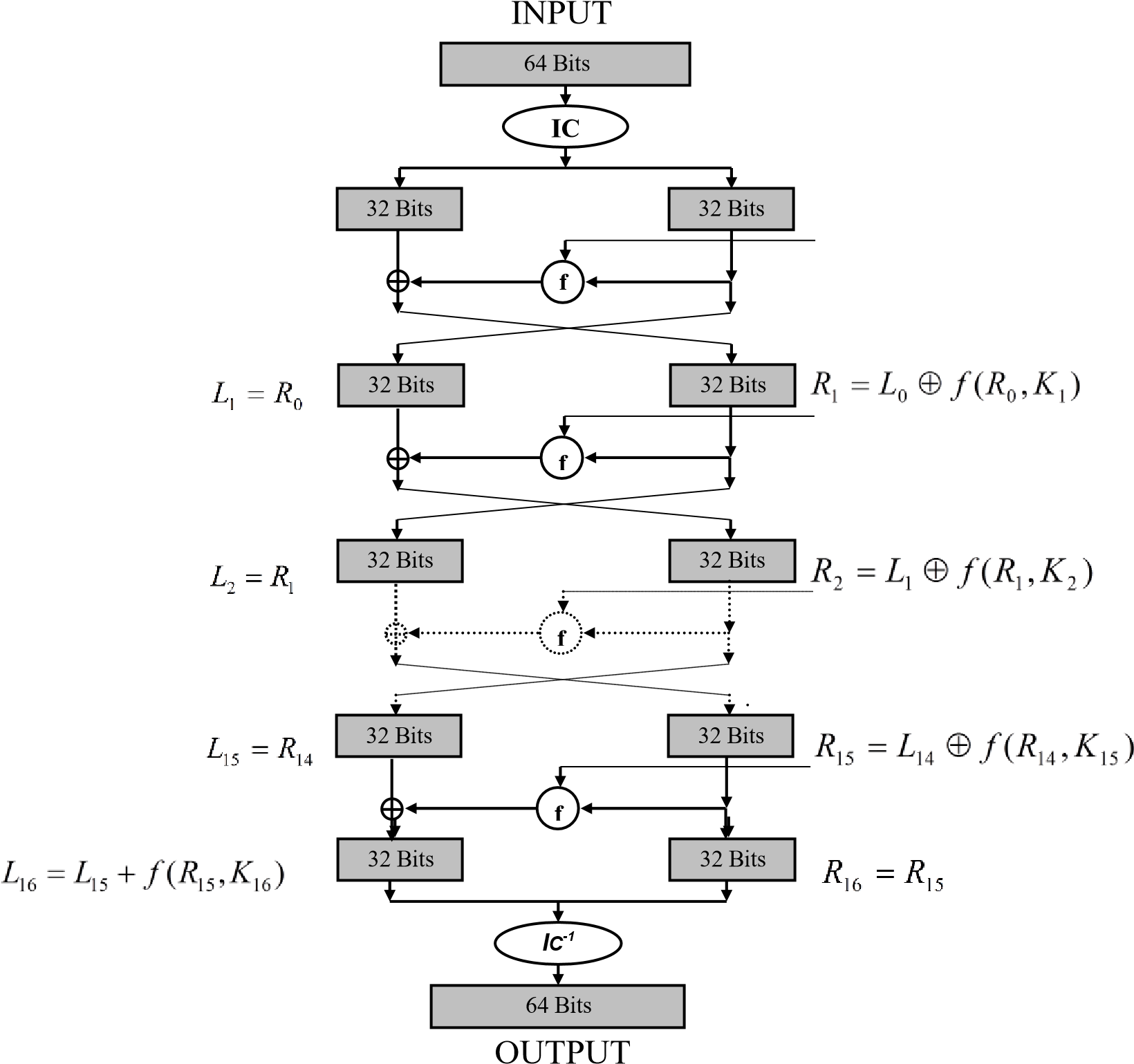
* Sự phát triển của công nghệ thông tin và của nhu cầu an toàn & bảo mật thông tin: sự ra đời của các mạng máy tính tiền thân của Internet đã cho phép khả năng hợp tác và liên lạc số hóa giữa nhiều công ty, tổ chức trong các dự án lớn của chính phủ Mỹ.
* Các thuật toán „cây nhà lá vườn‟ (ad hoc) không thể đảm bảo được tính tin cậy đòi hỏi cao.
* Các thiết bị khác nhau đòi hỏi sự trao đổi thông tin mật mã thống nhất, chuẩn.

Một chuẩn chung cần thiết phải có với các thuộc tính như:

1. Bảo mật ở mức cao
2. Thuật toán được đặc tả và công khai hoàn toàn, tức là tính bảo mật không được phép dựa trên những phần che giấu đặc biệt của thuật toán.
3. Việc cài đặt phải dễ dàng để đem lại tính kinh tế
4. Phải mềm dẻo để áp dụng được cho muôn vàn nhu cầu ứng dụng

#### 4. Các thuật toán và lưu đồ hoạt động của DES

Thuật toán sinh khóa con

 16 vòng lặp của DES cùng gọi thực hiện *f* nhưng với các tham số khóa khác nhau. Tất cả 16 khóa khác nhau này, được gọi là khóa con, cùng sinh ra từ khóa chính của DES bằng một thuật toán sinh khóa con. Trong thuật toán sinh khóa con này (lập lịch khóa), khóa chính K, 64 bit, đi qua 16 bước biến đổi, tại mỗi bước này một khóa con được sinh ra với độ dài 48 bit.

Mỗi vòng lặp của DES thực hiện trên cơ sở công thức sau:

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

trong đó, (Li,Ri) là 2 nửa trái và phải thu được từ biến đổi của vòng lặp thứ *i*. Ta cũng có thể viết lại

(Li,Ri) = T F (Ri-1,Ki))

Trong đó *F* là phép thay thế Li-1 bằng Li-1  *f* (Ri-1,Ki), còn T là phép đổi chỗ hai thành phần L và R. Tức là mỗi biến đổi vòng lặp của DES có thể coi là một tích hàm số của F và T (trừ vòng cuối cùng không có T).

Ta có thể viết lại toàn bộ *thuật toán sinh mã DES* dưới dạng công thức tích hàm số như sau:

DES = (IP)-1F16TF15T ... F2TF1 (IP)

*Thuật toán giải mã*

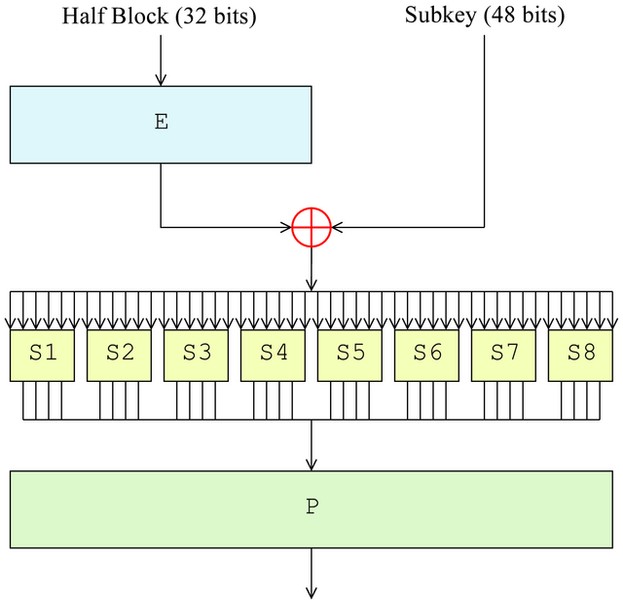
*DES* được xây dựng giống hệt như thuật toán sinh mã nhưng có các khóa con được sử dụng theo thứ tự ngược lại, tức là dùng khóa K16 cho vòng lặp 1, khóa K15 cho vòng lặp 2 ... Vì vậy, thuật toán giải mã có thể được viết lại dưới dạng công thức sau:

DES-1 = (IP)-1F1TF2T ... F15TF16 (IP) -1,

Bây giờ chú ý rằng mỗi hàm T hoặc F đều là các hàm có tính chất đối hợp (f=f hay f(f(x) =x). Do đó nếu ta thực hiện phép tích hàm DES-1DEShay DES DES-1 thì sẽ thu được phép đồng nhất. Điều đó giải thích tại sao thuật toán giải mã lại giống hệt như sinh mã chỉ có khác về thứ từ trong chuỗi khóa con.

*Cấu trúc cụ thể hàm f*

Sơ đồ biến đổi cụ thể của hàm *f* được minh họa trong hình 2.5. Trước hết, 32 bit của thành phần Ri-1 được mở rộng thành 48 bit thông qua biến đổi E (expansion: mở rộng với sự lặp lại một số bit) rồi đem XOR với 48 bit của khóa Ki. Tiếp theo, 48 bit kết quả sẽ được phân thành 8 nhóm 6 bit. Mỗi nhóm này sẽ đi vào một biến đổi đặc biệt gọi là biến đổi S-box (có 8 S-box khác nhau ứng với mỗi nhóm 6 bit) và cho ra kết quả là 8 nhóm 4 bit. Từ đó, 32 bit hợp thành (sau khi qua 8 S-box khác nhau) sẽ được hoán vị lại theo hàm hoán vị P để đưa ra kết quả cuối cùng của hàm *f* (tức nhân của Fi).



#### 5. Cấu trúc của các S-Box

Như ta biết mỗi một trong 8 nhóm 6 bit sẽ đi vào mỗi trong 8 bộ biến đổi S1,S2 ... S8. Mỗi S-box bao gồm 4 bảng biến đổi dòng, thực chất là một biến đổi hoán vị cho 16 tổ hợp của 4 bits. Trong 6 bits đầu vào thì hai bit ngoài cùng (bit 1 và 6) được dùng để chỉ định 1 trong 4 bảng biến đổi dòng này; vì thế chúng được gọi là các bit điều khiển trái và phải (CL và CR). Còn lại 4 bit chính (các bit 2-5) của nhóm 6 bit đầu vào sẽ là tổ hợp 4 bits bị biến đổi.

**Middle 4 bits of input**

**S5**

**0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111**

**00** 0010 1101 0000 1110 1001

**Outer 01** 1110 1011 0010 1100 0100 0111 1101 0001 0101 0000 1111 1010 0011 1001 1000 0110 **bits 10** 0100 0010 0001 1011 1010 1101 0111 1000 1111 1001 1100 0101 0110 0011 0000 1110

1100

0100

0001

0111

1010

1011

0110

1000

0101

0011

1111

**11** 1011 1000 1100 0111 0001 1110 0010 1101 0110 1111 0000 1001 1010 0100 0101 0011

#### 6.Các thuộc tính của S-Box

Các nguyên tắc thiết kế của 8 S-box được đưa vào lớp thông tin mật „Classified information‟ ở Mỹ. Mặc dù vây, NSA đã tiết lộ 3 thuộc tính của S-boxes, những thuộc tính này bảo đảm tính confusion & diffusion của thuật toán.

1. Các bit vào (output bit) luôn phụ thuộc không tuyến tính vào các bít ra (input bit).
2. Sửa đổi ở một bit vào làm thay đổi ít nhất là hai bit ra.
3. Khi một bit vào được giữ cố định và 5 bit con lại cho thay đổi thì S-boxes thể hiện một tính chất được gọi là „phân bố đồng nhất „ (uniform distribution): so sánh số lượng bit số 0 và 1 ở các đầu ra luôn ở mức cân bằng. Tính chất này khiến cho việc áp dụng phân tích theo lý thuyết thông kê để tìm cách phá S-boxes là vô ích.

Rõ ràng, 3 tính chất này đảm bảo tốt confusion & diffusion. Thực tế, sau 8 vòng lặp tất cả các bit ra của DES sẽ chịu ảnh hưởng của tất cả các bit vào và tất cả các bit của khóa. Hơn nữa sự phụ thuộc này là rất phức tạp. Tuy nhiên sau này một số tấn công mới đã được đề xuất và cho thấy 8 vòng lặp này là chưa đủ để bảo mật (điều này cho thấy NSA đã biết trước các dạng tấn công này nên mới qui định số vòng lặp là 16 ngay từ đầu).

Chính cấu tạo của S-box đã gây tranh luận mạnh mẽ trong các thập kỷ 70-90 về khả năng cơ quan NSA (National Security Agency), Mỹ, vẫn còn che dấu các một số đặc tính của S-box hay cài bên trong những cửa bẫy (trapdoor) mà qua đó họ có thể dễ dàng phá giải mã hơn người bình thường (biết các bí mật này có thể giản lược không gian khóa 256 để tìm kiếm vét cạn nhanh hơn). Sự phát hiện sau đó của các tấn công mới, rất mạnh như tấn công vi phân, đã củng cố sự nghi ngờ của giới khoa học.

#### 7. Các điểm yếu của DES

- Tính bù.

Ký hiệu u là phần bù của u (ví dụ 0100101 và 1011010 là bù của nhau) thì DES có tính chất sau:

y = DESz (x)  y  DESz(x)

Cho nên nếu biết MÃ y được mã hóa từ TIN x với khóa z thì ta suy ra y được mã hóa từ TIN x với khóa z . Tính chất này chính là một điểm yếu của DES bởi vì nhờ đó kẻ địch có thể loại trừ một nửa số khóa cần phải thử khi tiến hành phép thử-giải mã theo kiểu tìm kiếm vét cạn không gian khóa.

* Khóa yếu

Các khóa yếu là các khóa mà theo thuật toán sinh khóa con thì tất cả 16 khóa con đều như nhau

Z1 = Z2 = Z3 = ...=Z15 = Z16

điều đó khiến cho phép sinh mã và giải mã đối với các khóa yếu này là giống hệt nhau

DESz = DES-1z

Có tất cả 4 khóa yếu như sau:

1. [00000001 00000001 ... ... 00000001]
2. [11111110 11111110 ... ... 11111110]
3. [11100000 11100000 11100000 11100000

11110001 11110001 11110001 11110001]

1. [00011111 00011111 00011111 00011111 00001110 00001110 00001110 00001110]

Đồng thời có 10 khóa yếu với thuộc tính là tồn tại Z, Z‟ sao cho

DES-1z = DESz‟ hay là DES-1z‟ = DESz

#### 8. Cách tấn công bằng phương pháp vét cạn (hay là brute-force attack)

DES có 256=1017 khóa. Nếu như biết một cặp plaintext-ciphertext thì chúng ta có 17 khả năng này để tìm ra khóa cho kết quả khớp. Giả sử như một phép thể thử tất cả 10

-6 11 thử mất quãng 10 s (trên một máy PC thông thường), thì chúng ta sẽ thử mất 10 s tức là 7300 năm!

Nhưng nhớ rằng đấy mới chỉ là sử dụng các máy tính thông thường, còn có các máy tính được chế tạo theo nguyên lý xử lý song song. Chẳng hạn nếu như làm được

7 con chip mật mã DES chạy song song thì bây giờ mỗi con chip chỉ một thiết bị với 10

10 phép thử. Chip mã DES ngày nay có thể xử lý phải chịu trách nhiệm tính toán với 10

7 5 phép mã DES trong một tới tốc độ là 4.5 x 10 bits/s tức là có thể làm được hơn 10 giây.

Diffie và Hellman (1977) đã ước lượng rằng có thể chế được một máy tính chuyên dụng để vét cạn không gian khóa DES trong1/2 ngày với cái giá cho chiếc máy này là 20 triệu đô la. Cái giá này được tính toán lại và giảm xuống $200,000 vào năm 1987. Vì vậy DES đã bị phê bình ngay từ khi ra đời vì có kích thước khóa quá ngắn!

Hiện nay đã có những thiết kế cụ thể cho loại máy tính chuyên dụng phá khóa này dựa trên kỹ thuật xử lý song song tiên tiến và cho biết một thiết bị kiểu này có giá khoảng $10,000 có thể cho kết quả trong 1 ngày.

#### 9. Các dạng tấn công khác

*Differential Cryptanalysis*. Được công bố lần đầu bởi E. Biham và A. Shamir vào cuối những năm 80 (thế kỷ trước), tuy nhiên thực tế đã được biết đến từ lâu nhưng không công bố bởi IBM và NSA (Cục An ninh Quốc gia Mỹ). Để phá được DES với 49 bản rõ chọn trước (chosen plaintext). Để đầy đủ 16 vòng lặp, tấn công này cần tới 2 có được khối lượng bản rõ này là không thể xảy ra trên thực tế, điều đó cũng cho thấy là DES đã được thiết kế ban đầu để tránh được tấn công này.

*Linear Cryptanalysis*. Tấn công này được phát hiện bởi Matsui vào năm 1994, 43 bản rõ chọn trước và cần 2

#### 10. Các mật mã khối khác (Cho đến năm 1999)

Qua thời gian, có nhiều thuật toán mật mã khối khác nhau được đề xuất bởi cộng đồng khoa học mật mã như

* FEAL (-4, -8, -N, -NX),
* NewDES,
* LOKI91,
* Blowfísh,
* RC2,
* MMB,
* IDEA ...

Tuy nhiên, khá nhiều trong số đó đã bị phá giải hoặc chỉ ra có những điểm yếu nhất định. Điều đó chứng tỏ đề xuất thuật toán mã khối tốt có thể thay thế được DES không phải là đơn giản.

Trong số nói trên IDEA (1990) có thể được xem là thuật toán có độ an toàn cao nhất, cho đến giờ vẫn chưa có một công bố nào nói lên một điểm yếu đáng kể nào của DES, mặc dù kể từ năm 1990 đã có nhiều loại tấn công rất mạnh được sử dụng để thử phá giải. IDEA chính là một trong các thuật toán được dùng trong PGP (Pretty Good Privacy) - một giải pháp bảo mật không thương mại gần như duy nhất cho phép các người dùng trên Internet sử dụng cho các nhu cầu thỏa mãn bí mật riêng như e-mail.

IDEA làm việc với dữ liệu khối 64 bit, nhưng với khóa128 bit nên việc thay thế sử dụng IDEA cho DES là một khó khăn lớn.

#### 11. Mật mã AES là gì?

AES được xây dựng trên nguyên lý thiết kế *lưới giao hoán – thay thế* (substitution-permutation network). Đây là một hệ mã có tốc độ tốt trong cả cài đặt phần mềm cũng như phần cứng. Khác với DES, AES không theo mẫu thiết kế mạng Feistel.

Thay vào đó các thao tác cơ bản được thực hiện trên các khối ma trận dữ liệu 4\*4 (bytes), được gọi là các trạng thái (state). Số vòng lặp của AES là một tham số xác định trên cơ sở kích thước khóa: 10 vòng lặp cho khóa 128bit, 12 cho 192 bit, 14 cho 256bit.

#### 12. Các chế độ sử dụng mã khối

* *Chế độ bảng tra mã điện tử (Electronic code book - ECB)*

Trong chế độ này, các khối được tạo mật mã riêng biệt, độc lập. Do đó, những khối tin giống nhau sẽ được mã hóa thành những khối mã giống nhau. Điều này trở nên nguy hiểm, tạo miếng đất màu mỡ cho kẻ địch vận dụng tấn công replay cũng như thao tác biên tập theo khối. Kẻ thù có thể nghe trộm và tìm cách thu thập các mẫu tin-mã phổ biến, sau đó cắt ghép và trộn lẫn để tạo ra các bản mã giả mã bên nhận không phát hiện được.

Nhược điểm nói trên khiến cho việc truyền tin mật theo chế độ mã này là không có lợi, tuy nhiên chế độ này thường được dùng trong mã hóa thông tin lưu trữ, ví dụ như các cơ sở dữ liệu vì nó cho phép từng đơn vị dữ liệu được mã hóa độc lập và do đó có thể cập nhật thay đổi dễ dàng từng phần mà không động chạm đến các phần khác của cơ sở dữ liệu.

* *Chế độ mã móc xích (Cipher Block Chaining - CBC)*

Trong chế độ này, mỗi khối tin trước khi được mã hóa thì được XOR với khối mã sinh ra từ bước trước đó.

Như vậy các khối mã đều phụ thuộc rất chặt vào nhau theo kiểu “*móc xích*”. Cũng qua đó có thể thấy rằng CBC sẽ tạo ra các khối bản mã khác nhau khi các khối tin đưa vào là giống nhau tức là che giấu được các mẫu tin-mã phổ biến khỏi sự theo dõi của kẻ thù, chặn đứng khả năng phá hoại bằng tấn công replay và biên tập nói trên.

Tại bước đầu tiên, khi chưa có khối mã sinh ra từ bước trước, khối tin đầu sẽ được XOR với một vecto khỏi đầu, chọn ngẫu nhiên, ký hiệu là IV (initial vector).

Tính chất phụ thuộc lẫn nhau của các khối bản mã còn đem lại một ưu thế nữa là ngăn chặn kẻ thù sửa đổi cắt xén mã truyền tin, vì dù chỉ thay đổi 1 bit trên mã cùng làm ảnh hưởng đến toàn bộ thông tin mà được giải mã từ đó, đến mức người nhận có thể phát hiện được dễ dàng do đoạn thông tin giải mã sẽ bị hoàn toàn vô nghĩa.

Tuy nhiên tính chất đó cũng đem lại một mối hại là nếu như mã truyền đi bị sai 1 ít do nhiễu thì giải mã sẽ bị ảnh hưởng lan truyền nhiều, dẫn đến phải phát lại. Ngoài ra chế độ CBC mặc định sự xử lý tuần tự, do đó không thể thực hiện tính toán song song, tức là không thể cải tiến được tốc độ cho hệ máy tính song song.

*Liệu có tồn tại một cơ chế tấn công khác, thông minh hơn loại đã áp dụng cho ECB, để phá mã hoặc lợi dụng CBC*? Lý luận về sự phụ thuộc móc xích mới chỉ cho ta một cảm giác an toàn chứ chưa phải là một chứng minh chặt chẽ. Tuy nhiên tính an toàn trong truyền tin mật của chế CBC đã được chứng minh chặt chẽ bằng phương pháp toán học

* *Chế độ Mã phản hồi k-bit (k-bit Cipher Feedback Mode - CFB)*

Với một số ứng dụng thời gian thực yêu cầu dòng dữ liệu truyền đến phải liên tục hơn là gián đoạn (như là chuỗi ký tự truyền giữa host và terminal phải tạo thành dòng ký tự liên tục). Do đó các chế độ mật mã khối xử lý và truyền theo từng khối một trở nên không thích hợp; các mã stream cipher với đơn vị xử lý là ký tự - khối 8 bit sẽ là thích hợp hơn với dạng ứng dụng này.

Chế độ CFB là một cải tiến cho phép tạo ra khả năng truyền khối nhỏ k-bit (với k tùy ý) trong khi vẫn dùng thuật toán mã khối. Dòng tin đi vào được „múc‟ bằng từng „gầu‟ với dung lượng *k* bit mà *k* là tham số thay đổi được. Thuật toán mật mã khối **E** chạy liên tục như một lò nấu: ở mỗi bước người ta lấy k bit (bên trái nhất) của vector đầu ra từ **E** để bỏ vào „gầu‟ k bit tin, chúng được XOR với nhau. Kết quả k bit vừa được đem truyền đi, vừa được bỏ lại vào đầu vào của thuật toán mã khối: vecto đầu vào được dịch trái k vị trí và k bit phải nhất sẽ được thay thế bởi k bit lấy từ gầu tin.

Như vậy có thể thấy rằng thuật toán mã khối được thực hiện như một hàm sinh các số giả ngẫu nhiên k-bit, các gía trị này lại được XOR với các phần tử k-bit tin lấy vào để tạo ra mã truyền đi.

Qua trình giải mã thì được tiến hành theo nguyên tắc đối xứng. Rõ ràng chế độ này cũng cung cấp các khả năng như của chế độ CBC, thêm vào đó nó cho phép truyền tin với khối ngắn tùy ý, đảm bảo các ứng dụng về truyền-xử lý liên tục.

* *Chế độ mật mã kết quả phản hồi (Output Feedback Mode – OFB)*

Chế độ này cũng khá gần với hai chế độ trên đây, nhưng các phép XOR để tạo ra khối ciphertext là độc lập riêng rẽ, chứ không có sự phụ thuộc (móc xích) như trước.

Các khối plaintext được XOR với các đầu ra – output – của các hàm sinh mã (thuật toán mật mã khối) mà riêng các phần tử output của hàm mã hóa này là vẫn phụ thuộc móc xích (nên được gọi là output feedback).

Tuy nhiên chuỗi móc xích này có thể được thực hiện off-line thông qua tiền xử lý, trước khi thực sự có thông tin văn bản cần gửi đi. Chính vì vậy khả năng thời gian tính toán có thể được rút ngắn nhiều. Ngoài ra, chế độ này cũng cho phép mã khối nhỏ, như stream cipher, giống như với chế độ CFB vậy.

* *Chế độ mật mã con đếm (Counter mode – CTR)*

Đây là chế độ mật mã mới được phát minh không lâu lắm (2000) và được cho là ưu tú nhất. Sơ đồ của nó đơn giản một cách đáng ngạc nhiên! Sự móc xích (feedback) giữa các khối đã được loại trừ hoàn toàn, làm cho CTR có những hiệu năng tính toán cao đáng mong ước

 Có thể xử lý song song dễ dàng vì các khối tính toán hòan tòan độc lập; ngoài ra cũng cho phép tiền xử lý để tính toán trước chuỗi phần tử output của hàm sinh mã (chẳng qua là chuỗi mã hóa của dãy số tự nhiên liên tiếp từ giá trị IV ban đầu).  Không có sự phụ thuộc lẫn nhau nên có thể dùng vào mã hóa dữ liệu lưu trữ giống như với ECB: cho phép truy nhập ngẫu nhiên (random access) thay vì truy nhập tuần tự như với CBC chẳng hạn.

Mặc dù có sơn đồ tính toán rất đơn giản, tính an toàn của chế độ này đã được chứng minh đầy đủ bằng công cụ toán học hình thức, trên cơ sở thông qua so sánh với mật mã one-time-pad (đạt bí mật tuyệt đối.