ĐẠI HỌC QUỐC GIA HÀ NỘI

KHOA CÔNG NGHỆ

Phan Đình Diệu

Lý thuyết mật mã
&
AN TOÀN THÔNG TIN

NXB ĐẠI HỌC QUỐC GIA HÀ NỘI - 2002

Lý thuyết mật mã

&

An toàn thông tin

Lý thuyết mật mã
&
An toàn thông tin

Phan Đình Diệu

Đại học Quốc gia Hà Nội

Khoa Công nghệ- ĐHQG Hà nội

NỘI DUNG

Lời mở đầu

Từ khi con người có nhu cầu trao đổi thông tin, thư từ cho nhau thì nhu cầu giữ bí mật và bảo vệ tính riêng tư của những thông tin, thư từ được trao đổi đó cũng nẩy sinh. Hình thức thông tin được trao đổi phổ biến và sớm nhất là dưới dạng các văn bản, để giữ bí mật của thông tin người ta đã sớm nghĩ đến cách che dấu nội dung các văn bản bằng cách biến dạng các văn bản đó để người ngoài không đọc hiểu được, đồng thời có cách khôi phục lại nguyên dạng ban đầu để người trong cuộc vẫn đọc hiểu được; theo cách gọi ngày nay thì dạng biến đổi của văn bản được gọi là *mật mã* của văn bản, cách lập mật mã cho một văn bản được gọi là *phép lập mật mã*, còn cách khôi phục lại nguyên dạng ban đầu của văn bản từ bản mật mã được gọi là *phép giải mã*. Phép lập mật mã và phép giải mã được thực hiện nhờ một chìa khoá riêng nào đó mà chỉ những người trong cuộc được biết, sau đây ta sẽ gọi là *khoá mật mã*. Người ngoài cuộc không được biết khoá mật mã, nên dù có "ăn cắp" được bản mật mã trên đường truyền tin, về nguyên tắc cũng không thể giải mã để hiểu được nội dung của văn bản truyền đi.

Hiển nhiên, tiêu chuẩn của một bản mật mã là tạo được tính bí mật cho văn bản; vì vậy khái niệm *bí mật* là khái niệm cốt lõi nhất đối với một lý thuyết về mật mã. Có thể có một định nghĩa khoa học cho khái niệm *bí mật* hay không? Đã có nhiều cách tiếp cận để tìm hiểu nội dung của khái niệm bí mật, nhưng một định nghĩa khoa học, hay hơn nữa, một định nghĩa toán học cho khái niệm đó thì chưa có. Một cách tiếp cận khá phổ biến là gắn khái niệm bí mật với khái niệm "ngẫu nhiên", nếu một văn bản rõ có một nội dung xác định thì điều ta mong muốn là bản mật mã của nó phải là một bản gồm các ký tự được sắp xếp hỗn độn, có vẻ như ngẫu nhiên khiến

người ngoài nhìn vào không thể xác định được nội dung của văn bản gốc. Tuy nhiên, nếu "bí mật" là khái niệm chưa định nghĩa được, thì khái niệm "ngẫu nhiên", hay cụ thể hơn, khái niệm "dãy bit ngẫu nhiên", cũng khó định nghĩa như vậy, ta chưa qui định được một tiêu chuẩn toán học để xác định một dãy bit có là "ngẫu nhiên" hay không, mà chỉ mới tìm hiểu được một số thuộc tính gần với "ngẫu nhiên", dùng làm căn cứ để tạm xác định một dãy bit có là "giả ngẫu nhiên" theo nghĩa có các thuộc tính đó hay không mà thôi.

Từ mấy thập niên gần đây, bước vào kỷ nguyên máy tính, cũng như đối với nhiều lĩnh vực khác, lĩnh vực mật mã cũng đã có những chuyển biến to lớn từ giai đoạn mật mã truyền thống sang giai đoạn *mật mã máy tính;* máy tính điện tử được sử dụng ngày càng phổ biến trong việc lập mật mã, giải mật mã, và những chuyển biến đó đã kích thích việc nghiên cứu các giải pháp mật mã, biến việc nghiên cứu mật mã thành một khoa học có đối tượng ngày càng rộng lớn và được sử dụng có hiệu quả trong nhiều phạm vi hoạt động của cuộc sống. Vì các nghiệp vụ chủ yếu của mật mã được thực hiện bằng máy tính, nên các khái niệm bí mật, ngẫu nhiên cũng dần được "máy tính hoá", và với sự ra đời của *Lý thuyết về độ phức tạp tính toán* vào giữa những năm 1960, các khái niệm đó tìm được một nội dung chung có thể được nghiên cứu một cách toán học là tính *phức tạp*. Bây giờ ta có thể nói, một bản mật mã đối với anh là *bí mật*, nếu từ bản mật mã đó để tìm ra bản rõ anh phải thực hiện một tiến trình tính toán mà độ phức tạp của nó vượt quá mọi năng lực tính toán (kể cả mọi máy tính) của anh; một dãy bit có thể xem là *ngẫu nhiên* , nếu dựa vào một đoạn bit đã biết để tìm một bit tiếp theo của dãy anh cũng phải thực hiện một tiến trình tính toán có độ phức tạp cực lớn tương tự như nói trên.

Việc chuyển sang giai đoạn mật mã máy tính trước hết đã có tác dụng phát triển và hiện đại hoá nhiều hệ thống mật mã theo kiểu truyền thống, làm cho các hệ thống đó có các cấu trúc tinh tế hơn, đòi hỏi lập mật mã và giải mã phức tạp hơn, do đó hiệu quả giữ bí mật của các giải pháp mật mã được nâng cao hơn trước rất nhiều. Tuy nhiên, một bước chuyển có tính chất cách mạng mà mật mã máy tính mang lại là việc phát minh ra các hệ mật mã *có khoá công khai*, bắt đầu từ cuối những năm 1970, cơ sở lý thuyết của các phát

minh đó là sự tồn tại của các *hàm một phía* (one-way function), tức là những hàm số số học $y = f(x)$ mà việc tính theo phía thuận từ x tính y là tương đối dễ, nhưng việc tính theo phía ngược từ y tìm lại x ($x = f^{-1}(y)$) là cực kỳ phức tạp. Các hệ mật mã có khoá công khai đã làm thay đổi về bản chất việc tổ chức các hệ truyền thông bảo mật, làm dễ dàng cho việc bảo mật trên các hệ truyền thông công cộng, và do tính chất đặc biệt đó chúng đã là cơ sở cho việc phát triển nhiều giao thức an toàn thông tin khác khi sử dụng mạng truyền thông công cộng, chẳng hạn các loại giao thức về xác nhận nguồn tin và định danh người gửi, chữ ký điện tử, các giao thức xác nhận không để lộ thông tin gì khác ngoài việc xác nhận, các giao thức trao đổi khoá trong tổ chức truyền tin bảo mật và trong xác nhận, v.v..., và gần đây trong việc phát triển nhiều giao thức đặc thù khác trong các giao dịch ngân hàng và thương mại điện tử, phát hành và mua bán bằng tiền điện tử,... Cũng cần nói thêm là lý thuyết mật mã hiện đại, tức là mật mã máy tính trên cơ sở lý thuyết về độ phức tạp tính toán tuy có nhiều ứng dụng đặc sắc và có triển vọng to lớn, nhưng cũng mới đang trong giai đoạn phát triển bước đầu, còn phải khắc phục nhiều khó khăn và tìm kiếm thêm nhiều cơ sở vững chắc mới để tiếp tục hoàn thiện và phát triển. Chẳng hạn, như trên đã nói, một cơ sở quan trọng của lý thuyết mật mã hiện đại là sự tồn tại của các hàm một phía, nhưng ngay có thật tồn tại các hàm một phía hay không cũng còn là một bài toán chưa có câu trả lời! Ta chỉ mới *đang có* một số hàm một phía *theo sự hiểu biết của con người hiện nay*, nhưng chưa chứng minh được có một hàm cụ thể nào đó *chắc chắn* là hàm một phía! Tuy nhiên, nếu theo quan điểm khoa học hiện đại, ta không xem mục đích khoa học là đi tìm những chân lý chắc chắn tuyệt đối, mà là đi tìm những cách giải quyết vấn đề (problem solving) gặp trong thực tiễn, thì ta vẫn có thể tin vào những giải pháp "tương đối" rất có hiệu quả mà lý thuyết hiện đại về mật mã đang cống hiến cho con người hiện nay.

Tập giáo trình *Lý thuyết mật mã và an toàn thông tin* này được soạn để phục vụ cho việc học tập của sinh viên các lớp theo chương trình đại học hoặc cao học thuộc ngành Công nghệ thông tin của Đại học Quốc gia Hà nội. Trong khoảng mươi năm gần đây, trên thế giới đã xuất hiện nhiều sách và tài liệu có tính chất giáo khoa

hoặc tham khảo về lý thuyết mật mã hiện đại và ứng dụng. Người viết tập giáo trình này chỉ có cố gắng lựa chọn và sắp xếp một số nội dung mà mình nghĩ là cần thiết và thích hợp nhất để trong một phạm vi hạn chế về thời gian (và không gian) trình bày và giới thiệu được cho người học một cách tương đối hệ thống những kiến thức cơ bản về lý thuyết mật mã hiện đại, bao gồm cả một số kiến thức toán học cần thiết. Giáo trình này đã được giảng dạy cho sinh viên các khoá cao học về Công nghệ thông tin thuộc Đại học Bách khoa Hà nội và khoa Công nghệ Đại học Quốc gia Hà nội từ năm 1997 đến 2004. Người viết chân thành cảm ơn các bạn đồng nghiệp và người đọc chỉ cho những chỗ thiếu sót để có thể kịp thời sửa chữa cho những lần in sau, nếu có.

Tháng 12 năm 2002

Phan Đình Diệu

CHƯƠNG I

Giới thiệu chung về mật mã

1.1. Sơ lược lịch sử về mật mã.

Như đã giới thiệu trong *Lời mở đầu*, nhu cầu sử dụng mật mã đã xuất hiện từ rất sớm, khi con người biết trao đổi và truyền đưa thông tin cho nhau, đặc biệt khi các thông tin đó đã được thể hiện dưới hình thức ngôn ngữ, thư từ. Lịch sử cho ta biết, các hình thức mật mã sơ khai đã được tìm thấy từ khoảng bốn nghìn năm trước trong nền văn mịnh Ai cập cổ đại. Trải qua hàng nghìn năm lịch sử, mật mã đã được sử dụng rộng rãi trên khắp thế giới từ Đông sang Tây để giữ bí mật cho việc giao lưu thông tin trong nhiều lĩnh vực hoạt động giữa con người và các quốc gia, đặc biệt trong các lĩnh vực quân sự, chính trị, ngoại giao. Mật mã trước hết là một loại hoạt động thực tiễn, nội dung chính của nó là để giữ bí mật thông tin (chẳng hạn dưới dạng một văn bản) từ một người gửi A đến một người nhận B, A phải tạo cho văn bản đó một bản mã mật tương ứng, và thay vì gửi văn bản rõ thì A chỉ gửi cho B bản mã mật, B nhận được bản mã mật và sẽ có cách từ đó khôi phục lại văn bản rõ để hiểu được thông tin mà A muốn gửi cho mình. Vì bản gửi đi thường được chuyển qua các con đường công khai nên người ngoài có thể "lấy trộm" được, nhưng do đó là bản mật mã nên không đọc hiểu được, còn A có thể tạo ra bản mã mật và B có thể giải bản mã mật thành bản rõ để hiểu được là do giữa hai người đã có một thỏa thuận về một *chìa khóa chung*, chỉ với chìa khóa chung này thì A mới tạo được bản mã mật từ bản rõ, và B mới từ bản mã mật khôi phục lại được bản rõ. Sau này ta sẽ gọi đơn giản chìa khóa chung đó là *khóa mật mã.* Tất nhiên để thực hiện được một phép mật mã, ta

còn cần có một thuật toán biến bản rõ, cùng với khóa mật mã, thành bản mã mật, và một thuật toán ngược lại, biến bản mã mật, cùng với khóa mật mã, thành bản rõ. Các thuật toán đó được gọi tương ứng là thuật toán *lập mật mã* và thuật toán *giải mật mã*. Các thuật toán này thường không nhất thiết phải giữ bí mật, mà cái cần được giữ tuyệt mật luôn luôn là khóa mật mã. Trong thực tiễn, đã có hoạt động bảo mật thì cũng có hoạt động ngược lại là khám phá bí mật từ các bản mã mật "lấy trộm" được, ta thường gọi hoạt động này là *mã thám*, hoạt động này quan trọng không kém gì hoạt động bảo mật! Vì các thuật toán lập mật mã và giải mật mã không nhất thiết là bí mật, nên mã thám thường được tập trung vào việc tìm khóa mật mã, do đó cũng có người gọi công việc đó là *phá khóa.*

Suốt mấy nghìn năm lịch sử, các thông báo, thư từ được truyền đưa và trao đổi với nhau thường là các văn bản, tức là có dạng các dãy ký tự trong một ngôn ngữ nào đó; vì vậy, các thuật toán lập mật mã thường cũng đơn giản là thuật toán xáo trộn, thay đổi các ký tự được xác định bởi các phép chuyển dịch, thay thế hay hoán vị các ký tự trong bảng ký tự của ngôn ngữ tương ứng; khóa mật mã là thông tin dùng để thực hiện phép lập mật mã và giải mật mã cụ thể, thí dụ như số vị trí đối với phép chuyển dịch, bảng xác định các cặp ký tự tương ứng đối với phép thay thế hay hoán vị,... Mật mã chưa phải là một khoa học, do đó chưa có nhiều kiến thức sách vở để lại, tuy nhiên hoạt động bảo mật và thám mã trong lịch sử các cuộc đấu tranh chính trị, ngoại giao và quân sự thì hết sức phong phú, và mật mã đã có nhiều tác động rất quan trọng đưa đến những kết quả lắm khi có ý nghĩa quyết định trong các cuộc đấu tranh đó. Do trong một thời gian dài, bản thân hoạt động mật mã cũng được xem là một bí mật, nên các tài liệu kỹ thuật về mật mã được phổ biến đến nay thường chỉ ghi lại các kiến thức kinh nghiệm, thỉnh thoảng mới có một vài "phát minh" như các hệ mật mã Vigenère vào thế kỷ 16 hoặc hệ mật mã Hill ra đời năm 1929 là các hệ mã thực hiện phép chuyển dịch (đối với mã Vigenère) hay phép thay thế (mã Hill) đồng thời trên một nhóm ký tự chứ không phải trên từng ký tự riêng rẽ. Vấn đề thám mã, ngược lại, khi thành công thường đưa đến những cống hiến nổi trội và ấn tượng trong những

tình huống gay cấn của các cuộc đấu tranh, và cũng thường đòi hỏi nhiều tài năng phát hiện với những kinh nghiệm và suy luận tinh tế hơn, nên để lại nhiều chuyện hấp dẫn hơn. Nhiều câu chuyện kỳ thú của lịch sử thám mã đã được thuật lại trong quyển sách nổi tiếng của David Kahn *The Codebreakers . The Story of Secret Writing* , xuất bản năm 1967 (sách đã được dịch ra nhiều thứ tiếng, có bản dịch tiếng Việt *Những người mã thám*, 3 tập, xuất bản tại Hà nội năm 1987).

Bước sang thế kỷ 20, với những tiến bộ liên tục của kỹ thuật tính toán và truyền thông, ngành mật mã cũng đã có những tiến bộ to lớn. Vào những thập niên đầu của thế kỷ, sự phát triển của các kỹ thuật biểu diễn, truyền và xử lý tín hiệu đã có tác động giúp cho các hoạt động lập và giải mật mã từ thủ công chuyển sang cơ giới hóa rồi điện tử hóa. Các văn bản, các bản mật mã trước đây được viết bằng ngôn ngữ thông thường nay được chuyển bằng kỹ thuật số thành các dãy tín hiệu nhị phân, tức các dãy bit, và các phép biến đổi trên các dãy ký tự được chuyển thành các phép biến đổi trên các dãy bit, hay các dãy số, việc thực hiện các phép lập mã, giải mã trở thành việc thực hiện các hàm số số học. Toán học và kỹ thuật tính toán bắt đầu trở thành công cụ cho việc phát triển khoa học về mật mã. Khái niệm trung tâm của khoa học mật mã là khái niệm *bí mật*. Đó là một khái niệm phổ biến trong đời sống, nhưng liệu có thể cho nó một nội dung có thể định nghĩa được một cách toán học không? Như đã lược qua trong *Lời mở đầu,* khái niệm *bí mật* thoạt đầu được gắn với khái niệm *ngẫu nhiên*, rồi về sau trong những thập niên gần đây, với khái niệm *phức tạp*, cụ thể hơn là khái niệm *độ phức tạp tính toán*. Việc sử dụng lý thuyết xác suất và ngẫu nhiên làm cơ sở để nghiên cứu mật mã đã giúp C.Shannon đưa ra khái niệm *bí mật hoàn toàn* của một hệ mật mã từ năm 1948, khởi đầu cho một lý thuyết xác suất về mật mã. Trong thực tiễn làm mật mã, các *dãy bit ngẫu nhiên* được dùng để trộn với bản rõ (dưới dạng một dãy bit xác định) thành ra bản mật mã. Làm thế nào để tạo ra các dãy bit ngẫu nhiên? Có thể tạo ra bằng phương pháp vật lý đơn giản như sau: ta tung đồng xu lên, nếu đồng xu rơi xuống ở mặt sấp thì ta ghi bit 0, ở mặt ngửa thì ta ghi bit 1; tung n lần ta sẽ được một dãy n

bit, dãy bit thu được như vậy có thể được xem là dãy bit ngẫu nhiên. Nhưng tạo ra theo cách như vậy thì khó có thể sử dụng một cách phổ biến, vì không thể tìm ra *qui luật* để theo đó mà sinh ra dãy bit ngẫu nhiên được. Ở đây ta gặp một khó khăn có tính bản chất: nếu có qui luật thì đã không còn là ngẫu nhiên nữa rồi! Như vậy, nếu ta muốn tìm theo qui luật, thì không bao giờ có thể tìm ra các dãy bit ngẫu nhiên, mà cùng lắm cũng chỉ có thể được các dãy bit gần ngẫu nhiên, hay *giả ngẫu nhiên*, mà thôi. Từ nhiều chục năm nay, người ta đã nghiên cứu đề xuất nhiều thuật toán toán học để sinh ra các dãy bit giả ngẫu nhiên, và cũng đã đưa ra nhiều thuộc tính để đánh giá một dãy bit giả ngẫu nhiên có đáng được xem là "gần" ngẫu nhiên hay không. Một vài thuộc tính chủ yếu mà người ta đã đề xuất là: cho một dãy bit $X = (x_1,x_2,.....,x_n,...)$; dãy đó được xem là giả ngẫu nhiên "tốt" nếu xác suất xuất hiện bit 0 hay bit 1 trong toàn dãy đó cũng như trong mọi dãy con bất kỳ của nó đều bằng 1/2; hoặc một tiêu chuẩn khác: nếu mọi chương trình sinh ra được đoạn đầu n bit của dãy đều phải có độ phức tạp (hay độ dài) cỡ n ký tự ! Về sau này, khi lý thuyết về độ phức tạp tính toán đã được phát triển thì tiêu chuẩn về ngẫu nhiên cũng được qui về tiêu chuẩn phức tạp tính toán, cụ thể một dãy bit X được xem là giả ngẫu nhiên "tốt" nếu mọi thuật toán tìm được bit thứ n (x_n) khi biết các bit trước đó ($x_1,,...,x_{n-1}$) với xác suất đúng > 1/2 đều phải có độ phức tạp tính toán thuộc lớp *NP*-khó!

Lý thuyết về độ phức tạp tính toán ra đời từ giữa những năm 1960 đã cho ta một cách thích hợp để qui yêu cầu bí mật hoặc ngẫu nhiên về một yêu cầu có thể định nghĩa được là yêu cầu về *độ phức tạp tính toán.* Bây giờ ta có thể nói: một giải pháp mật mã là bảo đảm bí mật, nếu mọi thuật toán thám mã, nếu có, đều phải được thực hiện với độ phức tạp tính toán cực lớn! Cực lớn là bao nhiêu? Là vượt quá giới hạn khả năng tính toán (bao gồm cả máy tính) mà người thám mã có thể có. Về lý thuyết, có thể xem đó là những độ phức tạp tính toán với tốc độ tăng vượt quá hàm mũ, hoặc thuộc loại *NP*-khó. Tuy nhiên, lý thuyết độ phức tạp tính toán không chỉ cống hiến cho ta một khái niệm để giúp chính xác hóa tiêu chuẩn bí mật của các giải pháp mật mã, mà còn mở ra một giai đoạn mới của ngành mật mã, biến ngành mật mã thành một khoa học có nội dung

lý luận phong phú và có những ứng dụng thực tiễn quan trọng trong nhiều lĩnh vực của đời sống hiện đại. Bước ngoặt có tính cách mạng trong lịch sử khoa học mật mã hiện đại xẩy ra vào năm 1976 khi hai tác giả Diffie và Hellman đưa ra khái niệm về *mật mã khóa công khai* và một phương pháp trao đổi c*ông khai* để tạo ra một khóa bí mật chung mà tính an toàn được bảo đảm bởi độ khó của một bài toán toán học cụ thể (là bài toán tính "lôgarit rời rạc"). Hai năm sau, năm 1978, Rivest, Shamir và Adleman tìm ra một hệ mật mã khóa công khai và một sơ đồ *chữ ký điện tử* hoàn toàn có thể ứng dụng trong thực tiễn, tính bảo mật và an toàn của chúng được bảo đảm bằng độ phức tạp của một bài toán số học nổi tiếng là bài toán phân tích số nguyên thành các thừa số nguyên tố. Sau phát minh ra hệ mật mã đó (mà nay ta thường gọi là hệ RSA), việc nghiên cứu để phát minh ra các hệ mật mã khóa công khai khác, và ứng dụng các hệ mật mã khóa công khai vào các bài toán khác nhau của an toàn thông tin đã được tiến hành rộng rãi, lý thuyết mật mã và an toàn thông tin trở thành một lĩnh vực khoa học được phát triển nhanh trong vài ba thập niên cuối của thế kỷ 20, lôi cuốn theo sự phát triển của một số bộ môn của toán học và tin học. Trong các chương về sau của tập giáo trình này ta sẽ lần lượt làm quen với một số thành quả chủ yếu của lý thuyết đó.

1.2. Các hệ thống mật mã.

1.2.1. Sơ đồ hệ thống mật mã.

Mật mã được sử dụng để bảo vệ tính bí mật của thông tin khi thông tin được truyền trên các kênh truyền thông công cộng như các kênh bưu chính, điện thoại, mạng truyền thông máy tính, mạng Internet, v.v... Giả thử một người gửi A muốn gửi đến một người nhận B một văn bản (chẳng hạn, một bức thư) p, để bảo mật A lập cho p một bản mật mã c, và thay cho việc gửi p, A gửi cho B bản mật mã c, B nhận được c và "giải mã" c để lại được văn bản p như A định gửi. Để A biến p thành c và B biến ngược lại c thành p, A và B phải thỏa thuận trước với nhau các thuật toán lập mã và giải mã, và đặc biệt một *khóa mật mã chung K* để thực hiện các thuật toán đó. Người ngoài, không biết các thông tin đó (đặc biệt, không biết khóa

K), cho dù có lấy trộm được *c* trên kênh truyền thông công cộng, cũng không thể tìm được văn bản *p* mà hai người A, B muốn gửi cho nhau. Sau đây ta sẽ cho một định nghĩa hình thức về sơ đồ mật mã và cách thức thực hiện để lập mật mã và giải mật mã.

Định nghĩa 1.2.1. *Một sơ đồ hệ thống mật mã là một bộ năm*

$$\mathcal{S} = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D}) \qquad (1)$$

thỏa mãn các điều kiện sau đây:

$\mathcal{P}$ *là một tập hữu hạn các ký tự bản rõ,*

$\mathcal{C}$ *là một tập hữu hạn các ký tự bản mã,*

$\mathcal{K}$ *là một tập hữu hạn các khóa,*

$\mathcal{E}$ *là một ánh xạ từ* $\mathcal{K}x\mathcal{P}$ *vào* $\mathcal{C}$, *được gọi là phép lập mật mã; và* $\mathcal{D}$ *là một ánh xạ từ* $\mathcal{K}x\mathcal{C}$ *vào* $\mathcal{P}$, *được gọi là phép giải mã. Với mỗi* $K \in \mathcal{K}$, *ta định nghĩa* $e_K : \mathcal{P} \to \mathcal{C}$, $d_K : \mathcal{C} \to \mathcal{P}$ *là hai hàm cho bởi*:

$$\forall x \in \mathcal{P} : e_K(x) = \mathcal{E}(K,x) ; \forall y \in \mathcal{C} : d_K(y) = \mathcal{D}(K,y).$$

e_K *và* d_K *được gọi lần lượt là hàm lập mã và hàm giải mã ứng với khóa mật mã K. Các hàm đó phải thỏa mãn hệ thức*:

$$\forall x \in \mathcal{P} : d_K(e_K(x)) = x.$$

Về sau, để thuận tiện ta sẽ gọi một danh sách (1) thoả mãn các tính chất kể trên là một *sơ đồ hệ thống mật mã* , còn khi đã chọn cố định một khoá *K*, thì danh sách ($\mathcal{P}$, $\mathcal{C}$, e_K , d_K) là một *hệ mật mã* thuộc sơ đồ đó.

Trong định nghĩa này, phép lập mật mã (giải mã) được định nghĩa cho từng ký tự bản rõ (bản mã). Trong thực tế, bản rõ của một thông báo thường là một dãy ký tự bản rõ, tức là phần tử của tập $\mathcal{P}^*$, và bản mật mã cũng là một dãy các ký tự bản mã, tức là phần tử của tập $\mathcal{C}^*$, việc mở rộng các hàm e_K và d_K lên các miền tương ứng $\mathcal{P}^*$ và $\mathcal{C}^*$ để được các thuật toán lập mật mã và giải mã dùng trong thực tế sẽ được trình bày trong tiết sau. Các tập ký tự bản rõ và bản mã thường dùng là các tập ký tự của ngôn ngữ thông thường như tiếng Việt, tiếng Anh (ta ký hiệu tập ký tự tiếng Anh là $\boldsymbol{A}$ tức $\boldsymbol{A}$ = {*a,b,c,...,x,y,z* } gồm 26 ký tự; tập ký tự nhị phân $\boldsymbol{B}$ chỉ gồm hai ký tự

0 và 1; tập các số nguyên không âm bé hơn một số n nào đó (ta ký hiệu tập này là $\boldsymbol{Z_n}$ **tức** $\boldsymbol{Z_n = \{0,1,2,....,n-1\}}$**).** Chú ý rằng có thể xem $\boldsymbol{B = Z_2}$. Để thuận tiện, ta cũng thường đồng nhất tập ký tự tiếng Anh $\boldsymbol{A}$ với tập gồm 26 số nguyên không âm đầu tiên $\boldsymbol{Z_{26}}$ = {0,1,2,...., 24,25} với sự tương ứng sau đây:

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

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25.

Đôi khi ta cũng dùng với tư cách tập ký tự bản rõ hay bản mã là các tập tích của các tập nói trên, đặc biệt là các tập $\boldsymbol{A}^m$, $\boldsymbol{B}^m$, $\boldsymbol{Z}_n^m$.

1.2.2. Mã theo khối và mã theo dòng.

Như nói ở trên, bản rõ của thông báo mà ta muốn gửi đi thường là một dãy ký tự, trong khi theo định nghĩa của sơ đồ mật mã, hàm lập mật mã và hàm giải mã được định nghĩa cho từng ký tự. Từ các định nghĩa của hàm lập mật mã và hàm giải mã, ta mở rộng thành thuật toán lập mã (và giải mã) xác định cho mọi bản rõ (bản mã) như sau:

Theo cách *mã theo khối* (block cipher), trước hết ta xác định một độ dài khối (chẳng hạn là k), tiếp đó mở rộng không gian khóa từ $\mathcal{K}$ thành $\mathcal{K}^k$, và với mỗi $K = K_1...K_k \in \mathcal{K}^k$, ta mở rộng e_K và d_K thành các thuật toán $e_K : \mathcal{P}^k \to \mathcal{C}^k$ và $d_K : \mathcal{C}^k \to \mathcal{P}^k$ như sau: với mọi $x_1...x_k \in \mathcal{P}^k$ và $y_1...y_k \in \mathcal{C}^k$ ta có

$$e_K(x_1....x_k) = e_{K_1}(x_1)....e_{K_k}(x_k); \quad d_K(y_1....y_k) = d_{K_1}(y_1)....d_{K_k}(y_k).$$

Giả thử bản rõ mà ta muốn lập mật mã cho nó là dãy ký tự $X \in \mathcal{P}^*$. Ta cắt X thành từng khối, mỗi khối có độ dài k, khối cuối cùng có thể có độ dài $< k$, ta luôn có thể giả thiết là có thể bổ sung vào phần cuối của khối một số ký tự qui ước nào đó để nó cũng có độ dài k. Do đó ta có thể giả thiết $X = X_1....X_m$, trong đó mỗi $X_1,...,X_m$ là một khối có độ dài k. Và ta định nghĩa bản mật mã của X là:

$$e_K(X) = e_K(X_1....X_m) = e_K(X_1)....e_K(X_m).$$

Đặt $Y = e_K(X_1)....e_K(X_m)$, ta có thể viết $Y = Y_1....Y_m$ với $Y_i = e_K(X_i)$, và do đó có

$$d_K(Y) = d_K(Y_1)....d_K(Y_m) = X_1....X_m = X.$$

Cách mã theo khối đơn giản và thông dụng nhất là khi ta chọn độ dài khối k=1. Khi đó với mọi bản rõ $X = x_1...x_m \in \mathcal{P}^*$ ta có

$$e_K(X) = e_K(x_1....x_m) = e_K(x_1)....e_K(x_m).$$

Với cách *mã theo dòng* (stream cipher), trước hết ta phải xác định một *dòng khóa*, tức là một phần tử $K = K_1...K_m \in \mathcal{K}^*$, với dòng khóa đó ta xác định với mọi bản rõ $X = x_1...x_m \in \mathcal{P}^*$ bản mã tương ứng là

$$e_K(X) = e_K(x_1...x_m) = e_{K_1}(x_1)...e_{K_m}(x_m).$$

Giải mã $Y = e_K(X)$ ta được

$$d_K(Y) = d_{K_1}(e_{K_1}(x_1))....d_{K_m}(e_{K_m}(x_m)) = x_1....x_m = X .$$

Để sử dụng cách lập mật mã theo dòng, ngoài sơ đồ mật mã gốc ta còn phải có một dòng khóa, tức là một dãy có độ dài tùy ý các ký tự khóa. Đó thường là các dãy các ký tự khóa được sinh ra bởi một bộ "tạo dãy ngẫu nhiên" nào đó xuất phát từ một "mầm" chọn trước. Trong các ứng dụng thực tế, người ta thường dùng cách mã theo dòng có sơ đồ mật mã gốc là sơ đồ Vernam với

$$\mathcal{P} = \mathcal{C} = \mathcal{K} = \{0,1\}$$

và các hàm lập mã và giải mã được xác định bởi

$$e_K(x) = x + K \bmod 2,\ \ d_K(y) = y + K \bmod 2\ (K = 0 \text{ hoặc } 1);$$

dòng khóa là dãy bit ngẫu nhiên được sinh ra bởi một bộ tạo dãy bit ngẫu nhiên nào đó.

1.3. Mật mã khóa đối xứng và mật mã có khóa công khai.

Theo định nghĩa 1.2.1 về sơ đồ mật mã, cứ mỗi lần truyền tin bảo mật, cả người gửi A và người nhận B phải cùng thỏa thuận trước với nhau một khóa chung K, sau đó người gửi dùng e_K để lập mật mã cho thông báo gửi đi, và người nhận dùng d_K để giải mã bản mật mã nhận được. Người gửi và người nhận cùng có một khóa

chung K, được giữ như bí mật riêng của hai người, dùng cả cho lập mật mã và giải mã, ta gọi những hệ mật mã với cách sử dụng đó là *mật mã khóa đối xứng*, đôi khi cũng gọi là mật mã truyền thống, vì đó là cách đã được sử dụng từ hàng ngàn năm nay.

Tuy nhiên, về nguyên tắc hai hàm lập mã và giải mã là khác nhau, không nhất thiết phải phụ thuộc cùng một khóa. Nếu ta xác định mỗi khóa K gồm có hai phần $K = (K', K'')$, K' dành cho việc lập mật mã (và ta có hàm lập mã $e_{K'}$), K'' dành cho việc giải mã (và có hàm giải mã $d_{K''}$), các hàm lập mã và giải mã thỏa mãn hệ thức

$$d_{K''}(e_{K'}(x)) = x \text{ với mọi } x \in \mathcal{P},$$

thì ta được một hệ *mật mã khóa phi đối xứng*. Như vậy, trong một hệ mật mã khóa phi đối xứng, các khóa lập mã và giải mã (K' và K'') là khác nhau, nhưng tất nhiên có quan hệ với nhau. Trong hai khóa đó, khóa cần phải giữ bí mật là khóa giải mã K'', còn khóa lập mã K' có thể được công bố công khai; tuy nhiên điều đó chỉ có ý nghĩa thực tiễn khi việc *biết* K' *tìm* K'' là cực kỳ khó khăn đến mức hầu như không thể thực hiện được. Một hệ mật mã khóa phi đối xứng có tính chất nói trên, trong đó khóa lập mật mã K' của mỗi người tham gia đều được công bố công khai, được gọi là *hệ mật mã khóa công khai*. Khái niệm mật mã khóa công khai mới được ra đời vào giữa những năm 1970, và ngay sau đó đã trở thành một khái niệm trung tâm của khoa học mật mã hiện đại. Ta sẽ dành phần lớn nội dung giáo trình này cho các hệ mật mã đó và những ứng dụng của chúng vào các vấn đề an toàn thông tin.

1.4. Các bài toán về an toàn thông tin.

Chúng ta đang sống trong một thời đại bùng nổ thông tin. Nhu cầu trao đổi thông tin và các phương tiện truyền đưa thông tin phát triển một cách nhanh chóng. Và cùng với sự phát triển đó, đòi hỏi bảo vệ tính bí mật và an toàn của thông tin cũng càng ngày càng to lớn và có tính phổ biến. Có nhiều bài toán khác nhau về yêu cầu an toàn thông tin tùy theo những tình huống khác nhau, nhưng tựu

trung có một số bài toán chung nhất mà ta thường gặp trong thực tiễn là những bài toán sau đây:

- *bảo mật* : giữ thông tin được bí mật đối với tất cả mọi người, trừ một ít người có thẩm quyền được đọc, biết thông tin đó;

- *toàn vẹn thông tin* : bảo đảm thông tin không bị thay đổi hay xuyên tạc bởi những kẻ không có thẩm quyền hoặc bằng những phương tiện không được phép;

- *nhận thực một thực thể*: xác nhận danh tính của một thực thể, chẳng hạn một người, một máy tính cuối trong mạng, một thẻ tín dụng,... ;

- *nhận thực một thông báo* : xác nhận nguồn gốc của một thông báo được gửi đến ;

- *chữ ký* : một cách để gắn kết một thông tin với một thực thể, thường dùng trong bài toán nhận thực một thông báo cũng như trong nhiều bài toán nhận thực khác ;

- *ủy quyền* : chuyển cho một thực thể khác quyền được đại diện hoặc được làm một việc gì đó ;

- *cấp chứng chỉ* : cấp một sự xác nhận thông tin bởi một thực thể được tín nhiệm ;

- *báo nhận* : xác nhận một thông báo đã được nhận hay một dịch vụ đã được thực hiện ;

- *làm chứng* : kiểm thử việc tồn tại một thông tin ở một thực thể khác với người chủ sở hữu thông tin đó ;

- *không chối bỏ được* : ngăn ngừa việc chối bỏ trách nhiệm đối với một cam kết đã có (thí dụ đã ký vào một văn bản) ;

- *ẩn danh* : che giấu danh tính của một thực thể tham gia trong một tiến trình nào đó (thường dùng trong giao dịch tiền điện tử) ;

- *thu hồi* : rút lại một giấy chứng chỉ hay ủy quyền đã cấp;

- vân vân.....

Cơ sở của các giải pháp cho các bài toán kể trên là các phương pháp mật mã, đặc biệt là mật mã khóa công khai, ta sẽ xem xét kỹ một vài bài toán đó trong các chương tiếp theo.

1.5. Thám mã và tính an toàn của các hệ mật mã.

1.5.1. Vấn đề thám mã.

Mật mã được sử dụng trước hết là để bảo đảm tính bí mật cho các thông tin được trao đổi, và do đó bài toán quan trọng nhất của thám mã cũng là bài toán phá bỏ tính bí mật đó, tức là từ bản mật mã có thể thu được dễ dàng (trên các kênh truyền tin công cộng) người thám mã phải phát hiện được nội dung thông tin bị che giấu trong bản mật mã đó, mà tốt nhất là tìm ra được bản rõ gốc của bản mật mã đó. Tình huống thường gặp là bản thân sơ đồ hệ thống mật mã, kể cả các phép lập mã và giải mã (tức các thuật toán $\mathcal{E}$ và $\mathcal{D}$), không nhất thiết là bí mật, do đó bài toán qui về việc *tìm chìa khóa mật mã K*, hay chìa khóa giải mã K', nếu hệ mật mã có khóa phi đối xứng. Như vậy, ta có thể qui ước xem bài toán thám mã cơ bản là bài toán tìm khóa mật mã K (hay khóa giải mã K'). Để giải bài toán đó, giả thiết người thám mã biết thông tin về sơ đồ hệ mật mã được dùng, kể cả các phép lập mã và giải mã tổng quát $\mathcal{E}$ và $\mathcal{D}$. Ngoài ra, người thám mã có thể biết thêm một số thông tin khác, tùy theo những thông tin được biết thêm này mà ta có thể phân loại bài toán thám mã thành các bài toán cụ thể như sau:

- bài toán thám mã *chỉ biết bản mã* : là bài toán phổ biến nhất, khi người thám mã chỉ biết một bản mật mã Y;
- bài toán thám mã khi *biết cả bản rõ* : người thám mã biết một bản mật mã Y cùng với bản rõ tương ứng X;
- bài toán thám mã khi *có bản rõ được chọn* : người thám mã có thể chọn một bản rõ X, và biết bản mật mã tương ứng Y . Điều này có thể xẩy ra khi người thám mã chiếm được (tạm thời) máy lập mã;
- bài toán thám mã khi *có bản mã được chọn* : người thám mã có thể chọn một bản mật mã Y, và biết bản rõ tương ứng X. Điều này có thể xẩy ra khi người thám mã chiếm được tạm thời máy giải mã.

1.5.2. Tính an toàn của một hệ mật mã.

Tính an toàn của một hệ thống mật mã phụ thuộc vào độ khó khăn của bài toán thám mã khi sử dụng hệ mật mã đó. Người ta đã đề xuất một số cách hiểu cho khái niệm an toàn của hệ thống mật mã, để trên cơ sở các cách hiểu đó nghiên cứu tính an toàn của nhiều hệ mật mã khác nhau, sau đây ta giới thiệu vài cách hiểu thông dụng nhất:

- *An toàn vô điều kiện* : giả thiết người thám mã có được thông tin về bản mã. Theo quan niệm lý thuyết thông tin, nếu những hiểu biết về bản mã không thu hẹp được độ bất định về bản rõ đối với người thám mã, thì hệ mật mã là an toàn vô điều kiện, hay theo thuật ngữ của C. Shannon, hệ là *bí mật hoàn toàn*. Như vậy, hệ là an toàn vô điều kiện, nếu độ bất định về bản rõ sau khi người thám mã có được các thông tin (về bản mã) bằng độ bất định về bản rõ trước đó. Tính an toàn vô điều kiện đã được nghiên cứu cho một số hệ mật mã khóa đối xứng mà ta sẽ trình bày trong chương 3.

- *An toàn được chứng minh* : một hệ thống mật mã được xem là có độ an toàn được chứng minh nếu ta có thể chứng minh được là bài toán thám mã đối với hệ thống đó *khó* tương đương với một bài toán khó đã biết, thí dụ bài toán phân tích một số nguyên thành tích các thừa số nguyên tố, bài toán tìm lôgarit rời rạc theo một môđuyn nguyên tố, v.v... (*khó tương đương* có nghĩa là nếu bài toán này giải được thì bài toán kia cũng giải được với cùng một độ phức tạp như nhau).

- *An toàn tính toán* : hệ mật mã được xem là an toàn (về mặt) tính toán, nếu mọi phương pháp thám mã đã biết đều đòi hỏi một nguồn năng lực tính toán vượt mọi khả năng (kể cả phương tiện thiết bị) tính toán của một kẻ thù giả định. An toàn theo nghĩa này, nói theo ngôn ngữ của lý thuyết về độ phức tạp tính toán, là bao hàm cả khái niệm an toàn theo nghia "được chứng minh" nói trên.

Tính an toàn theo nghĩa được chứng minh hay tính toán được sử dụng nhiều trong việc nghiên cứu các hệ thống mật mã hiện đại, đặc biệt là các hệ thống mật mã khóa công khai, ta sẽ trình bày riêng cho từng hệ mật mã được trình bày trong các chương về sau. Ở mục

1,4 ta đã giới thiệu một số bài toán về an toàn thông tin nói chung. Các bài toán đó đều có hạt nhân là tính an toàn của một hệ mật mã nào đó, cho nên việc nghiên cứu tính an toàn của các hệ mật mã cũng góp phần giải quyết các vấn đề an toàn thông tin kể trên.

CHƯƠNG II

Cơ sở toán học của lý thuyết mật mã

2.1. Số học các số nguyên. Thuật toán Euclide.

Ta ký hiệu Z là tập hợp các số nguyên, Z = {.....,-2,-1,0,1,2,....}, **và** Z^+ là tập hợp các số nguyên không âm, Z^+= {0,1,2,.....}. Trong mục này ta sẽ nhắc lại một số kiến thức về số học của các số nguyên cần cho việc trình bày lý thuyết mật mã. Vì để tập giáo trình không quá dài dòng, các kiến thức sẽ được nhắc đến chủ yếu là các khái niệm, các mệnh đề sẽ được sử dụng, v.v..., còn các phần chứng minh sẽ được lược bỏ, bạn đọc nào muốn tìm hiểu kỹ hơn có thể tham khảo các sách chuyên về Số học.

2.1.1. Tính chia hết của các số nguyên.

Tập hợp Z là đóng kín đối với các phép cộng, trừ và nhân, nhưng không đóng kín đối với phép chia: chia một số nguyên cho một số nguyên không phải bao giờ cũng được kết quả là một số nguyên! Vì vậy, trường hợp chia hết, tức khi chia số nguyên a cho số nguyên b được thương là một số nguyên q , $a = b.q$, có một ý nghĩa đặc biệt. Khi đó, ta nói *a chia hết cho b, b chia hết a, a* là *bội số* của *b, b* là *ước số* của *a*, và ký hiệu là $b|a$. Dễ thấy ngay rằng số 1 là ước

số của mọi số nguyên bất kỳ, số 0 là bội số của mọi số nguyên bất kỳ, mọi số nguyên a là ước số, đồng thời là bội số, của chính nó.

Cho hai số nguyên bất kỳ a và b, $b > 1$. Thực hiện phép chia a cho b ta sẽ được hai số q và r sao cho

$$a = b.q + r, \ 0 < r < b.$$

Số q được gọi là *số thương* của phép chia a cho b, ký hiệu a div b, và số r được gọi là *số dư* của phép chia a cho b, ký hiệu a mod b. Thí dụ: 25 div 7 = 3 và 25 mod 7 = 4, -25 div 7 = -4 và -25 mod 7 = 3.

Một số nguyên d được gọi là *ước số chung* của hai số nguyên a và b nếu $d \mid a$ và $d \mid b$. Số nguyên d được gọi là *ước số chung lớn nhất* của a và b nếu $d > 0$, d là ước số chung của a và b, và mọi ước số chung của a và b đều là ước số của d. Ta ký hiệu ước số chung lớn nhất của a và b là gcd(a,b). Thí dụ gcd(12,18) = 6, gcd(-18, 27) = 3.

Dễ thấy rằng với mọi số nguyên dương a ta có gcd(a,0) = a, ta cũng sẽ qui ước xem rằng gcd(0, 0) = 0.

Một số nguyên $a > 1$ được gọi là *số nguyên tố*, nếu a không có ước số nào ngoài 1 và chính a; và được gọi là *hợp số*, nếu không phải là nguyên tố. Thí dụ các số 2 ,3 , 5, 7 là số nguyên tố; các số 4, 6, 8, 10, 12, 14, 15 là hợp số. Hai số a và b được gọi là *nguyên tố với nhau*, nếu chúng không có ước số chung nào khác 1, tức là nếu gcd(a,b) = 1. Một số nguyên $n > 1$ bất kỳ đều có thể viết dưới dạng:

$$n = p_1^{\alpha_1}.p_2^{\alpha_2}...p_k^{\alpha_k}$$

trong đó p_1 , p_2 ,..., p_k là các số nguyên tố khác nhau, α_1 , α_2 ,..., α_k là các số mũ nguyên dương. Nếu không kể thứ tự các thừa số nguyên tố, thì dạng biểu diễn đó là duy nhất, ta gọi đó là *dạng khai triển chính tắc* của n. Thí dụ dạng khai triển chính tắc của 1800 là $2^3 3^2 5^2$.

Các số nguyên tố và các vấn đề về số nguyên tố có một vai trò quan trọng trong số học và trong ứng dụng vào lý thuyết mật mã, ta sẽ xét riêng trong một mục sau.

Định lý 2.1.1. *Nếu $b > 0$ và $b \mid a$ thì* gcd(a ,b) = b.

Nếu $a = bq + r$ *thì* $\gcd(a,b) = \gcd(b,r)$.

Một số nguyên m được gọi là *bội số chung* của a và b nếu $a \mid m$ và $b \mid m$. Số m được gọi là *bội số chung bé nhất* của a và b, và được ký hiệu là lcm(a ,b), nếu $m > 0$, m là bội số chung của a và b, và mọi bội số chung của a và b đều là bội của m. Thí dụ lcm(14,21) = 42.

Với hai số nguyên dương a và b bất kỳ ta có quan hệ

$$\text{lcm}(a,b).\gcd(a,b) = a.b.$$

Từ định lý 2.1.1 ta suy ra thuật toán sau đây thực hiện việc tìm ước số chung lớn nhất của hai số nguyên bất kỳ:

Thuật toán Euclide tìm ước số chung lớn nhất :

INPUT: hai số nguyên không âm a và b, với $a \geq b$.

OUTPUT: ước số chung lớn nhất của a và b.

1. Trong khi còn $b > 0$, thực hiện:

1.1. đặt $r \leftarrow a \bmod b$, $a \leftarrow b$, $b \leftarrow r$.

2. Cho ra kết quả (a).

Thí dụ: Dùng thuật toán Euclide tìm gcd(4864, 3458), ta lần lượt được các giá trị gán cho các biến a, b và r như sau:

4864 = 1. 3458 + 1406
3458 = 2. 1406 + 646
1406 = 2. 646 + 114
646 = 5. 114 + 76
114 = 1. 76 + 38
76 = 2. 38 + 0

a	b	r
4864	3458	
3458	1406	1406
1406	646	646
646	114	114
114	76	76
76	38	38
38	0	0

Và thuật toán cho ta kết quả: gcd(4864, 3458) = 38.

Ta biết rằng nếu gcd(a,b) = d, thì phương trình bất định

$$a.x + b.y = d$$

có nghiệm nguyên (x,y), và một nghiệm nguyên (x,y) như vậy có thể tìm được bởi thuật toán Euclide mở rộng như sau:

Thuật toán Euclide mở rộng :

INPUT: hai số nguyên không âm a và b với $a \geq b$.

OUTPUT: d = gcd(a,b) và hai số x,y sao cho $a.x + b.y = d$.

1. Nếu $b = 0$ thì đặt $d \leftarrow a$, $x \leftarrow 1$, $y \leftarrow 0$, và cho ra (d,x,y).
2. Đặt $x_2 = 1,\ x_1 = 0$, $y_2 = 0$, $y_1 = 1$.
3. Trong khi còn $b > 0$, thực hiện:

   3.1. $q \leftarrow a \operatorname{div} b,\ r \leftarrow a \bmod b$, $x \leftarrow x_2 - qx_1$, $y \leftarrow y_2 - qy_1$.

   3.2. $a \leftarrow b,\ b \leftarrow r$, $x_2 \leftarrow x_1$, $x_1 \leftarrow x$, $y_2 \leftarrow y_1$ và $y_1 \leftarrow y$.
4. Đặt $d \leftarrow a$, $x \leftarrow x_2$, $y \leftarrow y_2$, và cho ra kết quả (d,x,y).

Thí dụ: Dùng thuật toán Euclide mở rộng cho các số $a = 4864$ và $b = 3458$, ta lần lượt được các giá trị sau đây cho các biến $a, b, q, r, x, y, x_1, x_2, y_1, y_2$ (sau mỗi chu trình thực hiện hai lệnh 3.1 và 3.2) :

a	***b***	q	***r***	***x***	***y***	x_1	x_2	y_1	y_2
4864	3458					0	1	1	0
3458	1406	1	1406	1	-1	1	0	-1	1
1406	646	2	646	-2	3	-2	1	3	-1
646	114	2	114	5	-7	5	-2	-7	3
114	76	5	76	-27	38	-27	5	38	-7

76	38	1	38	32	-45	32	-27	-45	38
38	0	2	0	-91	128	-91	32	128	-45

Ta dễ thử lại rằng sau mỗi lần thực hiện chu trình gồm hai lệnh 3.1 và 3.2, các giá trị x,y,r thu được luôn thoả mãn $4864.x + 3458.y = r$, và do đó khi kết thúc các vòng lặp (ứng với giá trị $b = 0$), thực hiện tiếp lệnh 4 ta được kết quả $d = 38$, $x = 32$ và $y = -45$, cặp số (32,-45) thoả mãn: $4864.32 + 3458.(-45) = 38$.

2.1.2. Đồng dư và phương trình đồng dư tuyến tính.

Cho n là một số nguyên dương. Ta nói hai số nguyên a và b là *đồng dư với nhau theo môđuyn* n, và viết $a \equiv b \pmod{n}$, nếu $n \mid a-b$ (tức cũng là nếu $a - b$ chia hết cho n, hay khi chia a và b cho n ta được cùng một số dư như nhau).

Thí dụ: $23 \equiv 8 \pmod 5$, vì $23 - 8 = 5.3$, $-19 \equiv 9 \pmod 7$ vì $-19 - 9 = -4 . 7$.

Quan hệ đồng dư (theo một môđuyn n) trên tập hợp các số nguyên có các tính chất phản xạ, đối xứng và bắc cầu,tức là một quan hệ tương đương, do đó nó tạo ra một phân hoạch trên tập hợp tất cả các số nguyên $\mathbb{Z}$ thành ra các lớp tương đương: hai số nguyên thuộc cùng một lớp tương đương khi và chỉ khi chúng cho cùng một số dư nếu chia cho n. Mỗi lớp tương đương như vậy được đại diện bởi một số duy nhất trong tập hợp $\mathbb{Z}_n = \{0, 1, 2, 3,...., n-1\}$, là số dư chung khi chia các số trong lớp đó cho n. Vì vậy, ta có thể đồng nhất $\mathbb{Z}_n$ với tập hợp tất cả các lớp tương đương các số nguyên theo mod$n$; trên tập đó ta có thể xác định các phép tính cộng, trừ và nhân theo mod$n$.

Thí dụ: $\mathbb{Z}_{25} = \{0, 1, 2, ..., 24\}$. Trong $\mathbb{Z}_{25}$, $15 + 14 = 4$, vì $15 + 14 = 29 = 4 \pmod{25}$. Tương tự, $15.14 = 10$ trong $\mathbb{Z}_{25}$.

Cho $a \in Z_n$. Một số nguyên $x \in Z_n$ được gọi là *nghịch đảo* của a theo mod n , nếu $a.x \equiv 1 \pmod{n}$. Nếu có số x như vậy thì ta nói a là khả nghịch, và ký hiệu x là $a^{-1} \bmod n$. Thí dụ $22^{-1} \bmod 25 = 8$, vì $22 .8 = 176 \equiv 1 \pmod{25}$. Từ định nghĩa ta có thể suy ra rằng a là khả nghịch theo mod$n$ khi và chỉ khi gcd(a,n) = 1, tức là khi a và n nguyên tố với nhau.

Ta định nghĩa phép chia trong Z_n như sau: $a : b \pmod{n} = a.b^{-1} \bmod n$. Phép chia chỉ thực hiện được khi b là khả nghịch theo mod$n$. Thí dụ $15 : 22 \pmod{25} = 15.22^{-1} \bmod 25 = 20$.

Bây giờ ta xét các *phương trình đồng dư tuyến tính.*

Phương trình đồng dư tuyến tính có dạng

$$a.x \equiv b \pmod{n}, \qquad (1)$$

trong đó a, b, n là các số nguyên, $n > 0$, x là ẩn số. Phương trình đó có nghiệm khi và chỉ khi d = gcd(a,n)$|b$, và khi đó nó có đúng d nghiệm theo mod$n$. Thực vậy, đặt $a' = a/d$, $b' = b/d$, $n' = n/d$, ta thấy phương trình đồng dư (1) tương đương với phương trình

$$a'.x \equiv b' \pmod{n'},$$

Vì gcd(a',n') = 1, nên phương trình này có một nghiệm theo modn' :

$$x = x_0 \equiv b'.a'^{-1} \pmod{n'},$$

và do đó phương trình (1) có d nghiệm theo mod$n$ là :

$$x = x_0 , x_0 + n', \ldots , x_0 + (d-1)n' \pmod{n}.$$

Tất cả d nghiệm đó khác nhau theo mod$n$, nhưng cùng đồng dư với nhau theo modn'.

Bây giờ ta xét hệ thống các phương trình đồng dư tuyến tính. Một hệ như vậy có thể đưa về dạng

$$\begin{cases} x_1 \equiv a_1 \pmod{n_1} \\ x_2 \equiv a_2 \pmod{n_2} \\ ..... \\ x_k \equiv a_k \pmod{n_k} \end{cases} \qquad (2)$$

Ta ký hiệu: $n = n_1.n_2....n_k$, $N_i = n/n_i$. Ta có định lý sau đây:

Định lý 2.2.1 (định lý số dư Trung quốc). *Giả sử* ***các số nguyên*** $n_1, n_2,....,n_k$ *là từng cặp nguyên tố với nhau. Khi đó, hệ phương trình đồng dư tuyến tính* (2) *có một nghiệm duy nhất theo* mod$n$.

Nghiệm duy nhất nói trong định lý 2.2.1 được cho bởi biểu thức:

$$x = \sum_{i=1}^{k} a_i.N_i.M_i \bmod n,$$

trong đó $M_i = N_i^{-1} \bmod n_i$ (có M_i vì N_i và n_i nguyên tố với nhau).

Thí dụ: Cặp phương trình $x \equiv 3 \pmod 7$ và $x \equiv 7 \pmod{13}$ có một nghiệm duy nhất $x \equiv 59 \pmod{91}$.

Nếu $(n_1 , n_2) = 1$, thì cặp phương trình $x \equiv a \pmod{n_1}$ và $x \equiv a \pmod{n_2}$ có nghiệm duy nhất $x \equiv a \pmod{n}$ theo mod$n$ với $n = n_1 n_2$.

2.1.3.Thặng dư thu gọn và phần tử nguyên thuỷ.

Tập $Z_n = \{ 0,1,2,..., n-1\}$ thường được gọi là *tập các thặng dư đầy đủ theo* mod$n$, vì mọi số nguyên bất kỳ đều có thể tìm được trong Z_n một số đồng dư với mình (theo mod$n$). Tập Z_n là đóng đối với các phép tính cộng, trừ và nhân theo mod$n$, nhưng không đóng đối với phép chia, vì phép chia cho a theo mod$n$ chỉ có thể thực hiện được khi a và n nguyên tố với nhau, tức khi gcd(a ,n) =1.

Bây giờ ta xét tập $Z_n^* = \{ a \in Z_n : \gcd(a, n) = 1\}$, tức Z_n^* là tập con của Z_n bao gồm tất cả các phần tử nguyên tố với n. Ta gọi tập đó là *tập các thặng dư thu gọn theo* mod$n$. Mọi số nguyên nguyên tố với n đều có thể tìm thấy trong Z_n^* một đại diện đồng dư với mình theo mod$n$. Chú ý rằng nếu p là một số nguyên tố thì $Z_p^* = \{1,2,...,p-1\}$.

Tập Z_n^* lập thành một nhóm con đối với phép nhân của Z_n, vì trong Z_n^* phép chia theo mod$n$ bao giờ cũng thực hiện được, ta sẽ gọi Z_n^* là *nhóm nhân* của Z_n.

Theo đại số học, ta gọi số các phần tử trong một nhóm là *cấp* của nhóm đó. Ta ký hiệu $\phi(n)$ là số các số nguyên dương bé hơn n và nguyên tố với n. Như vậy, nhóm Z_n^* có cấp $\phi(n)$, và nếu p là số nguyên tố thì nhóm Z_p^* có cấp p-1.

Ta nói một phần tử $g \in Z_n^*$ có *cấp* m, nếu m là số nguyên dương bé nhất sao cho $g^m = 1$ trong Z_n^*. Theo một định lý trong Đại số, ta có $m \mid \phi(n)$. Vì vậy, với mọi $b \in Z_n^*$ ta luôn có $b^{\phi(n)} \equiv 1 \bmod n$.

Nếu p là số nguyên tố, thì do $\phi(p) = p - 1$, ta có với mọi $b \in Z_p^*$:

$$b^{p-1} \equiv 1 \pmod{p} \qquad (3)$$

Nếu b có cấp p - 1, tức p - 1 là số mũ bé nhất thoả mãn công thức (3), thì các phần tử $b, b^2, ..., b^{P-1}$ đều khác nhau và theo mod$p$, chúng lập thành Z_p^*. Theo thuật ngữ đại số, khi đó ta nói Z_p^* là một *nhóm cyclic* và b là một phần tử sinh, hay *phần tử nguyên thuỷ* của nhóm đó. Trong lý thuyết số, người ta đã chứng minh được các tính chất sau đây của các phần tử nguyên thuỷ:

1. Với mọi số nguyên tố p, Z_p^* là nhóm cyclic, và có $\phi(p-1)$ phần tử nguyên thuỷ.
2. Nếu $p - 1 = p_1^{\alpha_1}.p_2^{\alpha_2}....p_s^{\alpha_s}$ là khai triển chính tắc của p -1, và nếu

$$a^{\frac{p-1}{p_1}} \equiv 1(\text{mod } p), ..., a^{\frac{p-1}{p_s}} \equiv 1(\text{mod } p),$$

thì a là phần tử nguyên thuỷ theo mod$p$ (tức của Z_p^*).

3. Nếu g là phần tử nguyên thuỷ theo mod$p$, thì $\beta = g^i \text{ mod} p$ với mọi i mà gcd(i, p -1) = 1, cũng là phần tử nguyên thuỷ theo mod$p$.

Ba tính chất đó là cơ sở giúp ta tìm các phần tử nguyên thuỷ theo mod$p$, với p là số nguyên tố bất kỳ. Ngoài ra, ta cũng chú ý một số tính chất sau đây, có thể được sử dụng nhiều trong các chương sau:

a) Nếu p là số nguyên tố và gcd(a,p) =1, thì $a^{p-1} \equiv 1$ (mod$p$) (*định lý Fermat*).

b) Nếu $a \in Z_n^*$, thì $a^{\phi(n)} \equiv 1(\text{mod } n)$. Nếu $r \equiv s(\text{mod } \phi(n))$ thì $a^r \equiv a^s(\text{mod } n)$ (*định lý Euler*).

2.1.4. Phương trình đồng dư bậc hai và thặng dư bậc hai.

Ta xét phương trình đồng dư bậc hai có dạng đơn giản sau đây:

$$x^2 \equiv a(\text{mod } n),$$

trong đó n là một số nguyên dương, a là số nguyên với gcd(a,n) =1, và x là ẩn số. Phương trình đó không phải bao giờ cũng có nghiệm, khi nó có nghiệm thì ta nói a là một *thặng dư bậc hai* mod$n$; nếu không thì nói a là một *bất thặng dư bậc hai* mod$n$. Tập các số nguyên nguyên tố với n được phân hoạch thành hai tập con: tập Q_n các thặng dư bậc hai mod$n$, và tập $\overline{Q_n}$ các bất thặng dư mod$n$.

Khi n = p là số nguyên tố, ta có *tiêu chuẩn Euler* sau đây: Số a là thặng dư bậc hai mod$p$ nếu và chỉ nếu $a^{(p-1)/2} \equiv 1(\text{mod } p)$. Tiêu chuẩn đó được chứng minh như sau:

Giả sử có x sao cho $x^2 \equiv a(\text{mod } p)$, khi đó ta cũng sẽ có

$$a^{(p-1)/2} \equiv (x^2)^{(p-1)/2} \equiv x^{p-1} \equiv 1(\text{mod } p).$$

Ngược lại, giả sử $a^{(p-1)/2} \equiv 1 (\text{mod } p)$. Khi đó $a \in Z_p^*$. Lấy b là một phần tử nguyên thuỷ mod$p$, ắt có một số i nào đó sao cho $a = b^i \text{ mod } p$ .Từ đó,

$$a^{(p-1)/2} \equiv b^{i(p-1)/2} \equiv 1 (\text{mod } p).$$

Phần tử b có cấp p - 1, do đó (p - 1) chia hết $i(p$ - 1)/2, i phải là số chẵn, $i = 2j$, và a có căn bậc hai là $\pm b^j \text{mod} p$.

Cho p là một số nguyên tố lẻ. Với mọi $a \geq 0$ ta định nghĩa *ký hiệu Legendre* $\left(\frac{a}{p}\right)$ như sau:

$$\left(\frac{a}{p}\right) = \begin{cases} 0\ ,\ khi\ a \equiv 0 (\text{mod } p); \\ 1\ ,\ khi\ a \in Q_p\ ; \\ -1,\ khi\ a \notin Q_p\ . \end{cases}$$

Từ định nghĩa ta suy ra ngay *a là thặng dư bậc hai modp khi và chỉ khi* $\left(\frac{a}{p}\right) = 1$. Và theo tiêu chuẩn Euler nói trên, với mọi $a \geq 0$, ta có:

$$\left(\frac{a}{p}\right) \equiv a^{(p-1)/2} (\text{mod } p).$$

Bây giờ ta mở rộng ký hiệu Legendre để được *ký hiệu Jacobi* đối với mọi số nguyên lẻ $n \geq 1$ và mọi số nguyên $a \geq 0$, cũng được ký hiệu bởi $\left(\frac{a}{n}\right)$ và được định nghĩa như sau: Giả sử a có khai triển chính tắc thành thừa số nguyên tố là $n = p_1^{\alpha_1}.p_2^{\alpha_2}....p_k^{\alpha_k}$ thì

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p_1}\right)^{\alpha_1}.\left(\frac{a}{p_2}\right)^{\alpha_2}....\left(\frac{a}{p_k}\right)^{\alpha_k}.$$

Khi $n = p$ là số nguyên tố thì giá trị của các ký hiệu Legendre và Jacobi là như nhau. Việc tính ký hiệu Legendre có thể phức tạp khi p rất lớn, trong khi việc tính ký hiệu Jacobi có thể thuận lợi hơn do có thể sử dụng các tính chất 1-4 sau đây:

1. Nếu $m_1 \equiv m_2 \pmod{n}$, thì $\left(\frac{m_1}{n}\right) = \left(\frac{m_2}{n}\right)$.

2. $\left(\frac{2}{n}\right) = \begin{cases} 1, & khi\ \ n \equiv \pm 1 \pmod 8, \\ -1, & khi\ n \equiv \pm 3 \pmod 8. \end{cases}$

3. $\left(\frac{m_1.m_2}{n}\right) = \left(\frac{m_1}{n}\right).\left(\frac{m_2}{n}\right)$.

4. Nếu m và n đều là số lẻ, thì

$$\left(\frac{m}{n}\right) = \begin{cases} -\left(\frac{n}{m}\right), khi\ m \equiv 3 \pmod 4 \,\&\, n \equiv 3 \pmod 4, \\ \left(\frac{n}{m}\right), khi\ m \equiv 1 \pmod 4 \vee n \equiv 1 \pmod 4. \end{cases}$$

Thí dụ: Dùng các tính chất đó, ta tính được:

$$\left(\frac{7411}{9283}\right) = \left(\frac{9283}{7411}\right) = \left(\frac{1872}{7411}\right) = \left(\frac{2}{7411}\right)^4 . \left(\frac{117}{7411}\right) =$$
$$\left(\frac{117}{7411}\right) = -\left(\frac{7411}{117}\right) = -\left(\frac{40}{117}\right) = -\left(\frac{2}{117}\right)^3 . \left(\frac{5}{117}\right) =$$
$$\left(\frac{5}{117}\right) = \left(\frac{117}{5}\right) = \left(\frac{2}{5}\right) = -1.$$

9283 là một số nguyên tố. Do đó, giá trị -1 của ký hiệu Jacobi $\left(\frac{7411}{9283}\right)$ cũng là giá trị của cùng ký hiệu Legendre đó, và ta kết luận được rằng 7411 là bất thặng dư bậc hai mod 9283 , hay phương trình

$$x^2 \equiv 7411 \pmod{9283}$$

là vô nghiệm.

Bây giờ ta xét việc giải phương trình đồng dư bậc hai

$$x^2 \equiv a \pmod{n} \qquad (4)$$

trong một trường hợp đặc biệt khi $n = p$ là số nguyên tố có dạng $p = 4m + 3$, tức p đồng dư với 3 theo mod4, và a là một số nguyên nguyên tố với p. Theo tiêu chuẩn Euler ta biết phương trình (4) có nghiệm khi và chỉ khi $a^{(p-1)/2} \equiv 1 \pmod{p}$. Khi đó ta có:

$$a^{\frac{p-1}{2}+1} \equiv a \pmod{p},$$
$$a^{2(m+1)} \equiv a \pmod{p},$$

do đó $x \equiv \pm a^{m+1} \pmod{p}$ là hai nghiệm của phương trình (4).

2.2. Xác suất và thuật toán xác suất.

2.2.1. Khái niệm xác suất.

Ta xét một tập hợp Ω, được gọi là *không gian các sự kiện sơ cấp* (hay không gian mẫu). Các phần tử của Ω, tức các sự kiện sơ cấp hay các mẫu, có thể được xem như các kết quả có thể có (và loại trừ lẫn nhau) của một thực nghiệm nào đó. Về sau ta chỉ xét các không gian rời rạc, tức tập Ω là hữu hạn, giả sử $\Omega = \{s_1, s_2, ..., s_n\}$.

Một *phân bố xác suất* P trên Ω được định nghĩa là một tập các số thực không âm $P = \{p_1, p_2, ..., p_n\}$ có tổng $\sum p_i = 1$. Số p_i được coi là xác suất của sự kiện sơ cấp s_i.

Một tập con $E \subseteq \Omega$ được gọi là một *sự kiện*. Xác suất của sự kiện E được định nghĩa bởi $p(E) = \sum_{s \in E} p(s)$.

Giả sử E là một sự kiện trong không gian xác suất Ω. Ta định nghĩa *sự kiện bù* của E, ký hiệu $\overline{E}$, là sự kiện gồm tất cả các sự kiện sơ cấp

trong Ω mà không thuộc E. Dùng các thuật ngữ của lý thuyết tập hợp, ta có thể định nghĩa các *sự kiện hợp* $E_1 \cup E_2$ và *sự kiện giao* $E_1 \cap E_2$ của hai sự kiện E_1 và E_2 bất kỳ. Và ta có:

1) Giả sử E là một sự kiện. Khi đó $0 \leq p(E) \leq 1$ và $p(\bar{E}) = 1 - p(E)$. Ngoài ra, $p(\Omega) = 1$ và $p(\varnothing) = 0$.

2) Giả sử E_1 và E_2 là hai sự kiện. Nếu $E_1 \subseteq E_2$ thì $p(E_1) \leq p(E_2)$. Và có $p(E_1 \cup E_2) + p(E_1 \cap E_2) = p(E_1) + p(E_2)$. Do đó $p(E_1 \cup E_2) = p(E_1) + p(E_2)$ khi và chỉ khi $E_1 \cap E_2 = \varnothing$, tức là khi E_1 và E_2 là hai sự kiện loại trừ lẫn nhau.

Cho E_1 và E_2 là hai sự kiện, với $p(E_2) > 0$. Ta định nghĩa *xác suất có điều kiện* của E_1 khi có E_2, ký hiệu $p(E_1|E_2)$, là

$$p(E_1|E_2) = \frac{p(E_1 \cap E_2)}{p(E_2)}.$$

Từ định nghĩa ta suy ra *công thức Bayes*:

$$p(E_1|E_2) = \frac{p(E_1).p(E_2|E_1)}{p(E_2)}. \quad .$$

Ta nói hai sự kiện E_1 và E_2 là *độc lập* với nhau, nếu $p(E_1 \cap E_2) = p(E_1).p(E_2)$. Khi đó ta có: $p(E_1|E_2) = p(E_1)$ và $p(E_2|E_1) = p(E_2)$.

Giả sử Ω là một không gian mẫu với một phân bố xác suất P. Ta gọi một *đại lượng ngẫu nhiên* ξ trên Ω là một ánh xạ gán cho mỗi $s \in \Omega$ một số thực $\xi(s)$. Hiển nhiên, nếu ξ và η là các đại lượng ngẫu nhiên trên Ω, thì $\xi + \eta$, $\xi.\eta$ được định nghĩa bởi :

$\forall s \in \Omega$: $(\xi+\eta)(s) = \xi(s) + \eta(s)$, $(\xi.\eta)(s) = \xi(s).\eta(s)$.

cũng là các đại lương ngẫu nhiên trên Ω.

Giả sử ξ là một đại lượng ngẫu nhiên trên không gian mẫu Ω. Điều đó có nghĩa là với mọi $s \in \Omega$, ξ lấy giá trị bằng $\xi(s)$ với xác suất $p(s)$. Ta định nghĩa *giá trị kỳ vọng* (hay *trung bình*, hay *kỳ vọng toán học*) của ξ là

$$E(\xi) = \sum_{s \in \Omega} \xi(s).p(s).$$

Phương sai của đại lượng ngẫu nhiên ξ có giá trị trung bình μ được định nghĩa là Var $(\xi) = E((\xi - \mu)^2)$.

Căn bậc hai không âm của Var (ξ) được gọi là *độ lệch chuẩn* của ξ.

2.2.2. Tính bí mật hoàn toàn của một hệ mật mã.

Năm 1949, C. Shannon công bố công trình *Lý thuyết truyền thông của các hệ bí mật*, đưa ra nhiều quan niệm làm cơ sở cho việc đánh giá tính bí mật của các hệ mật mã, trong đó có khái niệm *tính bí mật hoàn toàn* của một hệ mật mã được định nghĩa như sau: Cho hệ mật mã $\mathcal{S} = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$. Giả thử trên các tập $\mathcal{P}$, $\mathcal{C}$ và $\mathcal{K}$ được xác định tương ứng các phân bố xác suất $p_P(.)$, $p_C(.)$ và $p_K(.)$. Như vậy, với mọi $x \in \mathcal{P}$, $y \in \mathcal{C}$ và $K \in \mathcal{K}$, $p_P(x)$, $p_C(y)$ và $p_K(K)$ tương ứng là các xác suất để ký tự bản rõ là x, ký tự bản mã là y và khoá là K. Xác suất có điều kiện, chẳng hạn, xác suất của việc bản rõ là x khi bản mã là y, được ký hiệu là $p_P(x|y)$. Một hệ mật mã được gọi là *bí mật hoàn toàn*, nếu với mọi $x \in \mathcal{P}$, $y \in \mathcal{C}$ có $p_P(x|y) = p_P(x)$. Điều đó có nghĩa là việc biết xác suất bản rõ là x là như nhau dù biết hay không biết bản mã là y; nói cách khác, có thông tin về bản mã

không cho ta biết gì thêm về bản rõ; bản rõ và bản mã, với tư cách các biến ngẫu nhiên, là độc lập với nhau. Ta có định lý sau đây:

Định lý 2.2.1. *Giả sử* $\mathcal{S} = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ *là một hệ mật mã với điều kiện* $|\mathcal{P}| = |\mathcal{C}| = |\mathcal{K}|$, *tức các tập* $\mathcal{P}, \mathcal{C}, \mathcal{K}$ *có số các phần tử bằng nhau. Khi đó, hệ là bí mật hoàn toàn nếu và chỉ nếu mỗi khoá* $K \in \mathcal{K}$ *được dùng với xác suất bằng nhau là* $1/|\mathcal{K}|$, *và với mọi* $x \in \mathcal{P}$, $y \in \mathcal{C}$ *có một khoá duy nhất* $K \in \mathcal{K}$ *sao cho* $e_K(x) = y$.

Chứng minh. a) Giả thử hệ $\mathcal{S}$ là bí mật hoàn toàn. Khi đó, với mọi $x \in \mathcal{P}$ và $y \in \mathcal{C}$ có $p_P(x|y) = p_P(x)$. Ngoài ra ta có thể giả thiết $p_C(y) > 0$ với mọi $y \in \mathcal{C}$. Từ đó theo công thức Bayes ta có $p_C(y|x) = p_C(y) > 0$. Điều đó có nghĩa là có ít nhất một khoá K sao cho $e_K(x) = y$. Vì vậy, nếu cố định một $x \in \mathcal{P}$ thì ta có

$$|\mathcal{C}| = |\{ e_K(x): K \in \mathcal{K} \}| \le |\mathcal{K}| \ .$$

Theo giả thiết của định lý, $|\mathcal{C}| = |\mathcal{K}|$, do đó

$$|\{ e_K(x): K \in \mathcal{K} \}| = |\mathcal{K}| \ .$$

Nhưng điều này lại có nghĩa là không thể có hai khoá $K_1 \neq K_2$ sao cho $e_{K_1}(x) = e_{K_2}(x)$. Vậy ta đã chứng minh được với mọi $x \in \mathcal{P}$ và $y \in \mathcal{C}$ có đúng một khoá K sao cho $e_K(x) = y$.

Ký hiệu $n = |\mathcal{K}|$ và đặt $\mathcal{K} = \{K_1, ..., K_n\}$. Cố định một $y \in \mathcal{C}$ và giả thử $e_{K_i}(x_i) = y$ với $\mathcal{P} = \{x_1, ..., x_n\}$, $1 \le i \le n$. Dùng công thức Bayes ta lại có

$$p_P(x_i|y) = \frac{p_C(y|x_i).p_P(x_i)}{p_C(y)} = \frac{p_K(K_i).p_P(x_i)}{p_C(y)}.$$

Do giả thiết hệ là bí mật hoàn toàn, ta có $p_P(x_i|y) = p_P(x_i)$. Từ đó suy ra với mọi i, $1 \le i \le n$, $p_K(K_i) = p_C(y)$. Vậy các $p_K(K_i)$ $(1 \le i \le n)$ đều bằng nhau, và do đó đều bằng $1/|\mathcal{K}|$.

b) Bây giờ ta chứng minh điều ngược lại. Giả thiết $p_K(K) = 1/|\mathcal{K}|$ với mọi $K \in \mathcal{K}$, và với mọi $x \in \mathcal{P}$, $y \in \mathcal{C}$ có đúng một khoá $K \in \mathcal{K}$ sao cho $e_K(x) = y$. Ta tính:

$$p_C(y) = \sum_{K \in \mathcal{K}} p_K(K).p_P(d_K(y)) = \sum_{K \in \mathcal{K}} \frac{1}{|\mathcal{K}|} p_P(d_K(y)) =$$
$$= \frac{1}{|\mathcal{K}|} \sum_{K \in \mathcal{K}} p_P(d_K(y)).$$

Khi K chạy qua tập khoá $\mathcal{K}$ thì $d_K(y)$ chạy qua tập $\mathcal{P}$, do đó

$$\sum_{K \in \mathcal{K}} p_P(d_K(y)) = \sum_{x \in \mathcal{P}} p_P(x) = 1,$$

và ta được $p_C(y) = 1/|\mathcal{K}|$ với mọi $y \in \mathcal{C}$.

Mặt khác, gọi K là khoá duy nhất mà $e_K(x) = y$, ta có

$$p_C(y|x) = p_K(K) = 1/|\mathcal{K}| \ .$$

Dùng công thức B**ayes ta lại được** với mọi $x \in \mathcal{P}$, $y \in \mathcal{C}$:

$$p_P(x|y) = \frac{p_P(x).p_C(y|x)}{p_C(y)} = \frac{p_P(x).1/|\mathcal{K}|}{1/|\mathcal{K}|} = p_P(x).$$

Vậy hệ là bí mật hoàn toàn. Định lý được chứng minh.

2.2.3. Thuật toán xác suất:

Khái niệm *thuật toán* mà ta thường hiểu là thuật toán tất định, đó là một tiến trình thực hiện các phép toán trên dữ liệu đầu vào và cho kết quả ở đầu ra. Theo D.E. Knuth, thuật toán có 5 thuộc tính cơ bản: tính *hữu hạn*, thuật toán luôn kết thúc sau một số hữu hạn bước; tính *xác định*, mỗi bước của thuật toán phải được xác định một cách chính xác; tập hợp *đầu vào* và *đầu ra* của mỗi thuật toán cũng được xác định rõ ràng; và tính *hiệu quả*, mọi phép toán trong thuật toán phải là cơ bản, có thể được thực hiện chính xác trong một thời gian xác định. Thuật toán là khái niệm cơ bản đối với việc lập trình trên máy tính, và đã được sử dụng rất phổ biến. Nhưng như ta biết, đối với nhiều bài toán trong thực tế, không phải bao giờ ta cũng tìm được thuật toán giải chúng với độ phức tạp tính toán chấp nhận được (ta sẽ xét qua vấn đề này trong một tiết sau). Vì vậy, cùng với các thuật toán tất định, đối với một số bài toán ta sẽ xét thêm các thuật toán xác suất, đó là những thuật toán mà cùng với dữ liệu đầu vào ta bổ sung thêm giá trị của một đại lượng ngẫu nhiên tương ứng nào đó, thường là các số ngẫu nhiên.

Các thuật toán xác suất thường được xây dựng cho các bài toán quyết định, tức các bài toán xác định trên một tập hợp dữ liệu sao cho ứng với mỗi dữ liệu bài toán có một trả lời *có* hoặc *không* . Người ta chia các thuật toán xác suất thành hai loại: loại thuật toán *Monte Carlo* và loại thuật toán *Las Vegas* . Thuật toán Monte Carlo luôn kết thúc với kết quả *có* hoặc *không* đối với mọi dữ liệu đầu vào bất kỳ; còn thuật toán Las Vegas tuy cũng kết thúc với mọi dữ liệu, nhưng có thể kết thúc với một thông báo *không có trả lời* có hoặc không. Thuật toán Monte Carlo được gọi là *thiên về có*, nếu nó cho trả lời *có* thì trả lời đó chắc chắn là đúng, còn nếu nó cho trả lời *không* thì trả lời đó có thể sai với một xác suất ε nào đó. Tương tự, một thuật toán Monte Carlo được gọi là *thiên về không*, nếu nó cho trả lời *không* thì trả lời đó chắc chắn là đúng, còn nếu nó cho trả lời *có* thì trả lời đó có thể sai với một xác suất ε nào đó. Còn với thuật toán Las Vegas, nếu nó kết thúc với trả lời *có* hoặc *không* , thì trả lời đó chắc chắn đúng, và nó có thể kết thúc với thông báo *không có trả*

lời với một xác suất ε nào đó. Trong tiết 2.8 sau đây ta sẽ cho vài thí dụ cụ thể về một số thuật toán xác suất thuộc cả hai loại đó.

2.3. Độ phức tạp tính toán.

2.3.1. Khái niệm về độ phức tạp tính toán.

Lý thuyết thuật toán và các hàm số tính được ra đời từ những năm 30 của thế kỷ 20 đã đặt nền móng cho việc nghiên cứu các vấn đề "tính được", "giải được" trong toán học, đưa đến nhiều kết quả rất quan trọng và lý thú. Nhưng từ cái "tính được" một cách trừu tượng, hiểu theo nghĩa tiềm năng,đến việc tính được trong thực tế của khoa học tính toán bằng máy tính điện tử, là cả một khoảng cách rất lớn. Biết bao nhiêu thứ được chứng minh là tính được một cách tiềm năng, nhưng không tính được trong thực tế, dù có sự hỗ trợ của những máy tính điện tử ! Vấn đề là do ở chỗ những đòi hỏi về không gian vật chất và về thời gian để thực hiện các tiến trình tính toán nhiều khi vượt quá xa những khả năng thực tế. Từ đó, vào khoảng giữa những năm 60 (của thế kỷ trước), một lý thuyết về độ phức tạp tính toán bắt đầu được hình thành và phát triển nhanh chóng, cung cấp cho chúng ta nhiều hiểu biết sâu sắc về bản chất phức tạp của các thuật toán và các bài toán, cả những bài toán thuần tuý lý thuyết đến những bài toán thường gặp trong thực tế. Sau đây ta giới thiệu sơ lược một số khái niệm cơ bản và vài kết quả sẽ được dùng đến của lý thuyết đó.

Trước hết, ta hiểu *độ phức tạp tính toán* (về không gian hay về thời gian) của một tiến trình tính toán là số ô nhớ được dùng hay số các phép toán sơ cấp được thực hiện trong tiến trình tính toán đó.

Dữ liệu đầu vào đối với một thuật toán thường được biểu diễn qua các từ trong một bảng ký tự nào đó. *Độ dài của một từ* là số ký tự trong từ đó.

Cho một thuật toán $\mathcal{A}$ trên bảng ký tự Σ (tức có đầu vào là các từ trong Σ) . *Độ phức tạp tính toán* của thuật toán $\mathcal{A}$ được hiểu là một hàm số $f_A(n)$ sao cho với mỗi số n, $f_A(n)$ là số ô nhớ, hay số phép toán sơ cấp tối đa mà $\mathcal{A}$ cần để thực hiện tiến trình tính toán của mình trên các dữ liệu vào có độ dài $\leq n$. Ta nói thuật toán $\mathcal{A}$ có độ phức tạp thời gian *đa thức*, nếu có một đa thức $P(n)$ sao cho với mọi n đủ lớn ta có $f_A(n) \leq P(n)$, trong đó $f_A(n)$ là độ phức tạp tính toán theo thời gian của $\mathcal{A}$.

Về sau khi nói đến các bài toán, ta hiểu đó là các *bài toán quyết định*, mỗi bài toán P như vậy được xác định bởi:

- một tập các dữ liệu vào I (trong một bảng ký tự Σ nào đó),

- một câu hỏi Q trên các dữ liệu vào, sao cho với mỗi dữ liệu vào $x \in I$, câu hỏi Q có một trả lời *đúng* hoặc *sai*.

Ta nói bài toán quyết định P là *giải được*, nếu có thuật toán để giải nó, tức là thuật toán làm việc có kết thúc trên mọi dữ liệu vào của bài toán, và cho kết quả *đúng* hoặc *sai* tuỳ theo câu hỏi Q trên dữ liệu đó có trả lời đúng hoặc sai. **Bài toán P** là *giải được trong thời gian đa thức*, nếu có thuật toán giải nó với độ phức tạp thời gian đa thức. Sau đây là vài thí dụ về các bài toán quyết định:

Bài toán *SATISFIABILITY* (viết tắt là *SAT*):

- mỗi dữ liệu vào là một công thức F của lôgich mệnh đề, được viết dưới dạng hội chuẩn tắc, tức dạng hội của một số các "clause".

- Câu hỏi là: công thức F có thoả được hay không ?

Bài toán *CLIQUE*:

- mỗi dữ liệu vào là một graph G và một số nguyên k.

- Câu hỏi là: Graph G có một clique với $\geq k$ đỉnh hay không ? (một clique của G là một graph con đầy đủ của G).

Bài toán *KNAPSACK*:

- mỗi dữ liệu là một bộ $n+1$ số nguyên dương $I = (s_1,...,s_n ; T)$.
- Câu hỏi là: có hay không một vectơ Boole $(x_1,...,x_n)$ sao cho

$$\sum_{i=1}^{n} x_i.s_i = T \ ?$$

(vectơ boole là vectơ có các thành phần là 0 hoặc 1).

Bài toán thặng dư bậc hai :

- mỗi dữ liệu gồm hai số nguyên dương (a , n).
- Câu hỏi là: a có là thặng dư bậc hai theo mod$n$ hay không ?

Bài toán hợp số :

- mỗi dữ liệu là một số nguyên dương N.
- Câu hỏi: N là hợp số hay không ? Tức có hay không hai số $m, n > 1$ sao cho $N = m . n$?

Tương tự, nếu đặt câu hỏi là "N là số nguyên tố hay không?" thì ta được bài toán số nguyên tố .

Đối với tất cả các bài toán kể trên, trừ bài toán hợp số và số nguyên tố, cho đến nay người ta đều chưa tìm được thuật toán giải chúng trong thời gian đa thức.

2.3.2. Lớp phức tạp.

Ta xét một vài lớp các bài toán được xác định theo độ phức tạp tính toán của chúng. Trước hết, ta định nghĩa $\mathcal{P}$ là lớp tất cả các bài toán có thể giải được bởi thuật toán trong thời gian đa thức.

Giả sử cho hai bài toán P_1 và P_2 với các tập dữ liệu trong hai bảng ký tự tương ứng là Σ_1 và Σ_2 . Một thuật toán $f : \Sigma_1^* \to \Sigma_2^*$ được gọi là một *phép qui dẫn* bài toán P_1 về bài toán P_2 , nếu nó biến mỗi dữ liệu x của bài toán P_1 thành một dữ liệu $f(x)$ của bài toán P_2 , và sao cho câu hỏi của P_1 trên x có trả lời đúng khi và chỉ khi câu hỏi của P_2 trên $f(x)$ cũng có trả lời đúng. Ta nói bài toán P_1 *qui dẫn được* về bài toán P_2 *trong thời gian đa thức* , và ký hiệu $P_1 \propto P_2$, nếu có thuật toán f với độ phức tạp thời gian đa thức qui dẫn bài toán P_1 về bài toán P_2 .Ta dễ thấy rằng, nếu $P_1 \propto P_2$ và $P_2 \in \mathcal{P}$, thì cũng có $P_1 \in \mathcal{P}$.

Một lớp quan trọng các bài toán đã được nghiên cứu nhiều là lớp các bài toán khá thường gặp trong thực tế nhưng cho đến nay

chưa có khả năng nào chứng tỏ là chúng có thể giải được trong thời gian đa thức. Đó là lớp các bài toán $\mathcal{NP}$-*đầy đủ* mà ta sẽ định nghĩa sau đây:

Cùng với khái niệm thuật toán tất định thông thường (có thể mô tả chính xác chẳng hạn bởi máy Turing tất định), ta xét khái niệm thuật toán *không đơn định* với một ít thay đổi như sau: nếu đối với máy Turing tất định, khi máy đang ở một trạng thái q và đang đọc một ký tự a thì cặp (q,a) xác định duy nhất một hành động kế tiếp của máy, còn đối với máy Turing không đơn định, ta qui ước rằng (q,a) xác định không phải duy nhất mà là một tập hữu hạn các hành động kế tiếp; máy *có thể* thực hiện trong bước kế tiếp một trong các hành động đó. Như vậy, đối với một dữ liệu vào x, một thuật toán không đơn định (được xác định chẳng hạn bởi một máy Turing không đơn định) không phải chỉ có một tiến trình tính toán duy nhất, mà có thể có một số hữu hạn những tiến trình tính toán khác nhau. Ta nói thuật toán không đơn định $\mathcal{A}$ *chấp nhận* dữ liệu x, nếu với dữ liệu vào x thuật toán $\mathcal{A}$ có ít nhất một tiến trình tính toán kết thúc ở trạng thái chấp nhận (tức với kết quả *đúng*). Một bài toán P được gọi là *giải được bởi thuật toán không đơn định trong thời gian đa thức* nếu có một thuật toán không đơn định $\mathcal{A}$ và một đa thức $p(n)$ sao cho với mọi dữ liệu vào x có độ dài n, $x \in P$ (tức câu hỏi của P có trả lời đúng trên x) khi và chỉ khi thuật toán $\mathcal{A}$ chấp nhận x bởi một tiến trình tính toán có độ phức tạp thời gian $\leq p(n)$. Ta ký hiệu lớp tất cả các bài toán giải được bởi thuật toán không đơn định trong thời gian đa thức là $\mathcal{NP}$.

Người ta đã chứng tỏ được rằng tất cả những bài toán trong các thí dụ kể trên và rất nhiều các bài toán tổ hợp thường gặp khác đều thuộc lớp $\mathcal{NP}$, dù rằng hầu hết chúng đều chưa được chứng tỏ là thuộc $\mathcal{P}$. Một bài toán P được gọi là $\mathcal{NP}$-*đầy đủ*, nếu $P \in \mathcal{NP}$ và với mọi $Q \in \mathcal{NP}$ đều có $Q \propto P$.

Lớp $\mathcal{NP}$ có một số tính chất sau đây:

1) $\mathcal{P} \subseteq \mathcal{NP}$,

2) Nếu $P_1 \propto P_2$ và $P_2 \in \mathcal{NP}$, thì $P_1 \in \mathcal{NP}$.

3) Nếu $P_1, P_2 \in \mathcal{NP}$, $P_1 \propto P_2$, và P_1 là $\mathcal{NP}$-đầy đủ, thì P_2 cũng là $\mathcal{NP}$-đầy đủ.

4) Nếu có P sao cho P là $\mathcal{NP}$-đầy đủ và $P \in \mathcal{P}$, thì $\mathcal{P} = \mathcal{NP}$.

Từ các tính chất đó ta có thể xem rằng trong lớp $\mathcal{NP}$, $\mathcal{P}$ là lớp con các bài toán " dễ " nhất, còn các bài toán $\mathcal{NP}$-đầy đủ là các bài toán " khó " nhất; nếu có ít nhất một bài toán $\mathcal{NP}$-đầy đủ được chứng minh là thuộc $\mathcal{P}$, thì lập tức suy ra $\mathcal{P} = \mathcal{NP}$, dù rằng cho đến nay tuy đã có rất nhiều cố gắng nhưng toán học vẫn chưa tìm được con đường nào hy vọng đi đến giải quyết vấn đề [$\mathcal{P} = \mathcal{NP}$?], thậm chí vấn đề đó còn được xem là một trong 7 vấn đề khó nhất của toán học trong thiên niên kỷ mới!

2.3.3. Hàm một phía và cửa sập một phía.

Khái niệm *độ phức tạp tính toán* cung cấp cho ta một cách tiếp cận mới đối với vấn đề *bí mật* trong các vấn đề bảo mật và an toàn thông tin. Dù ngày nay ta đã có những máy tính điện tử có tốc độ tính toán cỡ hàng tỷ phép tính một giây đồng hồ, nhưng với những thuật toán có độ phức tạp tính toán cỡ $f(n) = 2^n$, thì ngay với những dữ liệu có độ dài khoảng $n = 1000$, việc thực hiện các thuật toán đó đã không thể xem là khả thi, vì nó đòi hỏi thực hiện khoảng 10^{300} phép tính! Như vậy, một giải pháp mật mã chẳng hạn có thể xem là có độ bảo mật cao, nếu để giải mã cần phải thực hiện một tiến trình tính toán có độ phức tạp rất lớn. Do đó, việc phát hiện và sử dụng các hàm số có độ phức tạp tính toán rất lớn là có ý nghĩa hết sức quan trọng đối với việc xây dựng các giải pháp về mật mã và an toàn thông tin.

Hàm số số học $y = f(x)$ được gọi là *hàm một phía* (one-way function), nếu việc tính thuận từ x ra y là "dễ", nhưng việc tính

ngược từ y tìm lại x là rất "khó", ở đây các tính từ "dễ" và "khó" không có các định nghĩa chính xác mà được hiểu một cách thực hành, ta có thể hiểu chẳng hạn dễ là tính được trong thời gian đa thức (với đa thức bậc thấp), còn khó là không tính được trong thời gian đa thức! Thực tế thì cho đến hiện nay, việc tìm và chứng minh một hàm số nào đó là không tính được trong thời gian đa thức còn là việc rất khó khăn, cho nên "khó" thường khi chỉ được hiểu một cách đơn giản là chưa tìm được thuật toán tính nó trong thời gian đa thức! Với cách hiểu tương đối như vậy về "dễ" và "khó", người ta đã đưa ra một số thí dụ sau đây về các hàm một phía:

Thí dụ 1. Cho p là một số nguyên tố, và α là một phần tử nguyên thuỷ mod$p$. Hàm số $y = \alpha^x \bmod p$ (từ Z_p^* vào Z_p^*) là một hàm một phía, vì hàm ngược của nó, tính từ y tìm x mà ta ký hiệu $x = \log_\alpha(y)$, là một hàm có độ phức tạp tính toán rất lớn.

Thí dụ 2. Cho $n = p.q$ là tích của hai số nguyên tố lớn. Hàm số $y = x^2 \bmod n$ (từ Z_n vào Z_n) cũng được xem là một hàm một phía.

Thí dụ 3. Cho $n = p.q$ là tích của hai số nguyên tố lớn, và a là một số nguyên sao cho gcd(a, $\phi(n)$) =1. Hàm số $y = x^a \bmod n$ (từ Z_n vào Z_n) cũng là một hàm một phía, nếu giả thiết là biết n nhưng không biết p,q.

Hàm $y = f(x)$ được gọi là *hàm cửa sập một phía* (trapdoor one-way function), **nếu việc tính thuận từ *x* ra *y*** là "dễ", việc tính ngược từ y tìm lại x là rất "khó", nhưng có một cửa sập z để với sự trợ giúp của cửa sập z thì việc tính x từ y và z lại trở thành dễ.

Thí dụ 4 (tiếp tục thí dụ 3). Hàm số $y = x^a \bmod n$ khi biết p và q là hàm cửa sập một phía. Từ x tính y là dễ, từ y tìm x (nếu chỉ biết n, a) là rất khó, nhưng vì biết p và q nên biết $\phi(n) = (p-1)(q-1)$, và dùng thuật toán Euclide mở rộng tìm được b sao cho $a.b \equiv 1 \pmod{\phi(n)}$, từ đó dễ tính được $x = y^b \bmod n$. Ở đây, có thể xem b là cửa sập.

2.4. Số nguyên tố. Phân tích thành thừa số. Logarit rời rạc.

Trong tiết này ta sẽ xét ba bài toán có vai trò quan trọng trong lý thuyết mật mã, đó là ba bài toán: thử tính nguyên tố của một số nguyên, phân tích một số nguyên thành tích của các thừa số nguyên tố, và tính logarit rời rạc của một số theo một môđuyn nguyên tố.

2.4.1. Thử tính nguyên tố của một số.

Bài toán đặt ra rất đơn giản: Cho một số nguyên dương n bất kỳ. Hãy thử xem n có là số nguyên tố hay không? Bài toán được đặt ra từ những buổi đầu của số học, và trải qua hơn 2000 năm đến nay vẫn là một bài toán chưa có được những cách giải dễ dàng. Bằng những phương pháp đơn giản như phương pháp sàng Euratosthène, từ rất sớm người ta đã xây dựng được các bảng số nguyên tố đầu tiên, rồi tiếp tục bằng nhiều phương pháp khác tìm thêm được nhiều số nguyên tố lớn. Tuy nhiên, chỉ đến giai đoạn hiện nay của lý thuyết mật mã hiện đại, nhu cầu sử dụng các số nguyên tố và thử tính nguyên tố của các số mới trở thành một nhu cầu to lớn và phổ biến, đòi hỏi nhiều phương pháp mới có hiệu quả hơn. Trong mục này ta sẽ lược qua vài tính chất của số nguyên tố, sau đó giới thiệu môt vài phương pháp thử tính nguyên tố của một số nguyên bất kỳ. Ta đã biết một số tính chất sau đây của các số nguyên tố và hợp số (trong các phát biểu dưới đây, ký hiệu $|A|$ chỉ cho số phần tử của tập hợp A):

1. <u>Tiêu chuẩn Euler-Solovay-Strassen:</u>

a) Nếu n là số nguyên tố, thì với mọi số nguyên dương a [n -1:

$$\left(\frac{a}{n}\right) \equiv a^{(n-1)/2} \bmod n .$$

b) Nếu n là hợp số , thì

$$\left|\left\{a:1\le a\le n-1,\left(\frac{a}{n}\right)\equiv a^{(n-1)/2}\bmod n\right\}\right|\le\frac{n-1}{2}.$$

2. Tiêu chuẩn Solovay-Strassen-Lehmann :

a) Nếu n là số nguyên tố, thì với mọi số nguyên dương a [n -1:

$$a^{(n-1)/2}\equiv\pm1(\bmod n).$$

b) Nếu n là hợp số, thì

$$\left|\left\{a:1\le a\le n-1,a^{(n-1)/2}\equiv\pm1\bmod n\right\}\right|\le\frac{n-1}{2}.$$

3. Tiêu chuẩn Miller-Rabin :

a) Cho n là số nguyên lẻ, ta viết $n - 1 = 2^e .u$, với u là số lẻ. Nếu n là số nguyên tố, thì với mọi số nguyên dương a [n -1:

$$(a^u\equiv1\bmod n)\vee\exists k<e(a^{2^k.u}\equiv-1\bmod n).$$

b) Nếu n là hợp số, thì

$$\left|\left\{a:1\le a\le n-1,(a^u\equiv1\bmod n)\vee\exists k<e(a^{2^k.u}\equiv-1\bmod n)\right\}\right|\le\frac{n-1}{4}.$$

Các tiêu chuẩn kể trên là cơ sở để ta xây dựng các thuật toán xác suất kiểu Monte-Carlo thử tính nguyên tố (hay hợp số) của các số nguyên. Chẳng hạn, từ tiêu chuẩn thứ nhất ta có *thuật toán Euler-Solovay-Strassen* sau đây:

Dữ liệu vào: số nguyên dương n và t số ngẫu nhiên $a_1,...,a_t$ (1[a_i[n -1),

1. **for** $i = 1$ **to** t do
2. **if** $\left(\frac{a_i}{n}\right)\equiv a_i^{(n-1)/2}\bmod n$, **then**
3. **answer** "n là số nguyên tố "
4. else
5. **answer** "n là hợp số " and **quit**

Thuật toán này nếu cho trả lời "n là hợp số" thì đúng n là hợp số, nhưng nếu nó cho trả lời "n là số nguyên tố" thì trả lời đó có thể sai với một xác suất ε nào đó. Như vậy, thuật toán đó là một thuật toán xác suất Monte-Carlo *thiên về có* nếu xem nó là thuật toán thử tính *là hợp số*; còn nó là một thuật toán xác suất *thiên về không* nếu xem nó là thuật toán thử tính *nguyên tố* của các số nguyên.

Tương tự như vậy, dựa vào các tiêu chuẩn 2 và 3 ta cũng có thể xây dựng các thuật toán xác suất Solovay-Strassen-Lehmann và Miller-Rabin kiểu Monte-Carlo để thử tính nguyên tố (hay là hợp số) của các số nguyên. Hai thuật toán đó chỉ khác thuật toán Euler-Solovay-Strassen kể trên ở chỗ công thức trong hàng lệnh thứ 2 cần được thay tương ứng bởi

$$a^{(n-1)/2} \equiv \pm 1 \bmod n$$

hay

$$(a^u \equiv 1 \bmod n) \vee \exists k < e (a^{2^k.u} \equiv -1 \bmod n)$$

trong đó u và e được xác định bởi: $n - 1 = 2^e.u$, u là số lẻ.

Xác suất sai lầm ε khi nhận được kết quả "n là số nguyên tố" trong các thuật toán đó được tính như sau: Giả sử n là một số lẻ trong khoảng N và $2N$, tức $N < n < 2N$. Gọi A là sự kiện "n là hợp số", và B là sự kiện "thuật toán cho kết quả trả lời n là số nguyên tố". Ta phải tính xác suất $\varepsilon = p(A|B)$. Theo tính chất b) của tiêu chuẩn Euler-Solovay-Strassen, nếu n là hợp số, thì sự kiện

$$\left(\frac{a}{n}\right) \equiv a^{(n-1)/2} \bmod n$$

đối với mỗi a ngẫu nhiên (1[a[n - 1) có xác suất [1/2, vì vậy ta có

$$p(B/A) \leq \frac{1}{2^t}.$$

Theo công thức Bayes ta có

$$p(A/B)=\frac{p(B/A).p(A)}{p(B)}=\frac{p(B/A).p(A)}{p(B/A).p(A)+p(B/\bar{A}).p(\bar{A})}.$$

Theo định lý về số nguyên tố, số các số nguyên tố giữa N và $2N$ xấp xỉ $\frac{N}{\ln N}\approx\frac{n}{\ln n}$, số các số lẻ là $\frac{N}{2}\approx\frac{n}{2}$, do đó $p(\bar{A})\approx\frac{2}{\ln n}$, và $p(A)\approx 1-\frac{2}{\ln n}$. Dĩ nhiên ta có $p(B/\bar{A})=1$. Thay các giá trị đó vào công thức trên, ta được

$$p(A/B)\le\frac{2^{-t}(1-\frac{2}{\ln n})}{2^{-t}(1-\frac{2}{\ln n})+\frac{2}{\ln n}}=\frac{\ln n-2}{\ln n-2+2^{t+1}}. \qquad (5)$$

Đánh giá đó cũng đúng đối với thuật toán Solovay-Strassen-Lehmann, còn đối với thuật toán Miler-Rabin thì ta được một đánh giá tốt hơn, cụ thể là

$$p(A/B)=\frac{\ln n-2}{\ln n-2+2^{2t+1}}. \qquad (6)$$

Chú ý rằng khi t=50 thì đại lượng ở vế phải của (5) $\approx 10^{-13}$, và vế phải của (6) $\approx 10^{-28}$; do đó nếu chọn cho dữ liệu vào thêm khoảng 50 số ngẫu nhiên a_i thì các thuật toán Euler-Solovay-Strassen và Solovay-Strassen-Lehmann sẽ thử cho ta một số là nguyên tố với xác suất sai lầm [10^{-13} và thuật toán Miller-Rabin với xác suất sai lầm [10^{-28} !

Ta có thể tính được rằng độ phức tạp tính toán về thời gian của các thuật toán xác suất kể trên là vào cỡ đa thức của $\log n$, tức là đa thức của độ đài biểu diễn của dữ liệu vào (là số n), tuy nhiên các thuật toán đó chỉ cho ta thử tính nguyên tố của một số với một xác suất sai lầm ε nào đó, dù ε là rất bé. Trong nhiều ứng dụng, ta muốn có được những số nguyên tố với độ chắc chắn 100% là số nguyên tố. Do đó, dù đã có các thuật toán xác suất như trên, người ta vẫn không ngừng tìm kiếm những thuật toán tất định để thử tính nguyên tố với độ chính xác tuyệt đối. Trong mấy chục năm gần đây,

một số thuật toán đã được đề xuất, trong đó có những thuật toán đặc sắc như thuật toán thử tổng Jacobi, được phát hiện bởi Adleman, Pomerance và Rumely, sau đó được đơn giản hoá bởi Cohen và Lenstra; thuật toán thử bằng đường cong elliptic, được đề xuất bởi Goldwasser, Kilian, Adleman và Huang, được tiếp tục hoàn thiện bởi Atkin và Morain, các thuật toán này đã được dùng để tìm nhiều số nguyên tố rất lớn, thí dụ dùng thuật toán Atkin-Morain đã chứng tỏ được số $(2^{3539}+1)/3$ có 1065 chữ số thập phân là số nguyên tố. Gần đây, vào tháng 8/2002, các nhà toán học Ân độ Agrawal, Kayal và Saxena đã đưa ra một thuật toán tất định mới thử tính nguyên tố có độ phức tạp tính toán thời gian đa thức khá đơn giản, thuật toán đó được mô tả như sau:

Thuật toán Agrawal-Kayal-Saxena:

Input: integer $n > 1$

1. if (n is of the form a^b, $b > 1$) ouput COMPOSITE;
2. r=2;
3. while ($r < n$) {
4. if (gcd(n , r)$\neq 1$) ouput COMPOSITE;
5. if (r is prime)
6. let q be the largest prime factor of r - 1;
7. if ($q \geq 4\sqrt{r}\log n$) and ($n^{\frac{r-1}{q}} \neq 1 (\mathrm{mod}\, r)$)
8. break;
9. $r \leftarrow r + 1$;
10. }
11. for $a = 1$ to $2\sqrt{r}\log n$
12. if $((x-a)^n \neq (x^n - a)(\mathrm{mod}\, x^r - 1, n))$ ouput COMPOSITE;
13. output PRIME;

Thuật toán này đã được một số nhà toán học kiểm nghiệm , đánh giá cao và xem là một thuật toán đẹp, có thể dùng cho việc kiểm thử tính nguyên tố của các số nguyên.

Trong thực tiễn xây dựng các giải pháp mật mã, thường có nhu cầu có các số nguyên tố rất lớn. Để tìm được các số như vậy, người ta thường chọn ngẫu nhiên một số rất lớn, rồi dùng trước cho nó một thuật toán xác suất chẳng hạn như thuật toán Miller-Rabin; nếu thuật toán cho ta kết quả "là số nguyên tố" với một xác suất sai ε nào đó, thì sau đó ta dùng tiếp một thuật toán tất định (chẳng hạn như thuật toán trên đây) để bảo đảm chắc chắn 100% rằng số đó là số nguyên tố. Thuật toán Agrawal-Kayal-Saxena trên đây được chứng tỏ là có độ phức tạp thời gian đa thức cỡ $O((\log n)^{12})$ khi thử trên số n ; và nếu số nguyên tố được thử có dạng Sophie Germain, tức dạng $2p+1$, thì độ phức tạp thời gian sẽ chỉ là cỡ $O((\log n)^6)$.

2.4.2. Phân tích thành thừa số nguyên tố.

Bài toán phân tích một số nguyên > 1 thành thừa số nguyên tố cũng được xem là một bài toán khó thường được sử dụng trong lý thuyết mật mã. Biết một số n là hợp số thì việc phân tích n thành thừa số mới là có nghĩa; do đó thường khi để giải bài toán phân tích n thành thừa số, ta thử trước n có là hợp số hay không (chẳng hạn bằng một trong các thuật toán ở mục trước); và bài toán phân tích n thành thừa số có thể dẫn về bài toán *tìm một ước số của n,* vì khi biết một ước số d của n thì tiến trình phân tích n được tiếp tục thực hiện bằng cách phân tích d và n/d.

Bài toán phân tích thành thừa số, hay bài toán tìm ước số của một số nguyên cho trước, đã được nghiên cứu nhiều, nhưng cũng chưa có một thuật toán hiệu quả nào để giải nó trong trường hợp tổng quát; do đó người ta có khuynh hướng tìm thuật toán giải nó trong những trường hợp đặc biệt, chẳng hạn khi n có một ước số nguyên tố p với

$p-1$ là B-mịn với một cận $B>0$ nào đó, hoặc khi n là số Blum, tức là số có dạng tích của hai số nguyên tố lớn nào đó ($n=p.q$).

Ta xét trường hợp thứ nhất với (p-1)-thuật tóan Pollard như sau: Một số nguyên n được gọi là ***B-mịn***, nếu tất cả các ước số nguyên tố của nó đều $\leq B$. Ý chính chứa trong (p-1)- thuật toán Pollard là như sau: Giả sử n là B-mịn. Ký hiệu Q là bội chung bé nhất của tất cả các luỹ thừa của các số nguyên tố $\leq B$ mà bản thân chúng $\leq n$. Nếu $q^l \leq n$ thì $l \ln q \leq \ln n$, tức $l \leq \left\lfloor \frac{\ln n}{\ln q} \right\rfloor$ ($\lfloor x \rfloor$ là số nguyên bé nhất lớn hơn x).

Ta có

$$Q = \prod_{q \leq B} q^{\lfloor \ln n / \ln q \rfloor},$$

trong đó tích lấy theo tất cả các số nguyên tố khác nhau $q \leq B$. Nếu p là một thừa số nguyên tố của n sao cho p-1 là B-mịn, thì p-1$|Q$, và do đó với mọi a bất kỳ thỏa mãn gcd(a,p) = 1, theo định lý Fermat ta có

$a^Q \equiv 1 \pmod{p}$. Vì vậy, nếu lấy d =gcd(a^Q - 1, n) thì $p \mid d$. Nếu $d = n$ thì coi như thuật toán không cho ta điều mong muốn, tuy nhiên điều đó chắc không xẩy ra nếu n có ít nhất hai thừa số nguyên tố khác nhau. Từ những lập luận đó ta có:

(p - 1)-thuật toán Pollard phân tích thành thừa số:

INPUT: một hợp số n không phải là luỹ thừa của một số nguyên tố.

OUTPUT: một thừa số không tầm thường của n.

1. Chọn một cận cho độ mịn B.
2. Chọn ngẫu nhiên một số nguyên a, $2 \leq a \leq n$ - 1, và tính d = gcd(a,n). Nếu $d \geq 2$ thì cho ra kết quả (d).
3. Với mỗi số nguyên tố $q \leq B$ thực hiện:

   3.1 Tính $l = \left\lfloor \frac{\ln n}{\ln q} \right\rfloor$.

   3.2 Tính $a \leftarrow a^{q^l} \bmod n$.

4. Tính $d = \gcd(a-1, n)$.
5. Nếu $1 < d < n$ thì cho ra kết quả (d). Nếu ngược lại thì thuật toán coi như không có kết quả.

Thí dụ: Dùng thuật toán cho số $n = 19048567$. Ta chọn $B = 19$, và $a = 3$, và tính được $\gcd(3, n) = 1$. Chuyển sang thực hiện bước 3 ta được bảng sau đây (mỗi hàng ứng với một giá trị của q) :

q	l	a
2	24	2293244
3	15	13555889
5	10	16937223
7	8	15214586
11	6	9685355
13	6	13271154
17	5	11406961
19	5	554506

Sau đó ta tính $d = \gcd(554506-1, 19048567) = 5281$. Vậy ta được một thừa số $p = 5281$, và do đó một thừa số nữa là $q = n/p = 3607$. Cả hai thừa số đó đều là nguyên tố.

Chú ý rằng ở đây $p-1 = 5280 = 2^5.3.5.11$, có tất cả các ước số nguyên tố đều ≤ 19, do đó chắc chắn thuật toán sẽ kết thúc có kết quả. Thuật toán sẽ kết thúc không có kết quả khi độ mịn B được chọn quá bé để không một thừa số nguyên tố p nào của n mà $p-1$ chỉ chứa các ước số nguyên tố $\leq B$. Như vậy, có thể xem ($p-1$)-thuật toán Pollard phân tích n thành thừa số nguyên tố là có hiệu quả đối với những số nguyên n là B-mịn, người ta tính được thời gian cần để thực hiện thuật toán đó là cỡ $O(B \ln n / \ln B)$ phép nhân theo môđuyn.

Bây giờ ta xét trường hợp các số nguyên Blum, tức là các số có dạng $n = p.q$, tích của hai số nguyên tố lớn. Trước hết ta chú ý rằng

nếu ta biết hai số nguyên khác nhau x và y sao cho $x^2 \equiv y^2 \pmod{n}$ thì ta dễ tìm được một thừa số của n. Thực vậy, từ $x^2 \equiv y^2 \pmod{n}$ ta có $x^2 - y^2 = (x+y)(x-y)$ chia hết cho n, do n không là ước số của $x + y$ hoặc $x - y$, nên gcd($x - y$, n) phải là một ước số của n, tức bằng p hoặc q.

Ta biết nếu $n = p.q$ là số Blum, thì phương trình đồng dư

$$x^2 \equiv a^2 \pmod{n}$$

có 4 nghiệm, hai nghiệm tầm thường là $x = a$ và $x = -a$. Hai nghiệm không tầm thường khác là $\pm b$, chúng là nghiệm của hai hệ phương trình đồng dư bậc nhất sau đây:

$$\begin{cases} x \equiv a \pmod{p} \\ x \equiv -a \pmod{q} \end{cases} \qquad \begin{cases} x \equiv -a \pmod{p} \\ x \equiv a \pmod{q} \end{cases}$$

Bằng lập luận như trên, ta thấy rằng nếu n là số Blum, a là một số nguyên tố với n, và ta biết một nghiệm không tầm thường của phương trình $x^2 \equiv a^2 \pmod{n}$, tức biết một $x \neq \pm a$ sao cho $x^2 \equiv a^2 \pmod{n}$ thì gcd($x - a$, n) sẽ là một ước số của n. Những điều trên đây là căn cứ cho một số phương pháp tìm ước số nguyên tố của một số nguyên dạng Blum; ý chung của các phương pháp đó là dẫn về việc tìm một nghiệm không tầm thường của một phương trình dạng $x^2 \equiv a^2 \pmod{n}$, chẳng hạn như phương trình $x^2 \equiv 1 \pmod{n}$.

Một trường hợp khá lý thú trong lý thuyết mật mã là khi ta biết hai số a, b là nghịch đảo của nhau theo modϕ (n) (nhưng không biết ϕ (n)), và tìm một phân tích thành thừa số của n. Bài toán được đặt ra cụ thể là: Biết n có dạng B**lum**, biết a và b sao cho $ab \equiv 1 \pmod{\phi(n)}$. Hãy tìm một ước số nguyên tố của n, hay tìm một nghiệm không tầm thường của phương trình $x^2 \equiv 1 \pmod{n}$. Ta giả thiết $ab - 1 = 2^s.r$ với r là số lẻ. Ta phát triển một thuật toán xác suất kiểu Las Vegas như sau: Ta chọn một số ngẫu nhiên v ($1 \leq v \leq n - 1$). Nếu may mắn v là bội số của p hay q, thì ta được ngay một ước số của n là gcd(v,n). Nếu v *nguyên tố với* n, thì ta tính các bình phương liên tiếp kể từ v^r, được $v^r, v^{2r}, v^{4r}, \ldots$ cho đến khi được $v^{2^t.r} \equiv 1 \pmod{n}$ với một t nào

đó. Số t như vậy bao giờ cũng đạt được, vì có $2^s.\ r \equiv 0\ (\mathrm{mod}\phi(n))$ nên có $v^{2^s.r} \equiv 1(\mathrm{mod}\, n)$. Như vậy, ta đã tìm được một số $x = v^{2^{t-1}.r}$ sao cho $x^2 \equiv 1(\mathrm{mod}\, n)$. Tất nhiên có $x \neq 1\ \mathrm{mod}n$. Nếu cũng có $x \neq$ -1 (mod$n$) thì x là nghiệm không tầm thường của $x^2 \equiv 1(\mathrm{mod}\, n)$, từ đó ta có thể tìm ước số của n . Nếu không thì thuật toán coi như thất bại, cho ta kết quả *không đúng*. Người ta có thể ước lượng xác suất cho kết quả *không đúng* với một lần thử với một số v là < 1/2, do đó nếu ta thiết kế thuật toán với m số ngẫu nhiên $v_1,...v_m$, thì sẽ có thể đạt được xác suất cho kết quả *không đúng* là $< 1/2^m$!

2.4.3. Tính logarit rời rạc theo môđuyn nguyên tố.

Cho p là một số nguyên tố, và α là một phần tử nguyên thuỷ theo mod$p$, tức là phần tử nguyên thuỷ của nhóm Z_p^*. *Bài toán tính logarit rời rạc* theo mod$p$ là bài toán tìm, với mỗi số $\beta \in Z_p^*$,một số a ($1 \leq a \leq p$ -1) sao cho $\beta = \alpha^a \bmod p$, tức là $a = \log_\alpha \beta$ (mod$p$ -1). Một thuật toán tầm thường để giải bài toán này là thuật toán *duyệt toàn bộ* các số a từ 1 đến p - 1, cho đến khi tìm được a thoả mãn $\beta = \alpha^a$ mod$p$. Tất nhiên, thuật toán này là không hiệu quả nếu p là số nguyên tố rất lớn. Một biến dạng của thuật toán đó với ít nhiều hiệu quả hơn là *thuật toán Shanks* sau đây:

Đặt $m = \left\lceil \sqrt{p-1} \right\rceil$. Ta tìm a dưới dạng $a = mj + i, 0 \leq j, i \leq m-1$. Rõ ràng $\beta = \alpha^a \bmod p$ khi và chỉ khi $\alpha^{mj} \equiv \beta\alpha^i (\mathrm{mod}\, p)$. Ta lập hai danh sách gồm các cặp (j, α^{mj}) và các cặp $(i, \beta\alpha^{-i})$ với j và i chạy từ 0 đến m - 1. Khi phát hiện ra có hai cặp từ hai danh sách đó có hai phần tử thứ hai bằng nhau là ta được kết quả $a = mj + i$, đó chính là giá trị $\log_\alpha \beta$ mà ta cần tìm. Thuật toán Shanks có độ phức tạp cỡ $O(m)$ phép toán nhân và $O(m)$ bộ nhớ (chưa kể $O(m^2)$ phép so sánh).

Một thuật toán khác, *thuật toán Polig-Hellman*, thường được dùng có hiệu quả trong trường hợp p -1 chỉ có các thừa số nguyên tố

bé, có nội dung như sau: Giả thiết rằng p - 1 có dạng phân tích chính tắc là

$$p-1=\prod_{i=1}^{k} p_i^{c_i}.$$

Để tìm $a = \log_\alpha \beta \pmod{p\text{-}1}$, ta tìm các số a_i sao cho $a_i \equiv a \bmod p_i^{c_i}$ với i = 1,...,k. Sau khi tìm được các a_i như vậy, thì hệ phương trình $x \equiv a_i \bmod p_i^{c_i}\ (i=1,...,k)$,được giải theo định lý số dư Trung quốc, sẽ cho ta lời giải $x \equiv a \pmod{p-1}$ cần tìm. Vậy, vấn đề là xác định các $a_i \bmod p_i^{c_i}\ (i=1,...,k)$. Vấn đề này được phát biểu lại như sau: Giả sử q là một ước số nguyên tố của p - 1, và $q^c \mid p$ - 1 nhưng không còn $q^{c+1} \mid p$ - 1 . Ta cần tìm $x = a \bmod q^c$. Ta biểu diễn x dưới dạng số q - phân như sau:

$$x=\sum_{i=0}^{c-1} x_i q_i \ (0 \le x_i \le q-1).$$

Vì $x = a \bmod q^c$ nên a viết được dưới dạng $a = x + q^c . s$, và vì $\alpha^{p-1} \equiv 1 \pmod p$, nên ta có

$$\beta^{\frac{p-1}{q}} \equiv \alpha^{a\frac{p-1}{q}} \equiv (\alpha^{p-1})^{\frac{a}{q}} \equiv \alpha^{\frac{(p-1)x_0}{q}} \pmod p.$$

Ta đặt $\gamma = \alpha^{(p-1)/q}$, và tính lần lượt $\gamma^0, \gamma^1, \gamma^2, \ldots$, đồng thời so sánh với $\beta^{(p-1)/q} \bmod p$, ta sẽ tìm được i sao cho $\gamma^i \equiv \beta^{(p-1)/q} \bmod p$. Ta lấy số i đó là x_0, tức x_0 = i. Nếu c = 1 thì x = x_0 , ta tìm xong x. Nếu c >1 thì bằng cách đặt $\beta' = \beta\alpha^{-x_0}$ và $x' = \log_\alpha \beta' \bmod q^c$ ta dễ thấy rằng

$$x'=\sum_{i=1}^{c-1} x_i\, q_i\ .$$

Từ đó ta suy ra

$$\beta'^{(p-1)/q^2} \equiv \alpha^{(p-1)x_1/q} \bmod p.$$

Tương tự như ở bước trên, tính lần lượt $\gamma^0, \gamma^1, \gamma^2, \ldots$, đồng thời so sánh với $\beta'^{(p-1)/q^2}$, ta sẽ tìm được x_1.

Cứ làm như vậy, ta sẽ tìm được dần tất cả các giá trị x_i với i = 0,1,...,c-1, tức là tính được x. Sau khi tìm được tất cả các giá trị x ứng với mọi ước số nguyên tố q của p, thì theo một nhận xét ở trên, chỉ cần giải tiếp một hệ phương trình đồng dư bậc nhất theo các môđuyn từng cặp nguyên tố với nhau (bằng phương pháp số dư Trung quốc), ta sẽ tìm được số a cần tìm, $a = \log_{\alpha}\beta$ theo mod$p$.

Thí dụ: Cho p = 29 và α = 2. Hãy tính $a = \log_2 18$ theo mod29. Ta có p - 1 = 28 = $2^2 . 7^1$. Theo thuật toán Polig-Hellman, ta tìm lần lượt a mod 4 và a mod 7. Theo các bước tính toán như mô tả ở trên, ta sẽ tìm được a mod 4 = 3 và a mod 7 =4 .Từ đó giải hệ phương trình

$$\begin{cases} x \equiv 3 (\text{mod}\, 4), \\ x \equiv 4 (\text{mod}\, 7) \end{cases}$$

ta được nghiệm $x \equiv 11$ (mod28), tức được $11 = \log_2 18$ theo mod29. Thuật toán Polig-Hellman cho ta một cách tính logarit rời rạc khá hiệu quả, nhưng chỉ khi p-1 chỉ có các thừa số nguyên tố bé. Vì vậy, nếu p-1 có ít nhất một thừa số nguyên tố lớn thì thuật toán đó khó được thực hiện có hiệu quả, tức trong trường hợp đó bài toán tính logarit rời rạc theo mod$p$ vẫn là một bài toán khó. Một lớp các số nguyên tố p mà p - 1 có ít nhất một ước số nguyên tố lớn là lớp các số nguyên tố dạng $p = 2q + 1$, trong đó q là nguyên tố. Những số nguyên tố dạng đó được gọi là số nguyên tố Sophie Germain, có vai trò quan trọng trong việc xây dựng một lớp khá thông dụng các hệ mật mã có khoá công khai.

Người ta cũng đã nghiên cứu phát triển nhiều thuật toán khác, cả thuật toán tất định, cả thuật toán xác suất, để tính logarit rời rạc, nhưng chưa có thuật toán nào được chứng tỏ là có độ phức tạp tính toán với thời gian đa thức.

CHƯƠNG III

Các hệ mật mã khóa đối xứng

3.1. Một số hệ mật mã cổ điển.

Trong chương này ta sẽ giới thiệu một số hệ mật mã có khóa đối xứng, tức là những hệ mật mã mà khóa lập mật mã và khóa giải mật mã là trũng nhau, và vì vậy khóa mật mã chung đó phải được giữ bí mật, chỉ riêng hai đối tác (người lập mật mã để gửi đi và ngườn nhận mật mã gửi đến) được biết mà thôi. Trong suốt một thời kỳ lịch sử dài từ thời cổ đại cho đến vài ba thập niên gần đây, các phương pháp mật mã được sử dụng trong thực tế đều là mật mã khoá đối xứng, từ hệ mật mã Ceasar đã được dùng hơn nghìn năm trước cho đến các hệ mật mã được sử dụng với sự trợ giúp của kỹ thuật máy tính hiện đại trong thời gian gần đây. Trước hết ta hãy bắt đầu với một số hệ mật mã cổ điển.

3.1.1. Mã chuyển dịch (shift cipher)

Các hệ mật mã dùng phép chuyển dịch nói trong mục này cũng như nhiều hệ mật mã tiếp sau đều có bảng ký tự bản rõ và bảng ký tự bản mã là bảng ký tự của ngôn ngữ viết thông thường. Vì bảng ký tự tiếng Việt có dùng nhiều dấu phụ làm cho cách xác định ký tự khó thống nhất, nên trong tài liệu này ta sẽ lấy bảng ký tự tiếng Anh để minh hoạ, bảng ký tự này gồm có 26 ký tự, được đánh số từ 0 đến 25 như trình bày ở tiết 1.2.1, ta có thể đồng nhất nó với tập $\boldsymbol{Z}_{26}$. Như vậy, *sơ đồ các hệ mật mã chuyển dịch* được định nghĩa như sau:

$$\mathcal{S} = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D}),$$

trong đó $\mathcal{P} = \mathcal{C} = \mathcal{K} = Z_{26}$, các ánh xạ $\mathcal{E}$ và $\mathcal{D}$ được cho bởi:

với mọi $K, x, y \in Z_{26}$: $\mathcal{E}(K, x) = x + K \bmod 26$,

$\mathcal{D}(K, y) = y - K \bmod 26$.

Các hệ mật mã được xác định như vậy là đúng đắn, vì với mọi $K, x, y \in Z_{26}$ ta đều có:

$$d_K(e_K(x)) = (x+K) - K \bmod 26 = x.$$

Các hệ mật mã chuyển dịch đã được sử dụng từ rất sớm, theo truyền thuyết, hệ mã đó với K=3 đã được dùng bởi J. Caesar từ thời đế quốc La mã, và được gọi là *hệ mã Caesar*.

Thí dụ: Cho bản rõ *hengapnhauvaochieuthubay*, chuyển dãy ký tự đó thành dãy số tương ứng ta được:

x = 7 4 13 6 0 15 13 7 0 20 21 0 14 2 7 8 4 20 19 7 20 1 0 24.

Nếu dùng thuật toán lập mật mã với khoá K = 13, ta được bản mã là:

y = 20 17 0 19 13 2 0 20 13 7 8 13 1 15 20 21 17 7 6 20 7 14 13 11,

chuyển dưới dạng ký tự thông thường ta được bản mật mã là:

uratncaunhinbpuv rhguhonl .

Để giải bản mật mã đó, ta chỉ cần chuyển nó lại dưới dạng số (để được dãy y), rồi thực hiện thuật toán giải mã, tức trừ từng số hạng với 13 (theo môđuyn 26), được lại dãy x, chuyển thành dãy ký tự là được bản rõ ban đầu.

Các hệ mật mã chuyển dịch tuy dễ sử dụng, nhưng việc thám mã cũng khá dễ dàng, số các khoá có thể có là 26; nhận được một bản mã, người thám mã chỉ cần thử dùng lần lượt tối đa là 26 khoá đó để giải mã, ắt sẽ phát hiện ra được khoá đã dùng và cả bản rõ!

3.1.2. Mã thay thế (substitution cipher).

Sơ đồ các hệ mật mã thay thế được định nghĩa như sau:

$$\mathcal{S} = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D}),$$

trong đó $\mathcal{P} = \mathcal{C} = Z_{26}$, $\mathcal{K}$ là tập hợp tất cả các phép hoán vị trên Z_{26} các ánh xạ $\mathcal{E}$ và $\mathcal{D}$ được cho bởi:

$$e_\pi(x) = \pi(x),$$
$$d_\pi(y) = \pi^{-1}(y),$$

với mọi $x \in \mathcal{P}$, $y \in \mathcal{C}$, $\pi \in \mathcal{K}$ là một phép hoán vị trên $\boldsymbol{Z}_{26}$.

Ta thường đồng nhất $\boldsymbol{Z}_{26}$ với bảng ký tự tiếng Anh, do đó phép hoán vị trên $\boldsymbol{Z}_{26}$ cũng được hiểu là một phép hoán vị trên tập hợp các ký tự tiếng Anh, thí dụ một phép hoán vị π được cho bởi bảng :

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r
x	n	y	a	h	p	o	g	z	q	w	b	t	s	f	l	r	c

s	t	u	v	w	x	y	z
v	m	u	e	k	j	d	i

Với hệ mật mã thay thế có khoá π, bản rõ

x = hengapnhauvaochieuthubay

sẽ được chuyển thành bản mật mã

y = ghsoxlsgxuexfygzhumgunxd .

Thuật toán giải mã với khoá π, ngược lại sẽ biến *y* thành bản rõ *x*.

Sơ đồ hệ mật mã có số khoá có thể bằng số các phép hoán vị trên tập $\boldsymbol{Z}_{26}$ **, tức là 26! khoá, đó là một số rất lớn (26!> 4.10^{26}). Do đó, việc duyệt lần lượt tất cả các khoá có thể để thám mã là không thực tế, ngay cả dùng máy tính. Tuy vậy, có những phương pháp thám mã khác hiệu quả hơn, làm cho các hệ mật mã thay thế không thể được xem là an toàn.**

3.1.3. Mã apphin.

Sơ đồ các hệ mật mã apphin được định nghĩa như sau:

$$\mathcal{S} = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D}),$$

trong đó $\mathcal{P} = \mathcal{C} = \mathbf{Z}_{26}$, $\mathcal{K} = \{ (a,b) \in \mathbf{Z}_{26} \times \mathbf{Z}_{26} \mid \gcd(a, 26) = 1\}$,

các ánh xạ $\mathcal{E}$ và $\mathcal{D}$ được cho bởi:

$$e_K(x) = ax + b \bmod 26,$$
$$d_K(y) = a^{-1}(y - b) \bmod 26,$$

với mọi $x \in \mathcal{P}$, $y \in \mathcal{C}$, $K = (a, b) \in \mathcal{K}$.

Có điều kiện gcd (a, 26) = 1 là để bảo đảm có phần tử nghịch đảo $a^{-1} \bmod 26$ của a, làm cho thuật toán giải mã d_K luôn thực hiện được. Có tất cả $\phi(26) = 12$ số $a \in \mathbf{Z}_{26}$ nguyên tố với 26, đó là các số

1, 3, 5, 7 ,9, 11, 15, 17, 19, 21, 23, 25,

và các số nghịch đảo theo mod26 tương ứng của chúng là

1, 9, 21, 15, 3, 19, 7, 23, 11, 5, 17, 25.

Thí dụ với bản rõ *hengapnhauvaochieuthubay*, có dãy số tương ứng là:

x = 7 4 13 6 0 15 13 7 0 20 21 0 14 2 7 8 4 20 19 7 20 1 0 24.

Nếu dùng hệ mật mã apphin với khoá K=(5, 6) ta sẽ được bản mật mã

y = 15 0 19 10 6 3 19 15 6 2 7 6 24 16 15 20 0 2 23 15 2 11 6 22,

chuyển sang dòng ký tự tiếng Anh ta được bản mật mã dưới dạng

patkgdtpgchgyqpuacxpclgw .

Vì có 12 số thuộc $\mathbf{Z}_{26}$ nguyên tố với 26, nên số các khoá có thể có (do đó , số các hệ mật mã apphin) là bằng 12x26 =312, một con số không lớn lắm nếu ta sử dụng máy tính để thực hiện việc thám mã bằng cách duyệt lần lượt tất cả các khoá có thể; như vậy, mã apphin cũng không còn được xem là mã an toàn !

3.1.4. Mã Vigenère.

Sơ đồ mật mã này lấy tên của Blaise de Vigenère, sống vào thế kỷ 16. Khác với các hệ mật mã đã kể trước, các hệ mật mã Vigenère không thực hiện trên từng ký tự một, mà được thực hiện trên từng bộ m ký tự (m là số nguyên dương).

Sơ đồ các hệ mật mã Vigenère được định nghĩa như sau:

$$\mathcal{S} = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D}),$$

trong đó $\mathcal{P} = \mathcal{C} = \mathcal{K} = Z_{26}^m$, các ánh xạ $\mathcal{E}$ và $\mathcal{D}$ được cho bởi:

$$e_K(x_1,..., x_m) = (x_1+k_1,...., x_m+k_m) \mod 26$$

$$d_K(y_1,..., y_m) = (y_1-k_1 ,..., y_m-k_m) \mod 26$$

với mọi $x=(x_1,..., x_m) \in \mathcal{P}$, $y=(y_1,..., y_m) \in \mathcal{C}$, $K=(k_1,...,k_m) \in \mathcal{K}$.

Sơ đồ mã Vigenère có thể được xem là mở rộng của sơ đồ mã chuyển dịch, nếu mã chuyển dịch thực hiện việc chuyển dịch từng ký tự một thì mã Vigenère thực hiện đồng thời từng bộ m ký tự liên tiếp. Thí dụ lấy $m = 6$ và $K = (2, 8, 15, 7, 4, 17)$. Để lập mật mã cho bản rõ

hengapnhauvaochieuthubay,

ta cũng chuyển nó thành dãy số và tách thành từng đoạn 6số liên tiếp:

x = 7 4 13 6 0 15| 13 7 0 20 21 0| 14 2 7 8 4 20 |19 7 20 1 0 24.

(nếu độ dài của x không phải là bội số của 6, ta có thể qui ước thêm vào đoạn cuối của x một số phần tử nào đó, chẳng hạn là các số 0, để bao giờ cũng có thể xem là x tách được thành các đoạn có 6 số liên tiếp). Cộng theo mod26 các số trong từng đoạn đó với các số tương ứng trong khoá K ta sẽ được bản mật mã

y = 9 12 2 13 4 6 |15 15 15 1 25 17| 16 10 22 15 8 11| 21 15 9 8 4 15

chuyển sang dãy ký tự ta được bản mã là

jmcnegpppbzrqkwpilvpjiep .

Từ bản mã đó, dùng thuật toán giải mã tương ứng ta lại thu được bản rõ ban đầu.

Tập $\mathcal{K}$ có tất cả là 26^m phần tử, do đó với mỗi m có tất cả là 26^m hệ mật mã Vigenère khác nhau (với m = 6 thì số đó là 308,915,776), duyệt toàn bộ chừng ấy khoá để thám mã bằng tính thủ công thì khó, nhưng nếu dùng máy tính đủ mạnh thì cũng không đến nỗi khó lắm!

3.1.5. Mã Hill.

Sơ đồ mật mã này được đề xuất bởi Lester S. Hill năm 1929. Cũng giống như sơ đồ mã Vigenère, các hệ mã này được thực hiện trên từng bộ m ký tự liên tiếp, điều khác là mỗi ký tự của bản mã được xác định bởi một tổ hợp tuyến tính (trên vành $\boldsymbol{Z_{26}}$) của m ký tự trong bản rõ. Như vậy, khoá sẽ được cho bởi một ma trận cấp m, tức là một phần tử của $K \in Z^{m \times m}$. Để phép biến đổi tuyến tính xác định bởi ma trận K có phép nghịch đảo, bản thân ma trận K cũng phải có ma trận nghịch đảo K^{-1} theo mod26; mà điều kiện cần và đủ để K có nghịch đảo là định thức của nó, ký hiệu det$K$, nguyên tố với 26. Vậy, sơ đồ mật mã Hill được định nghĩa là sơ đồ

$$\mathcal{S} = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D}),$$

trong đó $\mathcal{P} = \mathcal{C} = Z_{26}^m$, $\mathcal{K} = \left\{K \in Z_{26}^{m \times m} : \gcd(\det K, 26) = 1\right\}$,

các ánh xạ $\mathcal{E}$ và $\mathcal{D}$ được cho bởi:

$$e_K(x_1,..., x_m) = (x_1,..., x_m).K \mod 26,$$

$$d_K(y_1,..., y_m) = (y_1,..., y_m).K^{-1} \mod 26$$

với mọi $x=(x_1,..., x_m) \in \mathcal{P}$, $y=(y_1,..., y_m) \in \mathcal{C}$, $K \in \mathcal{K}$.

Thí dụ : Chọn m = 2, và $K = \begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix}$.

Với bộ hai ký tự $x = (x_1, x_2)$ ta có mã $y = (y_1, y_2)$. K được tính bởi:

$$y_1 = 11x_1 + 3x_2 \bmod 26$$
$$y_2 = 8x_1 + 7x_2 \bmod 26 .$$

Ta lấy lại bản rõ *hengapnhauvaochieuthubay*, ta cũng chuyển nó thành dãy số và tách thành từng đoạn 2 số liên tiếp:

x = 7 4 |13 6| 0 15| 13 7 |0 20| 21 0| 14 2 |7 8| 4 20 |19 7| 20 1| 0 24.

Lập mật mã cho từng đoạn hai số liên tiếp, rồi nối ghép lại ta được

y = 11 6|5 16| 19 1|8 21|8 2|23 12|4 22|23 8|0 16|22 19|15 11|20 12.

Và từ đó ta được bản mật mã dưới dạng dãy ký tự là

lgfqtbivicxmewxiaqwtplum .

Chú ý rằng

$$K^{-1} = \begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix}^{-1} (\bmod 26) = \begin{pmatrix} 7 & 18 \\ 23 & 11 \end{pmatrix},$$

và giải mã bằng cách nhân từng đoạn hai số liên tiếp của y với K^{-1} ta sẽ được lại dãy x, và từ đó được lại bản rõ.

Với mỗi số m cho trước, số các khoá có thể có là bằng số các ma trận K có $\det K$ nguyên tố với 26. Ta không có công thức để tính con số đó, tuy biết rằng khi m lớn thì số đó cũng là rất lớn, và tất nhiên việc thám mã bằng cách duyệt lần lượt toàn bộ các hệ mã Hill có cùng số m là không khả thi. Mặc dù vậy, từ lâu người ta cũng đã tìm được những phương pháp thám mã khác đối với hệ mã Hill một cách khá hiệu quả mà ta sẽ giới thiệu trong một phần sau.

3.1.6. Mã hoán vị.

Các hệ mã hoán vị cũng được thực hiện trên từng bộ m ký tự liên tiếp, nhưng bản mật mã chỉ là một hoán vị của các ký tự trong từng bộ m ký tự của bản rõ. Ta ký hiệu $\boldsymbol{S}_m$ là tập hợp tất cả các phép hoán vị của tập hợp { 1,2, ... ,m }. Sơ đồ các phép mã hoán vị được cho bởi

$$\mathcal{S} = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D}) ,$$

trong đó $\mathcal{P} = \mathcal{C} = Z_{26}^m$, $\mathcal{K} = \boldsymbol{S}_m$, các ánh xạ $\mathcal{E}$ và $\mathcal{D}$ được cho bởi:

$$e_K(x_1,..., x_m) = (x_{\pi(1)},..., x_{\pi(m)}),$$

$$d_K(y_1,..., y_m) = (y_{\pi^{-1}(1)},..., y_{\pi^{-1}(m)}),$$

với mọi $x=(x_1,..., x_m) \in \mathcal{P}$, $y=(y_1,..., y_m) \in \mathcal{C}$, $K = \pi \in \boldsymbol{S}_m$, π^{-1} là hoán vị nghịch đảo của π .

Thí dụ: Chọn $m = 6$ và phép hoán vị $\pi \in \boldsymbol{S}_6$ được cho bởi:

i	1	2	3	4	5	6
$\pi(i)$	3	5	1	6	4	2 .

Khi đó phép hoán vị π^{-1} sẽ là

j	1	2	3	4	5	6
$\pi^{-1}(j)$	3	6	1	5	2	4 .

Với bản rõ *hengapnhauvaochieuthubay*, tức cũng là với

x = 7 4 13 6 0 15| 13 7 0 20 21 0| 14 2 7 8 4 20 |19 7 20 1 0 24.

ta sẽ có bản mã tương ứng là:

y = 13 0 7 15 6 4 0 21 13 0 20 7 7 4 14 20 8 2 20 0 19 24 1 7

chuyển thành dãy ký tự là *nahpgeavnauhheouicuatybh* . Dùng cho từng bộ 6 ký tự liên tiếp của bản mật mã này (tức là của y) phép giải mã d_K ta sẽ thu lại được x và bản rõ ban đầu.

Chú ý rằng mã hoán vị là một trường hợp riêng của mã Hill. Thực vậy, cho phép hoán vị π trên $\{1,2,...,m\}$, ta xác định ma trận $K_\pi = (k_{ij})$ với $k_{ij} = 1$ nếu $i = \pi(j)$, và = 0 nếu ngược lại, thì dễ thấy rằng mã Hill với khoá K_π cho cùng một phép mật mã như mã loán vị với khoá π. Với mỗi m cho trước, số các hệ mật mã hoán vị có thể có là $m!$

3.2. Thám mã đối với các hệ mật mã cổ điển.

3.2.1. Một vài nhận xét chung.

Như đã trình bày trong tiết 1.5 chương 1, mục đích của việc thám mã là dựa vào thông tin về bản mật mã có thể thu thập được trên đường truyền tin mà phát hiện lại được bản rõ của thông báo. Vì sơ đồ của hệ mật mã được sử dụng thường khó mà giữ được bí mật, nên ta thường giả thiết thông tin xuất phát của bài toán thám mã là sơ đồ hệ mật mã được sử dụng và bản mật mã của thông báo, nhiệm vụ của thám mã là tìm bản rõ của thông báo đó. Ngoài các thông tin xuất phát đó, tuỳ trường hợp cụ thể, còn có thể có thêm các thông tin bổ sung khác, vì vậy bài toán thám mã được phân thành các loại bài toán khác nhau như: thám mã chỉ dựa vào bản mã, thám mã khi biết cả bản rõ, thám mã khi có bản rõ được chọn, thám mã khi có bản mã được chọn (xem mục 1.5, chương 1).

Trong tiết này ta sẽ trình bày một vài phương pháp thám mã đối với các hệ mật mã cổ điển mô tả trong tiết trước. Và ta cũng giả thiết các bản rõ cũng như bản mã đều được xây dựng trên bảng ký tự tiếng Anh, và hơn nữa các thông báo là các văn bản tiếng Anh. Như vậy, ta luôn có $\mathcal{P} = \mathcal{C} = \boldsymbol{Z}_{26}$ hay Z_{26}^m, và có thêm thông tin là các bản rõ tuân theo các qui tắc từ pháp và cú pháp của ngôn ngữ tiếng Anh. Đây là một căn cứ quan trọng của các phương pháp thám mã đối với các hệ mật mã cổ điển. Tiếc là việc dùng mật mã để truyền đưa thông tin tiếng Việt không để lại cho ta nhiều tư liệu để nghiên cứu, và những nghiên cứu về từ pháp và cú pháp cũng chưa cho ta những qui tắc thống kê xác suất đủ tin cậy, nên trong tài liệu này ta chưa trình bày được trên các thí dụ mật mã bằng ngôn ngữ Việt, ta đành tạm mượn các thí dụ bằng văn bản tiếng Anh để minh hoạ, mong được bạn đọc bổ sung sau. Các kết quả chủ yếu được sử dụng nhiều nhất trong thám mã là các qui tắc thống kê tần suất xuất hiện các ký tự hay các bộ đôi, bộ ba,...ký tự liên tiếp trong các văn bản tiếng Anh. Trên cơ sở phân tích các số liệu thống kê từ một lượng rất lớn các văn bản thư từ, sách vở, báo chí, v.v... người ta đã

thu được những kết quả mà các tác giả Beker và Piper đã tổng hợp lại như sau:

Phân bố xác suất xuất hiện của các ký tự được sắp xếp theo thứ tự: 1. Ký tự *e* có xác suất xuất hiện cao nhất là 0. 127,
2. Các ký tự *t, a, o, i, n, s, h, r* có xác suất từ 0. 060 đến 0. 090,
3. Các ký tự *d , l* có xác suất khoảng 0. 04,
4. Các ký tự *c, u, m,w, f, g, y, p, b* có xác suất từ 0. 015 đến 0.028,
5. Các ký tự *v, k, j, x, q, z* có xác suất dưới 0. 01.
Ba mươi bộ đôi ký tự có xác suất xuất hiện cao nhất là (kể từ cao xuống): *th, he, in, er, an, re, ed, on, es, st, en, at, to, nt, ha, nd, ou, ea, ng, as, or, ti, is, et, it, ar, te, se, hi, of.*
Mười hai bộ ba ký tự có xác suất xuất hiện cao nhất là: *the, ing, and, her, ere, ent, tha, nth, was, eth, for, dth.*

Sau đây là bảng phân bố xác suất của tất cả các ký tự:

A (0) 0.082	B (1) 0.015	C (2) 0.028	D (3) 0.043
E (4) 0.127	F (5) 0. 022	G (6) 0.020	H (7) 0. 061
I (8) 0.070	J (9) 0.002	K (10) 0.008	L (11) 0.040
M (12) 0.024	N (13) 0.067	O (14) 0.075	P (15) 0.019
Q (16) 0.001	R (17) 0.060	S (18) 0.063	T (19) 0.091
U (20) 0.028	V (21) 0.010	W (22) 0.023	X (23) 0.001
Y (24) 0.020	Z (25) 0.001.		

3.2.2. Thám mã đối với mã apphin.

Khoá mã apphin có dạng $K = (a,b)$ với $a, b \in Z_{26}$ và gcd(a,26)=1. Ký tự mã y và ký tự bản rõ x tương ứng có quan hệ

$$y = a.x + b \bmod 26.$$

Như vậy, nếu ta biết hai cặp (x, y) khác nhau là ta có được hai phương trình tuyến tính để từ đó tìm ra giá trị hai ẩn số a,b, tức là tìm ra K.

Thí dụ: Ta có bản mật mã:
fmxvedkaphferbndkrxrsrefmorudsdkdvshvufedkaprkdlyevlrhhrh .
Hãy tìm khoá mật mã và bản rõ tương ứng.

Ta thấy trong bản mật mã nói trên, r xuất hiện 8 lần, d 7 lần, e, k, h mỗi ký tự 5 lần, f, s, v mỗi ký tự 4 lần, v.v...; vậy có thể phán đoán r là mã của e, d là mã của t, khi đó có

$$4a + b = 17 \mod 26,$$
$$19a + b = 3 \mod 26,$$

giải ra được $a = 6$, $b = 19$. Vì gcd(a, 26) = 2 $\neq$ 1, nên (a, b) không thể là khoá được, phán đoán trên không đúng. Ta lại thử chon một phán đoán khác: r là mã của e, h là mã của t. Khi đó có:

4
$$4a + b = 17 \mod 26,$$
$$19a + b = 7 \mod 26,$$

ta giải ra được a = 3, b = 5. Vì (a, 26) = 1 nên K = (3,5) có thể là khóa cần tìm. Khi đó phép lập mật mã là $e_K(x) = 3x + 5 \mod 26$, và phép giải mã tương ứng là $d_K(y) = 9) = 9y - 19 \mod 26$. Dùng phép giải mã đó cho bản mã ta sẽ được (dưới dạng ký tự) bản rõ là:

algorithmsarequitegeneraldefinitionsofarithmeticprocesses .

Ta có thể kết luận khoá đúng là K = (3, 5) và dòng trên là bản rõ cần tìm.

3.2.3. Thám mã đối với mã Vigenère.

Mã Vigenère có thể coi là mã chuyển dịch đối với từng bộ m ký tự. Khoá mã là một bộ $K = (k_1..., k_m)$ gồm m số nguyên mod 26. Việc thám mã gồm hai bước: bước thứ nhất xác định độ dài m, bước thứ hai xác định các số $k_1..., k_m$.

Có hai phương pháp để xác định độ dài m : phép thử Kasiski và phương pháp dùng chỉ số trùng hợp.

Phép thử Kasiski (đề xuất từ 1863). Phép thử dựa vào nhận xét rằng hai đoạn trùng nhau của bản rõ sẽ được mã hoá thành hai đoạn trùng nhau của bản mã, nếu khoảng cách của chúng trong văn bản rõ (kể từ ký tự đầu của đoạn này đến ký tự đầu của đoạn kia) là bội số của m. Mặt khác, nếu trong bản mã, có hai đoạn trùng nhau và có độ dài khá lớn (≥ 3 chẳng hạn) thì rất có khả năng chúng là mã của hai đoạn trùng nhau trong bản rõ. Vì vậy, ta thử tìm một đoạn mã (có ba ký tự trở lên) xuất hiện nhiều lần trong bản mã, tính khoảng cách của các lần xuất hiện đó, chẳng hạn được $d_1, d_2..., d_t$; khi đó ta có thể phán đoán $m = d = \gcd(d_1, d_2..., d_t)$- ước số chung lớn nhất của $d_1, d_2..., d_t$; hoặc m là ước số của d.

Phương pháp dùng chỉ số trùng hợp: (định nghĩa chỉ số trùng hợp do W.Friedman đưa ra năm 1920).

Định nghĩa 3.1. Cho $x = x_1, x_2... x_n$ là một dãy gồm n ký tự. Xác suất của việc hai phần tử của x trùng nhau được gọi là *chỉ số trùng hợp* của x, ký hiệu là $I_C(x)$.

Ký hiệu $f_0, f_1,..., f_{25}$ lần lượt là tần suất xuất hiện của $a, b, ..., z$ trong x, ta có:

$$I_C(x) = \frac{\sum_{i=0}^{25}\binom{f_i}{2}}{\binom{n}{2}} = \frac{\sum_{i=0}^{25} f_i(f_i+1)}{n(n+1)}.$$

Giả sử x là một dãy ký tự (tiếng Anh). Ta có thể hy vọng rằng:

$$I_C(x) \approx \sum_{i=0}^{25} p_i^2 = 0{,}065,$$

trong đó p_i là xác suất của ký tự ứng với số hiệu i cho bởi bảng phân bố xác suất các ký tự (trang 61)

Nếu x là một dãy ký tự hoàn toàn ngẫu nhiên thì ta có:

$$I_C \approx 26.\ (1/26)^2 = 1/26 = 0{,}038.$$

Dựa vào các điều nói trên, ta có phương pháp đoán độ dài m của mã Vigenère như sau: Cho bản mã $y = y_1y_2..., y_n$. Ta viết lại y theo bảng có m ($m \geq 1$) hàng như sau:

$$y = \begin{matrix} y_1\ y_{m+1}\cdots\ y_{tm+1} \\ y_2\ y_{m+2}\cdots\ y_{tm+2} \\ \cdots\cdots\cdots \\ y_m\ y_{em}\cdots\ y_{(tm+1)m} \end{matrix}$$

nghĩa là viết lần lượt theo các cột m ký tự cho đến hết. Ta ký hiệu $\boldsymbol{y}_1$, $\boldsymbol{y}_2$,..., $\boldsymbol{y}_m$ là các xâu ký tự theo m hàng trong bảng đó. Chú ý rằng các ký tự ở mỗi hàng $\boldsymbol{y}_i$ đều thu được từ các ký tự ở văn bản gốc bằng cùng một phép dịch chuyển nếu m đúng là độ dài của khoá, do đó nếu m là độ dài của khoá thì ta có thể hy vọng rằng với mọi i, $1 \leq i \leq m$:

$$I_C(\boldsymbol{y}_i) \approx 0{,}065\ .$$

Để đoán độ dài m, ta lần lượt chia y theo cách trên thành m = 1, 2, 3... hàng, và tính các $I_C(y_i)$ ($1 \leq i \leq m$), cho đến khi nào được một số m mà với mọi i, $1 \leq i \leq m$, đều có $I_C(y_i) \approx 0{,}065$ thì ta có thể chắc m là độ dài của khoá.

Thí dụ: Cho bản mã

*chr*eevoahmaeratbiaxxwtnxbeeophbsbqmqeqerbwrvxuoakxa
osxxweahbwgjmmqmnkgrfvgxwtrzxwiaklxfpskautemndemg
tsxmxbtuiadngmgpsrelxnjelxvrvprtulhdnqwtwdtygbphxtfalj
hasvbfxngll*chr*zbwelekmsjiknbhwrignmgjsglxfeyphagnbieqjt
mrvlcrremndglxrrimgnsnrw*chr*qhaeyevtaqebbipeewevkakoe
wadremxmtbhh*chr*tkdnvrz*chr*clqohpwqaiiwxnrmgwoiifkee.

Dùng phép thử Kasiski, ta nhận thấy rằng *chr* xuất hiện 5 lần, khoảng cách của các lần xuất hiện liên tiếp là 165, 70, 50, 10. Ước số

chung của các số đó là 5. Vậy ta có thể phán đoán độ dài khoá mã là 5.

Dùng phương pháp chỉ số trùng hợp, với m = 1 ta có một chỉ số trùng hợp là 0,045; với m = 2 có hai chỉ số là 0,046 và 0,041; với m = 3 có ba chỉ số là 0,043; 0,050 và 0,047 ; với m = 4 có bốn chỉ số là 0,042; 0,039; 0,046 và 0,043; với m = 5, ta thu được năm chỉ số là 0,063; 0,068; 0,069; 0,061 và 0,072, đều khá gần với 0,065. Vậy có thể phán đoán độ dài khoá là 5. Cả hai phương pháp cho kết quả như nhau.

Bây giờ đến bước thứ hai là xác định các giá trị k_1, k_2,...k_m. Ta cần một khái niệm mới là *chỉ số trùng hợp tương hỗ*, được định nghĩa như sau:

Định nghĩa 3.2. Giả sử $\boldsymbol{x} = x_1x_2...\ x_n$ và $\boldsymbol{y} = y_1y_2...\ y_n$ là hai dãy ký tự có độ dài n và n'. *Chỉ số trùng hợp tương hỗ* của $\boldsymbol{x}$ và $\boldsymbol{y}$, ký hiệu $MI_C(\boldsymbol{x},\boldsymbol{y})$, được định nghĩa là xác suất của việc một phần tử của $\boldsymbol{x}$ trùng với một phần tử của $\boldsymbol{y}$.

Ký hiệu $f_0, f_1,..., f_{25}$ và $f_0', f_1',..., f_{25}'$ là tần suất xuất hiện của a, b,...,z trong $\boldsymbol{x}$ và $\boldsymbol{y}$ tương ứng. Khi đó, ta có:

$$MI_C(\boldsymbol{x},\boldsymbol{y}) = \frac{\sum_{i=0}^{25} f_i \cdot f_i'}{n.n'}.$$

Bây giờ với m đã xác định, ta viết bản mã $\boldsymbol{y}$ lần lượt theo từng cột để được m hàng $y_1...y_m$ như ở phần trên. Ta tìm khoá mã $K = (k_1,k_2...k_m)$.

Giả sử $\boldsymbol{x}$ là bản rõ và $x_1...,x_m$ là các phần bản rõ tương ứng với $y_1...,y_m$. Ta có thể xem phân bố xác suất của các ký tự trên $\boldsymbol{x}$, và cũng trên các $x_1...,\ x_m$ là xấp xỉ với phân bố xác suất của các ký tự trên văn bản tiếng Anh nói chung. Do đó, xác suất của việc một ký tự ngẫu

nhiên của y_i bằng a là p_{-k_i}, bằng b là p_{1-k_i}, v.v... Và ta có thể đánh giá

$$MI_C(y_i, y_j) \approx \sum_{h=0}^{25} p_{h-k_i} \cdot p_{h-k_j} = \sum_{h=0}^{25} p_h \cdot p_{h+k_i-k_j}.$$

Đại lượng đó chỉ phụ thuộc vào $k_i - k_j$, ta gọi là *dịch chuyển tương đối* của y_i và y_j. Ta chú ý rằng biểu thức:

$$\sum_{h=0}^{25} p_h \cdot p_{h+l}$$

có giá trị lớn nhất khi $l = 0$ là 0,065, và có giá trị biến thiên giữa 0,031 và 0,045 với mọi $l > 0$.

Nhận xét rằng y_j phải dịch chuyển $l = k_i - k_j$ bước (hay dịch chuyển l ký tự trong bảng chữ cái) để được y_i, nên nếu ký hiệu y_j^g là dịch chuyển g bước của y_j, thì ta có hy vọng khi tính lần lượt các đại lượng $MI_C(y_i, y_j^g)$ với $0 \le g \le 25$, ta sẽ đạt được một giá trị xấp xỉ 0,065 với $g = l$, và các giá trị khác đều ở khoảng giữa 0,031 và 0,045. Điều đó cho ta một phương pháp để ước lượng các dịch chuyển $k_i - k_j$, tức là được một số phương trình dạng $k_i - k_j = l$, từ đó giúp ta tính ra các giá trị $k_1, k_2, \ldots, k_m$.

Trong thí dụ của bản mã đang xét, ta tính được các giá trị $MI_C(y_i, y_j^g)$ với $1 \le i \le j \le 5$, $0 \le g \le 25$, như trong bảng ở trang sau đây (trong bảng đó, ở bên phải mỗi cặp (i, j) là một ngăn gồm có 26 giá trị của $MI_C(y_i, y_j^g)$ ứng với các giá trị của $g = 0,1,2,\ldots, 25$).

Nhìn bảng đó, ta thấy các giá trị $MI_C(y_i, y_j^g)$ xấp xỉ 0.065 (như được in đậm và gạch dưới ở trong bảng) ứng với các bộ giá trị (i, j, g) lần lượt bằng (1,2,9), (1,5,16), (2,3,13), (2,5,7), (3,5,20) và (4,5,11).

i	j	Giá trị của $MI_C(y_i, y_j^g)$
1	2	.028 .027 .028 .034 .039 .037 .026 .025 .052 **.068** .044 .026 .037 .043 .037 .043 .037 .028 .041 .041 .034 .037 .051 .045 .042 .036
1	3	.039 .033 .040 .034 .028 .053 .048 .033 .029 .056 .050 .045 .039 .040 .036 .037 .032 .027 . 037 .036 .031 .037 .055 .029 .024 .037
1	4	.034 .043 .025 .027 .038 .049 .040 .032 .029 .034 .039 .044 .044 .034 .039 .045 .044 .037 .055 .047 .032 .027 .039 .037 .039 .035
1	5	.043 .033 .028 .046 .043 .044 .039 .031 .026 .030 .036 .040 .041 .024 .019 .048 **.070** .044 .028 .038 .044 .043 .047 .033 .026 .046
2	3	.046 .048 .041 .032 .036 .035 .036 030 .024 .039 .034 .029 .040 **.067** .041 .033 .037 .045 .033 .033 .027 .033 .045 .052 .042 .030
2	4	.046 .034 .043 .044 .034 .031 .040 .045 040 .048 .044 .033 .024 .028 .042 .039 .026 .034 .050 .035 ,032 .040 .056 .043 .028 .028
2	5	.033 .033 .036 .046 .026 .018 .043 **.080** .050 .029 .031 .045 .039 .037 .027 .026 .031 .039 .040 .037 .041 .046 .045 .043 .035 .030
3	4	.038 .036 .040 .033 .036 .060 .035 .041 .029 .058 .035 .035 .034 .053 .030 .032 .035 .036 .036 .028 .046 .032 .051 .032 .034 .030
3	5	.035 .034 .034 .036 .030 .043 .043 .050 .025 .041 .051 .050 .035 .032 .033 .033 .052 .031 .027 .030 **.072** .035 .034 .032 .043 .027
4	5	.052 .038 .033 .038 .041 .043 .037 .048 .028 .028 .036 **.061** .033 .033 .032 .052 .034 .027 .039 .043 .033 .027 .030 .039 .048 .035

Từ đó ta có các phương trình (theo mod26):

$k_1 - k_2 = 9$ $\quad$ $k_2 - k_5 = 7$

$k_1 - k_5 = 16$ $\quad$ $k_3 - k_5 = 20$

$k_2 - k_3 = 13$ $\quad$ $k_4 - k_5 = 11$.

Hệ phương trình đó chỉ có 4 phương trình độc lập tuyến tính, mà có 5 ẩn số, nên lời giải phụ thuộc một tham số, ta chọn là k_1, và được

$$(k_1, k_2, k_3, k_4, k_5) = (k_1, k_1 + 17, k_1 + 4, k_1 + 21, k_1 + 10) \text{mod} 26.$$

Thử với các giá trị có thể của k_1 ($0 \le k_1 \le 26$), cuối cùng ta có thể tìm được bản rõ như sau đây với khoá là JANET ($k_1 = 9$):

the *almond tree was in tentative blossom the days were longer often ending with magnificent evenings of corrugated pink skies the hunting season was over with hounds and guns put away for six months* the *vineyards were busy again as* the *well organized farmers treated their vines and* the *more lackadaisical neighbors hurried to do the pruning* the*y should have done in november.*

3.2.4. Thám mã đối với mã Hill.

Mật mã Hill khó bị khám phá bởi việc thám mã *chỉ dựa vào bản mã*, nhưng lại là dễ bị khám phá nếu có thể sử dụng phép thám mã kiểu *biết cả bản rõ*. Trước hết ta giả thiết là đã biết giá trị *m*. Mục đích của thám mã là phát hiện được khoá mật mã *K*, trong trường hợp mã Hill là một ma trận cấp *m* có các thành phần trong $\boldsymbol{Z_{26}}$.

Ta chọn một bản rõ có chứa ít nhất *m* bộ *m* khác nhau các ký tự:

$$x_1 = (x_{11}, ..., x_{1m}), ..., x_m = (x_{m1}, ..., x_{mm}),$$

và giả thiết biết mã tương ứng của chúng là:

$$y_1 = (y_{11}, ..., y_{1m}), ..., y_m = (y_{m1}, ..., y_{mm}).$$

Ta ký hiệu X và Y là hai ma trận cấp *m* , X=(x_{ij}), Y= (y_{ij}). Theo định nghĩa mã Hill, ta có phương trình Y =X.*K*. Nếu các x_i được chọn sao cho ma trận X có nghịch đảo X^{-1} thì ta tìm được $K = X^{-1}.Y$, tức là tìm được khoá của hệ mã được sử dụng.

Thí dụ: Giả sử mã Hill được sử dụng có *m* =2, và ta biết bản rõ *friday* cùng bản mã tương ứng *pqcfku*. Như vậy ta biết

$e_K(5,17) = (15,16)$, $e_K(8,3) = (2,5)$, và $e_K(0,24) = (10,20)$.

Từ hai phương trình đầu ta được

$$\begin{pmatrix} 15 & 16 \\ 2 & 5 \end{pmatrix} = \begin{pmatrix} 5 & 17 \\ 8 & 3 \end{pmatrix}.K,$$

từ đó được $K = \begin{pmatrix} 7 & 19 \\ 8 & 3 \end{pmatrix}$. Với K đó phương trình thứ ba cũng nghiệm đúng.

Trở lại với vấn đề xác định m. Nếu m không qua lớn, ta có thể thử cách trên lần lượt với $m = 2,3,4,...$ cho đến khi tìm được khoá, và khoá K xem là tìm được nếu ngoài m cặp bộ m $(x_1,y_1),...,(x_m, y_m)$ dùng để tìm khoá, K vẫn nghiệm đúng với các cặp bộ m khác mà ta có thể chọn để thử.

3.3. Mật mã theo dòng và các dãy số giả ngẫu nhiên.

3.3.1. Mật mã theo dòng.

Các hệ mật mã được xét trong các tiết trên đều thuộc loại mật mã theo khối, văn bản rõ được chia thành từng khối và việc lập mật mã cho văn bản đó được thực hiện cho từng khối rồi sau đó nối ghép lại, lập mật mã cho tất cả các khối đều theo cùng một khoá chung K. Với cách lập mật mã theo dòng, theo mô tả trong tiết 1.2, các khoa dùng cho các khối văn bản nói trên có thể khác nhau, do đó, cùng với sơ đồ mật mã gốc, ta còn cần có một *bộ sinh dòng khoá* để với mỗi mầm khoá s cho trước nó sinh ra một dòng khoá $K_1K_2K_3...$, mỗi K_i dùng để lập mật mã cho khối x_i của văn bản. Mỗi từ khoá K_i, ngoài việc phụ thuộc vào mầm khoá s còn có thể phụ thuộc vào đoạn từ khoá $K_1...K_{i-1}$ đã được sinh ra trước đó và cả vào các yếu tố khác, chẳng hạn như đoạn văn bản $x_1...x_{i-1}$ đã được lập mật mã trước đó. Như vậy, ta có thể định nghĩa lại như sau: Một sơ đồ hệ mật mã theo dòng được cho bởi một bộ

$$\mathcal{S} = (\mathcal{P}, \mathcal{C}, \mathcal{R}, \mathcal{K}, \mathcal{F}, \mathcal{E}, \mathcal{D}) \qquad (1)$$

thỏa mãn các điều kiện sau đây:

$\mathcal{P}$ *là một tập hữu hạn các ký tự bản rõ,*

$\mathcal{C}$ *là một tập hữu hạn các ký tự bản mã,*

$\mathcal{R}$ *là một tập hữu hạn các mầm khoá,*

$\mathcal{K}$ *là một tập hữu hạn các khóa,*

$\mathcal{F} = \{ f_1, f_2,....\}$ *là bộ sinh dòng khoá, trong đó mỗi* f_i *là một ánh xạ từ* $\mathcal{R}3\mathcal{K}^{i-1}3\mathcal{P}^{i-1}$ *vào* $\mathcal{K}$,

$\mathcal{E}$ *là một ánh xạ từ* $\mathcal{K}3\mathcal{P}$ *vào* $\mathcal{C}$, *được gọi là phép lập mật mã; và* $\mathcal{D}$ *là một ánh xạ từ* $\mathcal{K}3\mathcal{C}$ *vào* $\mathcal{P}$, *được gọi là phép giải mã. Với mỗi* $K\varepsilon\mathcal{K}$, *ta định nghĩa* $e_K : \mathcal{P} \rightarrow \mathcal{C}$, $d_K : \mathcal{C} \rightarrow \mathcal{P}$ *là hai hàm cho bởi*:

$$)x\varepsilon\mathcal{P} : e_K(x) = \mathcal{E}(K,x);)y\varepsilon\mathcal{C} : d_K(y) = \mathcal{D}(K,y).$$

e_K *và* d_K *được gọi lần lượt là hàm lập mã và hàm giải mã ứng với khóa mật mã K. Các hàm đó phải thỏa mãn hệ thức*:

$$)x\varepsilon\mathcal{P} : d_K(e_K(x)) = x.$$

Khi cho trước mầm khoá $r\varepsilon\mathcal{R}$, *với mỗi bản rõ* $x = x_1x_2....x_m \in \mathcal{P}^*$, *ta có bản mật mã tương ứng là* $y = y_1 y_2.... y_m$, *với*

$$y_i = \mathcal{E}(K_i, x_i)\text{, trong đó } K_i = f_i(r, K_1,...,K_{i-1}, x_1x_2....x_{i-1}), (i=1,2,...,m).$$

Điều đó có nghĩa là từ mầm khoá r và bản rõ x sinh ra được dòng khoá $K_1K_2...K_m$, và với dòng khoá đó lập được bản mật mã y theo từng ký tự một.

Nếu bộ sinh dòng khoá không phụ thuộc vào văn bản rõ, tức là nếu mỗi f_i là một ánh xạ từ $\mathcal{R}3\mathcal{K}^{i-1}$ vào $\mathcal{K}$, thì ta gọi bộ sinh dòng khoá đó là *đồng bộ*; dòng khoá chỉ phụ thuộc vào mầm khoá và là như nhau đối với mọi văn bản rõ. Một dòng khóa $K = K_1K_2K_3..$ được gọi là *tuần hoàn* với chu kỳ d nếu có số nguyên N sao cho $K_{i+d} = K_i$ với mọi $i \geq N$. Chú ý rằng mã Vigenère với độ dài khóa m có thể được coi là mã dòng với dòng khoá có chu kỳ m, và có các phép lập mã và giải mã theo mã chuyển dịch.

Đối với các hệ mã theo dòng, độ bảo mật thường được quyết định bởi độ ngẫu nhiên của dòng khoá, tức là tính ngẫu nhiên của

việc xuất hiện các ký tự trong dòng khoá, mà ít phụ thuộc vào bản thân phép lập mật mã, do đó các phép lập mật mã e_K (và cả phép giải mã d_K) đều có thể được chọn là các phép đơn giản; trong các ứng dụng thực tế, người ta thường dùng hệ mã với $\mathcal{P} = \mathcal{C} = \mathcal{K} = \boldsymbol{Z}_2$, và với các phép lập mã và giải mã được cho bởi

$$\left.\begin{aligned} e_K(x) &= x + K \bmod 2, \\ d_K(y) &= y + K \bmod 2 \end{aligned}\right\} \qquad (2)$$

3.3.2. Mã dòng với dòng khoá sinh bởi hệ thức truy toán.

Các hệ mật mã dòng với dòng khoá sinh bởi hệ thức truy toán là các hệ mã theo sơ đồ (1) với $\mathcal{P} = \mathcal{C} = \mathcal{K} = \boldsymbol{Z}_2$, $\mathcal{R} = Z_2^m\ (m \geq 1)$, mỗi mầm khoá $r = r_1....r_m$ tạo ra một dòng khoá đồng bộ $K = z_1z_2....z_i....$ với

$$\begin{cases} z_i = r_i, (i = 1, ..., m) \\ z_i = c_1 z_{i-m} + ... + c_m z_{i-1} \bmod 2, (i \geq m+1), \end{cases} \qquad (3)$$

trong đó c_1 ,..., c_m là các hằng số thuộc $\boldsymbol{Z}_2$; các phép lập mật mã và giải mã cho từng ký tự được cho bởi các công thức (2).

Các dòng khoá sinh bởi hệ thức truy toán như trên là các dòng khoá tuần hoàn, ta có thể chọn mầm sao cho đạt được dòng khoá có chu kỳ lớn nhất là 2^m-1.

Hệ tạo sinh các dòng khoá bởi hệ thức truy toán có thể được thực hiện bởi một thiết bị kỹ thuật đơn giản bằng cách dùng một *thanh ghi chuyển dịch phản hồi tuyến tính* (linear feedback shift register); và như vậy chỉ cần thêm một bộ cộng mod2 nữa là ta có được một máy lập mật mã và giải mã tự động; do đó các máy mật mã kiểu này đã được sử dụng khá phổ biến trong một giai đoạn trước đây.

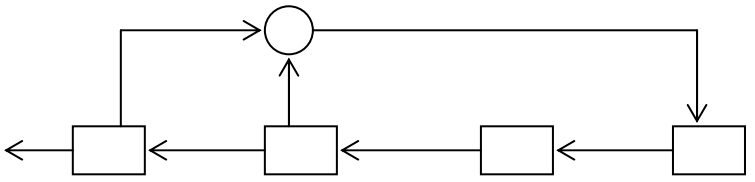
Thí dụ: chọn $m = 4$ và hệ thức truy toán

$$z_i = z_{i-4} + z_{i-3} \bmod 2 \quad (i > 4)$$

ta sẽ được với mọi mầm $K = z_1z_2z_3z_4 \neq 0000$ một dòng khoá tuần hoàn có chu kỳ 15. Chẳng hạn, với $r = 1000$ ta sẽ được dòng khoá:

100010011010111100010001.....

Dòng khoá đó được sinh bởi thanh ghi chuyển dịch phản hồi tuyến tính sau đây:



3.3.3. Mã dòng với dòng khoá là dãy số giả ngẫu nhiên.

Như đã xét trong các mục trên, sơ đồ mã theo dòng có thể được xem là bao gồm hai bộ phận: một *sơ đồ mật mã nền* (cho việc lập mật mã và giải mã trên từng ký tự),và một *cơ chế tạo dòng khóa.* Tương tự như với hệ mã dòng có dòng khoá sinh bởi thanh ghi chuyển dịch trong mục trên, ta sẽ xét sơ đồ mật mã nền là sơ đồ

$$\mathcal{S} = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D}),$$

trong đó $\mathcal{P} = \mathcal{C} = \mathcal{K} = \mathbf{Z}_2$, $\mathcal{E}$ và $\mathcal{D}$ được cho bởi:

$$\mathcal{E}(K, x) = x + K \bmod 2, \qquad \mathcal{D}(K, y) = y + K \bmod 2.$$

Cơ chế tạo dòng khoá có thể xem là một ánh xạ $\varphi : \mathcal{R} \times Z \rightarrow \mathcal{K}$, xác định với mỗi mầm khoá $r \in \mathcal{R} = Z_2^m (m \geq 1)$ và mỗi số nguyên $i \geq 0$, một số hạng $z_i = \varphi(r, i) \in \mathcal{K}$ của dòng khoá đồng bộ $K = z_1z_2....z_i.....$

Một hệ mật mã dòng là có độ bảo mật cao, nếu bản thân sơ đồ mật mã nền có độ bảo mật cao (chẳng hạn, là bí mật hoàn toàn theo định nghĩa Shannon), và cơ chế tạo dòng khoá tạo ra được các dòng khoá là các dãy bit ngẫu nhiên. Dễ thấy rằng, sơ đồ mật mã nền mô tả ở trên thoả mãn các điều kiện của định lý 2.2.1 , do đó nó là bí mật

hoàn toàn; vì vậy để có được các hệ mã dòng với độ bảo mật cao ta chỉ cần chọn được các cơ chế tạo dòng khoá bảo đảm sinh ra các dãy bit ngẫu nhiên. Một dãy bit $z_1z_2....z_i.....$ được xem là *ngẫu nhiên*, nếu mỗi z_i là một biến ngẫu nhiên với $p(z_i = 0) = p(z_i = 1) = 0.5$, và các biến ngẫu nhiên z_i và z_j $(i \neq j)$ là độc lập với nhau. Với nghĩa đó, ta không có cách nào để đoán nhận một dãy bit cho trước có là ngẫu nhiên hay không, vả chăng một dãy bit, nếu đã được sinh ra bởi một số hữu hạn qui tắc nào đó, thì không còn có thể xem là ngẫu nhiên được nữa. Vì vậy, thay cho đòi hỏi phải tạo ra các dãy bit ngẫu nhiên, thường ta chỉ yêu cầu tạo ra được các dãy bit *giả ngẫu nhiên*, tức là có một tính chất nào đó gần với ngẫu nhiên, mà thôi. Yêu cầu thông dụng nhất đối với tính giả ngẫu nhiên của một dãy bit $z_1z_2....z_i.....$ là "biết trước một đoạn đầu $z_1z_2....z_{i-1}$ khó mà đoán được bit tiếp theo z_i". Ta thử chính xác hoá ý tưởng này như sau:

Không gian các mầm khoá $\mathcal{R} = Z_2^m$ $(m \geq 1)$ có tất cả là 2^m mầm khoá khác nhau, giả sử tất cả chúng đều có xác suất xuất hiện như nhau, tức là bằng $1/2^m$. Ta xét tập hợp tất cả các dòng khoá có thể có với độ dài l $(l > m)$, tức là tập Z^l, và trên tập đó ta xác định một phân bố xác suất p_1 sao cho $p_1(z_1....z_l) = 1/2^m$ nếu $z_1....z_l$ là một dòng khoá sinh ra được từ một mầm khoá $r \in \mathcal{R}$ nào đó, và $p_1(z_1....z_l) = 0$ nếu ngược lại. Ta nói phân bố xác suất p_1 đó trên Z^l là được cảm sinh từ phân bố xác suất đều trên không gian các mầm khoá $\mathcal{R}$. Còn chính phân bố xác suất đều trên Z^l sẽ được ký hiệu là p_0.

Giả sử $\varphi : \mathcal{R} \times Z \rightarrow \mathcal{K}$ là cơ chế tạo dòng khoá của một hệ mật mã dòng, và $r \in \mathcal{R}$. Ta nói B là một thuật toán *đoán bit tiếp theo* (đối với φ và r) nếu với mọi số nguyên i $(0 \leq i \leq l)$ và mọi từ $z_1...z_{i-1} \in Z^{i-1}$, ta có : $B(i, z_1...z_{i-1}) = \varphi(r, i)$. Rõ ràng nếu ta muốn cơ chế φ tạo ra các dòng khoá giả ngẫu nhiên *tốt* thì ta không mong có thuật toán đoán bit tiếp theo làm việc có hiệu quả (chẳng hạn tính toán được trong thời gian đa thức). Giảm nhẹ yêu cầu "đoán đúng bit tiếp theo", ta sẽ

nói thuật toán B là *ε-đoán bit tiếp theo* (đối với φ và r) nếu có

$$\sum_{z_1...z_{i-1}\in Z^{i-1}} p_1(z_1...z_{i-1}).p(B(i,z_1...z_{i-1})=\varphi(r,i))\geq\frac{1}{2}+\varepsilon. \qquad (4)$$

(chú ý rằng biểu thức ở vế trái là kỳ vọng toán học của việc đoán đúng bit thứ i tiếp theo của các dòng khoá gồm i-1 bit).

Như vậy, ta có thể xem một cơ chế tạo dòng khoá φ là an toàn để sử dụng cho các hệ mật mã theo dòng, nếu với mọi mầm khoá r và mọi $\varepsilon > 0$ bất kỳ, không thể có thuật toán ε-đoán bit tiếp theo làm việc trong thời gian đa thức.

Dưới đây, ta sẽ dựa vào các hàm số học một phía để xây dựng một số cơ chế tạo các dãy số giả ngẫu nhiên cơ hồ có thể dùng làm cơ chế để tạo dòng khoá cho các hệ mật mã theo dòng mà ta đang xét.

Tạo bit giả ngẫu nhiên RSA.

Cơ chế tạo dãy bit giả ngẫu nhiên RSA được mô tả như sau : Chọn số nguyên $n = p.q$ là tích của hai số nguyên tố p và q có biểu diễn nhị phân với độ dài cỡ $m/2$ bit (như vậy n có biểu diễn nhị phân cỡ m bit), và một số b sao cho gcd(b,ϕ (n)) = 1. Lấy $\mathcal{R} = Z_n^*$, và với mỗi $r \in \mathcal{R}$ xác định dãy số s_0, s_1, s_2,.... như sau:

$$\begin{cases} s_0 = r, \\ s_{i+1} = s_i^b \bmod n, \end{cases}$$

và sau đó định nghĩa $z_i = \varphi(r, i) = s_i \bmod 2$, tức z_i là bit thấp nhất trong biểu diễn nhị phân của số s_i. Dãy $K = z_1 z_2 .... z_i ....$ là dòng bit đồng bộ được tạo ra bởi mầm r.

Thí dụ : Lấy n = 91261 = 263.347, b =1547, r =75634. Có thể tính các số s_1,...,s_{20} lần lượt là:

31483, 31238, 51968, 39796, 28716, 14089, 5923, 44891, 62284, 11889, 43467, 71215, 10401, 77444, 56794, 78147, 72137, 89592, 29022, 13356.

Và 20 bit đầu tiên của dòng bit giả ngẫu nhiên được sinh ra là:

$z_1...z_{20}$ = 10000111011110011000.

Tạo bit giả ngẫu nhiên BBS (Blum-Blum-Shub) :

Cơ chế tạo bit giả ngẫu nhiên BBS được mô tả như sau : Chọn $n = p.q$ là tích của hai số nguyên tố dạng $4m+3$, tức $p \equiv 3 (\text{mod} 4)$ và $q \equiv 3 \ (\text{mod} 4)$. Gọi $QR(n)$ là tập các thặng dư bậc hai theo mod$n$. Lấy $\mathcal{R} = QR(n)$, và với mỗi $r \in \mathcal{R}$ xác định dãy số s_0, s_1, s_2,.... như sau:

$$\begin{cases} s_0 = r, \\ s_{i+1} = s_i^2 \bmod n, \end{cases}$$

và sau đó định nghĩa $z_i = \varphi(r, i) = s_i \bmod 2$, tức z_i là bit thấp nhất trong biểu diễn nhị phân của số s_i. Dãy $K = z_1 z_2 .... z_i ...$ là dòng bit đồng bộ được tạo ra bởi mầm r.

Thí dụ : Lấy n = 192649 = 383.503, r = 20749 (= 101355^2 mod$n$). Có thể tính 20 số đầu của dãy $s_1,...,s_{20},...$ lần lượt là:

143135, 177671, 97048, 89992, 174051, 80649, 45663,

69442, 186894, 177046, 137922, 123175, 8630, 114386,

14863, 133015, 106065, 45870, 137171, 18460.

Và 20 bit đầu của dòng bit giả ngẫu nhiên được sinh ra là:

$z_1...z_{20}$ = 11001110000100111010.

Tạo bit giả ngẫu nhiên dựa vào bài toán logarit rời rạc :

Chọn p là một số nguyên tố lớn, và α là một phần tử nguyên thuỷ theo mod$p$. Tập các mầm khoá là $\mathcal{R} = Z_p^*$. Với mỗi mầm khoá $r \in \mathcal{R}$ ta xác định dãy số $s_0,...,s_i$ .... bởi :

$$s_0 = r,$$
$$s_{i+1} = \alpha^{s_i} \bmod p.$$

Sau đó định nghĩa $z_i = \varphi(r, i)(i=1,2,....)$ như sau: $z_i = 1$ nếu $s_i > p/2$, và $z_i = 0$ nếu $s_i < p/2$. Và $K = z_1....z_i.....$ là dòng khoá, tức dòng bit giả ngẫu nhiên, được tạo ra.

Trên đây là một vài cơ chế tạo dòng khoá, và để các dòng khoá được tạo ra đó là những dòng bit giả ngẫu nhiên *tốt*, ta đã cố ý dựa vào một số bài toán số học *khó* theo nghĩa là chưa tìm được những thuật toán làm việc trong thời gian đa thức để giải chúng, như các bài toán RSA, bài toán thặng dư bậc hai và bài toán lôgarit rời rạc. Các cơ chế tạo dòng khoá đó được xem là an toàn nếu ta chứng minh được rằng không thể có các thuật toán ε-đoán bit tiếp theo đối với chúng; hay một cách khác, nếu có thuật toán ε-đoán bit tiếp theo đối với chúng thì cũng sẽ có thuật toán (tất định hoặc xác suất) giải các bài toán số học tương ứng. Tiếc thay, đến nay ta chưa chứng minh được một kết quả nào theo hướng mong muốn đó; tuy nhiên cũng đã có một vài kết quả yếu hơn, thí dụ, đối với bộ tạo bit giả ngẫu nhiên BBS người ta đã chứng minh được rằng : nếu với mọi $\varepsilon > 0$ có thuật toán ε- đoán bit có trước (đối với φ và r) thì với mọi $\delta > 0$ cũng có thể xây dựng một thuật toán xác suất giải bài toán thặng dư bậc hai với xác suất trả lời sai là $< \delta$ (Định nghĩa của thuật toán ε- đoán bit có trước tương tự như với thuật toán ε- đoán bit tiếp theo, chỉ khác là thay công thức (4) bởi công thức sau đây

$$\sum_{z_1...z_{i-1} \in Z^{i-1}} p_1(z_1...z_{i-1}).p(B(i, z_1...z_{i-1}) = z_0) \geq \frac{1}{2} + \varepsilon.$$

trong đó $z_0 = s_0 \bmod 2$ là bit có trước dãy $z_1...z_{i-1}$).

Trong thực tiễn, các hệ mã dòng với dòng khoá là dãy bit ngẫu nhiên đã được sử dụng từ lâu và còn được sử dụng cho đến ngày nay, với dòng bit ngẫu nhiên được tạo ra một cách cơ học như việc tung đồng xu liên tiếp và ghi liên tiếp các kết quả "sấp, ngửa" của các lần tung. Các hệ mã dòng với dòng khoá ngẫu nhiên và với sơ đồ mật mã nền cho bởi các hệ thức (2) có thể được xem là "bí mật hoàn toàn" theo nghĩa Shannon, do đó rất được ưa chuộng trong ứng dụng thực tế, chúng thường được gọi là các hệ *đệm một lần* (one-time pad), được mô tả và sử dụng đầu tiên bởi Gilbert Vernam năm 1917. Tuy nhiên, việc tạo các dòng bit ngẫu nhiên một cách thủ công là không hiệu quả, việc giữ bí mật các dòng khoá như vậy lại

càng khó hơn, nên không thể sử dụng một cách phổ biến được, do đó ngày nay các hệ mã như vậy chỉ còn được sử dụng trong những trường hợp thật đặc biệt.

3.4. Hệ mật mã chuẩn DES.

3.4.1. Giới thiệu hệ mã chuẩn.

Bước sang kỷ nguyên máy tính, việc sử dụng máy tính nhanh chóng được phổ cập trong mọi hoạt động của con người, và tất nhiên việc dùng máy tính trong truyền tin bảo mật đã được hết sức chú ý. Các hệ mật mã với các thuật toán lập mật mã và giải mã thực hiện bằng máy tính được phát triển nhanh chóng, đồng thời các lĩnh vực truyền tin cần sử dụng mật mã cũng được mở rộng sang nhiều địa hạt kinh tế xã hội ngoài các địa hạt truyền thống. Vào đầu thập niên 1970, trước tình hình phát triển đó đã nẩy sinh nhu cầu phải chuẩn hoá các giải pháp mật mã được sử dụng trong xã hội, để một mặt, hướng dẫn các thành viên trong xã hội thực hiện quyền truyền tin bảo mật hợp pháp của mình, mặt khác, bảo đảm sự quản lý và giám sát của nhà nươc đối với các hoạt động bảo mật đó. Do đó, tại Hoa kỳ, ngày 15/5/1973, Văn phòng quốc gia về Chuẩn (NBS - National Bureau of Standards) công bố một yêu cầu công khai xây dựng và đề xuất một *thuật toán mật mã chuẩn*, đáp ứng các đòi hỏi chủ yếu là:

- Thuật toán phải được định nghĩa đầy đủ và dễ hiểu;
- Thuật toán phải có độ an toàn cao, độ an toàn đó phải không phu thuộc vào sự giữ bí mật của bản thân thuật toán, mà chỉ nằm ở sự giữ bí mật của khoá;
- Thuật toán phải được sẵn sàng cung cấp cho mọi người dùng;
- Thuật toán phải thích nghi được với việc dùng cho các ứng dụng khác nhau;
- Thuật toán phải cài đặt được một cách tiết kiệm trong các thiết bị điện tử;
- Thuật toán phải sử dụng được có hiệu quả;
- Thuật toán phải có khả năng được hợp thức hoá;

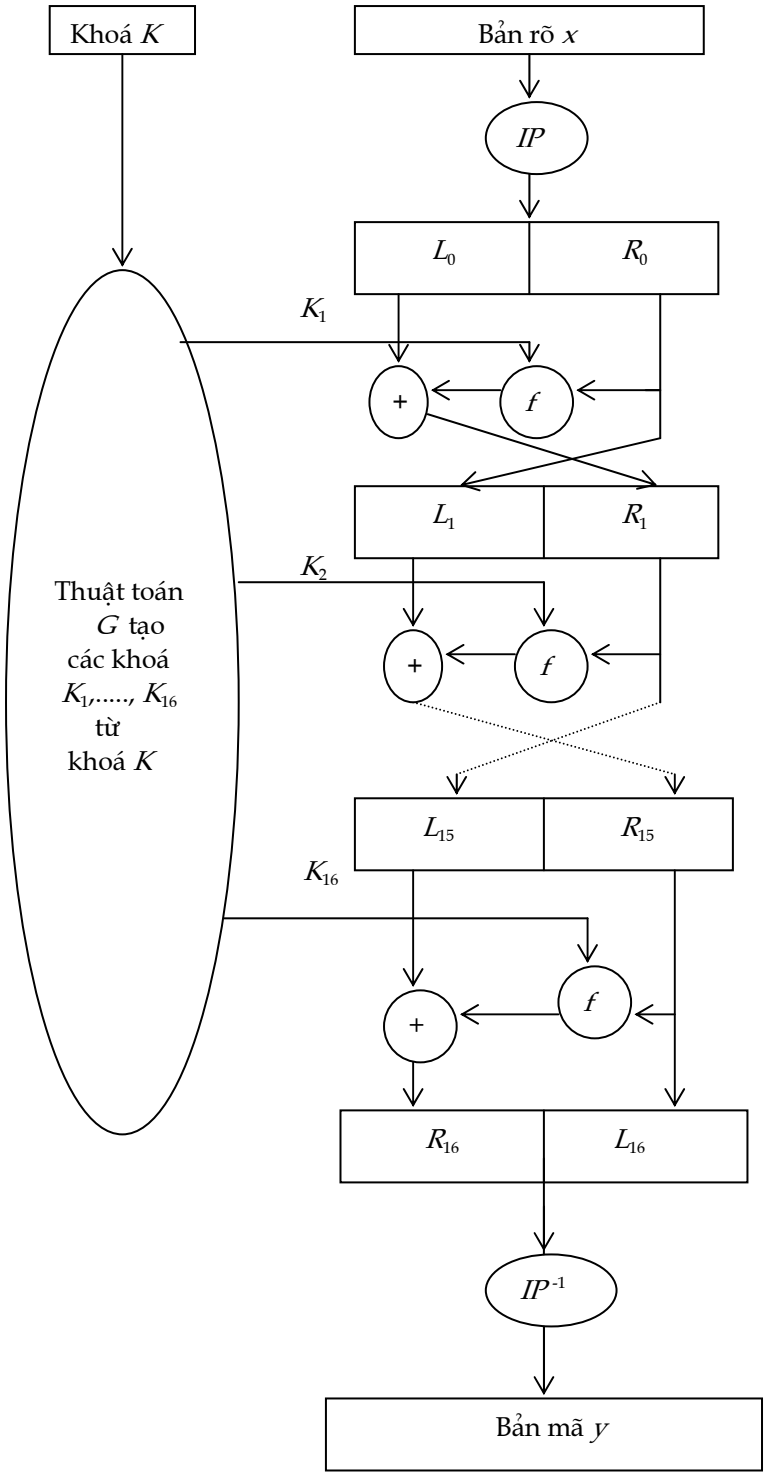
- Thuật toán phải xuất khẩu được.

Vào thời điểm NBS đưa ra yêu cầu nói trên, chưa có một cơ quan nào đề xuất được một giải pháp đáp ứng tất cả các đòi hỏi đó. Một năm sau, ngày 27/4/1974, yêu cầu đó lại được nhắc lại; và lần này hãng IBM chấp nhận dự tuyển với sản phẩm sẽ được đệ trình là một thuật toán cải tiến từ một thuật toán đã được phát triển trước đó là LUCIFER. Kết quả là, sản phẩm DES (Data Encryption Standard) được công bố, lần đầu tiên vào ngày 17/3/1975. Sau nhiều tranh luận, cuối cùng DES được chấp nhận như một chuẩn liên bang vào ngày 23/11/1976, và được công bố ngày 15/1/1977; đến năm 1980 lại công bố thêm «các cách dùng DES», cho phép người dùng có thể sử dụng DES theo nhiều cách khác nhau. Từ đó, DES được cài đặt sẵn vào các thiết bị cứng thành các máy mã, hoặc được cài đặt như một phần mềm trong các thiết bị tính toán đa dụng, và đã được sử dụng rộng rãi trong các lĩnh vực quản lý hành chính, kinh tế, thương mại, ngân hàng, v.v... không những ở Hoa kỳ mà còn ở nhiều quốc gia khác. Theo qui định của NBS, văn phòng quốc gia về chuẩn của Hoa kỳ, cứ khoảng 5 năm DES lại phải được xem xét lại một lần để được cải tiến và bổ sung. Sau khi các hệ mật mã có khoá công khai được phát triển và sử dụng rộng rãi, cũng đã có nhiều ý kiến đề nghị thay đổi chuẩn mới cho các hệ mật mã, nhưng trên thực tế, DES vẫn còn được sử dụng như một chuẩn cho đến ngày nay trong nhiều lĩnh vực hoạt động.

3.4.2. Mô tả hệ mật mã chuẩn DES.

Sơ đồ khái quát. Dưới đây ta sẽ trình bày sơ đồ của thuật toán lập mật mã DES. Hệ mật mã DES là một hệ mật mã theo khối, mỗi khối bản rõ là một từ 64 bit, tức là một phần tử thuộc Z_2^{64}, và các khối bản mã cũng là các từ 64 bit, như vậy $\mathcal{P} = \mathcal{C} = Z_2^{64}$. DES có tập khoá $\mathcal{K} = Z_2^{56}$, tức mỗi khoá là một từ 56 bit. Với mỗi khoá K và bản rõ x, quá trình lập mật mã diễn ra như sau: Thoạt đầu, dùng một phép hoán vị ban đầu IP, từ x 64 bit sẽ biến thành một từ mới $IP(x)$, từ này được chia thành hai nửa L_0 và R_0 , mỗi nửa là một từ 32 bit. Từ đây, sẽ dùng 15 lần những phép toán giống nhau để liên tiếp được các cặp (L_1,R_1),...., (L_{15},R_{15}), sau đó dùng phép hoán vị nghịch

đảo IP^{-1} cho từ đảo ngược $R_{15}L_{15}$ ta sẽ được bản mã y tương ứng. Sơ đồ khái quát của phép lập mật mã được cho bởi hình vẽ sau đây:



Sơ đồ khái quát của thuật toán lập mật mã DES

Để hoàn chỉnh sơ đồ thuật toán lập mật mã, ta còn phải trình bày các thuật toán IP (và do đó, cả IP^{-1}), thuật toán f, và thuật toán G tạo ra các khoá $K_1,...,K_{16}$.

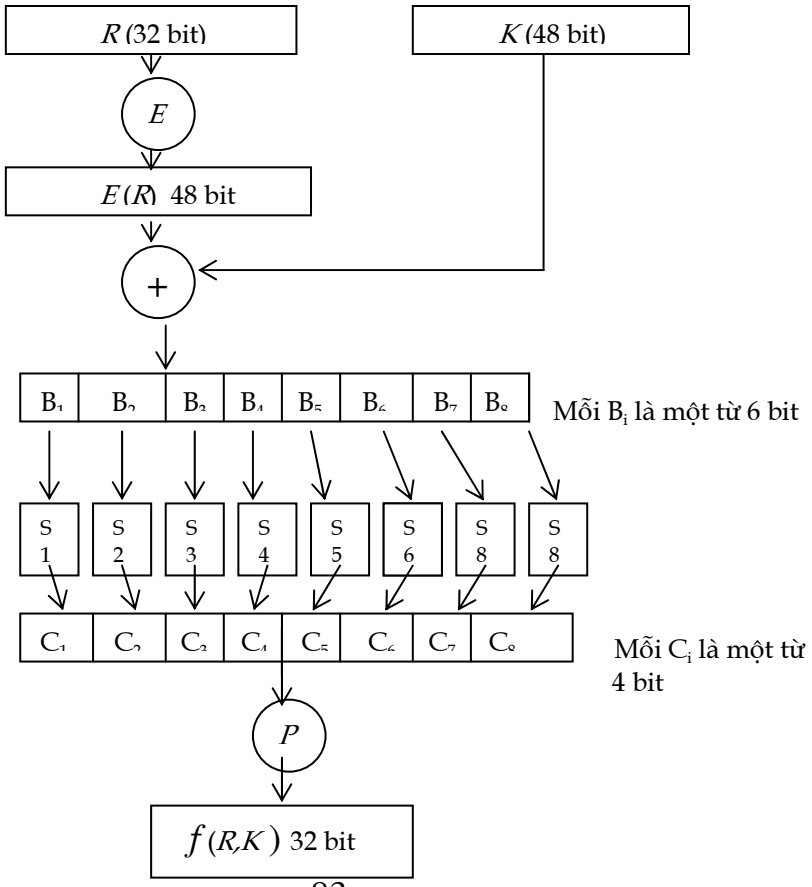
IP là một phép hoán vị vị trí của các ký tự trong mỗi từ 64 bit, từ vị trí thứ 1 đến vị trí thứ 64. Bảng dưới đây cho ta phép hoán vị IP, với cách hiểu là bit thứ nhất của $IP(x)$ là bit thứ 58 của từ x (có

64 bit), bit thứ hai của $IP(x)$ là bit thứ 50 của x, v.v... Bảng của phép hoán vị IP^{-1} cũng được hiểu tương tự.

IP							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

IP^{-1}							
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

<u>*Sơ đồ hàm f*:</u> Hàm f lấy đầu vào là hai từ : R có 32 bit và K có 48 bit, và có kết quả ở đầu ra là từ $f(R,K)$ có 32 bit, được xác định bởi sơ đồ sau đây:



Trong sơ đồ ở trên của hàm f, E là một phép hoán vị "mở rộng" theo nghĩa là nó biến mỗi từ R 32 bit thành từ $E(R)$ bằng cách hoán vị 32 bit của R nhưng có một số cặp bit được lặp lại để $E(R)$ thành một từ có 48 bit, cụ thể phép hoán vị "mở rộng" đó được cho bởi bảng sau đây :

Phép hoán vị "mở rộng" E					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Theo định nghĩa đó, mỗi từ $R = a_1a_2a_3.....a_{32}$ sè biến thành từ

$$E(R) = a_{32}\, a_1a_2a_3a_4a_5a_4a_5a_6a_7a_8a_9a_8a_9 \,.....a_{32}a_1 \,.$$

Sau khi thực hiện E, $E(R)$ sẽ được cộng (từng bit theo mod2) với K, được một từ 48 bit, chia thành 8 đoạn B_1, ..., B_8 . Mỗi hộp S_i (i = 1,...,8) là một phép thay thế, biến mỗi từ B_j 6 bit thành một từ C_j 4 bit; các hộp S_i được cho bởi các bảng dưới đây với cách hiểu như sau: mỗi từ $B_j = b_1b_2b_3b_4b_5b_6$ ứng với một vị trí (r,s) ở hàng thứ r và cột thứ s trong bảng, các hàng được đánh số từ thứ 0 đến thứ 3 ứng với biểu diễn nhị phân b_1b_6 và các cột được đánh số từ thứ 0 đến thứ 15 ứng với biểu diễn nhị phân $b_2b_3b_4b_5$. Giá trị của $S_i(B_j) = C_j = c_1c_2c_3c_4$ là một từ 4 bit, biểu diễn nhị phân của số tại hàng r cột s trong bảng. Thí dụ ta có S_1(101110) = 0101, S_2(011000) = 1110, v.v...

S_1															
14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S_2															
15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S_3															
10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S_4															
7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S_5															
2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S_6															
12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

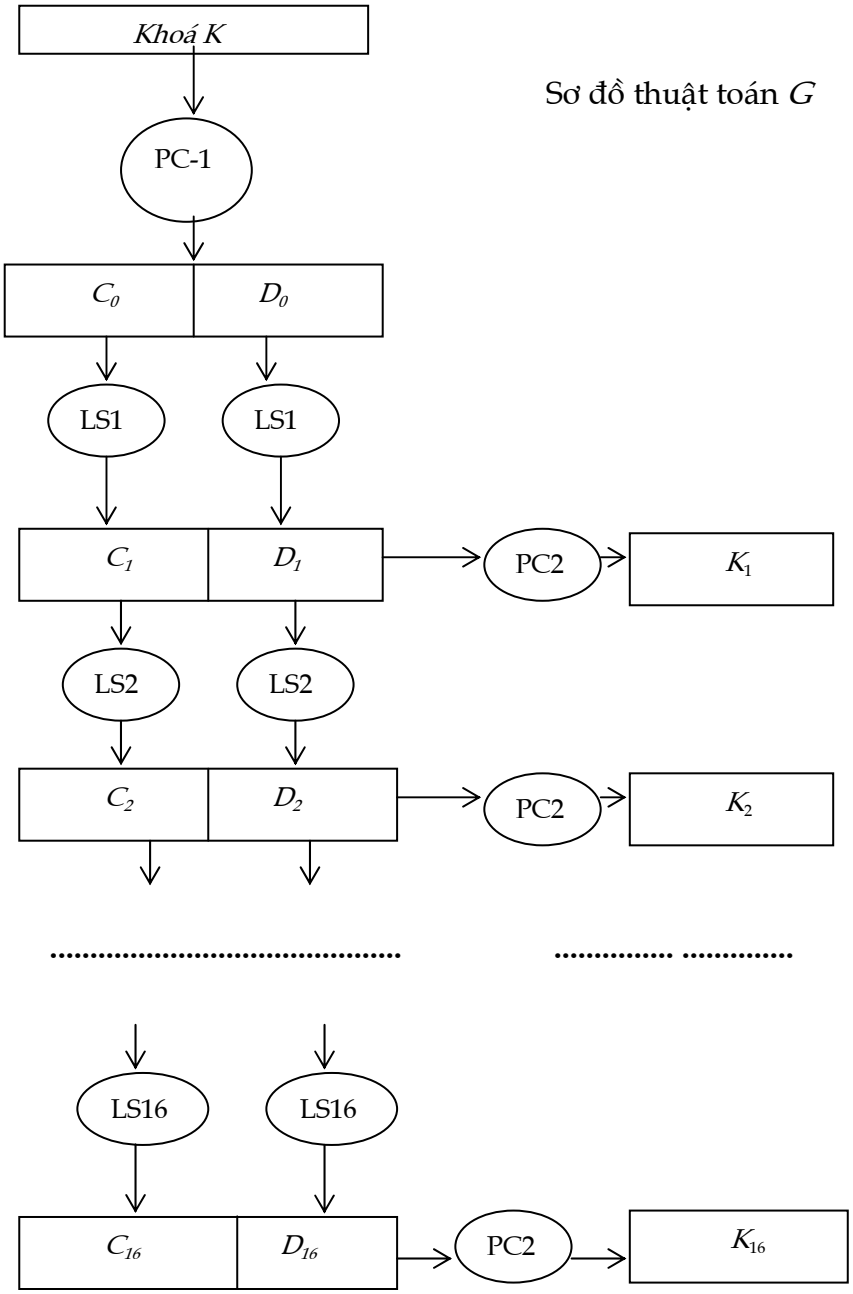
S_7															
4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S_8															
13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Phép hoán vị P trong sơ đồ của hàm f được cho bởi bảng ở trang sau đây. Như vậy, hàm f đã được xác định hoàn toàn. Chú ý rằng các hộp $S_1,\ldots, S_8$ là phần quan trọng nhất trong việc bảo đảm tính bí mật của hệ mã DES.

P			
16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Sơ đồ thuật toán G tạo các từ khoá K_1,...,K_{16}:



Sơ đồ thuật toán *G*

Thuật toán *G* tạo ra các từ khoá K_1,...,K_{16} từ khoá mật mã *K* được thực hiện theo sơ đồ thuật toán mô tả ở trên. Khoá mật mã *K* là một từ 56 bit, ta chia thành 8 đoạn, mỗi đoạn 7 bit, ta thêm cho mỗi đoạn 7 bit đó một bit thử tính chẵn lẻ vào vị trí cuối để được một từ 64 bit, ta vẫn ký hiệu là *K*, từ mới *K* này là từ xuất phát cho quá trình tính toán của thuật toán *G* (như sẽ thấy về sau, các bit thử tính chẵn lẻ mà ta thêm vào chỉ được dùng để phát hiện sai trong từng đoạn bit của khoá chứ thực tế không tham gia vào chính quá trình tính toán của *G*).

Trước tiên, thuật toán *PC*-1 biến K thành một từ 56 bit mà ta chia thành hai nửa C_0D_0, mỗi nửa có 28 bit. Phép hoán vị *PC*-1 được xác định bởi bảng sau đây (chú ý là trong bảng không có các số 8,16,24,32,40,48,56,64 là vị trí của những bit được thêm vào khi hình thành từ mới K). Nhớ rằng theo qui ước của phép hoán vị, bit thứ nhất của *PC*-1(x) là bit thứ 57 của x, bit thứ hai của
PC-1(x) là bit thứ 49 của x, v.v...

PC-1						
57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

Với mỗi i = 1,2,...16, LS_i là phép chuyển dịch vòng sang trái, chuyển dịch một vị trí nếu i = 1,2,9,16, và chuyển dịch hai vị trí với những giá trị i còn lại.
Cuối cùng, phép hoán vị *PC*-2 biến mỗi từ 56 bit C_iD_i (i =1,2,...16) thành từ 48 bit K_i theo bảng dưới đây:

PC-2					
14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Như vậy, ta đã mô tả đầy đủ quá trình tính toán của thuật toán G để từ khoa mã ban đầu K thu được các từ khoá K_1 ,..., K_{16} cung cấp cho thuật toán f, và từ đó cho toàn bộ thuật toán lập mật mã DES. Ta chú ý rằng mỗi K_i có 48 bit đều do hoán vị 56 bit (có bỏ bớt 8 bit) của K mà thành, do đó có thể cho trực tiếp bằng cách cho các bảng mô tả các phép hoán vị đó. Bạn đọc có thể tìm đọc 16 bảng ứng với 16 K_i đó trong sách của D.R. Stinson (có trong phần Sách tham khảo).

Với việc trình bày sơ đồ khái quát cùng với các bảng, các sơ đồ của các thuật toán phụ, ta đã hoàn thành việc giới thiệu thuật

toán lập mật mã $\mathcal{E}$ của hệ mật mã *DES*, cho ta $y = \mathcal{E}(K,x)$ với mối khoá K và bản rõ x.

Thuật toán giải mã $\mathcal{D}$, cho ta $x = \mathcal{D}(K, y)$, được thực hiện bằng cùng một quá trình tính toán như quá trình lập mã, chỉ khác là thứ tự dùng các K_i được đảo ngược lại theo thứ tự $K_{16}, K_{15}, ..., K_1$.

Có thể thực hiện thử các thuật toán lập mã và giải mã kể trên với thí dụ sau đây: Cho K và x là

K = 12695BC9B7B7F8

x = 0123456789ABCDEF,

ở đây các số được viết theo cơ số 16 (hexadecimal), mỗi ký tự thay cho 4 bit. Bản mã y tương ứng sẽ là

y = 85E813540F0AB405.

3.4.3. Các cách dùng DES.

Năm 1981, NBS công bố các chuẩn xử lý thông tin liên bang có liên quan đến DES, trong đó đã hợp thức hoá bốn cách dùng DES trong thực tế là các cách: ECB (electronic codebook mode), CFB (cipher feedback mode), CBC (cipher block chaining mode) và OFB (output feedback mode).

ECB là cách sử dụng thông thường và đơn giản của DES. Với cách sử dụng đó, ta chia bản rõ (là một dãy bit) thành từng khối 64 bit $x = x_1x_2....x_n$, và dùng cùng một khoá K để mã các khối đó rồi ghép lại để được bản mã $y = y_1y_2... y_n$, trong đó $y_i = e_K(x_i)$.

Với cách dùng CFB, để được khối mã y_i ta dùng DES cho không phải x_i mà là cho $x_i \oplus y_{i-1}$,tức là có $y_i = e_K(x_i \oplus y_{i-1})$ với mọi $i > 1$.

Trong hai cách CBC và OFB, ta dùng DES để tạo ra một dòng từ khoá $z_1...z_i.....$, rồi sau đó lập mã $y_i = x_i \oplus z_i$ ($i \geq 1$). Dòng khoá $z_1...z_i.....$ trong cách CBC được xác định bởi

$z_0 = K^*$ (là một từ 64 bit được chọn từ khoá K),

$z_i = e_K(z_{i-1})$;

còn trong cách OFB được xác định bởi

$y_0 = K^*$

$z_i = e_K(y_{i-1})$

$y_i = x_i \oplus z_i (i \geq 1)$.

Trong thực tế, các cách ECB và CBC được nhiều ngân hàng dùng làm chuẩn mật mã của mình, còn các cách CFB và òB thường được dùng cả với các mục đích xác nhận.

3.4.4. Về tính an toàn và việc thám mã đối với DES.

1. *Về tính an toàn bảo mật của DES.* Sau khi DES được công bố như một chuẩn chính thức cho truyền tin bảo mật của quốc gia, nhiều vấn đề về tính an toàn và khả năng bảo mật của DES được đặt ra và nhiều biện pháp thám mã cũng được nghiên cứu, trong suốt hơn hai mươi năm qua và cho đến nay.

Ta chú ý rằng trong cấu trúc của thuật toán DES, ở mỗi vòng lặp đều có các phép chuyển dịch và thay thế xen kẽ liên tiếp nhau, có tác dụng tăng thêm độ bảo mật của mật mã. Thuật toán DES nói chung đáp ứng các yêu cầu mà NBS đề ra từ đầu cho một chuẩn mật mã, và do đó yếu tố bảo mật chủ yếu tập trung vào việc giữ bí mật của khoá, hay nói cách khác, thám mã chủ yếu phải là phát hiện khoá được sử dụng. Trong các khâu của sơ đò DES thì các yếu tố phi tuyến duy nhất nằm ở các hộp S_1,..., S_8. Người ta không biết người thiết kế các hộp đó đã chọ chúng theo những tiêu chuẩn nào, và Cục an ninh quốc gia NSA có cài vào đó những "cửa sập" nào không; nhưng sau nhiều cố gắng thám mã không thành công, người ta đã công bố một số các tiêu chuẩn chon các hộp S_1,..., S_8 như sau:

1. Mỗi hàng của một hộp S_i phải là một hoán vị của 0,1,...,15;

2. Không một hộp S_i nào là một hàm tuyến tính hay apphin đối với các đầu vào của nó;

3. Với mỗi hộp S_i, việc thay đổi một bit ở đầu vào gây ra sự thay đổi ít nhất hai bit ở đầu ra của nó;

4. Nếu hai từ vào của một hộp S_i giống nhau ở hai bit đầu và hai bit cuối, thì hai từ ra phải khác nhau ở hai bit;

5. Nếu hai từ vào của một hộp S_i khác nhau ở hai bit đầu và giống nhau ở hai bit cuối, thì hai từ ra phải khác nhau;

6. Với mỗi hộp S_i, nếu ta cố định giá trị một bit vào và xét giá trị của bit ra ở một vị trí nào đó, thì số các từ vào tạo ra giá trị 0 và số các từ vào tạo ra giá trị 1 ở cùngvị trí đó phải xấp xỉ bằng nhau.

Nói chung, độ bảo mật của DES đã được thử thách qua hơn hai mươi năm sử dụng và được chứng tỏ là tin cậy. Các phương pháp thám mã, tuy đã được tìm kiếm khá nhiều, nhưng gần như không tránh được độ phức tạp của cách tầm thường là duyệt toàn bộ, mà theo cách này thì dù là thám mã theo kiểu "biết cả bản rõ" ta cũng phải duyệt qua 2^{56} khoá có thể có, điều đó đòi hỏi một lượng tính toán khổng lồ khó mà khắc phục nổi !

Về việc thám mã đối với DES.

Hệ mã chuẩn DES có thể xem là hệ mã đầu tiên được dùng phổ biến một cách rộng rãi không chỉ trong một quốc gia mà cả trên phạm vi toàn thế giới, toàn bộ cấu trúc thuật toán được công bố công khai, cả phép lập mã và giải mã, thậm chí các sản phẩm phần cứng cũng như phần mềm của nó được thương mại hoá; do đó bí mật của thông tin được truyền đi chỉ còn nằm ở chìa khoá được

chon, đó là một từ 56 bit. Việc thám mã đối với DES dã hấp dẫn nhiều nhà toán học và chuyên gia mật mã nghiên cứu, đề xuất nhiều phương pháp khác nhau. Ngoài phương pháp "duyệt toàn bộ" như nói trên, người ta đã đề xuất một số phương pháp khác, như:

- phương pháp phân tích độ chênh lệch (differential analysis) do Biham và Shamir đề xuất năm 1990,
- phương pháp phân tích liên quan đến khoá, do Biham đề xuất vào khaỏng 1992-1994,
- phương pháp phân tích tuyến tính, do Matsui đưa ra năm 1993-1994,
- phương pháp phân tích chênh lệch-tuyến tính, do Langfort và Hellman đưa ra năm 1994,
- v.v...

Các phương pháp này đều chứa đựng nhiều ý tưởng sâu sắc và tinh tế, nhưng vẫn đòi hỏi những khối lượng tính toán rất lớn, nên trong thực tế vẫn chỉ dừng lại ở những minh hoạ tương đối đơn giản chứ chưa được sử dụng thực sự.

CHƯƠNG IV

Các hệ mật mã khoá công khai

4.1. Giới thiệu mở đầu.

4.1.1. Sự ra đời của mật mã khoá công khai.

Trong chương I ta đã giới thiệu qua định nghĩa của các khái niệm hệ mật mã khoá đối xứng và hệ mật mã khoá công khai. Sự ra đời của khái niệm hệ mật mã khoá công khai là một tiến bộ có tính chất bước ngoặt trong lịch sử mật mã nói chung, gắn liền với sự phát triển của khoa học tính toán hiện đại. Người ta có thể xem thời điểm khởi đầu của bước ngoặt đó là sự xuất hiện ý tưởng của W. Diffie và M.E. Hellman được trình bày vào tháng sáu năm 1976 tại Hội nghị quốc gia hàng năm của AFIPS (Hoa kỳ) trong bài *Multiuser cryptographic techniques*. Trong bài đó, cùng với ý tưởng chung, hai tác giả cũng đã đưa ra những thí dụ cụ thể để thực hiện ý tưởng đó, và mặc dù các thí dụ chưa có ý nghĩa thuyết phục ngay đối với tác giả, thì ý tưởng về các hệ mật mã khoá công khai cũng đã rất rõ ràng và có sức hấp dẫn đối với nhiều người. Và ngay sau đó, công việc tìm kiếm những thể hiện cụ thể có khả năng ứng dụng trong thực tế đã bắt đầu thu hút sự quan tâm của nhiều chuyên gia. Một năm sau, năm 1977, R.L. Rivest, A. Shamir và L.M. Adleman đề xuất một hệ cụ thể về mật mã khoá công khai mà độ an toàn của hệ dựa vào bài toán khó "phân tích số nguyên thành thừa số nguyên tố", hệ này về sau trở thành một hệ nổi tiếng và mang tên là hệ RSA, được sử dụng rộng rãi trong thực tiễn bảo mật và an toàn thông tin. Cũng vào thời gian đó, M.O. Rabin cũng đề xuất một hệ mật mã khoá công khai dựa vào cùng bài toán số học khó nói trên. Liên tiếp sau đó, nhiều hệ mật mã khóa công khai được đề xuất, mà khá nổi tiếng và được quan tâm nhiều là các hệ: hệ McEliece được đưa ra năm 1978 dựa trên độ *NP*-khó của bài toán giải mã đối với các hệ mã cyclic tuyến tính, hệ Merkle-Hellman dựa trên tính *NP*- đầy đủ của bài toán xếp ba lô(knapsack problem), hệ mật mã nổi tiếng ElGamal dựa trên độ khó của bài toán lôgarit rời rạc, hệ này về sau được mở rộng để phát triển nhiều

hệ tương tự dựa trên độ khó của các bài toán tương tự lôgarit rời rạc trên các cấu trúc nhóm cyclic hữu hạn, nhóm các điểm nguyên trên đường cong eliptic, v.v... Để tăng độ bảo mật, hệ mật mã ElGamal còn dùng với tư cách đầu vào cho thuật toán lập mật mã của mình, ngoài khoá công khai và bản rõ, một yếu tố ngẫu nhiên được chọn tuỳ ý, điều đó làm cho hệ mật mã trở thành một hệ mật mã xác suất khoá công khai. Một số hệ mật mã xác suất khoá công khai cũng được phát triển sau đó bởi Goldwasser-Micali và Blum-Goldwasser. Tất cả các hệ mật mã khoá công khai kể trên sẽ được trình bày trong chương này cùng với một số tính chất liên quan của chúng.

4.1.2. Một số bài toán cơ bản.

Sau đây ta sẽ nhắc lại một số bài toán số học được sử dụng đến khi xây dựng các hệ mật mã khoá công khai như nói ở trên. Các bài toán này phần lớn đã được trình bày trong chương II, một số được phát triển thêm cho các ứng dụng trực tiếp khi xây dựng các hệ mã cụ thể, ta liệt kê dưới đây một lần để thuận tiện cho các chỉ dẫn về sau.

Bài toán phân tích số nguyên (thành thừa số nguyên tố):

Cho số nguyên dương n, tìm tất cả các ước số nguyên tố của nó, hay là tìm dạng phân tích chính tắc của $n = p_1^{\alpha_1}.p_2^{\alpha_2}...p_k^{\alpha_k}$, trong đó p_i là các số nguyên tố từng cặp khác nhau và các $\alpha_i \geq 1$.

Bài toán này có liên hệ mật thiết với các bài toán thử tính nguyên tố hay thử tính hợp số của một số nguyên, nhưng với những gì mà ta biết đến nay, nó dường như khó hơn nhiều so với hai bài toán thử tính nguyên tố và tính hợp số.

Trong lý thuyết mật mã, bài toán này thường được sử dụng với các dữ liệu n là số nguyên Blum, tức các số nguyên dương có dạng tích của hai số nguyên tố lớn nào đó.

Bài toán RSA (Rivest-Shamir-Adleman) :

Cho số nguyên dương n là tích của hai số nguyên tố lẻ khác nhau, một số nguyên dương e sao cho gcd(e,ϕ (n)) =1, và một số nguyên c; tìm một số nguyên m sao cho $m^e \equiv c \pmod n$.

Điều kiện gcd(e,ϕ (n)) =1 bảo đảm cho việc với mỗi số nguyên $c \in \{0,1,...,n-1\}$ có đúng một số $m \in \{0,1,...,n-1\}$ sao cho $m^e \equiv c \pmod n$.

Dễ thấy rằng nếu biết hai thừa số nguyên tố của n, tức là biết $n = p.q$ thì sẽ biết $\phi(n) = (p-1)(q-1)$, và từ đó, do gcd(e,ϕ (n)) =1 sẽ

tìm được $d = e^{-1} \bmod \phi(n)$, và do đó sẽ tìm được $m = c^d \bmod n$. Như vậy, bài toán RSA có thể qui dẫn trong thời gian đa thức về bài toán phân tích số nguyên. Tuy rằng cho đến nay chưa có một chứng minh nào cho việc qui dẫn ngược lại nhưng nhiều người vẫn tin rằng hai bài toán đó là tương đương với nhau về độ phức tạp tính toán.

Bài toán thặng dư bậc hai :

Cho một số nguyên lẻ n là hợp số, và một số nguyên $a \in J_n$, tập tất cả các số a có ký hiệu Jacobi $\left(\frac{a}{n}\right)=1$. Hãy quyết định xem a có là thặng dư bậc hai theo mod$n$ hay không?

Trong lý thuyết mật mã, bài toán này cũng thường được xét với trường hợp n là số nguyên Blum, tức n là tích của hai số nguyên tố p và q, $n = p.q$. Ta chú ý rằng trong trường hợp này, nếu $a \in J_n$, thì a là thặng dư bậc hai theo mod$n$ khi và chỉ khi $\left(\frac{a}{p}\right)=1$, điều kiện này có thể thử được dễ dàng vì nó tương đương với điều kiện $a^{(p-1)/2} \equiv 1 \pmod{p}$. Như vậy, trong trường hợp này, bài toán thặng dư bậc hai có thể qui dẫn trong thời gian đa thức về bài toán phân tích số nguyên. Mặt khác, nếu không biết cách phân tích n thành thừa số nguyên tố thì cho đến nay, không có cách nào giải được bài toán thặng dư bậc hai trong thời gian đa thức. Điều đó củng cố thêm niềm tin rằng bài toán thặng dư bậc hai và bài toán phân tích số nguyên là có độ khó tương đương nhau.

Bài toán tìm căn bậc hai mod$n$:

Cho một số nguyên lẻ n là hợp số Blum, và một số $a \in Q_n$, tức a là một thặng dư bậc hai theo mod$n$. Hãy tìm một căn bậc hai của a theo mod$n$, tức tìm x sao cho $x^2 \equiv a \pmod{n}$.

Nếu biết phân tích n thành thừa số nguyên tố, $n = p.q$, thì bằng cách giải các phương trình $x^2 \equiv a$ theo các mod$p$ và mod$q$, rồi sau đó kết hợp các nghiệm của chúng lại theo định lý số dư Trung quốc ta sẽ được nghiệm theo mod$n$, tức là căn bậc hai của a theo mod$n$ cần tìm. Vì mỗi phương trình $x^2 \equiv a$ theo mod$p$ và mod$q$ có hai nghiệm (tương ứng theo mod$p$ và mod$q$), nên kết hợp lại ta được bốn nghiệm, tức bốn căn bậc hai của a theo mod$n$. Người ta đã tìm được một số thuật toán tương đối đơn giản (trong thời gian đa thức) giải phương trình $x^2 \equiv a \pmod{p}$ với p là số nguyên tố.

Như vậy, bài toán tìm căn bậc hai mod$n$ có thể qui dẫn trong thời gian đa thức về bài toán phân tích số nguyên. Ngược lại, nếu có thuật toán $\triangleq$ giải bài toán tìm căn bậc hai mod$n$ thì cũng có thể xây dựng một thuật toán giải bài toán phân tích số nguyên như sau: Chọn ngẫu nhiên một số x với gcd(x,n) =1, và tính $a = x^2 \bmod n$. Dùng thuật toán $\triangleq$ cho a để tìm một căn bậc hai mod$n$ của a. Gọi căn bậc hai tìm được đó là y. Nếu $y \equiv \pm x \pmod n$, thì phép thử coi như thất bại, và ta phải chọn tiếp một số x khác. còn nếu $y \not\equiv \pm x \pmod n$, thì gcd($x$-$y$, n) chắc chắn là một ước số không tầm thường của n, cụ thể là p hay là q. Vì n có 4 căn bậc hai mod$n$ nên xác suất của thành công ở mỗi lần thử là 1/2, và do đó số trung bình (kỳ vọng toán học) các phép thử để thu được một thừa số p hay q của n là 2, từ đó ta thu được một thuật toán giải bài toán phân tích số nguyên (Blum) với thời gian trung bình đa thức. Tóm lại, theo một nghĩa không chặt chẽ lắm, ta có thể xem hai bài toán phân tích số nguyên và tìm căn bậc hai mod$n$ là khó tương đương nhau.

Bài toán lôgarit rời rạc :

Cho số nguyên tố p, một phần tử nguyên thuỷ α theo mod$p$ (hay α là phần tử nguyên thuỷ của Z_p^*), và một phần tử $\beta \in Z_p^*$.Tìm số nguyên x ($0 \le x \le p-2$) sao cho $\alpha^x \equiv \beta \pmod p$.

Trong mục 2.4.3 ta đã giới thiệu qua bài toán này, và biết rằng trong trường hợp chung, cho đến nay chưa có một thuật toán nào giải bài toán này trong thời gian đa thức.

Bài toán này cũng được suy rộng cho các nhóm cyclic hữu hạn như sau:

Bài toán lôgarit rời rạc suy rộng :

Cho một nhóm cyclic hữu hạn G cấp n, một phần tử sinh (nguyên thuỷ) α của G, và một phần tử $\beta \in G$. Tìm số nguyên x ($0 \le x \le n-1$) sao cho $\alpha^x = \beta$.

Các nhóm được quan tâm nhiều nhất trong lý thuyết mật mã là: nhóm nhân của trường hữu hạn $GF(p)$ - đẳng cấu với nhóm Z_p^* của trường Z_p ,nhóm nhân $\mathbb{F}_{2^m}^*$ của trường hữu hạn $GF(2^m)$, nhóm nhân $Z_n^* = \{a : 0 \le a \le n-1, \gcd(a,n) = 1\}$ của trường Z_n với n là hợp số, nhóm gồm các điểm trên một đường cong elliptic xác định trên một trường hữu hạn, v.v...

Bài toán Diffie-Hellman :

Cho số nguyên tố p, một phần tử nguyên thuỷ α theo mod$p$ (tức phần tử sinh của Z_p^*), và các phần tử $\alpha^a \bmod p$ và $\alpha^b \bmod p$.

Hãy tìm giá trị $\alpha^{ab} \bmod p$.

Có thể chứng minh được rằng bài toán Diffie-Hellman qui dẫn được về bài toán lôgarit rời rạc trong thời gian đa thức. Thực vậy, giả sử có thuật toán $\triangleq$ giải bài toán lôgarit rời rạc. Khi đó, cho một bộ dữ liệu vào của bài toán Diffie-Hellman gồm p, α, $\alpha^a \bmod p$ và $\alpha^b \bmod p$; trước hết dùng thuật toán $\triangleq$ cho (p, α, $\alpha^a \bmod p$) ta tìm được a, và sau đó tính được $\alpha^{ab} \bmod p = (\alpha^b)^a \bmod p$. Người ta cũng chứng minh được hai bài toán lôgarit rời rạc và Diffie-Hellman là tương đương về mặt tính toán trong một số trường hợp, ví dụ p-1 là B-mịn với $B = O((\ln p)^c)$, c là hằng số.

Tương tự như với bài toán lôgarit rời rạc, ta cũng có thể định nghĩa các bài toán Diffie-Hellman suy rộng cho các nhóm cyclic hữu hạn khác.

Bài toán tổng tập con (hay bài toán KNAPSACK) :

Cho một tập các số nguyên dương $\{a_1, a_2, ..., a_n\}$ và một số nguyên dương s. Hãy xác định xem có hay không một tập con các a_j mà tổng của chúng bằng s. Một cách tương đương, hãy xác định xem có hay không các $x_i \in \{0,1\}$ $(1 \le i \le n)$ sao cho $\sum_{i=1}^{n} a_i x_i = s$.

Bài toán này là một bài toán $\mathcal{NP}$- đầy đủ, tức là thuộc lớp những bài toán khó mà cho đến nay chưa tìm được thuật toán giải chúng trong thời gian đa thức !

Bài toán giải mã đối với mã tuyến tính :

Mã tuyến tính là một lớp mã truyền tin có tính chất tự sửa sai được sử dụng trong kỹ thuật truyền tin số hoá. Không đi vào chi tiết của lớp mã này, ta có thể phát biểu trực tiếp bài toán giải mã đối với mã tuyến tính như sau:

Cho một ma trận cấp $n \times m$ A=(a_{ij}) gồm các thành phần là 0 hoặc 1, một vectơ $\boldsymbol{y}$=($y_1, y_2, ..., y_m$) các giá trị 0 và 1, và một số nguyên dương K. Hỏi: có hay không một vectơ $\boldsymbol{x}$=($x_1, x_2, ..., x_n$) gồm các số 0 hoặc 1 và có không nhiều hơn K số 1 sao cho với mọi j $(1 \le j \le m)$:

$$\sum_{i=1}^{n} x_i . a_{ij} \equiv y_j \pmod 2 \ ?$$

Chú ý rằng ở đây, $\boldsymbol{x}$ là vectơ thông tin, và $\boldsymbol{y}$ là vectơ mã, phép giải mã là tìm lại $\boldsymbol{x}$ khi nhận được $\boldsymbol{y}$, bài toán này tiếc thay lại là một bài toán khó; Berlekamp, McEliece và Tilborg năm 1978 đã chứng minh nó thuộc lớp các bài toán $\mathcal{NP}$- đầy đủ !

4.2. Hệ mật mã khoá công khai RSA.

4.2.1. Mô tả hệ mật mã RSA.

Sơ đồ chung của hệ mật mã khoá công khai được cho bởi

$$\mathcal{S} = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D}) \qquad (1)$$

trong đó $\mathcal{P}$ là tập ký tự bản rõ, $\mathcal{C}$ là tập ký tự bản mã, $\mathcal{K}$ là tập các khoá K, mỗi khoá K gồm có hai phần $K=(K',K'')$, K' là khoá công khai dành cho việc lập mật mã, còn K'' là khoá bí mật dành cho việc giải mã. Với mỗi ký tự bản rõ $x\in\mathcal{P}$, thuật toán lập mã $\mathcal{E}$ cho ta ký tự mã tương ứng $y=\mathcal{E}(K', x) \in \mathcal{C}$, và với ký tự mã y thuật toán giải mã $\mathcal{D}$sẽ cho ta lại ký tự bản rõ x: $\mathcal{D}(K'', y) = \mathcal{D}(K'', \mathcal{E}(K', x)) = x$.

Để xây dựng một hệ mật mã khoá công khai RSA, ta chọn trước một số nguyên $n=p.q$ là tích của hai số nguyên tố lớn, chọn một số e sao cho $\gcd(e, \phi(n)) = 1$, và tính số d sao cho

$$e.d \equiv 1 (\mathrm{mod}\, \phi(n)).$$

Mỗi cặp $K=(K',K'')$, với $K' =(n,e)$ và $K'' = d$ sẽ là một cặp khoá của một hệ mật mã RSA cụ thể cho một người tham gia.

Như vậy, <u>sơ đồ chung của hệ mật mã RSA</u> được định nghĩa bởi danh sách (1), trong đó:

$\mathcal{P} = \mathcal{C} = Z_n$, trong đó n là một số nguyên Blum, tức là tích của hai số nguyên tố;

$\mathcal{K} = \{K=(K',K''): K' =(n,e)$ và $K'' = d$, $\gcd(e, \phi(n)) = 1$, $e.d \equiv 1 (\mathrm{mod}\, \phi(n))\}$;

$\mathcal{E}$ và $\mathcal{D}$ được xác định bởi:

$$\mathcal{E}(K', x) = x^e \bmod n, \text{ với mọi } x \in \mathcal{P},$$
$$\mathcal{D}(K'', y) = y^d \bmod n, \text{ với mọi } y \in \mathcal{C}.$$

Để chứng tỏ định nghĩa trên là hợp thức, ta phải chứng minh rằng với mọi cặp khoá $K=(K',K'')$, và mọi $x \in \mathcal{P}$, ta đều có

$$\mathcal{D}(K'', \mathcal{E}(K', x)) = x .$$

Thực vậy, do $e.d \equiv 1 (\mathrm{mod}\, \phi(n))$ ta có thể viết $e.d = t.\phi(n) + 1$. Nếu x nguyên tố với n, thì dùng định lý Euler (xem 2.1.3) ta có

$$\mathcal{D}(K'', \mathcal{E}(K', x)) = x^{ed} \equiv x^{t\phi(n)+1} \equiv x^{t\phi(n)}.x \,(\mathrm{mod}\, n) = x.$$

Nếu x không nguyên tố với n, thì do $n=p.q$, hoặc x chia hết cho p và nguyên tố với q, hoặc x chia hết cho q và nguyên tố với p, và $\phi(n) = (p-1).(q-1)$,trong cả hai trường hợp ta đều có

$$x^{t\phi(n)+1} \equiv x \,(\mathrm{mod}\, p),$$
$$x^{t\phi(n)+1} \equiv x \,(\mathrm{mod}\, q);$$

từ đó suy ra $x^{t\phi(n)+1} \equiv x (\mathrm{mod}\, n)$, tức $\mathcal{D}(K'', \mathcal{E}(K', x)) = x$.
Thí dụ: Giả sử chọn $n = p.q$ = 2357.2551 = 6012707, ta sẽ có $\phi(n) = (p-1).(q-1)$=2356.2550 = 6007800. Chọn e = 3674911, và tính được d = 422191 sao cho $e.d \equiv 1 (\mathrm{mod}\, \phi(n))$. Một người dùng A có thể chọn khoá công khai là K'=(n =6012707, e = 3674911) và giữ khoá bí mật K''=d =422191. Một đối tác B muốn gửi cho A một thông báo x =5234673, sẽ dùng khoá công khai để tạo bản mật mã $y = x^e$ = $5234673^{3674911} \mathrm{mod} 6012707$ = 3650502. A nhận được y, giải mã sẽ được bản rõ $x = 3650502^{422191} \mathrm{mod}\ 6012707$ =5234673.

4.2.2. Thực hiện hệ mật mã RSA.

Để thực hiện hệ mật mã RSA cho một mạng truyền tin bảo mật, ngoài việc xây dựng các chương trình tính toán hàm $\mathcal{E}$ (với tham biến đầu vào là n ,e và x) và hàm $\mathcal{D}$ (với tham biến đầu vào là n ,d và y), ta còn phải chọn cho mỗi người tham gia một bộ (n,e,d) để tạo các khoá công khai K' và khoá bí mật K''. Hệ mã của mỗi người tham gia chỉ có khả năng bảo mật khi $n = p.q$ là số nguyên rất lớn (và do đó p,q cũng phải là những số nguyên tố rất lớn); rất lớn có nghĩa là p,q phải có biểu diễn thập phân cỡ hơn 100 chữ số, do đó n có cỡ hơn 200 chữ số thập phân, hay $n \geq 10^{200}$!

Tính toán các số e,d , hay thực hiện các hàm $\mathcal{E}$, $\mathcal{D}$, đều chủ yếu là thực hiện các phép tính số học trên các số nguyên rất lớn; về vấn đề này trong mấy chục năm qua, khoa lập trình máy tính đã đề xuất nhiều chương trình máy tính làm việc rất có hiệu quả, ta có thể tham khảo để sử dụng khi thực thi các hệ mật mã RSA cũng như nhiều hệ mật mã khác.

4.2.3. Tính bảo mật của mật mã RSA.

Bài toán thám mã (khi chỉ biết bản mã) đối với mật mã RSA là: biết khoá công khai K'=(n,e), biết bản mã $y = x^e \mathrm{mod} n$, tìm x. Bài toán này chính là bài toán RSA được trình bày trong mục 4.1.2. Trong mục đó ta đã chứng tỏ rằng nếu biết hai thừa số p,q của n thì dễ tìm được x từ y, và nói chung có bằng chứng để coi rằng bài toán RSA (hay bài toán thám mã RSA) là có độ khó tương đương với bài toán phân tích số nguyên (Blum) thành thừa số nguyên tố. Do đó, giữ tuyệt mật khoá bí mật d , hay giữ tuyệt mật các thừa số p,q , là có ý nghĩa rất quyết định đến việc bảo vệ tính an toàn của hệ mật mã RSA.

Một mạng truyền tin bảo mật sử dụng sơ đồ các hệ mật mã RSA được xem là an toàn, nếu tuân thủ các điều kiện cơ bản: mỗi

người tham gia phải độc lập lựa chọn các tham số n, e,d của riêng mình, chọn n cũng có nghĩa là chọn các thừa số p,q của n (n =$p.q$), và do có p,q nên tính được ϕ (n) = (p -1).(q -1), và từ đó tìm được e,d tương đối dễ dàng; nhưng cũng chính vì vậy mà sau khi đã chọn thì mỗi người tham gia phải giữ tuyệt đối bí mật các giá trị p,q,d , chỉ công bố khoá công khai (n,e) mà thôi.

Tuy nhiên, đó là điều kiện chung, còn trong thực tế vẫn có thể còn nhiều sơ hở mà người thám mã có thể lợi dụng để tấn công vào tính bảo mật của các hệ mã RSA khó mà lường trước hết được; sau đây là một số trường hợp đơn giản đã biết mà ta cần chú ý:

1.*Dùng môđuyn n chung*. Giả sử có hai người tham gia A và B cùng sử dụng một môđuyn chung n trong khoá công khai của mình, chẳng hạn A chọn khoá công khai (n,e) và giữ khoá bí mật d, B chọn khoá công khai (n,a) và giữ khoá bí mật b. Một người tham gia thứ ba C gửi một văn bản cần bảo mật x đến cả A và B thì dùng các khoá công khai nói trên để gửi đến A bản mật mã $y = x^e \bmod n$ và gửi đến B bản mật mã $z = x^a \bmod n$. Ta sẽ chứng tỏ rằng một người thám mã O có thể dựa vào những thông tin n,e,a,y,z trên đường công khai mà phát hiện ra bản rõ x như sau:

a. Tính $c = e^{-1} \bmod a$,
b. Sau đó tính $h = (ce - 1)/a$,
c. Và ta được $x = y^c (z^h)^{-1} \bmod n$.

Thực vậy, theo định nghĩa trên, ce -1 chia hết cho a, và tiếp theo ta có: $y^c (z^h)^{-1} \bmod n = x^{ec} . (x^{a(ce-1)/a})^{-1} \bmod n = x^{ce} . (x^{ce-1})^{-1} \bmod n = x$. Như vậy, trong trường hợp này việc truyền tin bảo mật không còn an toàn nữa. Vì vậy, ta cần nhớ khi dùng các hệ RSA để tổ chức mạng truyền tin bảo mật, cần tránh dùng môđuyn n chung cho các người tham gia khác nhau!

2. *Dùng số mũ lập mã e bé.* Để cho việc tính toán hàm lập mã được hiệu quả, ta dễ có xu hướng chọn số mũ e của hàm lập mã là một số nguyên bé, chẳng hạn e =3. Tuy nhiên, nếu trong một mạng truyền tin bảo mật dùng các hệ mật mã RSA, nếu có nhiều người cùng chọn số mũ lập mã e bé giống nhau thì sẽ có nguy cơ bị tấn công bởi việc thám mã như sau : Giả sử có ba người tham gia chọn ba khoá công khai là (n_1, e), (n_2, e), (n_3, e) với cùng số mũ e =3. Một người tham gia A muốn gửi một thông báo x cho cả ba người đó, và để bảo mật, gửi bản mã $c_i = x^3 \bmod n_i$ cho người thứ i. Ba môđuyn n_i là khác nhau, và có phần chắc là từng cặp nguyên tố với nhau. Một người thám mã có thể dùng định lý số dư Trung quốc để tìm một số m ($0 \le m \le n_1 n_2 n_3$) thoả mãn

$$\begin{cases} m \equiv c_1 \bmod n_1 \\ m \equiv c_2 \bmod n_2 \\ m \equiv c_3 \bmod n_3 \end{cases}$$

Vì $x \le n_i$, nên $x^3 \le n_1 n_2 n_3$, do đó ắt có $m = x^3$. Vậy là ta đã đưa được bài toán tìm căn bậc ba theo nghĩa đồng dư modn_i về bài toán tìm căn bậc ba theo nghĩa số học thông thường: tìm căn bậc ba của m ta được x, tức được bản rõ!

Với những lý do khác, người ta đã có những bằng chứng để chứng tỏ rằng hệ RSA cũng không bảo đảm an toàn nếu ta dùng các khoá có số mũ giải mã d là số nguyên bé, dù rằng khi đó thuật toán giải mã có làm việc hiệu quả hơn. Vì thế, khi sử dụng các hệ mật mã RSA, để bảo đảm an toàn ta nên chọn các số mũ e và d là những số nguyên lớn, có kích cỡ lớn gần như bản thân số n.

3. *Lợi dụng tính nhân của hàm lập mã*. Ta chú ý rằng hàm lập mã $f(x) = x^e \bmod n$ có tính nhân (multiplicative property), nghĩa là $f(x.y) = f(x).f(y)$. Dựa vào tính chất đó, ta thấy rằng nếu c là mật mã của bản rõ x, thì $\overline{c} = c.u^e \bmod n$ sẽ là mật mã của bản rõ xu. Do đó, khi lấy được bản mật mã c, để phát hiện bản rõ x người thám mã có thể chọn ngẫu nhiên một số u rồi tạo ra bản mã $\overline{c}$, và nếu người thám mã có khả năng thám mã theo kiểu « có bản mã được chọn » (xem 1.5.1), tức có khả năng với $\overline{c}$ được chọn tìm ra bản rõ tương ứng là $\overline{x} = xu$, thì bản rõ gốc cần phát hiện sẽ là $x = \overline{x}.u^{-1} \bmod n$. Tất nhiên, khả năng người thám mã có năng lực giải quyết bài toán thám mã theo kiểu có bản mã được chọn là rất hiếm, nhưng dầu sao đấy cũng là một trường hợp mà vấn đề bảo mật dễ bị tấn công, ta không thể không tính đến để tìm cách tránh!

4. *Tấn công bằng cách lặp phép mã*. Ta cũng chú ý rằng hàm lập mã $f(x) = x^e \bmod n$ là một phép hoán vị trên tập $Z_n = \{0,1,...,n-1\}$, do đó với mọi $c \in Z_n$ nếu ta thực hiện lặp phép lập mã để được

$$c_0 = c, c_1 = c^e \bmod n, c_2 = c^{e^2} \bmod n, ..., c_i = c^{e^i} \bmod n, ...$$

ắt sẽ tìm được số $k \ge 1$ sao cho $c_k = c^{e^k} \bmod n = c$. Nếu c là bản mã của một bản rõ x nào đó, $c = x^e \bmod n$, thì người thám mã có thể xuất phát từ c thực hiện lặp phép lập mã như trên sẽ tìm được số $k \ge 1$ bé nhất sao cho $c_k = c$. Và khi đó ta sẽ có số hạng trước đó $c_{k-1} = x$, là bản rõ cần phát hiện. Thuật toán về hình thức là khá đơn giản, nhưng hiệu quả thực hiện không đáng hy vọng lắm, vì số phép lặp cần thực hiện nói chung có thể là rất lớn, cỡ bằng số các phép hoán vị trên Z_n, tức là bằng $n!$, với số n có khoảng 200 chữ số thập phân. Trên thực tế, phỏng theo thuật toán nói trên ta có thể dễ dàng có một thuật toán phân tích n thành thừa số nguyên tố, mà một thuật

toán như vậy làm việc có hiệu quả thiết thực, như đã trình bày trong một phần trên, là chưa có! Vì vậy, nguy cơ bị thám mã bằng thuật toán đơn giản nói trên đối với tính an toàn của hệ mật mã RSA là không đáng ngại lắm.

5. *Về khả năng che giấu của bản mật mã*. Mật mã, sở dĩ nó giữ được bí mật, là do khả năng che giấu thông tin của nó, tức là biết bản mã y khó lòng tìm được thông tin nào để phát hiện ra bản rõ x. Một cách thô thiển, ta nói bản rõ x là *không che giấu được* qua phép lập mật mã RSA $e_K(x) = x^e \bmod n$, nếu $e_K(x) = x$. Nói cách khác, x là không che giấu được nếu bản mã của x cũng chính là x. Tiếc rằng với bất kỳ hệ mật mã RSA nào cũng có những bản rõ không che giấu được, đó là những bản rõ $x = -1, 0, 1 \bmod n$ (vì số mũ e luôn luôn là số lẻ). Người ta chứng minh được rằng nếu $n = p.q$, thì số các bản rõ $x \in Z_n$ không che giấu được là bằng

$$(1+\gcd(e-1, p-1)).(1+\gcd(e-1, q-1)).$$

Vì $e-1$, $p-1$, $q-1$ là các số chẵn, nên số đó ít nhất là 9, nên mỗi hệ RSA có ít nhất 9 bản rõ không che giấu được. Tuy nhiên, thường n, và do đó cả p và q, đều rất lớn, nên tỷ lệ các bản rõ không che giấu được nói chung là bé không đáng kể, và do đó khả năng gặp các bản rõ không che giấu được không tạo nên một nguy cơ đáng kể nào đối với việc dùng các hệ mật mã RSA.

4.3. Hệ mật mã khoá công khai Rabin.

4.3.1. Mô tả hệ mật mã Rabin.

Sơ đồ hệ mật mã khoá công khai Rabin được cho bởi

$$\mathcal{S} = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D}),$$

trong đó: $\mathcal{P} = \mathcal{C} = Z_n$, trong đó n là một số nguyên Blum, $n = p.q$, với p và q là hai số nguyên tố có tính chất $p \equiv 3 (\text{mod} 4)$, $q \equiv 3 (\text{mod} 4)$,

$$\mathcal{K} = \{K = (K', K'') : K' = (n, B), K'' = (p, q), 0 \le B \le n-1\},$$

các thuật toán $\mathcal{E}$ và $\mathcal{D}$ được xác định bởi

$$\mathcal{E}(K', x) = x(x+B) \bmod n,$$

$$\mathcal{D}(K'', y) = \sqrt{\frac{B^2}{4} + y} - \frac{B}{2} \bmod n.$$

(ký hiệu căn bậc hai sẽ được giải thích sau).

Trong một mạng truyền tin bảo mật với sơ đồ mật mã Rabin, mỗi người tham gia chọn cho mình các yếu tố n,B,p,q để lập nên khoá công khai và khoá bí mật của mình.

Ta chú ý rằng với mỗi bộ khoá K, các thuật toán $e_{K'} = \mathcal{E}(K',.)$ và $d_{K''} = \mathcal{D}(K'',.)$ không lập thành một cặp song ánh, cụ thể là $e_{K'}$ không phải là một đơn ánh, vì nếu w là một căn bậc hai của 1 theo mod$n$ thì $e_{K'}(w(x+\frac{B}{2}) - \frac{B}{2}) = e_{K'}(x)$, mà ta có đến 4 căn bậc hai của 1 theo mod$n$,tức là ta có 4 giá trị khác nhau của đối số x cho cùng một giá trị $e_{K'}(x)$.

Bây giờ nói đến thuật toán giải mã $d_{K''} = \mathcal{D}(K'',.)$. Đặt $C = B^2/4 + y$, ta có $d_{K''}(y) = \sqrt{C} - B/2 \bmod n$, do đó để có $d_{K''}(y)$, ta cần tính $\sqrt{C} \bmod n$, tức cần giải phương trình $z^2 \equiv C \bmod n$. Phương trình đó tương đương với hệ thống gồm hai phương trình sau đây:

$$\begin{cases} z^2 \equiv C \bmod p, \\ z^2 \equiv C \bmod q. \end{cases} \qquad (2)$$

Vì p và q là các số nguyên tố nên ta có $C^{\frac{p-1}{2}} \equiv 1 \bmod p$, $C^{\frac{q-1}{2}} \equiv 1 \bmod q$. Theo giả thiết, $p \equiv 3 (\text{mod} 4)$ và $q \equiv 3 (\text{mod} 4)$, nên $\frac{p+1}{4}$ và $\frac{q+1}{4}$ là các số nguyên; và ta có

$$(\pm C^{\frac{p+1}{4}})^2 \equiv C (\text{mod}\, p),\ (\pm C^{\frac{q+1}{4}})^2 \equiv C (\text{mod}\, q).$$

Do đó,phương trình $z^2 \equiv C \bmod n$, hay hệ phương trình (2), có 4 nghiệm theo mod$n$, tương ứng với 4 hệ phương trình sau đây :

$$\begin{cases} z \equiv C^{(p+1)/4} (\text{mod}\, p) \\ z \equiv C^{(q+1)/4} (\text{mod}\, q) \end{cases} \quad \begin{cases} z \equiv C^{(p+1)/4} (\text{mod}\, p) \\ z \equiv -C^{(q+1)/4} (\text{mod}\, q) \end{cases}$$

$$\begin{cases} z \equiv -C^{(p+1)/4} (\text{mod}\, p) \\ z \equiv C^{(q+1)/4} (\text{mod}\, q) \end{cases} \quad \begin{cases} z \equiv -C^{(p+1)/4} (\text{mod}\, p) \\ z \equiv -C^{(q+1)/4} (\text{mod}\, q) \end{cases}$$

Cả 4 nghiệm của 4 hệ phương trình đó theo mod$n$ đều được viết chung dưới một ký hiệu là $\sqrt{C} \bmod n$, và vì vậy thuật toán giải mã $d_{K''}(y)$ thực tế sẽ cho ta 4 giá trị khác nhau theo mod$n$ mà bản rõ là một trong 4 giá trị đó. Việc chọn giá trị nào trong 4 giá trị tìm được làm bản rõ là tuỳ thuộc vào những đặc trưng khác của bản rõ mà người giải mã nhận biết (thí dụ bản rõ dưới dạng số phải có biểu diễn nhị phân là mã của một văn bản tiếng Anh thông thường).

Thí dụ : Giả sử n =77 = 7.11, B =9 (ở đây p =7, q =11). Ta có

$$e_{K'}(x) = x^2 + 9x \bmod 77,$$

$$d_{K''}(y) = \sqrt{1+y} - 43 \bmod 77,$$

vì 2^{-1}=39mod77, 9.2^{-1} =9.39 =43mod77, B^2=4mod77, $B^2/4$ =1mod 77.

Với x =44 ta có $e_{K'}(x)$ = 44^2+9.44 =2332 =22mod77, bản mã tương ứng với x là y = 22. Bây giờ giải mã với bản mã y =22, bằng thủ tục nói trên ta có thể tìm được 4 giá trị của $\sqrt{1+y} = \sqrt{1+22} = \sqrt{23}$ theo mod77 là 10,67,32,45, từ đó 4 giá trị có thể có của $d_{K''}(y)$ là

$$d_{K''}(y) = 44, 24, 66, 2.$$

Bản rõ nằm trong 4 giá trị đó, trong trường hợp này là 44.

4.3.2. Tính an toàn của hệ mật mã Rabin.

Trong định nghĩa của hệ mật mã Rabin, khoá công khai là (n,B), khoá bí mật là (p,q) tức là cặp thừa số nguyên tố của n . Như vậy, tính an toàn của hệ mật mã nằm ở việc giữ bí mật các thừa số p và q. Định nghĩa của phép giải mã cũng cho ta thấy rằng yếu tố có ý nghĩa quyết định trong phép giải mã là việc tính căn bậc hai của một số theo mod$n$. Trong mục 4.1.2 bài toán tìm căn bậc hai theo mod$n$ (với n là hợp số Blum) đã được chứng tỏ là có độ khó tương đương với bài toán phâ n tích n thành thừa số nguyên tố. Vì vậy, bài toán giải mã đối với hệ mật mã Rabin, cũng là bài toán giữ bí mật khoá bí mật (p,q), và bài toán phân tích số nguyên thành thừa số nguyên tố là có độ khó tương đương nhau. Và đó cũng là yếu tố bảo đảm tính an toàn của hệ mật mã Rabin !

4.4. Hệ mật mã khoá công khai ElGamal.

4.4.1. Mô tả hệ mật mã ElGamal.

Hệ mật mã ElGamal được T. ElGamal đề xuất năm 1985, dựa vào độ phức tạp của bài toán tính lôgarit rời rạc, và sau đó đã nhanh chóng được sử dụng rộng rãi không những trong vấn đề bảo mật truyền tin mà còn trong các vấn đề xác nhận và chữ ký điện tử.

Sơ đồ hệ mật mã khoá công khai ElGamal được cho bởi

$$\mathcal{S} = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D}),$$

trong đó: $\mathcal{P} = Z_p^*$, $\mathcal{C} = Z_p^* \times Z_p^*$, với p là một số nguyên tố;

$$\mathcal{K} = \{K = (K', K'') : K' = (p, \alpha, \beta),\ K'' = a,\ \beta \equiv \alpha^a \bmod p\},$$

ở đây α là một phần tử nguyên thuỷ theo mod$p$, tức của Z_p^*.

Các thuật toán lập mã $e_{K'} = \mathcal{E}(K',.)$ và giải mã $d_{K''} = \mathcal{D}(K'',.)$ được xác định như sau: Với mỗi $x \in \mathcal{P} = Z_p^*$, để lập mật mã cho x trước hết ta chọn thêm một số ngẫu nhiên $k \in Z_{p-1}$ rồi tính:

$$e_{K'}(x,k) = (y_1, y_2), \quad \text{với} \quad \begin{cases} y_1 = \alpha^k \bmod p, \\ y_2 = x.\beta^k \bmod p. \end{cases}$$

Với mọi số ngẫu nhiên k bất kỳ, ta đều xem $e_{K'}(x,k)$ là mật mã của x. Và thuật toán giải mã được xác định bởi

$$d_{K''}(y_1, y_2) = y_2.(y_1^a)^{-1} \bmod p.$$

Các phép lập mật mã và giải mã được xác định như vậy là hợp thức, vì ta có với mọi $x \in \mathcal{P} = Z_p^*$ và mọi $k \in Z_{p-1}$:

$$d_{K''}(e_{K'}(x,k)) = x.\beta^k.(\alpha^{k.a})^{-1} \bmod p = x.\beta^k.\beta^{-k} \bmod p = x.$$

Ta chú ý rằng trong một mạng truyền thông bảo mật với việc dùng sơ đồ mật mã ElGamal, mỗi người tham gia tự chọn cho mình các tham số $p, \alpha, a,$ rồi tính β, sau đó lập và công bố khoá công khai $K' = (p, \alpha, \beta)$, nhưng phải giữ tuyệt mật khoá bí mật $K'' = a$. Bài toán biết khoá công khai tìm ra khoá bí mật chính là bài toán tính lôgarit rời rạc được kể đến trong mục 4.1.2, một bài toán khó cho đến nay chưa có một thuật toán nào làm việc trong thời gian đa thức giải được nó.

Thí dụ : Chọn p = 2579, α =2, a =765, ta tính được $\beta = 2^{765} = 949$ mod2579. Ta có khoá công khai (2579, 2, 949) và khoá bí mật 765. Giả sử để lập mật mã cho x =1299, ta chọn ngẫu nhiên k =853, sẽ có

$$e_{K'}(1299, 853) = (2^{853}, 1299.\ 949^{853}) \bmod 2579$$
$$= (453, 2396).$$

Và giải mã ta được lại

$$d_{K''}(453, 2396) = 2396.\ (453^{765})^{-1} \bmod 2579 = 1299.$$

4.4.2. Tính an toàn của hệ mật mã ElGamal.

Như đã trình bày ở trên, nếu ta xem tính an toàn của hệ mật mã ElGamal là ở việc giữ tuyệt mật khoá bí mật K'', thì ta có thể yên tâm vì bài toán phát hiện khoá bí mật có độ khó tương đương với bài toán tính lôgarit rời rạc, mà bài toán này thì như ở các mục 4.1.2 và 2.4.3 đã chứng tỏ, cho đến nay chưa có một thuật toán nào làm việc trong thời gian đa thức giải được nó. Có một điều cảnh báo là nên chú ý chọn môđuyn p là số nguyên tố sao cho p -1 có ít nhất một ước số nguyên tố lớn (xem 2.4.3). Điều đó là thực hiện được

nếu số nguuyên tố p được chọn là số nguyên tố Sophie Germain (tức có dạng $2q+1$, với q cũng là số nguyên tố lớn).

Ngoài ra, còn có khả năng khoá bí mật $K'' = a$ bị lộ do cẩu thả trong việc sử dụng số ngẫu nhiên k, đặc biệt là khi *để lộ số k được dùng*. Thực vậy, nếu để lộ số k, thì khoá bí mật a được tính ra ngay theo công thức sau đây:

$$a = (x - ky_2)y_1^{-1} \bmod (p-1).$$

Như vậy,một người thám mã có khả năng tấn công theo kiểu "biết cả bản rõ" (xem 1.5.1) có thể phát hiện ra khoá a nếu biết k.

Một trường hợp khác làm mất tính an toàn của hệ mật mã ElGamal là việc *dùng cùng một số k cho nhiều lần lập mật mã*. Thực vậy, giả sử dùng cùng một số ngẫu nhiên k cho hai lần lập mã, một lần cho x_1, một lần cho x_2, và được các bản mã tương ứng (y_1,y_2) và (z_1,z_2). Vì cùng dùng một số k nên $y_1=z_1$. Và do đó theo công thức lập mã ta có $z_2/y_2 = x_2/x_1$, tức là $x_2 = x_1.z_2/y_2$. Như vậy, một người thám mã, một lần "biết cả bản rõ" dễ dàng phát hiện được bản rõ trong các lần sau.

4.4.3. Các hệ mật mã tương tự ElGamal.

Hệ mật mã ElGamal được xây dựng dựa trên các yếu tố: một nhóm hữu hạn cyclic (Z_p^*), một phần tử nguyên thuỷ $(\alpha \in Z_p^*)$ sao cho bài toán tính lôgarit rời rạc (tính $a = \log_\alpha \beta$, tức cho β tìm a sao cho $\beta = \alpha^a \bmod p$) là rất khó thực hiện. Vì vậy, nếu có đủ các yếu tố đó thì ta có thể xây dựng các hệ mật mã tương tự ElGamal. Như vậy, sơ đồ của một hệ mật mã tương tự ElGamal được cho bởi

$$\mathcal{S} = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D}),$$

trong đó: $\mathcal{P}=G$, $\mathcal{C}=G\times G$, với G là một nhóm cyclic hữu hạn;

$$\mathcal{K} = \{K = (K', K'') : K' = (G,\alpha,\beta),\ K'' = a,\ \beta = \alpha^a\},$$

ở đây α là một phần tử nguyên thuỷ của nhóm G.

Các thuật toán lập mã $e_{K'} = \mathcal{E}(K',.)$ và giải mã $d_{K''} = \mathcal{D}(K'',.)$ được xác định như sau: Với mỗi $x\in\mathcal{P}=G$, để lập mật mã cho x trước hết ta chọn thêm một số ngẫu nhiên k $(0 \le k \le |G|)$ rồi tính:

$$e_{K'}(x,k) = (y_1, y_2), \quad \text{với} \quad \begin{cases} y_1 = \alpha^k \\ y_2 = x.\beta^k \end{cases}$$

Với mọi số ngẫu nhiên k bất kỳ, ta đều xem $e_{K'}(x,k)$ là mật mã của x. Và thuật toán giải mã được xác định bởi

$$d_{K''}(y_1, y_2) = y_2.(y_1^a)^{-1} \bmod p.$$

Phép nhân trong các biểu thức nói trên đều là phép nhân của G.

Có hai lớp nhóm thường được sử dụng để xây dựng các hệ mật mã tương tự ElGamal là nhóm nhân của trường Galois GF(p^n) và nhóm cộng của một đường cong elliptic xác định trên một trường hữu hạn.

1. *Nhóm nhân của trường Galois GF(p^n)*: Trường Galois GF(p^n) là trường của các đa thức với hệ số trong Z_p lấy theo môđuyn là một đa thức bậc n bất khả qui; với phép cộng và phép nhân là phép cộng và phép nhân đa thức theo môđuyn đó. Trường có p^n phần tử, có thể xem mỗi phần tử là một đa thức bậc n -1 với hệ số thuộc Z_p = {0,1,2,...,p -1}, thậm chí là một vectơ n chiều mà các thành phần là các hệ số của đa thức đó. Tập tất cả các đa thức khác 0 lập thành nhóm nhân của trường $GF(p^n)$,và người ta chứng minh được rằng nhóm nhân đó là cyclic.

Như vậy, nhóm $G = GF(p^n)\setminus\{0\}$ là nhóm cyclic cấp p^n-1. ta có thể chọn một phần tử nguyên thuỷ của nhóm đó, và thiết lập bài toán lôgarit rời rạc tương ứng, từ đó xây dựng được hệ mật mã tương tự ElGamal.

2. *Nhóm công của đường cong elliptic* : Giả sử p là một số nguyên tố > 3. *Đường cong elliptic* $y^2=x^3+a.x+b$ *trên* Z_p **,** trong đó $a,b \in Z_p$ là các hằng số thoả mãn $4a^3+27b^2 \not\equiv 0 \pmod p$, được định nghĩa là tập hợp tất cả các điểm $(x,y) \in Z_p \times Z_p$ thoả mãn phương trình

$$y^2 \equiv x^3+a.x+b \pmod p,$$

cùng với một phần tử đặc biệt mà ta ký hiệu là $\mathcal{O}$. Tập hợp đó được ký hiệu là E. Trên tập E ta xác định một phép cộng như sau : Giả sử $P=(x_1, y_1)$ và $Q = (x_2, y_2)$ là hai điểm của E. Nếu $x_1=x_2$ và $y_1= -y_2$ thì ta định nghĩa $P+Q=\mathcal{O}$; nếu không thì $P+Q = (x_3, y_3)$, trong đó

$$x_3 = \lambda^2-x_1-x_2\,, \quad y_3 = \lambda\,(x_1-x_3) - y_1\,,$$

với

$$\lambda = \begin{cases} (y_2 - y_1)/(x_2 - x_1), & khi\ P \neq Q; \\ (3x_1^2 + a)/2y_1, & khi\ P = Q. \end{cases}$$

Ngoài ra, ta định nghĩa thêm : $P+\mathcal{O} = \mathcal{O}+P = P$.

Tập E với phép toán cộng đó lập thành một nhóm. Nếu $|E| = q$ là số nguyên tố thì nhóm cộng đó là nhóm cyclic, và mọi phần tử khác không ($\neq\mathcal{O}$) đều là phần tử nguyên thuỷ. Ta nhớ rằng trong trường hợp này, phần tử nghịch đảo là phần tử đối, phép nâng lên luỹ thừa n là phép nhân với số n , phép lôgarit tương ứng với một kiểu phép chia. Ta có thể xuất phát từ nhóm E này để xây dựng hệ mật mã tương tự ElGamal.

4.5. Các hệ mật mã dựa trên các bài toán $\mathcal{NP}$-đầy đủ.

4.5.1. Nguyên tắc chung.

Như đã giới thiệu trong chương II, các bài toán $\mathcal{NP}$-đầy đủ là các bài toán mà cho đến nay chưa tìm được một thuật toán với độ phức tạp tính toán đa thức nào để giải chúng. Và tính « khó» của các bài toán đó lại được bảo đảm bằng sự kiện là chỉ cần có một thuật toán với độ phức tạp đa thức giải một bài toán $\mathcal{NP}$-đầy đủ nào đó thì lập tức mọi bài toán $\mathcal{NP}$-đầy đủ đều giải được trong thời gian đa thức.

Đối với một số bài toán $\mathcal{NP}$-đầy đủ, tuy không có thuật toán với độ phức tạp đa thức để giải đối với mọi dữ liệu của bài toán, nhưng có thể có một lớp các dữ liệu mà đối với chúng có thuật toán để giải với thời gian chấp nhận được. Với những bài toán như vậy ta có thể sử dụng để xây dựng các hệ mật mã khoá công khai với nguyên tắc chung như sau : Hệ mật mã sẽ có phép giải mã tương đương với việc tìm lời giải cho bài toán $\mathcal{NP}$-đầy đủ đó; tuy nhiên có một thủ tục để biến một dữ liệu nói chung của bài toán $\mathcal{NP}$-đầy đủ đó thành một dữ liệu thuộc lớp đặc biệt mà đối với nó có thể giải được bởi một thuật toán với độ phức tạp thời gian chấp nhận được. Như vậy, ta đã biến được phép lập mã thành một hàm « cửa sập một phía », và đó là cơ sở để xây dựng hệ mật mã khoá công khai tương ứng.

Ta sẽ xét sau đây hai trường hợp xây dựng được các hệ mật mã khoá công khai theo cách như vậy : một là hệ mật mã Merkle-Hellman dựa trên bài toán sắp ba lô (hay bài toán tổng tập con), và hai là hệ mật mã Mc-Eliece dựa trên bài toán giải mã tuyến tính tự sửa sai.

4.5.2. Hệ mật mã Merkle-Hellman.

Bài toán sắp ba lô (tức bài toán KNAPSACK, cũng được gọi là bài toán tổng tập con) được đặt ra như sau: Cho một tập các số nguyên dương $\{a_1, a_2, ..., a_n\}$ và một số nguyên dương s. Hãy xác định xem có hay không một tập con các a_j mà tổng của chúng bằng s. Một cách tương đương, hãy xác định xem có hay không các $x_i \in \{0,1\}$ $(1 \le i \le n)$ sao cho $\sum_{i=1}^{n} a_i x_i = s$.

Bài toán này là $\mathcal{NP}$-đầy đủ, tuy nhiên nếu ta hạn chế bài toán trên các dữ liệu $I = (\{a_1, a_2, ..., a_n\}, T)$, trong đó $\{a_1, a_2, ..., a_n\}$ là dãy *siêu tăng*, tức là dãy thoả mãn điều kiện

$$\forall j = 2, 3, ..., n: \; a_j > \sum_{i=1}^{j-1} a_i,$$

thì việc tìm trả lời là khá dễ dàng, chẳng hạn có thể bằng thuật toán đơn giản dưới đây:

1. **for** $i = n$ **downto** 1 **do**
   if $T > a_i$ **then** $T = T - a_i$, $x_i = 1$, **else** $x_i = 0$
2. **if** $\sum_{i=1}^{n} x_i.a_i = T$ **then** $X = (x_1, ..., x_n)$ is the solution of problem,
   else there is no solution.

Bây giờ, để chuẩn bị xây dựng một sơ đồ mật mã Merkle-Hellman, ta chọn trước một số nguyên dương n và một số nguyên tố p đủ lớn. Với mỗi người tham gia sẽ được chọn một bộ khoá $K = (K', K'')$, trong đó khoá bí mật $K'' = (A, p, a)$ gồm một dãy siêu tăng $A = \{a_1, a_2, ..., a_n\}$ thoả mãn $\sum_{i=1}^{n} a_i < p$, và một số a, $1 < a < p$; khoá công khai $K' = \{b_1, ..., b_n\}$ với $b_i = a.a_i \bmod p$.

Sơ đồ hệ mật mã Merkle-Hellman được định nghĩa bởi

$$\mathcal{S} = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D}),$$

trong đó $\mathcal{P} = \{0,1\}^n$, $\mathcal{C} = \{0, 1, ..., n(p-1)\}$, $\mathcal{K}$ là tập các bộ khoá $K = (K', K'')$ như được xây dựng ở trên. Các thuật toán lập mật mã và giải mã được xác định bởi:

Với mọi $x = (x_1, ..., x_n) \in \mathcal{P}$ thuật toán lập mã cho ta

$$\mathcal{E}(K', x) = \sum_{i=1}^{n} x_i.b_i \; ;$$

và với mọi $y \in \mathcal{C}$, ta tính $z = a^{-1}.y \bmod p$, rồi sau đó giải bài toán sắp balô đối với dữ liệu $I = (\{a_1, a_2, ..., a_n\}, z)$ ta sẽ được lời giải $(x_1, ..., x_n)$, lời giải đó là giá trị của $\mathcal{D}(K'', y)$.

Thí dụ: Chọn $n = 6$, khoá bí mật có $p = 737$, $A = \{12, 17, 33, 74, 157, 316\}$, $a = 635$. Tính được khoá công khai là {250, 477, 319, 559, 200, 196}. Với bản rõ $x = 101101$ ta có bản mã tương ứng là $y = 1324$. Để giải mã, trước hết tính $z = a^{-1}.y \bmod p = 635^{-1}.1324 \bmod 737 = 435$, sau đó giải bài toán sắp balô với dãy siêu tăng A và z ta được

$$435 = 12 + 33 + 74 + 316,$$

tức được lời giải $x = (1,0,1,1,0,1)$.

Hệ mật mã Merkle-Hellman được đề xuất khá sớm, từ năm 1978, đến năm 1985 Shamir tìm được một phương pháp thám mã trong thời gian đa thức dựa vào một thuật toán của Lenstra giải bài toán qui hoạch động. Tuy nhiên, sau đó, vào năm 1988, Chor và Rivest có đưa ra một cách khác xây dựng hệ mật mã cũng dựa vào bài toán sắp balô, cho đến nay vẫn giữ được an toàn.

4.5.3. Hệ mật mã McEliece.

Hệ mật mã McEliece được xây dựng dựa vào tính $\mathcal{NP}$-đầy đủ của bài toán giải mã tuyến tính tự sửa sai (trong lý thuyết truyền tin). Bài toán được đặt ra như sau: giả sử nguồn tin là tập các từ k bit nhị phân, tức tập hợp $\{0,1\}^k$, được truyền đi trên một kênh có nhiễu, tức là nếu truyền trực tiếp các dãy từ k bit thì thông tin mà ta nhận được có thể bị sai lệch và ta không nhận được đúng thông tin được truyền đi. Để khắc phục những sai lệch đó người ta tìm cách mã hoá nguồn tin gốc bằng cách thêm cho mỗi từ k bit mang thông tin một số bit dùng để tự hiệu chỉnh, tức là thực hiện một phép mã hoá biến mỗi từ k bit ban đầu thành một từ n bit, với $n > k$, được gọi là từ mã. Phép mã hoá tuyến tính là phép mã hoá được thực hiện bằng cách nhân từ k bit ban đầu x với một ma trận G cấp $k \times n$ để được từ mã n bit y, $y = x.G$ (các phép toán cộng và nhân được thực hiện theo mod2). Ta định nghĩa khoảng cách Hamming giữa hai từ mã n bit là số các vị trí mà tại đó hai từ mã có giá trị khác nhau; khoảng cách d của hệ mã là khoảng cách Hamming bé nhất giữa hai từ mã bất kỳ. Như vậy, một hệ mã tuyến tính được xác định bởi một ma trận G (gọi là ma trận sinh), và được đặc trưng bởi ba số $[n,k,d]$. Nếu $d = 2t+1$, thì hệ mã có khả năng tự sửa sai đến t sai ngẫu nhiên nhiễm phải do nhiễu của kênh truyền. Tuy nhiên, việc tự sửa sai (tức là khi nhận được từ mã có thể có đến t sai ta tìm lại được đúng từ k bit thông tin ban đầu) của các hệ mã tuyến tính như vậy nói chung khá phức tạp, và bài toán giải mã tuyến tính tự sửa sai đã được chứng minh là một bài toán $\mathcal{NP}$-khó, tức cho đến nay chưa biết có thuật toán nào làm việc trong thời gian đa thức giải được nó. Mặc dầu vậy, người ta đã tìm được một số lớp riêng các hệ mã tuyến tính mà đối với chúng có thể xây dựng được những thuật toán giải mã tự sửa sai làm việc có hiệu quả, các hệ mã Goppa là một lớp như vậy. Hệ mã Goppa là một loại hệ mã tuyến tính có các đặc trưng $n = 2^m$, $d = 2t+1$, $k = n - mt$, có ma trận sinh G cấp $k \times n$ được xây dựng dựa trên một số tính chất đại số của trường GF(2^m)-mà ở đây ta không đi vào các chi tiết.

Để có một hệ mật mã McEliece, trước hết ta chọn một hệ mã Goppa với ma trận sinh G và các đặc trưng trên, sau đó dùng một

ma trận S khả nghịch cấp $k \times k$ trên Z_2 và một ma trận hoán vị P cấp $n \times n$ (cũng có các phần tử trong Z_2) để biến hệ mã Goppa với ma trận sinh G thành một hệ mã tuyến tính "phổ biến" với ma trận sinh $G^* = SGP$; vậy là đã biến hệ mã Goppa có thuật toán giải mã hiệu quả thành một hệ mã tuyến tính nói chung mà ta chỉ biết việc giải mã tự sửa sai đối với nó là $\mathcal{NP}$-khó. Hệ mật mã mà ta xây dựng sẽ có thuật toán giải mã là "dễ" đối với người trong cuộc như giải mã Goppa, và là "khó" đối với người ngoài như giải mã tuyến tính nói chung!

Như vậy, một *hệ mật mã khoá công khai McEliece* được xác định bởi

$$\mathcal{S} = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D}),$$

trong đó $\mathcal{P} = \{0,1\}^k$, $\mathcal{C} = \{0,1\}^n$, $\mathcal{K}$ là tập hợp các bộ khoá $K = (K', K'')$, với khoá bí mật $K'' = (G,S,P)$ gồm một ma trận sinh G của một hệ mã Goppa, một ma trận khả nghịch S cấp $k \times k$ trên Z_2 và một ma trận hoán vị P cấp $n \times n$; khoá công khai $K' = G^*$ là ma trận "đã được biến đổi" nói trên.

Thuật toán lập mật mã $\mathcal{E}(K',.): \mathcal{P} \to \mathcal{C}$ được xác định bởi

$$\mathcal{E}(K', x) = x.G^* + \boldsymbol{e},$$

trong đó $\boldsymbol{e} \in \{0,1\}^n$ là một vectơ ngẫu nhiên có trọng số t, tức có t thành phần là 1. Thuật toán giải mã $\mathcal{D}(K'',.)$ được thực hiện theo ba bước như sau với mọi $y \in \mathcal{C} = \{0,1\}^n$:

1. Tính $y_1 = y.P^{-1}$,
2. Giải mã Goppa đối với y_1, giả sử được x_1.
3. Tính $\mathcal{D}(K'', y) = x_1.S^{-1}$.

Dễ thử lại rằng các thuật toán lập mật mã và giải mã xác định như trên là hợp thức, vì với mọi $x \in \mathcal{P} = \{0,1\}^k$, ta đều có

$$\mathcal{D}(K'', \mathcal{E}(K', x)) = x,$$

Đẳng thức đó đúng với mọi vectơ $\boldsymbol{e}$ bất kỳ có trọng số $\le t$. Hệ mật mã này cũng tương tự như hệ mật mã ElGamal ở chỗ khi lập mật mã ta có thể chọn thêm cho dữ liệu vào một yếu tố ngẫu nhiên; về sau ta sẽ gọi những hệ mật mã như vậy là hệ mật mã xác suất.

Yếu tố chủ yếu bảo đảm tính an toàn của các hệ mật mã McEliece là ở chỗ từ khoá công khai G^* khó phát hiện ra khoá bí mật (G,S,P) và ở tính $\mathcal{NP}$-khó của bài toán giải mã tuyến tính tự sửa sai nói chung. Cũng cần nhớ rằng độ an toàn còn phụ thuộc vào việc chọn các tham số k,n,t đủ lớn; theo gợi ý của các nghiên cứu thực nghiệm thì đủ lớn có nghĩa là $n \approx 1024$, $k \approx 644$, $t \approx 38$. Với những đòi hỏi đó thì kích cỡ của các ma trận G, S, P và G^* sẽ quá

lớn, khá bất tiện cho việc thực thi trong thực tế, vì vậy mà các hệ mật mã McEliece chưa được sử dụng phổ biến lắm.

4.6. Các hệ mật mã xác suất khoá công khai.

4.6.1. Đặt vấn đề và định nghĩa.

Mật mã xác suất là một ý tưởng được đề xuất bởi Goldwasser và Micali từ năm 1984, xuất phát từ yêu cầu giải quyết một vấn đề sau đây: Giả thiết ta có một hệ mật mã khoá công khai, và ta muốn lập mật mã cho bản rõ chỉ gồm một bit. Điều đó thường gặp khi ta muốn bí mật truyền đi một thông tin chỉ có nội dung là *có* hoặc *không*, tức là một thông tin đặc biệt quan trọng nhưng chỉ gồm một bit. Nếu ta dùng một hệ mật mã khoá công khai thông thường, thì bản mật mã được truyền đi sẽ là $e_{K'}(0)$ hoặc $e_{K'}(1)$, một người thám mã có thể không biết cách giải mã, nhưng lại hoàn toàn có thể tính trước các giá trị $e_{K'}(0)$ và $e_{K'}(1)$, và khi lấy được bản mã truyền đi trên kênh truyền tin công cộng, chỉ cần so sánh bản mã nhận được đó với hai bản $e_{K'}(0)$ và $e_{K'}(1)$ đã được tính sẵn là đủ biết được thông tin mật được truyền đi là 0 hay là 1. Các hệ mật mã khoá công khai sở dĩ có được tính bảo mật là vì từ thông tin về bản mã khó lòng khai thác được thông tin gì về bản rõ, nhưng rõ ràng điều đó không còn được bảo đảm nếu số các bản rõ là rất ít, chẳng hạn như khi các bản rõ có độ dài cực ngắn, hay như trường hợp trên, số các bản rõ chỉ là hai, cụ thể là 0 và 1.

Mục đích của việc xây dựng mật mã xác suất là để bảo đảm không một thông tin nào về bản rõ có thể khai thác được (trong thời gian đa thức) từ bản mã; điều này, đối với các hệ mật mã khoá công khai, có thể được thực hiện bằng cách tạo cho một bản rõ nhiều bản mã khác nhau thu được một cách ngẫu nhiên với việc sử dụng các số ngẫu nhiên trong tiến trình lập mã. Sau đây là định nghĩa về một hệ mật mã xác suất khoá công khai:

Định nghĩa. Một *hệ mật mã xác suất khoá công khai* được xác định bởi một bộ

$$\mathcal{S} = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D}, \mathcal{R}),$$

trong đó $\mathcal{P}$, $\mathcal{C}$, $\mathcal{K}$ được hiểu như đối với các hệ mật mã khoá công khai thông thường, $\mathcal{R}$ là một tập các phần tử ngẫu nhiên, và với mỗi $K = (K', K'') \in \mathcal{K}$, thuật toán lập mật mã $e_{K'} = \mathcal{E}(K', .): \mathcal{P} \times \mathcal{R} \to \mathcal{C}$ và giải mã $d_{K''} = \mathcal{D}(K'', .): \mathcal{C} \to \mathcal{P}$ thoả mãn đẳng thức:

với mọi $x \in \mathcal{P}$, $r \in \mathcal{R}$, $d_{K''}(e_{K'}(x,r)) = x$.

Ngoài ra, ta mong muốn một *điều kiện an toàn* như trong định nghĩa sau đây được thoả mãn: ta ký hiệu $p_{K,x}$ là phân bố xác

suất trên tập $\mathcal{C}$, trong đó $p_{K,x}(y)$ là xác suất của việc y là bản mã khi biết K là khoá và x là bản rõ (xác suất được tính cho tất cả $r \in \mathcal{R}$). Ta nói hai phân bố xác suất p_1 và p_2 trên $\mathcal{C}$ là ε-*phân biệt được* nếu có một thuật toán ε-phân biệt hai phân bố xác suất đó, tức là một thuật toán $A : \mathcal{C} \to \{0,1\}$ thoả mãn tính chất

$$| E_A(p_1) - E_A(p_2)| \geq \varepsilon,$$

trong đó

$$E_A(p_i) = \sum_{y \in \mathcal{C}} p_i(y).p(A(y) = 1).$$

Bây giờ *điều kiện an toàn* được phát biểu như sau: Hệ mật mã xác suất khoá công khai $\mathcal{S}$ là an toàn nếu có ε>0 sao cho với mọi $K \in \mathcal{K}$ và mọi $x \neq x'$, các phân bố xác suất $p_{K,x}$ và $p_{K,x}$ là không ε-phân biệt được.

4.6.2. Hệ mật mã xác suất Goldwasser-Micali.

Sau đây là mô tả sơ đồ của hệ mật mã xác suất khoá công khai trên tập văn bản một bit do Goldwasser và Micali đề xuất năm 1984. Một hệ như vậy được cho bởi một danh sách

$$\mathcal{S} = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D}, \mathcal{R}),$$

trong đó $\mathcal{P} = \{0,1\}$, $\mathcal{C} = \mathcal{R} = Z_n^*$, $n = p.q$ là tích của hai số nguyên tố lớn, $\mathcal{K}$ là tập hợp các bộ khoá $K = (K', K'')$, trong đó khoá công khai $K' = (n, m)$ với $m \in \widetilde{Q_n} = J_n - Q_n$ là một giả thặng dư bậc hai mod$n$, và khoá bí mật $K'' = (p,q)$. Các thuật toán lập mật mã và giải mã được xác định bởi

$$e_{K'}(x,r) = m^x.r^2 \bmod n,$$

$$d_{K''}(y) = \begin{cases} 0, \ khi \ y \in Q_n \\ 1, \ khi \ y \in \bar{Q}_n \end{cases}$$

với mọi $x \in \mathcal{P}$, $r \in \mathcal{R}$, $y \in \mathcal{C}$.

Hệ mật mã Goldwasser-Micali lập mật mã cho bản rõ một bit: mật mã của bit 0 luôn luôn là một thặng dư bậc hai mod$n$, và mật mã của bit 1 là một giả thặng dư bậc hai mod$n$. Việc giải mã là khá dễ dàng khi ta biết khoá bí mật $K'' = (p,q)$. Thực vậy, với mọi $y \in Q_n \cup \bar{Q}_n$ ta có $\left(\frac{y}{n}\right) = 1$. Vì biết $K'' = (p,q)$, nên ta tính được

$$\left(\frac{y}{p}\right) = y^{\frac{p-1}{2}} \bmod p,$$

và do đó dễ thử được $y \in Q_n \Leftrightarrow \left(\frac{y}{p}\right) = 1$, và tính được $d_{K''}(y)$.

4.6.3.Hệ mật mã xác suất Blum-Goldwasser.

Hệ mật mã xác suất khoá công khai Blum-Goldwasser được xây dựng trên nền của các hệ mật mã theo dòng với dòng khoá là dãy số giả ngẫu nhiên Blum-Blum-Shub (xem 3.3.3), yếu tố ngẫu nhiên $r \in \mathcal{R}$ ở đây sẽ được sử dụng như mầm sinh ra dãy số giả ngẫu nhiên của dòng khoá đó. Sơ đồ của *hệ mật mã xác suất khoá công khai Blum-Goldwasser* được cho bởi danh sách

$$\mathcal{S} = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D}, \mathcal{R}),$$

trong đó $\mathcal{P} = Z_2^*$, $\mathcal{C} = Z_2^* \times Z_n$, $\mathcal{R} = Q_n$, $n = p.q$ là tích của hai số nguyên tố lớn với $p \equiv q \equiv 3 \bmod 4$; $\mathcal{K}$ là tập hợp các bộ khoá $K = (K', K'')$, trong đó khoá công khai $K' = n$, và khoá bí mật $K'' = (p,q)$.

Thuật toán lập mã $e_{K'} = \mathcal{E}(K',.) : \mathcal{P} \times \mathcal{R} \to \mathcal{C}$ được tính theo các bước sau:

1. Cho $x = (x_1,...,x_l) \in \mathcal{P}$ và $r \in \mathcal{R}$. Từ mầm r theo thuật toán Blum-Blum-Shub tính dãy số $(s_0, s_1,...,s_{l+1})$ theo công thức

$$\begin{cases} s_0 = r, \\ s_{i+1} = s_i^2 \bmod n, \end{cases}$$

sau đó tính dãy số giả ngẫu nhiên $(z_1,...,z_l)$ bởi $z_i = s_i \bmod 2$.

2.Tính $y = (y_1,...,y_l)$ với $y_i = x_i + z_i \bmod 2$ $(1 \le i \le l)$.

3. Bản mã là $e_{K'}(x,r) = (y, s_{l+1}) = (y_1,...,y_l; s_{l+1})$.

Thuật toán giải mã $d_{K''} = \mathcal{D}(K'',.): \mathcal{C} \to \mathcal{P}$ được thực hiện theo các bước sau đây sau khi nhận được bản mã $(y_1,...,y_l; s_{l+1})$:

1. Tính

$$a_1 = ((p+1)/4)^{l+1} \bmod (p-1),$$

$$a_2 = ((q+1)/4)^{l+1} \bmod (q-1).$$

2. Tính $b_1 = s_{l+1}^{a_1} \bmod p$, $b_2 = s_{l+1}^{a_2} \bmod q$.

3. Tìm $s_0 = r$ bằng cách giải hệ phương trình

$$\begin{cases} s_0 \equiv b_1 \bmod p \\ s_0 \equiv b_2 \bmod q \end{cases}$$

4. Với s_0 theo thuật toán BBS ta tìm lại được dãy bit $(z_1,...,z_l)$.

5. Cuối cùng ta được

$d_{K''}(y_1,...,y_l; s_{l+1}) = (x_1,...,x_l)$, với $x_i = y_i + z_i \bmod 2$ $(1 \le i \le l)$.

Như vậy là hệ mật mã Blum-Goldwasser đã được định nghĩa đầy đủ. Ta chú ý rằng nếu bản rõ x gồm l bit thì trong bản mã tương ứng, ngoài các bit mã $y_1,...,y_l$ ta phải gửi thêm số s_{l+1}, số

đó được sử dụng trong các bước 1-3 của thuật toán giải mã để tìm lại mầm s_0 cần thiết cho việc tìm dòng khoá ngẫu nhiên $(z_1,...,z_l)$.

Ta chứng minh rằng số s_0 tính được theo thuật toán giải mã đúng là mầm s_0 mà ta cần tìm. Thực vậy, theo định nghĩa, ta có với mọi i =0,1,...,l +1, s_i đều là thặng dư bậc hai, và với mọi i =0,...,l , s_i đều là căn bậc hai của s_{i+1} theo mod$n$; điều đó cũng đúng đối với mod$p$ và mod$q$. Vì $p \equiv 3$ mod4, nên mỗi thặng dư bậc hai x theo mod$p$ đều có duy nhất một căn bậc hai mod$p$ cũng là thặng dư bậc hai mod$p$, đó là $x^{(p+1)/4}$mod$p$. Thực vậy, vì $x^{(p+1)/2} \equiv x$ mod$p$, nên $\pm x^{(p+1)/4}$mod$p$ là căn bậc hai theo mod$p$ của x ; mặt khác ta lại có $\left(\frac{x^{(p+1)/4}}{p}\right) = \left(\frac{x}{p}\right)^{(p+1)/4} = 1$, nên $x^{(p+1)/4}$mod$p$ cũng là một thặng dư bậc hai mod$p$. Từ nhận xét đó ta suy ra với mọi i (i = 0,1,..,l):

$$s_i \equiv s_{i+1}^{(p+1)/4} \pmod{p},$$

do đó,

$$s_0 = s_{l+1}^{((p+1)/4)^{l+1}} \bmod p = s_{l+1}^{a_1} \bmod p.$$

Xét tương tự đối với q, ta cũng được

$$s_0 = s_{l+1}^{a_2} \bmod q.$$

Vậy số s_0 tính theo các bước 1-3 của thuật toán giải mã đúng là mầm s_0=r mà ta cần tìm. Các thuật toán lập mật mã và giải mã như được định nghĩa ở trên là hợp thức.

Thí dụ : Chọn n = 192649 = 383.503.

Cho bản rõ x = 11010011010011101101. (l = 20)

Giả sử chọn ngẫu nhiên s_0=r = 20749. Ta tính được dãy z :

z = 11001110000100111010.

Ta tính thêm được s_{21}=94739, và bản mã được gửi đi là

$$e_{K'}(x,r) = (y, s_{l+1}) = (y, 94739),$$

trong đó y = 00011101010111010111.

Để giải mã, trước hết ta tìm s_0 từ s_{21} = 94739. Ta có

$$(p+1)/4 = 96,\ (q+1)/4 = 126.$$

Theo thuật toán giải mã:

$$a_1 = 96^{21} \bmod 382 = 266,$$
$$a_2 = 126^{21} \bmod 502 = 486.$$

Từ đó tính được

$$b_1 = 94739^{266} \bmod 383 = 67,$$
$$b_2 = 94739^{486} \bmod 503 = 126.$$

Giải hệ phương trình đồng dư:

$$\begin{cases} s_0 \equiv 67 \pmod{383} \\ s_0 \equiv 126 \pmod{503} \end{cases}$$

ta được s_0=20749, từ đó tính lại được dãy z, cộng mod2 từng bit với y ta lại thu được bản rõ x.

CHƯƠNG V

Bài toán xác nhận và chữ ký điện tử

5.1. Bài toán xác nhận và sơ đồ chữ ký.

5.1.1. Đặt vấn đề.

Trong chương I, tiết 1.3, ta đã liệt kê một số bài toán chủ yếu về an toàn thông tin, trong đó ngoài bài toán quan trọng nhất là bảo mật thông tin thì các bài toán kế tiếp là: xác nhận thông báo và xác nhận người gửi (cùng với thông báo), xưng danh và xác nhận danh tính của một chủ thể giao dịch, v.v... Bài toán bảo mật được đáp ứng bằng các giải pháp mật mã đã là nội dung của các chương III và IV, trong chương này và chương sau ta sẽ đề cập đến các bài toán xác nhận và nhận thức kể trên, chương V này sẽ dành cho bài toán xác nhận thông báo và người gửi thông báo, chương VI tiếp theo sẽ xét bài toán xưng danh và xác nhận danh tính.

Trong cách thức truyền thống, thông báo được truyền đi trong giao dịch thường dưới dạng các văn bản viết tay hoặc đánh máy được kèm thêm chữ ký (viết tay) của người gửi ở bên dưới văn bản. Chữ ký đó là bằng chứng xác nhận thông báo đúng là của người ký, tức là của chủ thể giao dịch, và nếu tờ giấy mang văn bản không bị cắt, dán, tẩy, xoá, thì tính toàn vẹn của thông báo cũng được chứng thực bởi chữ ký đó. Chữ ký viết tay có nhiều ưu điểm quen thuộc như dễ kiểm thử, không sao chép được, chữ ký của một người là giống nhau trên nhiều văn bản, nhưng mỗi chữ ký gắn liền với một văn bản cụ thể, v.v...

Khi chuyển sang cách thức truyền tin bằng phương tiện hiện đại, các thông báo được truyền đi trên các mạng truyền tin số hoá, bản thân các thông báo cũng được biểu diễn dưới dạng số hoá, tức dưới dạng các dãy bit nhị phân, "chữ ký" nếu có cũng ở dưới dạng các dãy bit, thì các mối quan hệ tự nhiên kể trên không còn giữ được nữa. Chẳng hạn, "chữ ký" của một người gửi trên những văn bản khác nhau phải thể hiện được sự gắn kết trách nhiệm của

người gửi đối với từng văn bản đó thì tất yếu phải khác nhau chứ không thể là những đoạn bit giống nhau như các chữ ký giống nhau trên các văn bản thông thường. Chữ ký viết tay có thể được kiểm thử bằng cách so sánh với nguyên mẫu, nhưng "chữ ký" điện tử thì không thể có "nguyên mẫu" để mà so sánh, việc kiểm thử phải được thực hiện bằng những thuật toán đặc biệt. Một vấn đề nữa là việc sao chép một văn bản cùng chữ ký. Nếu là văn bản cùng chữ ký viết tay thì dễ phân biệt bản gốc với bản sao, do đó khó mà dùng lại được một văn bản có chữ ký thật. Còn với văn bản điện tử cùng chữ ký điện tử thì có thể nhân bản sao chép tuỳ thích, khó mà phân biệt được bản gốc với bản sao, cho nên nguy cơ dùng lại nhiều lần là có thực, do đó cần có những biện pháp để tránh nguy cơ đó.

Một "chữ ký", nếu muốn thể hiện được trách nhiệm của người gửi trên toàn văn bản, thì phải mang được một chút gắn bó nào đó với từng bit thông tin của văn bản, vì vậy, theo hình dung ban đầu, độ dài của chữ ký cũng phải dài theo độ dài của văn bản; để có được "chữ ký ngắn" như trong trường hợp viết tay người ta phải dùng một kỹ thuật riêng gọi là *hàm băm* mà ta sẽ trình bày ở cuối chương. Bây giờ, trước hết ta sẽ giới thiệu định nghĩa về sơ đồ chữ ký (điện tử).

5.1.2. Định nghĩa sơ đồ chữ ký.

Định nghĩa 5.1. Một *sơ đồ chữ ký* $\mathcal{S}$ là một bộ năm

$$\mathcal{S} = (\mathcal{P}, \mathcal{A}, \mathcal{K}, \mathcal{S}, \mathcal{V}),$$

trong đó: $\mathcal{P}$ là một tập hữu hạn các thông báo có thể có,

$\mathcal{A}$ là một tập hữu hạn các chữ ký có thể có,

$\mathcal{K}$ là một tập hữu hạn các khoá, mỗi khoá $K \in \mathcal{K}$ gồm có hai phần $K=(K',K'')$, K' là khoá bí mật dành cho việc ký, còn K'' là khoá công khai dành cho việc kiểm thử chữ ký.

Với mỗi $K=(K',K'')$, trong $\mathcal{S}$ có một *thuật toán ký* $sig_{K'}: \mathcal{P} \to \mathcal{A}$, và trong $\mathcal{V}$ có một *thuật toán kiểm thử* $ver_{K''}: \mathcal{P} \times \mathcal{A} \to$ {đúng,sai} thoả mãn điều kiện sau đây đối với mọi thông báo $x \in \mathcal{P}$ và mọi chữ ký $y \in \mathcal{A}$:

$$ver_{K''}(x, y) = \text{đúng} \Leftrightarrow y = sig_{K'}(x).$$

Với sơ đồ trên, mỗi chủ thể sở hữu một bộ khoá $K=(K',K'')$, công bố công khai khoá K'' để mọi người có thể kiểm thử chữ ký của mình, và giữ bí mật khoá K' để thực hiện chữ ký trên các thông báo mà

mình muốn gửi đi. Các hàm $ver_{K''}$ và $sig_{K'}$ (khi biết K') phải tính được một cách dễ dàng (trong thời gian đa thức), tuy nhiên hàm $y = sig_{K'}(x)$ là khó tính được nếu không biết K' - điều đó bảo đảm bí mật cho việc ký, cũng tức là bảo đảm chống giả mạo chữ ký.

Bài toán xác nhận với chữ ký điện tử, theo một nghĩa nào đó, có thể xem là « đối ngẫu » với bài toán bảo mật bằng mật mã, như được minh hoạ bởi thí dụ sơ đồ chữ ký RSA, đối ngẫu với sơ đồ mật mã RSA, dưới đây :

5.1.3. Sơ đồ chữ ký RSA.

Sơ đồ chữ ký RSA được cho bởi bộ năm

$$\mathcal{S} = (\mathcal{P}, \mathcal{A}, \mathcal{K}, \mathcal{S}, \mathcal{V}),$$

trong đó $\mathcal{P} = \mathcal{A} = Z_n$, với $n = p.q$ là tích của hai số nguyên tố lớn p,q, $\mathcal{K}$ là tập các cặp khoá $K = (K', K'')$, với $K' = a$ và $K'' = (n,b)$, a và b là hai số thuộc Z_n^* thoả mãn $a.b \equiv 1 (\mathrm{mod}\, \phi(n))$. Các hàm $sig_{K'}$ và $ver_{K''}$ được xác định như sau:

$$sig_{K'}(x) = x^a \bmod n,$$

$$ver_{K''}(x,y) = \text{đúng} \Leftrightarrow x \equiv y^b \pmod{n}.$$

Dễ chứng minh được rằng sơ đồ được định nghĩa như vậy là hợp thức, tức là với mọi $x \in \mathcal{P}$ và mọi chữ ký $y \in \mathcal{A}$:

$$ver_{K''}(x, y) = \text{đúng} \Leftrightarrow y = sig_{K'}(x).$$

Chú ý rằng tuy hai vấn đề xác nhận và bảo mật theo sơ đồ RSA là có bề ngoài giống nhau, nhưng nội dung của chúng là hoàn toàn khác nhau: Khi A gửi thông báo x cho B, để B có căn cứ xác nhận đó đúng thực là thông báo do A gửi, A phải gửi kèm theo chữ ký $sig_{K'}(x)$, tức là A gửi cho B $(x, sig_{K'}(x))$, trong các thông tin gửi đi đó, thông báo x hoàn toàn không được giữ bí mật. Cũng tương tự như vậy, nếu dùng sơ đồ mật mã RSA, khi một chủ thể A nhận được một bản mật mã $e_{K'}(x)$ từ B thì A chỉ biết rằng thông báo x được bảo mật, chứ không có gì để xác nhận x là của B.

Nếu ta muốn hệ truyền tin của ta vừa có tính bảo mật vừa có tính xác nhận, thì ta phải sử dụng đồng thời cả hai hệ mật mã và xác nhận (bằng chữ ký). Giả sử trên mạng truyền tin công cộng, ta có cả hai hệ mật mã khoá công khai $\mathcal{S}_1$ và hệ xác nhận bằng chữ ký $\mathcal{S}_2$. Giả sử B có bộ khoá mật mã $K = (K', K'')$ với $K' = (n, e)$ và $K'' = d$ trong hệ $\mathcal{S}_1$, và A có bộ khoá chữ ký $K_s = (K_s', K_s'')$ với $K_s' = a$ và $K_s'' = (n,b)$ trong hệ $\mathcal{S}_2$. A có thể gửi đến B một thông báo vừa bảo

mật vừa có chữ ký để xác nhận như sau: A ký trên thông báo x trước, rồi thay cho việc gửi đến B văn bản cùng chữ ký $(x, sig_{K_s'}(x))$ thì A sẽ gửi cho B bản mật mã của văn bản đó được lập theo khoá công khai của B, tức là gửi cho B $e_{K'}((x, sig_{K_s'}(x))$. Nhận được văn bản mật mã đó B sẽ dùng thuật toán giải mã $d_{K''}$ của mình để thu được $(x, sig_{K_s'}(x))$, sau đó dùng thuật toán kiểm thử chữ ký công khai $ver_{K_s''}$ của A để xác nhận chữ ký $sig_{K_s'}(x)$ đúng là của A trên x.

5.2. Sơ đồ chữ ký ElGamal và chuẩn chữ ký điện tử.

5.2.1. Sơ đồ chữ ký ElGamal.

Sơ đồ chữ ký ElGamal được đề xuất năm 1985, gần như đồng thời với sơ đồ hệ mật mã ElGamal, cũng dựa trên độ khó của bài toán lôgarit rời rạc. Sơ đồ được thiết kế đặc biệt cho mục đích ký trên các văn bản điện tử, được mô tả như một hệ

$$\mathcal{S} = (\mathcal{P}, \mathcal{A}, \mathcal{K}, \mathcal{S}, \mathcal{V}),$$

trong đó $\mathcal{P} = Z_p^*$, $\mathcal{A} = Z_p^* \times Z_{p-1}$, với p là một số nguyên tố sao cho bài toán tính lôgarit rời rạc trong Z_p^* là rất khó. Tập hợp $\mathcal{K}$ gồm các cặp khoá $K=(K',K'')$, với $K' = a$ là một số thuộc Z_p^*, $K'' =(p, \alpha, \beta)$, α là một phần tử nguyên thuỷ của Z_p^*, và $\beta = \alpha^a \bmod p$. K' là khoá bí mật dùng để ký, và K'' là khoá công khai dùng để kiểm thử chữ ký. Các thuật toán ký và kiểm thử chữ ký được xác định như sau: Với mỗi thông báo x, để tạo chữ ký trên x ta chọn thêm môt số ngẫu nhiên $k \in Z_{p-1}^*$, rồi tính

$$sig_{K'}(x,k) = (\gamma, \delta), \quad \text{với}$$
$$\gamma = \alpha^k \bmod p,$$
$$\delta = (x - a\gamma).k^{-1} \bmod (p\text{-}1).$$

Thuật toán kiểm thử được định nghĩa bởi:

$$ver_{K''}(x,(\gamma,\delta)) = \text{đúng} \Leftrightarrow \beta^{\gamma}.\gamma^{\delta} \equiv \alpha^{x} \pmod p.$$

Dễ thấy rằng sơ đồ chữ ký được định nghĩa như trên là hợp thức. Thực vậy, nếu $sig_{K'}(x,k) = (\gamma, \delta)$, thì ta có :

$$\beta^{\gamma}.\gamma^{\delta} \equiv \alpha^{a\gamma}.\alpha^{k\delta} \bmod p$$
$$\equiv \alpha^{x} \bmod p,$$

vì $k\delta + a\gamma \equiv x \bmod (p\text{-}1)$. Do đó, $ver_{K''}(x,(\gamma,\delta))$ = đúng.

Thí dụ: Giả sử p = 467, α = 2, a = 127. Khi đó $\beta = 2^{127} \text{mod} 467 = 132$. Cho x =100; ta chọn ngẫu nhiên k =213 ($\in Z^*_{466}$) và được $k^{-1} \text{mod} 466$ =431. Chữ ký trên văn bản x =100 với số ngẫu nhiên k =213 là (γ, δ), trong đó $\gamma = 2^{213} \text{mod} 467 = 29$ và δ = (100 - 127.29).431mod466 =51.

Để kiểm thử ta tính :

$$\beta^{\gamma}.\gamma^{\delta} = 132^{29}.29^{51} \equiv 189 \ (\text{mod}467),$$

$$\alpha^{x} = 2^{100} \equiv 189 \ (\text{mod}467),$$

hai giá trị đó đồng dư với nhau theo mod467, chữ ký(γ, δ)=(29,51) được xác nhận là đúng.

5.2.2. Tính an toàn của sơ đồ chữ ký ElGamal.

Sơ đồ chữ ký ElGamal được xem là an toàn, nếu việc ký trên một văn bản là không thể giả mạo được, nói cách khác, không thể có một người nào ngoài chủ thể hợp pháp có thể giả mạo chữ ký của chủ thể hợp pháp đó trên một văn bản bất kỳ. Vì vậy, việc giữ bí mật khoá $K' = a$ dùng để tạo chữ ký là có ý nghĩa quyết định đối với việc bảo đảm tính an toàn của chữ ký. Có thể để lộ khoá bí mật $K' = a$ trong những trường hợp nào, và có thể không để lộ $K' = a$ mà vẫn giả mạo chữ ký được không? Ta sẽ xét sau đây một vài trường hợp đơn giản :

1) *Khả năng để lộ khoá* $K' = a$: Cũng như đối với sơ đồ hệ mật mã ElGamal, khoá bí mật a có thể bị phát hiện trong trường hợp để lộ số ngẫu nhiên k ở một lần ký nào đó, hoặc sử dụng cùng một số ngẫu nhiên k ở hai lần ký khác nhau.

Nếu số ngẫu nhiên k được sử dụng khi ký trên văn bản x bị lộ, thì khoá bí mật $K' = a$ được tính theo công thức sau đây:

$$a = (x - k\delta).\ \gamma^{-1} \ \text{mod}(p - 1).$$

Bây giờ ta xét trường hợp dùng cùng một số ngẫu nhiên k cho hai lần ký khác nhau, chẳng hạn cho x_1 và x_2. Khi đó ta có chữ ký trên x_1 là (γ ,δ_1), trên x_2 là (γ ,δ_2), với thành phần thứ nhất bằng nhau (và bằng $\gamma = \alpha^k \ \text{mod} p$), và các chữ ký đó thoả mãn

$$\beta^{\gamma}.\gamma^{\delta_1} \equiv \alpha^{x_1} \ (\text{mod}p),$$

$$\beta^{\gamma}.\gamma^{\delta_2} \equiv \alpha^{x_2} \ (\text{mod}p).$$

Từ đó ta có

$$\alpha^{x_1 - x_2} \equiv \gamma^{\delta_1 - \delta_2} \equiv \alpha^{k(\delta_1 - \delta_2)} \ (\text{mod}p),$$

điều đó tương đương với

$$x_1 - x_2 \equiv k(\delta_1 - \delta_2) \ \ (\text{mod}(p-1)).$$

Đặt d = gcd($\delta_1 - \delta_2$, p -1). Cả ba số $\delta_1 - \delta_2$, p -1 và $x_1 - x_2$ đều chia hết cho d, ta đặt

$$x' = \frac{x_1 - x_2}{d}\ ,\ \delta' = \frac{\delta_1 - \delta_2}{d}\ ,\ p' = \frac{p-1}{d}.$$

Khi đó đồng dư thức ở trên trở thành

$$x' \equiv k.\delta' \ (\text{mod}\ p').$$

Vì $\gcd(\delta', p') = 1$, nên có thể tính $\varepsilon = \delta'^{-1} \bmod p'$, và sau đó giá trị k theo mod p': $k = x'.\varepsilon \bmod p'$, tức là

$$k = x'.\varepsilon + i.p' \bmod (p\text{-}1)$$

với i là một giá trị nào đó, $0 \le i \le d - 1$. Thử lần lượt điều kiện

$$\gamma = \alpha^k \bmod p$$

với các giá trị đó của i, ta sẽ tìm được k; sau đó từ k tính được a cần tìm.

2) *Khả năng giả mạo chữ ký trên một văn bản cho trước* :

Giả sử chủ thể A chọn sơ đồ chữ ký ElGamal với cặp khoá $K = (K', K'')$, trrong đó $K' = a$ là khoá bí mật. Một người ngoài O không biết khoá bí mật $K' = a$ mà muốn giả mạo chữ ký của A trên một văn bản x thì phải có khả năng tạo ra được chữ ký (γ, δ) mà không cần biết a. Có hai cách : hoặc chọn trước γ rồi tìm δ tương ứng, hoặc ngược lại, chọn trước δ rồi tìm γ tương ứng.

Nếu chọn trước γ rồi tìm δ, thì δ phải là

$$\delta = (x - a\gamma)k^{-1} \bmod (p\text{-}1) = ((x - a\gamma)\log_\gamma \alpha \bmod (p\text{-}1)$$

$$= \log_\alpha(\alpha^x \beta^{-\gamma}).\log_\gamma \alpha = \log_\gamma \alpha^x \beta^{-\gamma} \bmod (p\text{-}1);$$

đó là một bài toán tính lôgarit rời rạc, mà ta biết rằng rất khó.

Nếu chọn trước δ rồi tìm γ thì phải giải phương trình

$$\beta^\gamma.\gamma^\delta \equiv \alpha^x \bmod p$$

với ẩn số γ. Ta chưa biết có cách giải hữu hiệu nào không, nhưng chắc là không dễ hơn bài toán tính lôgarit rời rạc.

Như vậy, ta có thể tin rằng khả năng giả mạo chữ ký trên một văn bản cho trước khi không biết khoá bí mật $K' = a$ là rất ít, do đó không có ảnh hưởng đáng kể đến tính an toàn của sơ đồ chữ ký.

3) *Giả mạo chữ ký cùng với văn bản được ký* :

Có một khả năng giả mạo khác là giả mạo cả văn bản gửi đi x cùng với chữ ký (γ, δ) trên x. Khả năng đó xẩy ra khi kẻ giả mạo chọn được x và (γ, δ) thoả mãn điều kiện kiểm thử, cụ thể khi chọn được x, γ, δ có dạng sau đây :

$$\gamma = \alpha^i.\beta^j \bmod p,$$

$$\delta = -\gamma.j^{-1} \bmod(p\text{-}1),$$
$$x = -\gamma.i.j^{-1} \bmod(p\text{-}1),$$

trong đó i, j là các số nguyên sao cho $0 \le i, j \le p-2$, $\gcd(j, p-1) = 1$, và j^{-1} được tính theo mod$(p-1)$. Thực vậy, khi đó ta có

$$\begin{aligned}\beta^{\gamma}.\gamma^{\delta} &\equiv \beta^{\gamma}(\alpha^{i}\beta^{j})^{-\gamma.j^{-1}} \bmod p \\ &\equiv \beta^{\gamma}.\alpha^{-i\gamma j^{-1}}.\beta^{-\gamma} \bmod p \\ &\equiv \alpha^{x} \bmod p,\end{aligned}$$

tức điều kiện kiểm thử được thoả mãn, (γ,δ) có thể được xác nhận hợp thức là chữ ký trên x.

Có thể có một cách giả mạo khác nữa, nếu kẻ giả mạo sử dụng chữ ký đúng (γ,δ) trên một văn bản x có từ trước để tạo ra một chữ ký (λ,μ) mới cho một văn bản "mới" x' như sau:

$$\lambda = \gamma^{h}.\alpha^{i}.\beta^{j} \bmod p,$$
$$\mu = \delta\lambda(h\gamma - j\delta)^{-1} \bmod(p\text{-}1),$$
$$x' = \lambda(hx + i\delta)(h\gamma - j\delta)^{-1} \bmod(p\text{-}1).$$

Có thể thử lại rằng điều kiện kiểm thử đúng đối với "chữ ký" (λ,μ) và "văn bản" x', tức là

$$\beta^{\lambda}.\lambda^{\mu} \equiv \alpha^{x'} \bmod p.$$

Cả hai cách giả mạo nói trên đều cho chữ ký thoả mãn điều kiện kiểm thử đối với văn bản tương ứng, tuy nhiên văn bản đó không phải là văn bản được chọn theo ý muốn của người giả mạo, cho nên khả năng sử dụng các cách giả mạo đó trong thực tế cũng không có giá trị , do đó không thể gây nguy hại đáng kể cho tính an toàn của sơ đồ chữ ký nói chung.

5.2.3. Chuẩn chữ ký số (Digital Signature Standard).

Chuẩn chữ ký số (DSS) được đề xuất từ năm 1991 và được chấp nhận vào cuối năm 1994 để sử dụng trong một số lĩnh vực giao dịch điện tử tại Hoa kỳ. DSS dựa vào sơ đồ chữ ký ElGamal, với một vài sửa đổi. Để bảo đảm an toàn , số nguyên tố p cần phải đủ lớn, biểu diễn nhị phân của p phải có từ 512 bit trở lên (cụ thể từ 512 đến 1024 bit, số bit là một bội của 64). Tuy nhiên, độ dài chữ ký theo sơ đồ ElGamal là gấp đôi số bit của p, mà trong nhiều ứng dụng người ta lại mong muốn có chữ ký độ dài ngắn, nên giải pháp sửa đổi được đề xuất là: trong khi vẫn dùng p lớn với độ dài biểu diễn 512 bit trở lên, thì sẽ hạn chế độ dài của γ và δ trong chữ ký (γ,δ) vào khoảng 160 bit (như vậy cả chữ ký sẽ có độ dài khoảng 320 bit); điều này được thực hiện bằng cách dùng một nhóm con cyclic Z_q^* của Z_p^* thay cho chính bản thân Z_p^*, do đó mọi tính toán

vẫn được thực hiện như trong Z_p^* nhưng các dữ liệu và thành phần chữ ký lại thuộc Z_q^* . Ta được *sơ đồ chuẩn chữ ký số DSS* như mô tả sau đây:

Chọn p là một số nguyên tố lớn có độ dài biểu diễn ≥ 512 bit sao cho bài toán tính logarit rời rạc trong Z_p là khó, q là một ước số nguyên tố của p-1, có độ dài biểu diễn cỡ 160 bit. Gọi $\alpha \in Z_p^*$ là một căn bậc q của 1 theo mod$p$.

Đặt $\mathcal{P} = Z_p^*$, $\mathcal{A} = Z_q^* \times Z_q^*$. Chọn $a \in Z_q^*$ và tính $\beta \equiv \alpha^a \bmod p$. Xác định khoá K =(K',K'') trong đó khoá bí mật K' = a, và khoá công khai K'' = (p,q,α,β). Thuật toán ký và thuật toán kiểm thử được định nghĩa như sau: Với $x \in \mathcal{P} = Z_p^*$, ta chọn thêm một số ngẫu nhiên k ($0 \le k \le q$-1), và định nghĩa chữ ký

$$sig_{K'}(x,k) = (\gamma, \delta), \quad \text{trong đó}$$
$$\gamma = (\alpha^k \bmod p) \bmod q,$$
$$\delta = (x + a\gamma).k^{-1} \bmod q.$$

Thuật toán kiểm thử được định nghĩa bởi:

$$ver_{K''}(x,(\gamma,\delta)) = \text{đúng} \Leftrightarrow (\alpha^{e_1}.\beta^{e_2} \bmod p) \bmod q = \gamma,$$

trong đó $e_1 = x.\delta^{-1} \bmod q$ và $e_2 = \gamma.\delta^{-1} \bmod q$.

Chú ý rằng ta phải có $\delta \neq 0 \bmod q$ để có thể tính được $\delta^{-1} \bmod q$ dùng trong thuật toán kiểm thử, vì vậy nếu chọn k mà được $\delta \equiv 0 \bmod q$ thì phải chọn lại số k khác để có được $\delta \neq 0 \bmod q$.

5.3. Hàm băm và chữ ký.

5.3.1. Hàm băm (hash function).

Trong các phần trên, ta đã giới thiệu một vài sơ đồ chữ ký điện tử. Theo các sơ đồ đó, chữ ký được xác định cho từng khối của văn bản, và nếu văn bản gồm nhiều khối thì chữ ký cho toàn văn bản cũng phải do ghép chữ ký trên từng khối lại với nhau mà thành; mà chữ ký trên từng khối văn bản thường có độ dài bằng (hoặc thậm chí gấp đôi) độ dài của khối văn bản, do đó chữ ký chung cũng có độ dài tương đương với độ dài văn bản. Đó là một điều bất tiện. Ta mong muốn, như trong trường hợp viết tay, chữ ký chỉ có độ dài ngắn và hạn chế cho dù văn bản có thể dài bao nhiêu cũng được. Đối với chữ ký điện tử, vì chữ ký phải được "ký" cho từng bit của văn bản, nên muốn có chữ ký độ dài hạn chế trên văn bản có độ dài tuỳ ý thì phải tìm cách rút ngắn độ dài văn bản. Nhưng bản thân văn bản không thể rút ngắn được, nên chỉ còn cách là tìm cho mỗi văn bản một bản "tóm lược" có độ dài hạn chế, rồi thay cho việc ký trên toàn bộ văn bản, ta ký trên bản tóm lược

đó, xem chữ ký trên bản tóm lược có tư cách là chữ ký trên văn bản. Giả sử Σ là tập hợp tất cả các văn bản có thể có (tất nhiên, trong một lĩnh vực nào đó), và Δ là tập hợp tất cả các bản "tóm lược" có thể được sử dụng. Việc tìm cho mỗi văn bản một bản tóm lược tương ứng xác định một hàm $h : \Sigma \to \Delta$. Một hàm h như vậy người ta gọi là một *hàm băm* (hash function). Thông thường, Σ là tập hợp các dãy bit có độ dài tuỳ ý, và Δ là tập hợp các dãy bit có một độ dài n cố định, nên người ta cũng định nghĩa *hàm băm* là các hàm $h : \Sigma \to \Delta$ với các tập hợp Σ và Δ đó (tức các hàm $h : \{0,1\}^* \to \{0,1\}^n$).

Dùng hàm băm h, ta xem $z = h(x)$ là "tóm lược" của x, đại diện cho x, và ta sẽ xem chữ ký trên z là chữ ký trên văn bản x; vì z có độ dài hạn chế, nên chữ ký trên x cũng có độ dài hạn chế.

Một vấn đề được đặt ra là: vậy hàm $h : \Sigma \to \Delta$ phải thoả mãn những điều kiện gì để $h(x)$ xứng đáng được xem là đại diện của x trong việc tạo lập chữ ký ? Hai điều kiện sau đây thường được người ta xem là hai điều kiện chủ yếu cho một hàm băm:

1. Hàm băm phải là hàm một phía, nghĩa là cho x tính $z = h(x)$ là việc dễ, nhưng ngược lại, biết z tính x là việc cực khó (có thể qui ước dễ hay khó theo nghĩa tính được trong thời gian đa thức hay không).
2. Hàm băm phải là hàm *không va chạm mạnh* theo nghĩa sau đây: không có thuật toán tính được trong thời gian đa thức giải bài toán " tìm x_1 và x_2 thuộc Σ sao cho $x_1 \neq x_2$ và $h(x_1) = h(x_2)$"; nói cách khác, tìm hai văn bản khác nhau có cùng một đại diện là cực kỳ khó.

(Còn có một khái niệm *không va chạm yếu* được định nghĩa như sau: Cho $x \in \Sigma$. Hàm h là không va chạm yếu đối với x nếu rất khó tìm được $x' \in \Sigma$, $x' \neq x$ và $h(x') = h(x)$).

Ta mong muốn độ dài của chữ ký là ngắn, tức là độ dài của các tóm lược cũng ngắn. Nhưng ngắn bao nhiêu là vừa? Ngắn bao nhiêu thì có thể bảo đảm tính không va chạm mạnh? Và ở đây ta gặp một kiểu "tấn công", thường được gọi là "tấn công ngày sinh" có liên quan đến khả năng va chạm mạnh, nói rằng trong một nhóm gồm 23 người được chọn một cách ngẫu nhiên thì ít nhất có hai người có cùng ngày sinh (tức có va chạm mạnh!). Một cách tổng quát, người ta chứng minh được rằng: *Nếu có tất cả n bản tóm lược, và* $k \approx \sqrt{2n \ln \frac{1}{1-\varepsilon}}$, *thì trong k văn bản được chọn ngẫu nhiên có ít nhất một va chạm mạnh (tức có* $x' \neq x$ *và* $h(x') = h(x)$*) với xác suất* ε.

Khi $\varepsilon = \frac{1}{2}$, ta có $k \approx 1{,}17\sqrt{n}$. Trong trường hợp ngày sinh, ta có n=365, do đó $k \approx 22{,}3 \approx 23$.

Trở lại với vấn đề chọn độ dài (của biểu diễn nhị phân) cho các tóm lược, nếu ta lấy chẳng hạn độ dài 40 bit, thì $n = 2^{40}$, và do đó từ $k \approx 2^{20}$ (khoảng một triệu) văn bản sẽ có một va chạm mạnh với xác suất 1/2, như vậy khó bảo đảm được an toàn. Nhưng nếu ta lấy độ dài của bản tóm lược là 128, tức $n = 2^{128}$, thì va chạm mạnh có thể xẩy ra với xác suất 1/2 khi số các văn bản có thể là $k \approx 2^{64}$, một con số khá lớn (so với số văn bản có thể nẩy sinh trong thực tế), do đó hy vọng tính an toàn sẽ được bảo đảm. Có thể vì vậy mà trong chuẩn DSS người ta chọn độ dài của các tóm lược là 160 bit.

5.3.2. Hàm băm Chaum-van Heijst-Pfitzmann.

Dưới đây ta sẽ giới thiệu một thí dụ cụ thể về một hàm băm được xây dựng dựa trên tính khó của bài toán lôgarit rời rạc, do các tác giả Chaum, van Heijst và Pfitzmann đề xuất năm 1992. Hàm băm đó được xây dựng như sau:

Giả sử p là một số nguyên tố lớn dạng Sophie Germain, tức có dạng $p = 2q + 1$, trong đó q cũng là số nguyên tố. Chọn α và β là hai phần tử nguyên thuỷ của Z_p^*. Việc tính $\log_\alpha \beta$, khi biết α và β, là rất khó. Hàm băm $h : Z_q \times Z_q \to Z_p - \{0\}$ được định nghĩa như sau: với mọi $x_1, x_2 \in Z_q$ ta có

$$h(x_1, x_2) = \alpha^{x_1}.\beta^{x_2} \bmod p.$$

Ta gọi hàm băm h được định nghĩa như vậy là hàm băm Chaum-van Heijst-Pfitzmann. Hàm băm đó có các tính chất là hàm một phía và không va chạm mạnh như yêu cầu đối với một hàm băm. Tính một phía của hàm đó được suy ra từ tính một phía của hàm lôgarit rời rạc. Còn tính không va chạm mạnh của h được chứng minh bởi <u>định lý</u> sau đây : *Nếu biết một va chạm mạnh đối với h thì có thể tính được* $\log_\alpha \beta$ *một cách có hiệu quả*.

Giả sử có một va chạm

$$h(x_1, x_2) = h(x_3, x_4),$$

trong đó $(x_1,x_2) \neq (x_3,x_4)$. Như vậy ta có

$$\alpha^{x_1}.\beta^{x_2} \equiv \alpha^{x_3}.\beta^{x_4} \pmod p,$$

tức là

$$\alpha^{x_1 - x_3} \equiv \beta^{x_4 - x_2} \pmod p.$$

Đặt d=gcd(x_4 - x_2, p-1). Vì p-1 = $2q$ và q là số nguyên tố, nên ta có $d \in \{1,2,q,p\text{-}1\}$. Ta xét lần lượt bốn khả năng đó của d.

Giả sử d=1. Khi đó, đặt $y = (x_4 - x_2)^{-1} \bmod (p\text{-}1)$, ta có

$$\beta \equiv \beta^{(x_4-x_2)y} \pmod{p}$$
$$\equiv \alpha^{(x_1-x_3)y} \pmod{p},$$

và ta có thể tính logarit rời rạc $\log_\alpha \beta$ như sau :

$$\log_\alpha \beta = (x_1 - x_3)(x_4 - x_2)^{-1} \bmod (p-1).$$

Bây giờ giả sử $d = 2$. Vì $p - 1 = 2q$ và q là số lẻ, ta phải có $\gcd(x_4 - x_2, q) = 1$. Cũng đặt $y = (x_4 - x_2)^{-1} \bmod q$, ta có

$$(x_4 - x_2)y = kq + 1$$

với k là một số nguyên nào đó, và ta có

$$\beta^{(x_4-x_2)y} \equiv \beta^{kq+1} \pmod{p}$$
$$\equiv (-1)^k \beta \pmod{p} \quad (\text{vì } \beta^q \equiv -1 \pmod{p})$$
$$\equiv \pm\beta \pmod{p}.$$

Như vậy ta có

$$\beta^{(x_4-x_2)y} \equiv \alpha^{(x_1-x_3)y} \pmod{p}$$
$$\equiv \pm\beta \pmod{p}.$$

Từ đó suy ra

$$\log_\alpha \beta = (x_1 - x_3)y \bmod (p-1)$$

hay là

$$\log_\alpha \beta = (x_1 - x_3)y + q \bmod (p-1).$$

Có thể thử để xác định giá trị nào trong hai giá trị đó đúng là $\log_\alpha \beta$.

Bây giờ ta xét trrường hợp $d = q$. Vì $0 \le x_2, x_4 \le q-1$, nên

$$-(q-1) \le x_4 - x_2 \le q-1.$$

Do đó không thể có $\gcd(x_4 - x_2, p-1) = q$, trường hợp này không thể xẩy ra.

Cuối cùng là trường hợp $d = p - 1$. Điều này chỉ xẩy ra nếu $x_2 = x_4$. Nhưng khi đó ta có

$$\alpha^{x_1}\beta^{x_2} \equiv \alpha^{x_3}\beta^{x_2} \pmod{p}$$
$$\alpha^{x_1} \equiv \alpha^{x_3} \pmod{p}$$

và $x_1 = x_3$. Như vậy $(x_1, x_2) = (x_3, x_4)$, mâu thuẫn với giả thiết. Vậy trường hợp này cũng không thể xẩy ra. Định lý nói trên được chứng minh. Hàm băm Chaum-van Heijst-Pfitzmann là không va chạm mạnh.

Chú ý rằng nếu p có độ dài biểu diễn nhị phân là t bit, tức Z_p là tập con của $\Delta = \{0,1\}^t$, thì q có độ dài $t-1$ bit, và $Z_q \times Z_q$ là tập con của $\Sigma = \{0,1\}^m$ với $m = 2(t-1)$. Hàm băm h được định nghĩa ở trên có thể xem là hàm $h : \Sigma \to \Delta$. Với mục đích chữ ký, ta muốn có những hàm băm $h : \Sigma \to \Delta$ với Δ là tập các từ có số bit hạn chế, nhưng Σ lại là tập các từ có độ dài tuỳ ý. Muốn vậy, ta phải có khả năng mở rộng hàm băm; định lý sau đây cho ta khả năng đó.

5.3.3. Mở rộng hàm băm.

Bây giờ giả sử $h: Z_2^m \to Z_2^t$ (ở đây Z_2 ={0,1}) là một hàm băm không va chạm mạnh thoả mãn $m \geq t+1$ (hàm băm trong mục trên thoã mãn điều kiện đó). Ta sẽ dùng h để xây dựng một hàm băm $h^*: Z_2^* \to Z_2^t$ như sau :

Giả sử $x \in Z_2^*$, ta cắt x thành các đoạn có cùng độ dài l bit, trong đó $l = m-t-1$, nếu đoạn cuối cùng chưa có đủ l bit, thì ta bổ sung thêm các bit 0 cho đủ, và để ghi nhớ sự bổ sung đó (chẳng hạn là d bit) ta thêm cho x một đoạn cuối x_{k+1} là biểu diễn nhị phân l bit của số d. Như vậy mỗi $x \in Z_2^*$ được viết lại dưới dạng

$$x = x_1 x_2 .... x_k x_{k+1},$$

trong đó với mọi i =1,2,...,k, k +1, $x_i \in Z_2^l$ (ta chú ý rằng nếu biết x dưới dạng này ta sẽ khôi phục lại được x ở dạng gốc ban đầu). Ta định nghĩa một cách đệ qui dãy từ $g_1, g_2, ..., g_{k+1} \in Z_2^t$ và hàm h^* như sau :

$$g_1 = h(0^{t+1} x_1),$$
$$g_{i+1} = h(g_i 1 x_{i+1}) \quad (i = 1,...,k)$$
$$h^*(x) = g_{k+1}.$$

Như vậy, giá trị của hàm băm h^* là một từ có độ dài t bit.

Người ta chứng minh được <u>định lý</u> sau đây : *Nếu hàm băm h có tính chất không va chạm mạnh thì hàm băm mở rộng* h^* *cũng có tính chất không va chạm mạnh.*

5.3.4. Xây dựng hàm băm từ các hệ mật mã.

Có một phương pháp chung để xây dựng hàm băm là sử dụng các hệ mật mã khoá đối xứng. Giả sử $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ là một hệ mật mã khoá đối xứng mà độ an toàn đã được thử nghiệm. Để tiện trình bày, ta có thể giả thiết rằng $\mathcal{P} = \mathcal{C} = \mathcal{K} = Z_2^n$. Nên chọn n khá lớn, cỡ $n \geq 128$ để tránh kiểu "tấn công ngày sinh". Chẳng hạn, có thể chọn hệ mật mã đó là hệ DES (có thể với những điều chỉnh cần thiết để có độ dài các ký tự trong $\mathcal{P}$, $\mathcal{C}$, $\mathcal{K}$ thích hợp). Xuất phát từ hàm lập mật mã $\mathcal{E}$ ta xác định một hàm $f: Z_2^n \times Z_2^n \to Z_2^n$ sao cho với mọi $(x, y) \in Z_2^n \times Z_2^n$, giá trị của $f(x, y)$ được tính theo x, y và hàm $\mathcal{E}$.

Bây giờ giả sử cho $x \in Z_2^*$. Như trong mục trên, ta có thể viết x dưới dạng ghép nối liên tiếp của k đoạn ký tự, mỗi đoạn có n bit :

$$x = x_1 x_2 .... x_k .$$

Tiếp đó, ta chọn một giá trị ban đầu $g_0 \in Z_2^n$, và xây dựng tiếp $g_1, g_2, ..., g_k$ theo qui tắc

$$g_i = f(x_i , g_{i-1}) \text{ với } i=1,2,...,k.$$

Và cuối cùng, ta định nghĩa giá trị hàm băm $h(x) = g_k$. Hàm băm h được định nghĩa như vậy là một hàm ánh xạ Z_2^* vào Z_2^n; trong trường hợp chung có thể không bảo đảm tính an toàn, nhưng người ta đã chứng tỏ được rằng nó là an toàn trong các trường hợp hàm f được chọn như sau:

$$f(x, y) = x \oplus \mathcal{E}(y,x),$$
$$f(x, y) = x \oplus y \oplus \mathcal{E}(y,x),$$
$$f(x, y) = x \oplus \mathcal{E}(y,x \oplus y),$$
$$f(x, y) = x \oplus y \oplus \mathcal{E}(y,x \oplus y),$$

trong đó $\oplus$ là phép cộng mod2 từng cặp bit một của hai từ có số bit bằng nhau.

5.4. Một số sơ đồ chữ ký khác.

5.4.1. Sơ đồ chữ ký Rabin.

Tương tư như sơ đồ chữ ký RSA, sơ đồ chữ ký Rabin cũng sử dụng số nguyên n là tích của hai số nguyên tố lớn p và q, $n = p.q$, với hàm một phía ở đây là hàm lấy bình phương của một số nguyên theo mod$n$, có hàm ngược là hàm tìm căn bậc hai theo mod$n$, một hàm không tính được một cách dễ dàng nếu không biết các thừa số p, q của n.

Như vậy, một cách đại thể, sơ đồ chữ ký Rabin có thể được mô tả là một bộ

$$\mathcal{S} = (\mathcal{P}, \mathcal{A}, \mathcal{K}, \mathcal{S}, \mathcal{V}),$$

trong đó $\mathcal{P} = Q_n$, $\mathcal{A} = Z_n$, $\mathcal{K}$ là tập các cặp khoá $K = (K', K'')$, trong đó $K'' = n$ là khoá công khai dùng để kiểm thử chữ ký, n là tích của hai số nguyên tố lớn p và q, $n = p.q$, với $p \equiv q \equiv 3 \pmod 4$, còn $K' = d$ với $d = (n - p - q + 5)/8$ là khoá bí mật dùng để ký. Các hàm $sig_{K'}$ và $ver_{K''}$ được xác định như sau:

$$sig_{K'}(x) = x^d \bmod n,$$
$$ver_{K''}(x,y) = \text{đúng} \Leftrightarrow x \equiv y^2 \pmod n.$$

Ta chú ý rằng nếu p và q được chọn với tính chất nói trên thì với mọi $x \in \mathcal{P} = Q_n$, $x^d \bmod n$ là một căn bậc hai của x theo mod$n$, vì

$$x^{2d} \equiv x^{2\frac{(p-1)(q-1)+4}{8}} \equiv x^{\frac{(p-1)(q-1)}{4}+1} \equiv x \pmod n;$$

và các hàm $sig_{K'}$ và $ver_{K''}$ được định nghĩa như trên là hợp thức.

Y tưởng cơ bản về một sơ đồ chữ ký Rabin chỉ đơn giản là như thế, tuy nhiên để có một sơ đồ chữ ký dùng được trong thực tế,

người ta muốn tập các văn bản $\mathcal{P}$ không hạn chế trong Q_n, mà rộng rãi hơn, là Z_n chẳng hạn, nhưng để được như vậy, ta phải dùng thêm một hàm R để chuyển một $x \in \mathcal{P}$ ban đầu về một giá trị m nào đó có quan hệ gần gũi với một thặng dư bậc hai theo mod$n$ để sơ đồ chữ ký theo ý tưởng nói trên có thể vận hành được. Để thực hiện được một sơ đồ chữ ký sửa đổi như vậy, người ta sẽ dùng một bổ đề toán học sau đây:

Bổ đề 5.4.1. Giả sử p và q là các số nguyên tố khác nhau cùng đồng dư với 3 theo mod4, và $n = p.q$. Khi đó ta có:

1) Nếu gcd(x,n) =1, thì $x^{(p-1)(q-1)/2} \equiv 1 \pmod{n}$
2) Nếu $x \in Q_n$, thì $x^{(n-p-q+5)/8} \bmod n$ là một căn bậc hai của x theo mod$n$.
3) Nếu x là số nguyên có $\left(\frac{x}{n}\right) = 1$, và $d = (n-p-q+5)/8$, thì
$$x^{2d} \bmod n = \begin{cases} x, & khi\ x \in Q_n, \\ n-x, & khi\ x \notin Q_n. \end{cases}$$
4) Nếu $p \neq q \pmod 8$ thì $\left(\frac{2}{n}\right) = -1$. Do đó, nhân một số nguyên x bất kỳ với 2 hay với $2^{-1} \bmod n$ đều đảo ngược ký hiệu Jacobi của x.

Người đọc có thể tự chứng minh lấy bổ đề trên.

Bây giờ một <u>sơ đồ chữ ký Rabin sửa đổi</u> có thể được xây dựng như sau : Trước hết ta xác định cho mỗi thực thể tham gia một cặp khoá $K = (K', K'')$, với khoá công khai $K' = n$, khoá bí mật $K'' = (p,q)$ hay $= d = (n-p-q+5)/8$,trong đóp và q là hai số nguyên tố có tính chất $p \equiv 3 \pmod 8$ và $q \equiv 7 \pmod 8$,$n = p.q$;p và q được chọn và giữ bí mật.

Thực thể A có khoá $K = (K', K'')$ sẽ tạo chữ ký trên một văn bản x ($x \in Z_n$, $x \leq (n-6)/16$) bằng các bước sau đây :

a. Tính $m = R(x) = 16x + 6$.
b. Tính ký hiệu Jacobi $J = \left(\frac{m}{n}\right)$.
c. Nếu J=1 thì tính $s = m^d \bmod n$,
   nếu J= -1 thì tính $s = (m/2)^d \bmod n$.
d. s là chữ ký của A trên x.

Việc kiểm thử chữ ký s của A bằng cách dùng khóa công khai n được thực hiện bởi các bước sau đây:

a. Tính $m^* = s^2 \bmod n$
b. Nếu $m^* \equiv 6 \pmod 8$, thì lấy $m = m^*$,
   nếu $m^* \equiv 3 \pmod 8$, thì lấy $m = 2m^*$,
   nếu $m^* \equiv 7 \pmod 8$, thì lấy $m = n - m^*$,

nếu $m^* \equiv 2 \pmod 8$, thì lấy $m = 2(n - m^*)$.

c. Thử điều kiện $m \equiv 6 \pmod{16}$, nếu sai thì bác bỏ chữ ký.

d. Nếu điều kiện trên đúng thì lấy $x = R^{-1}(m) = (m-6)/16$.

(Theo định nghĩa của phép kiểm thử thì ta có thể viết điều d là: thuật toán kiểm thử xác nhận s là chữ ký của A trên văn bản x nếu $x = R^{-1}(m) = (m-6)/16$).

Ta có thể chứng minh tính hợp thức của các thuật toán ký và kiểm thử như sau: Các bước tạo chữ ký b-c cho ta chữ ký Rabin của $v = m$ hay $v = m/2$ tuỳ theo ký hiệu Jacobi bằng 1 hay không. Theo điều 4 của bổ đề 5.4.1, có đúng một khả năng hoặc m, hoặc $m/2$ có giá trị ký hiệu Jacobi bằng 1. Giá trị v được ký là $\equiv 3$ hoặc $\equiv 6 \pmod 8$. Theo điều 3 của bổ đề đó, $s^2 \bmod n = v$ hoặc $= n - v$ là tuỳ theo $v \in Q_n$ hay không. Vì $n \equiv 5 \pmod 8$, có thể xác định một cách duy nhất một trong hai trường hợp đó.

Thí dụ: Giả thử chọn $p = 19$, $q = 31$, do đó $n = 589$ và $d = 68$. A có khoá công khai $n = 589$ và khoá bí mật $d = 68$. Không gian ký gồm các giá trị của m ứng với các giá trị $x = 0,1,2,...,32,33$ cùng với các giá trị của ký hiệu Jacobi tương ứng được cho bởi bảng sau đây:

m	6	22	54	70	86	102	118	134	150
$\left(\frac{m}{589}\right)$	-1	1	-1	-1	1	1	1	1	-1
m	166	182	198	214	230	246	262	278	294
$\left(\frac{m}{589}\right)$	1	-1	1	1	1	1	-1	1	-1
m	326	358	374	390	406	422	438	454	470
$\left(\frac{m}{589}\right)$	-1	-1	-1	-1	-1	1	1	1	-1
m	486	502	518	534	550	566	582		
$\left(\frac{m}{589}\right)$	-1	1	-1	-1	1	-1	1		

Ta tạo chữ ký với thông báo $x = 12$. Tính $m = R(12) = 198$, $\left(\frac{m}{n}\right) = \left(\frac{198}{589}\right) = 1$, và $s = 198^{68} \bmod 589 = 102$. Chữ ký là $s = 102$.

Dùng thuật toán kiểm thử ta có: $m^* = s^2 \bmod n = 102^2 \bmod 589 = 391$. Vì $m^* \equiv 7 \pmod 8$, ta lấy $m = n - m^* = 589 - 391 = 198$. Cuối cùng, tính $x = R^{-1}(m) = (198-6)/16 = 12$, và chữ ký được xác nhận.

5.4.2. Sơ đồ chữ ký Fiat-Shamir.

Mỗi sơ đồ chữ ký Fiat-Shamir sử dụng một hàm băm h : $Z_2^* \to Z_2^k$,biến mọi dãy ký tự nhị phân x độ dài tuỳ ý thành một dãy có độ dài k bit, được gọi là "tóm lược" của x .

Mỗi thực thể A tạo cho mình cặp khoá $K=(K',K'')$ bằng cách: chọn hai số nguyên tố khác nhau p và q, và đặt $n=p.q$; sau đó chọn ngẫu nhiên k số nguyên khác nhau $s_1,...,\ s_k \in Z_n^*$, và tính với mỗi j $(1\le j\le k)$ $v_j = s_j^{-2} \bmod n$. Xác định khoá bí mật K' là bộ k $(s_1,...,\ s_k)$, và khoá công khai K'' là gồm bộ k $(v_1,...,v_k)$ và môđuyn n.

Lấy $\mathcal{P} = Z_2^*$, $\mathcal{A} = Z_2^k \times Z_n$, và xác định các thuật toán ký và kiểm thử như sau:

Để tạo chữ ký trên văn bản $x \in \mathcal{P} = Z_2^*$, A chọn ngẫu nhiên một số nguyên dương $r \in Z_n$, tính $u = r^2 \bmod n$, tính $e = (e_1,...,\ e_k) = h(x \| u)$, trong đó $x \| u$ là dãy ký tự nhị phân thu được bằng cách nối ghép biểu diến nhị phân của số u tiếp sau biểu diễn nhị phân của số x. Chữ ký của A trên x được định nghĩa là (e,s), trong đó

$$s = r.\prod_{j=1}^{k} s_j^{e_j} \bmod n.$$

Để kiểm thử (e,s) có đúng là chữ ký của A trên x hay không, ta dùng khoá công khai $(v_1,...,v_k)$ và môđuyn n để tính

$$w = s^2.\prod_{j=1}^{k} v_j^{e_j} \bmod n,$$

rồi tính $e' = h(x \| w)$; và xác nhận (e,s) đúng là chữ ký của A trên x khi và chỉ khi $e = e'$.

Dễ chứng minh rằng nếu (e,s) là chữ ký của A trên x thì $e = e'$, và ngược lại, tức các thuật toán ký và kiểm thử xác định như trên là hợp thức.

5.4.3. Sơ đồ chữ ký Schnorr.

Sơ đồ chữ ký Schnorr cũng được xây dựng tương tự như sơ đồ Fiat-Shamir, nhưng ở đây ta dùng một hàm băm một phía dựa trên bài toán khó tính lôgarit rời rạc.

Mỗi thực thể A tạo cho mình cặp khoá $K=(K',K'')$ bằng cách: Chọn một số nguyên tố lớn p, một số nguyên tố q là ước số của p -1, một phần tử α cấp q của Z_p^*, và một số a , $1\le a\le q$ -1. Giữ $K'=a$ là khoá bí mật , và công bố khoá công khai $K'' = (p,q,\alpha,r)$, trong đó $r = \alpha^a \bmod p$.

Chọn một hàm băm $h: Z_2^* \to Z_q$. Lấy $\mathcal{P} = Z_2^*$ và $\mathcal{A} = Z_q \times Z_q$.

Để ký trên một thông báo $x \in \mathcal{P} = Z_2^*$ A chọn thêm một số ngẫu nhiên $k \in Z_q$ và tính $y = \alpha^k \bmod p$, $e = h(x \| y)$ và $s = ae+k \bmod q$. Chữ ký của A trên x được xác định là cặp số (s, e).

Để kiểm thử xem cặp số (s, e) có đúng là chữ ký của A trên x hay không, ta dùng khoá công khai $K'' = (p,q,\alpha,r)$ để tính

$$v = \alpha^s r^{-e} \bmod p \quad \text{và} \quad e' = h(x \| v),$$

và xác nhận (s, e) đúng là chữ ký của A trên x khi và chỉ khi $e' = e$.

Ta có thể chứng minh rằng các thuật toán ký và kiểm thử xác định như ậy là hợp thức. Thực vậy, nếu chữ ký(s, e) được ký bởi A trên x, thì

$$v = \alpha^s r^{-e} \bmod p = \alpha^s \alpha^{-ae} \bmod p = \alpha^k \bmod p = y,$$

do đó $e' = h(x \| v) = h(x \| y) = e$. Ngược lại, cũng dễ chứng tỏ rằng nếu $e' = e$ thì (s, e) đúng là chữ ký của A trên x.

5.5.Chữ ký không phủ định được và không chối bỏ được

5.5.1. Đặt vấn đề. Trong các phần trước ta đã trình bày một vài sơ đồ chữ ký điện tử ; trong các sơ đồ đó, việc kiểm thử tính đúng đắn của chữ kýlà do người nhận thực hiện. Như vậy, cả văn bản cùng chữ ký có thể được sao chép và tán phát cho nhiều người mà không được phép của người gửi. Để tránh khả năng đó, người ta đưa ra các *sơ đồ chữ ký không phủ định được* với một yêu cầu là chữ ký không thể được kiểm thử nếu không có sự hợp tác của người ký. Sự hợp tác đó được thực hiện thông qua một *giao thức mời hỏi và trả lời* giữa người nhận và người gửi (cũng là người ký), gọi là giao thức kiểm thử. Khi chữ ký đòi hỏi được xác nhận bằng một giao thức kiểm thử thì một vấn đề khác lại nẩy sinh là làm thế nào để ngăn cản người ký chối bỏ một chữ ký mà anh ta đã ký bằng cách tuyên bố rằng chữ ký đó là giả mạo? Để đáp ứng yêu cầu đó, cần có thêm một giao thức chối bỏ, thông qua giao thức này người ký có thể chứng minh một chữ ký không phải của mình *đúng thực là giả mạo*. Nếu anh ta từ chối không tham gia giao thức đó thì có bằng chứng để chứng tỏ rằng anh ta không chứng minh được đó là chữ ký giả mạo, tức không chối bỏ được chữ ký của mình!

Như vậy, một *sơ đồ chữ ký không phủ định được* sẽ gồm ba phần : một thuật toán ký, một giao thức kiểm thử và một giao thức chối bỏ.

5.5.2. Sơ đồ chữ ký Chaum-van Antverpen.

Sơ đồ chữ ký không phủ định được đầu tiên được Chaum và van Antverpen đề xuất năm 1989. Một chủ thể A chọn một số nguyên tố dạng Sophie Germain $p = 2q + 1$, trong đó q cũng là số

nguyên tố; chọn $\alpha \in Z_p^*$ là một phần tử cấp q. Gọi G là nhóm con (theo phép nhân) cấp q sinh bởi α của Z_p^*. Sơ đồ chữ ký Chaum - van Antverpen của A gồm có: $\mathcal{P} = \mathcal{A} = G$, cặp khoá $K=(K',K'')$ gồm có khoá bí mật $K' = a$ và khoá công khai $K'' = (p, \alpha, a, \beta)$, trong đó α là một số nguyên dương $< p$-1, và $\beta = \alpha^a \bmod p$.

Thuật toán ký: A ký trên văn bản $x \in \mathcal{P} = G$ với chữ ký

$$y = sig_{K'}(x) = x^a \bmod p.$$

Giao thức kiểm thử : Với văn bản x và chữ ký y người nhận B cùng người ký A thực hiện giao thức kiểm thử sau đây:

1. B chọn ngẫu nhiên hai số $e_1, e_2 \in Z_q^*$, tính $c = y^{e_1}.\beta^{e_2} \bmod p$ và gửi c cho A,
2. A tính $d = c^{a^{-1} \bmod q} \bmod p$ và gửi d cho B.
3. B chấp nhận y là chữ ký của A trên x nếu $d \equiv x^{e_1}.\alpha^{e_2} \bmod p$.

Giao thức chối bỏ: gồm các bước sau đây:

1. B chọn ngẫu nhiên hai số $e_1, e_2 \in Z_q^*$, tính $c = y^{e_1}.\beta^{e_2} \bmod p$ và gửi c cho A,
2. A tính $d = c^{a^{-1} \bmod q} \bmod p$ và gửi d cho B,
3. B thử điều kiện $d \not\equiv x^{e_1}.\alpha^{e_2} \pmod p$.
4. B chọn tiếp hai số $f_1, f_2 \in Z_q^*$, tính $C = y^{f_1}.\beta^{f_2} \bmod p$ và gửi C cho A,
5. A tính $D = C^{a^{-1} \bmod q} \bmod p$ và gửi D cho B,
6. B thử điều kiện $D \not\equiv x^{f_1}.\alpha^{f_2} \pmod p$.
7. B kết luận y là chữ ký giả mạo, nếu $(d\alpha^{-e_2})^{f_1} \equiv (D\alpha^{-f_2})^{e_1} \pmod p$.

5.5.3. Tính hợp thức của các giao thức.

Ta sẽ chứng minh hai định lý sau đây để chứng tỏ tính hợp thức của các giao thức kiểm thử và chối bỏ của sơ đồ chữ ký Chaum-van Antverpen.

Định lý 5.5.1. a)*Nếu y đúng là chữ ký của A trên x, tức* $y \equiv x^a \bmod p$, *thì việc B chấp nhận y là chữ ký của A trên x theo giao thức kiểm thử là đúng.*

b) *Nếu* $y \not\equiv x^a \pmod p$, *tức y không phải là chữ ký của A trên x, thì việc B, theo giao thức kiểm thử, chấp nhận y là chữ ký của A trên x, có thể xảy ra với xác suất* $1/q$.

Chứng minh. a) Giả sử $y \equiv x^a \bmod p$. Khi đó, $y^{a^{-1}} \equiv x \pmod p$. (chú ý rằng tất cả các số mũ đều được tính theo mod$q$). Ta cũng có

$\beta^{a^{-1}} \equiv \alpha \pmod p$. Do đó,

$$d \equiv c^{a^{-1}} \equiv y^{e_1 a^{-1}} \beta^{e_2 a^{-1}} \equiv x^{e_1} \alpha^{e_2} \pmod p,$$

và theo giao thức kiểm thử, B chấp nhận y là chữ ký của A trên x, việc chấp nhận đó là đúng.

b)Bây giờ giả thử $y \not\equiv x^a \pmod p$. Trước hêt ta chú ý rằng mỗi lời mời hỏi c tương ứng với đúng q cặp (e_1, e_2), vì y và β là các phần tử của nhóm nhân G cấp q. Khi A nhận được câu hỏi c, A không có cách gì để biết là B đã dùng cặp (e_1, e_2) nào trong q cặp có thể đó. Ta chứng minh rằng, do $y \not\equiv x^a \pmod p$, nên trong q cặp đó chỉ có đúng một cặp thoả mãn đồng dư thức $d \equiv x^{e_1} \alpha^{e_2} \pmod p$. Thực vậy, ta có thể đặt $c = \alpha^i, d = \alpha^j, x = \alpha^k, y = \alpha^l$ với $i, j, k, l \in Z_q$, vì α là phần tử sinh của G, và hai đồng dư thức $c \equiv y^{e_1} \beta^{e_2} \pmod p$ và $d \equiv x^{e_1} \alpha^{e_2} \pmod p$ tương đương với hai phương trình

$$i \equiv le_1 + ae_2 \pmod q$$
$$j \equiv ke_1 + e_2 \pmod q.$$

Từ giả thiết $y \not\equiv x^a \pmod p$ suy ra $l - ak \not\equiv 0 \pmod q$, tức định thức của hệ phương trình nói trên (với các ẩn số e_1, e_2) là $\not\equiv 0 \pmod q$. Như vậy, mỗi $d \in G$ là câu trả lời đúng (theo giao thức kiểm thử) chỉ với một cặp (e_1, e_2) trong q cặp có thể. Vì vậy, nếu $y \not\equiv x^a \pmod p$, thì xác suất để B chấp nhận y là chữ ký của A trên x (theo giao thức) là bằng $1/q$. Định lý được chứng minh.

Đối với giao thức chối bỏ, ta có định lý sau đây :

Định lý 5.5.2. a) *Nếu* $y \not\equiv x^a \pmod p$, *và cả A,B đều tuân theo giao thức chối bỏ, thì* $(d\alpha^{-e_2})^{f_1} \equiv (D\alpha^{-f_2})^{e_1} \pmod p$, *tức giao thức cho kết quả chính xác.*

b) *Nếu* $y \equiv x^a \bmod p$, *A và B đều tuân theo giao thức, và có*

$$d \not\equiv x^{e_1}.\alpha^{e_2} \pmod p.$$
$$D \not\equiv x^{f_1}.\alpha^{f_2} \pmod p.$$

Khi đó, đồng dư thức $(d\alpha^{-e_2})^{f_1} \equiv (D\alpha^{-f_2})^{e_1} \pmod p$ *đúng với xác suất* $1/q$, *tức nếu y đúng là chữ ký của A trên x, thì theo giao thức, B có thể kết luận rằng nó là giả mạo (một cách sai lầm) với xác suất* $1/q$.

Chứng minh. a) Giả thử $y \not\equiv x^a \pmod p$, và A,B cùng thực hiện giao thức chối bỏ. Do y không là chữ ký của A trên x nên B sẽ kiểm thử đúng các bất đồng dư thức trong các bước 3 và 6 của giao thức. Vì $\beta \equiv \alpha^a \pmod p$, nên ta có

$$(d\alpha^{-e_2})^{f_1} \equiv ((y^{e_1} \beta^{e_2})^{a^{-1}} \alpha^{-e_2})^{f_1} \pmod p$$

$$\equiv y^{a^{-1}e_1 f_1}\beta^{e_2 a^{-1} f_1}\alpha^{-e_2 f_1} \pmod{p}$$
$$\equiv y^{e_1 a^{-1} f_1} \pmod{p}.$$

Tương tự, ta cũng có

$$(D\alpha^{-f_2})^{e_1} \equiv y^{e_1 a^{-1} f_1} \pmod{p}.$$

Như vậy, đồng dư thức ở điểm 7 của giao thức được nghiệm đúng, và kết luận y là chữ ký giả mạo của A trên x là chính xác, không thể bác bỏ được.

b) Bây giờ giả thiết $y \equiv x^a \pmod{p}$, và A, B cùng thực hiện giao thức chối bỏ. Đặt $x_0 = d^{1/e_1}\alpha^{-e_2/e_1} \bmod p$, ta có

$$x_0^a \equiv d^{a/e_1}.\alpha^{-ae_2/e_1} \not\equiv (x^{e_1}\alpha^{e_2})^{a/e_1}\alpha^{-ae_2/e_1} \equiv x^a \equiv y \pmod{p}.$$

Theo điểm b) trong định lý 5.5.1, B có thể chấp nhận y là chữ ký của A trên x_0, tức là có đồng dư thức

$$D \equiv x_0^{f_1}\alpha^{f_2} \pmod{p},$$

với xác suất $1/q$. Nhưng đồng dư thức đó tương đương với đồng dư thức

$$(d\alpha^{-e_2})^{f_1} \equiv (D\alpha^{-f_2})^{e_1} \pmod{p},$$

tức đồng dư thức này cũng có thể xẩy ra với xác suất $1/q$. Định lý được chứng minh.

Ta chú ý rằng trong giao thức chối bỏ, cặp (e_1, e_2) được sử dụng để tạo ra x_0 với $x_0^a \not\equiv y \pmod{p}$; còn cặp ($f_1$, f_2) được dùng để kiểm thử xem y có là chữ ký của A trên x_0 hay không.

Thí dụ minh hoạ. Chọn p = 467, q =233 (p = 2q +1), α =4 là phần tử sinh của một nhóm con G cấp 233 của Z_{467}^*. Chọn a =101, khi đó ta có $\beta = \alpha^a \bmod p = 4^{101} \bmod 467 = 449$.

A có cặp khoá K =(K',K'') với K' =101, và K'' = (467, 4, 449). Giả thử A ký trên văn bản x =119 với chữ ký

$$y = 119^{101} \bmod 467 = 129.$$

1)B có thể dùng giao thức kiểm thử để biết y có đúng là chữ ký của A trên x hay không như sau: B chọn ngẫu nhiên e_1=38, e_2=397, và tính c =13; A sẽ trả lời lại bằng d =9. B thử điều kiện

$$d \equiv x^{e_1}.\alpha^{e_2} \bmod p,$$

tức là
$$9 \equiv 119^{38}.4^{397} \pmod{467}.$$

Đồng dư thức đó đúng. B chấp nhận 129 đúng là chữ ký của A trên văn bản 119.

2) Bây giờ ta thử thực hiện giao thức chối bỏ. Giả thử A gửi văn bản x =286 với chữ ký y = 83. B chọn ngẫu nhiên e_1=45, e_2=237, rồi tính c =305 và gửi cho A; A trả lời lại bằng d =109. B thử điều kiện $d \not\equiv x^{e_1}.\alpha^{e_2} \pmod{p}$, điều kiện đó được thoả mãn vì

$109 \neq 149 (= 286^{45}.4^{237} \bmod 467)$. B lại tiếp tục phần sau của giao thức bằng cách chọn ngẫu nhiên $f_1 = 125$, $f_2 = 9$, và tính $C = 270$, gửi cho A, A trả lời lại bằng $D = 68$. B lại thử điều kiện $D \equiv x^{f_1}.\alpha^{f_2} (\bmod p)$, điều kiện này cũng được thoả mãn vì $68 \neq 25 (= 286^{125}.4^{9} \bmod 467)$. Bây giờ B lại thử điều kiện cuối cùng của giao thức bằng cách tính

$$(d\alpha^{-e_2})^{f_1} \equiv (109.4^{-237})^{125} \equiv 188 (\bmod 467)$$

$$(D\alpha^{-f_2})^{e_1} \equiv (68.4^{-9})^{45} \equiv 188 (\bmod 467)$$

Hai giá trị đó bằng nhau. B có thể kết luận y không phải là chữ ký của A trên x với xác suất sai lầm là $1/233$!

Thí dụ này được trình bày với mục đích minh hoạ, nên chỉ sử dụng các số nguyên tố p, q bé cho dễ tính. Trong thực tế ứng dụng, để bảo đảm tính an toàn, ta phải dùng các số p, q rất lớn, chẳng hạn phải là các số có biểu diễn nhị phân cỡ 512 bit, khi đó ta có $q \geq 2^{510}$, tức là $1/q \leq 2^{-510}$, một xác suất rất bé, có thể bỏ qua; và vì vậy, các yêu cầu đối với các giao thức kiểm thử và giao thức chối bỏ như đề cập đến trong phần đặt vấn đề (5.5.1) có thể xem là được thoả mãn.

CHƯƠNG VI

Các sơ đồ xưng danh và xác nhận danh tính

6.1. Vấn đề xưng danh.

Trong chương trước ta đã thấy các kỹ thuật mật mã có thể được ứng dụng để xây dựng nhiều giải pháp an toàn cho vấn đề xác nhận các thông báo cùng với người gửi trên các mạng truyền tin công cộng. Trong chương này ta sẽ xét việc ứng dụng cũng các kỹ thuật đó cho bài toán xây dựng các sơ đồ xưng danh và xác nhận danh tính, cũng là một bài toán quan trọng và thường gặp trong mọi hoạt động giao lưu thông tin, đặc biệt giao lưu qua mạng. Việc xưng danh và xác nhận danh tính của một người thường là cần thiết trong những tình huống như:

- Để rút tiền từ các máy rút tiền tự động (ATM), ta cần xưng danh bằng cách dùng một thẻ rút tiền cùng với một số PIN (số xưng danh cá nhân) của mình

- Để mua hàng hoặc thanh toán một khoản tiền qua mạng điện thoại, ta cần thông báo số thẻ tín dụng (cùng ngày hết hạn) của mình.

- Để truy nhập vào một máy tính trên một mạng, ta cần khai báo tên người dùng cùng mật hiệu (password) của mình.

- v.v...

Trong thực tế cuộc sống, việc xưng danh theo thói quen thường không đòi hỏi tính an toàn, chẳng hạn các số PIN, mật khẩu thường không có gì để bảo đảm là được giữ kín, người ngoài không biết được. Tuy nhiên, cuộc sống càng ngày càng được tin học hoá, phần lớn các giao dịch được thực hiện trên các mạng tin học, việc xem thường các yêu cầu về an toàn trong các khâu xưng danh và xác nhận danh tính là không thể tiếp tục được; cần phải có những giải pháp bảo đảm tính an toàn cho các hoạt động đó.

Mục tiêu an toàn của việc xưng danh là bảo đảm sao cho khi "nghe" một chủ thể A xưng danh với một chủ thể B, bất kỳ một ai

khác A cũng không thể sau đó mạo nhận mình là A, kể cả chính B cũng không thể mạo xưng mình là A sau khi được A xưng danh với mình. Nói cách khác, A muốn chứng minh để được đối tác xác nhận danh tính của mình mà không để lộ bất cứ thông tin nào về việc chứng minh danh tính đó.

Việc xưng danh thường phải thông qua một giao thức hỏi-đáp nào đó, qua giao thức đó, để B có thể xác nhận danh tính của A, B đặt cho A một câu hỏi; A phải trả lời, trong trả lời đó A phải chứng tỏ cho B biết là A có sở hữu một *bí mật* riêng A mới có, điều đó thuyết phục B tin chắc rằng người trả lời đúng là A và do đó xác nhận danh tính của A. Vấn đề khó ở đây là A phải làm cho B biết là A có sở hữu một *bí mật* chỉ riêng A mới có, nhưng lại không được lộ cho B biết cái *bí mật* riêng A mới có đó là cái gì. Mặt khác, để cho việc "A có sở hữu một *bí mật* của riêng A" đó là đáng tin (dù là không biết) thì cần được chứng thực bởi một bên thứ ba nào đó,chẳng hạn bởi một cơ quan được uỷ thác (trusted authority). Tất nhiên cơ quan được uỷ thác này cũng không biết bản thân bí mật của A, nhưng biết và chứng nhận A là chủ sở hữu của một yếu tố công khai mà việc A sử dụng nó chứng tỏ A có cái bí mật nói trên.

Trong tiết ngay sau đây ta sẽ giới thiệu một sơ đồ xưng danh điển hình để minh hoạ các ý tưởng nói trên.

6.2. Sơ đồ xưng danh Schnorr.

Trong sơ đồ xưng danh này có sự tham gia của một cơ quan được uỷ thác mà ta ký hiệu là TA. TA sẽ chọn các tham số cho sơ đồ xưng danh như sau:

- một số nguyên tố lớn p sao cho bài toán tính lôgarit rời rạc theo mod$p$ là rất khó; và một ước số nguyên tố q của p-1 (người ta khuyên nên chọn $p \geq 2^{512}$ và $q \geq 2^{140}$).

- một phần tử $\alpha \in Z_p^*$ có cấp q (một phần tử α như vậy có thể lấy là một luỹ thừa bậc $(p-1)/q$ của một phần tử nguyên thuỷ theo mod$p$.

- một tham số an toàn t sao cho $q \geq 2^t$. Có thể lấy t=40.

- TA chọn cho mình một sơ đồ chữ ký gồm một thuật toán ký(bí mật) sig_{TA} và một thuật toán kiểm thử (công khai) ver_{TA}.

- một hàm băm an toàn (một phía và không va chạm mạnh). Ta giả thiết là mọi thông tin đều được "tóm lược" bởi hàm băm trước khi được ký; tuy nhiên trong mô tả sau đây để cho đơn giản ta sẽ bỏ qua các bước sử dụng hàm băm.

Các tham số p, q, α, thuật toán kiểm thử ver_{TA} và hàm băm đều có thể được công bố công khai.

Bây giờ, một chủ thể A cần xưng danh sẽ yêu cầu TA cấp cho mình một chứng chỉ. Thủ tục cấp chứng chỉ cho A được tiến hành như sau:

1.TA xác lập các thông tin về danh tính của A như họ,tên, ngày sinh, số chứng minh hoặc hộ chiếu, v.v... dưới dạng một dãy ký tự mà ta ký hiệu là I_A hay ID(A).

2. A chọn bí mật một số ngẫu nhiên a ($0 \le a \le q$-1), tính

$$v = \alpha^{-a} \bmod p$$

và chuyển số v cho TA.

3. TA tạo chữ ký $s = sig_{TA}(I_A, v)$ và cấp cho A chứng chỉ

$$C(A) = (ID(A), v, s).$$

Như vậy, chứng chỉ mà TA cấp cho A gồm (I_A, v) và chữ ký của TA trên thông tin (I_A, v) đó. Chú ý rằng TA cấp chứng chỉ cho A mà hoàn toàn không biết gì về thông tin bí mật của A là số a.

Bây giờ, với chứng chỉ C(A) đó, A có thể xưng danh với bất kỳ đối tác B nào bằng cách cùng B thực hiện một giao thức xác nhân danh tính như sau:

1. A chọn thêm một số ngẫu nhiên k ($0 \le k \le q$-1), tính

$$\gamma = \alpha^k \bmod p,$$

và gửi cho B các thông tin C(A) và γ.

2. B kiểm thử chữ ký của TA trong chứng chỉ C(A) bởi hệ thức $ver_{TA}(ID(A), v, s)$ =đúng. Kiểm thử xong, B chọn một số ngẫu nhiên r ($1 \le r \le 2^t$) và gửi r cho A.

3. A tính $y = k + ar \bmod q$ và gửi y cho B.

4. B thử điều kiện

$$\gamma \equiv \alpha^y v^r \pmod p$$

và nếu điều kiện đó được thoả mãn thì xác nhận danh tính của A.

Thực hiện giao thức đó, A sẽ chứng minh được danh tính của mình, vì

$$\alpha^y v^r \equiv \alpha^{k+ar} v^r \equiv \alpha^{k+ar} \alpha^{-ar} \equiv \alpha^k \pmod p$$
$$\equiv \gamma \pmod p,$$

tức điều kiện mà B cần thử là đúng.

Sơ đồ xưng danh cùng với giao thức xác nhận danh tính như mô tả ở trên có các tính chất đáp ứng các yêu cầu như đề ra từ phần đặt vấn đề ở tiết 6.1. Điều vừa chứng minh ở trên chứng tỏ rằng nếu A tuân thủ giao thức thì B xác nhận danh tính của A là đúng (B tin rằng A quả thực có sở hữu một bí mật a, dù B cũng không biết cái bí mật a đó là số nào).

Bây giờ ta xét khả năng một người O muốn giả danh A để giao dịch với B. Khả năng thứ nhất là O tạo ra một chứng chỉ giả mạo với danh tính của A, một chứng chỉ như vậy có dạng

$$C'(A) = (ID(A), v□, s'),$$

trong đó $v□ \neq v$. Để tạo ra một chứng chỉ như vậy thì O phải tạo ra được $s□$ là chữ ký của TA trên (ID(A), $v□$), O không biết thuật toán ký sig_{TA} nên không thể tạo ra chữ ký đúng của TA được, và nếu lấy s' là một chữ ký giả mạo, thì khi thực hiện điểm 2 của giao thức xác nhận danh tính thể nào B cũng phát hiện ra. Khả năng thứ hai là O vẫn dùng chứng chỉ thật C(A) của A, tự chọn một số k và tính số γ tương ứng theo điểm 1 của giao thức xác nhận danh tính. Vấn đề ở đây là khi B gửi đến số r, O phải trả lời lại bằng một số y sao cho điều kiện $\gamma \equiv \alpha^y v^r \pmod{p}$ được nghiệm đúng. Điều này xem ra là rất khó, ít nhất cũng khó như là O biết bí mật về số a của A vậy. Thực vậy, giả sử O có khả năng nói trên, khi đó ta cho hai lần hỏi r_1 và r_2 O sẽ có hai trả lời y_1 và y_2, và ta có

$$\gamma \equiv \alpha^{y_1} v^{r_1} \equiv \alpha^{y_2} v^{r_2} \pmod{p},$$

từ đó suy ra

$$\alpha^{y_1 - y_2} \equiv v^{r_2 - r_1} \pmod{p}.$$

Vì $v = \alpha^{-a}$, ta có

$$y_1 — y_2 \equiv a(r_2 — r_1) \pmod{q}.$$

Vì q là số nguyên tố $> 2^t$ và $0 < |r_2 - r_1| < 2^t$, nên gcd($r_2 - r_1$, q) =1, và O có thể tính được $a = (y_1 - y_2)(r_2 - r_1)^{-1} \bmod q$.

Thí dụ : Lấy p =88667, q = 1031 và t =10. Phần tử α = 70322 có cấp q trong Z_p^*. Giả sử A chọn số mũ bí mật là a = 755, khi đó v = 13136.

A và B có thể thực hiện giao thức xác định danh tính như sau: A chọn k = 543, và tính γ =70322^{543}mod88667 =84109 rồi gửi γ cho B. Giả sử B gửi r =1000 cho A, A trả lời lại bằng y =k +ar mod$q$ = =543+755.1000mod1031 = 851. B thử điều kiện $\gamma \equiv \alpha^y v^r \pmod{p}$, trong trường hợp này là:

$$84109 \equiv 70322^{851} 13136^{1000} \pmod{88667},$$

đó là đồng dư thức đúng. B xác nhận danh tính của A.

Bây giờ vẫn với các tham số trên, giả thiết O có khả năng trả lời đúng hai câu hỏi r_1=1000 và r_2=19 của B bằng y_1=851 và y_2=454. Khi đó O có thể tính được

$$a = (y_1 - y_2)(r_2 - r_1)^{-1} \bmod q$$
$$= (851\text{-}454)(19\text{-}1000)^{-1} \bmod 1031 = 755,$$

đúng là số bí mật của A.

Sơ đồ xưng danh Schnorr, với giao thức xác nhận danh tính như định nghĩa ở trên, là có tính chất *đầy đủ* (việc có bí mật a bảo đảm A chứng minh được danh tính của mình), và *đúng đắn* (việc giả danh A thành công cũng khó như biết bí mật của A); tuy nhiên như vừa trình bày trong thí dụ trên, sơ đồ đó chưa phải là *an toàn*,

việc giả danh là khó nếu O không hề biết gì về sơ đồ xưng danh đó, chứ nếu, chẳng hạn, O đã được A xưng danh với ít nhất hai lần (tức hai lần biết được hai cặp số (r_1, y_1) và (r_2, y_2)) thì có khả năng O phát hiện được bí mật của A, như vậy việc xưng danh của A không còn an toàn nữa!

Để khắc phục điểm yếu đó của sơ đồ Schnorr, Okamoto đã đề xuất một sửa đổi làm cho sơ đồ trở nên an toàn, sửa đổi này dựa trên tính khó của một bài toán đặc biệt về tính lôgarit rời rạc. Ta trình bày trong tiết sau đây sơ đồ được sửa đổi đó.

6.3. Sơ đồ xưng danh Okamoto.

Cũng như đối với sơ đồ Schnorr, sơ đồ xưng danh Okamoto cần có một cơ quan uỷ thác TA để cấp chứng chỉ cho các người tham gia.

TA chọ trước các số nguyên tố p và q như đối với sơ đồ Schnorr. Sau đó, TA chọn hai số $\alpha_1, \alpha_2 \in Z_p^*$, cùng có cấp q. Giá trị $c = \log_{\alpha_1} \alpha_2$ (tức giá trị c sao cho $\alpha_1^c = \alpha_2$) được giữ tuyệt mật đối với mọi người tham gia, kể cả A; nói cách khác, ta giả thiết rằng việc tính ra c là cực kỳ khó đối với bất kỳ ai (chẳng hạn, A,O, hoặc thậm chí liên minh của A và O,...).

Thủ tục <u>cấp chứng chỉ cho A</u> được tiến hành như sau:

1. TA xác lập các thông tin về danh tính của A dưới dạng một dãy ký tự mà ta ký hiệu là I_A hay ID(A).

2. A chọn bí mật hai số ngẫu nhiên a_1, a_2 ($0 \le a_1, a_2 \le q$-1), tính

$$v = \alpha_1^{-a_1} \alpha_2^{-a_2} \bmod p,$$

và chuyển số v cho TA.

3. TA tạo chữ ký $s = sig_{TA}(I_A, v)$ và cấp cho A chứng chỉ

$$C(A) = (ID(A), v, s).$$

Bây giờ, với chứng chỉ C(A) đó, A có thể xưng danh với bất kỳ đối tác B nào bằng cách cùng B thực hiện một <u>giao thức xác nhận danh tính</u> như sau:

1. A chọn thêm hai số ngẫu nhiên k_1, k_2 ($0 \le k_1, k_2 \le q$-1), tính

$$\gamma = \alpha_1^{k_1} \alpha_2^{k_2} \bmod p,$$

và gửi cho B các thông tin C(A) và γ.

2. B kiểm thử chữ ký của TA trong chứng chỉ C(A) bởi hệ thức $ver_{TA}(ID(A), v, s)$ =đúng. Kiểm thử xong, B chọn một số ngẫu nhiên r ($1 \le r \le 2^t$) và gửi r cho A.

3. A tính

$$y_1 = k_1 + a_1 r \bmod q,$$
$$y_2 = k_2 + a_2 r \bmod q,$$

và gửi y_1, y_2 cho B.

4. B thử điều kiện

$$\gamma \equiv \alpha_1^{y_1}\alpha_2^{y_2}v^r \pmod{p}$$

và nếu điều kiện đó được thoả mãn thì xác nhận danh tính của A.

Thực hiện giao thức đó, A sẽ chứng minh được danh tính của mình, vì

$$\begin{aligned}\alpha_1^{y_1}\alpha_2^{y_2}v^r &\equiv \alpha_1^{k_1+a_1r}\alpha_2^{k_2+a_2r}\alpha_1^{-a_1r}\alpha_2^{-a_2r} \pmod{p}\\ &\equiv \alpha_1^{k_1}\alpha_2^{k_2} \pmod{p}\\ &\equiv \gamma \pmod{p}\end{aligned}$$

tức điều kiện mà B cần thử là đúng. Như vậy, do biết cặp số bí mật (a_1, a_2), nên A có thể thực hiện thông suốt giao thức xác nhận để chứng minh danh tính của mình.

Ngược lại, một người khác A, do không biết cặp số bí mật (a_1, a_2), nên khó có khả năng tính đúng được (y_1,y_2) để trả lời B ở bước 3 của giao thức, tức là không vượt qua được sự kiểm thử của giao thức để mạo nhận mình là A.

Bây giờ giả sử có một người O có thể thực hiện thông suốt giao thức xác nhận để có thể được mạo nhận là A, chẳng hạn ít nhất hai lần. Điều đó có nghĩa là O biết được hai số $r \neq s$ và hai cặp số (y_1,y_2), (z_1,z_2) sao cho

$$\gamma \equiv \alpha_1^{y_1}\alpha_2^{y_2}v^r \equiv \alpha_1^{z_1}\alpha_2^{z_2}v^s \pmod{p}.$$

Đặt

$$b_1 = (y_1 - z_1)(r-s)^{-1} \bmod q,$$

$$b_2 = (y_2 - z_2)(r-s)^{-1} \bmod q,$$

ta sẽ được

$$v \equiv \alpha_1^{-b_1}\alpha_2^{-b_2} \pmod{p},$$

do đó

$$\alpha_1^{-b_1}\alpha_2^{-b_2} \equiv \alpha_1^{-a_1}\alpha_2^{-a_2} \pmod{p},$$

tức là

$$\alpha_1^{a_1-b_1} \equiv \alpha_2^{b_2-a_2} \pmod{p}.$$

Giả thiết rằng O và A liên minh với nhau, khi đó biết được cả các số a_1, a_2, b_1, b_2. Nếu giả thiết $(a_1, a_2) \neq (b_1, b_2)$ thì $a_2 \neq b_2$, và $(b_2 - a_2)^{-1} \bmod q$ tồn tại, và lôgarit rời rạc c được tính bởi

$$c = \log_{\alpha_1} \alpha_2 = (a_1 - b_1)(b_2 - a_2)^{-1} \bmod q.$$

Như vậy, nếu O có thể thực hiện thông suốt giao thức xác nhận để được mạo nhận là A thì khi O và A liên minh với nhau có thể tìm được khá dễ dàng lôgarit rời rạc c. Nhưng từ đầu ta đã giả thiết việc tìm ra c là cực kỳ khó đối với bất kỳ ai (là A, là O, thậm chí là liên minh của A và O,...), nên cũng sẽ cực kỳ khó để O thực hiện được thông suốt giao thức xác nhận với mục đích mạo xưng là A. Vậy là ta đã chứng minh được tính *an toàn* của sơ đồ xưng danh

Okamoto với giao thức xác nhận danh tính như mô tả ở trên. Trong chứng minh đó còn một số chỗ tinh tế cần được bổ sung thêm, chẳng hạn như vì sao có thể giả thiết $(a_1, a_2) \neq (b_1, b_2)$, thực ra người ta đã chứng minh được rằng xác suất của khả năng $(a_1, a_2) = (b_1, b_2)$ là rất bé, không đáng kể. Tuy nhiên, để đơn giản trình bày, xin phép được bỏ qua một vài chi tiết chứng minh tinh tế đó.

6.4. Sơ đồ xưng danh Guillou-Quisquater.

Sơ đồ Guillou-Quisquater cũng được xây dựng theo cùng một cách thức như các sơ đồ Schnorr và Okamoto kể trên, nhưng bài toán khó mà ta dựa vào ở đây không phải là bài toán tính lôgarit rời rạc mà là bài toán RSA.

Sơ đồ cũng cần có sự tham gia của một cơ quan uỷ thác TA để cấp chứng chỉ cho các người tham gia. TA chọn hai số nguyên tố lớn p và q và tính tích $n = pq$, giữ bí mật p, q và công khai n. Các tham số đó được chọn sao cho bài toán phân tích n thành thừa số là rất khó. TA cũng chọn thêm một số b là số nguyên tố có độ lớn khoảng 2^{40} như là một tham số an toàn. Số b cũng được xem là số mũ thoả mãn điều kiện RSA, nghĩa là việc tính $v = u^b \bmod n$ là dễ, nhưng việc tính ngược u từ v là rất khó, nếu không biết p, q.

Thủ tục cấp chứng chỉ cho một người tham gia A được tiến hành như sau:

1.TA xác lập các thông tin về danh tính của A dưới dạng một dãy ký tự mà ta ký hiệu là I_A hay $ID(A)$.

2. A chọn bí mật một số ngẫu nhiên u $(0 \leq u \leq n\text{-}1)$, tính

$$v = (u^{-1})^b \bmod n,$$

và chuyển số v cho TA.

3. TA tạo chữ ký $s = sig_{TA}(I_A, v)$ và cấp cho A chứng chỉ

$$C(A) = (ID(A), v, s).$$

Như vậy, chứng chỉ mà TA cấp cho A gồm (I_A, v) và chữ ký của TA trên thông tin (I_A, v) đó. Chú ý rằng TA cấp chứng chỉ cho A mà có thể không biết gì về thông tin bí mật của A là số u.

Bây giờ, với chứng chỉ C(A) đó, A có thể xưng danh với bất kỳ đối tác B nào bằng cách cùng B thực hiện một giao thức xác nhận danh tính như sau:

1. A chọn thêm một số ngẫu nhiên k $(0 \leq k \leq n\text{-}1)$, tính

$$\gamma = k^b \bmod n,$$

và gửi cho B các thông tin C(A) và γ.

2. B kiểm thử chữ ký của TA trong chứng chỉ C(A) bởi hệ thức $ver_{TA}(ID(A), v, s)$ =đúng. Kiểm thử xong, B chọn một số ngẫu nhiên r $(1 \leq r \leq b\text{-}1)$ và gửi r cho A.

3. A tính $y = k.u^r \bmod n$ và gửi y cho B.

4. B thử điều kiện

$$\gamma \equiv v^r y^b \pmod{n}$$

và nếu điều kiện đó được thoả mãn thì xác nhận danh tính của A.

Cũng như các trường hợp trước, việc chứng minh tính *đầy đủ* của sơ đồ là rất đơn giản:

$$\begin{aligned} v^r y^b &\equiv (u^{-b})^r (ku^r)^b \pmod{n} \\ &\equiv u^{-br} k^b u^{br} \pmod{n} \\ &\equiv k^b \equiv \gamma \pmod{n}. \end{aligned}$$

Một người khác A, do không biết số bí mật u, nên không thể tính đúng được số y ở bước 3 của giao thức để được B xác nhận (như là A) ở bước 4, tức không thể mạo nhận mình là A; đó là tính *đúng đắn* của sơ đồ.

Giả sử có một người O có thể thực hiện thông suốt giao thức xác nhận để có thể được mạo nhận là A, chẳng hạn ít nhất hai lần. Điều đó có nghĩa là O biết được hai số $r_1 \neq r_2$ và hai số y_1, y_2 sao cho

$$\gamma \equiv v^{r_1} y_1^b \equiv v^{r_2} y_2^b \pmod{n}.$$

Giả thiết $r_1 > r_2$, khi đó ta có

$$v^{r_1 - r_2} \equiv (y_2 / y_1)^b \pmod{n}.$$

Do $0 < r_1 - r_2 < b$ và b là số nguyên tố nên $\gcd(r_1 - r_2, b) = 1$, có thể tính được dễ dàng $t = (r_1 - r_2)^{-1} \bmod b$, và có

$$v^{(r_1 - r_2)t} \equiv (y_2 / y_1)^{bt} \pmod{n}.$$

Do $t = (r_1 - r_2)^{-1} \bmod b$ nên ta có

$$(r_1 - r_2)t = lb + 1$$

với l là một số nguyên dương nào đó, vì vậy,

$$v^{lb+1} \equiv (y_2 / y_1)^{bt} \pmod{n},$$

hay là

$$v \equiv (y_2 / y_1)^{bt} (v^{-1})^{lb} \pmod{n}.$$

Nâng cả hai về lên luỹ thừa bậc $b^{-1} \bmod \phi(n)$, ta được

$$u^{-1} \equiv (y_2 / y_1)^t (v^{-1})^l \pmod{n}.$$

cuối cùng, tính nghịch đảo của hai vế theo mod$n$ ta được

$$u = (y_1 / y_2)^t v^l \bmod n.$$

Như vậy, O tính được số bí mật u trong thời gian đa thức! Theo giả thiết, điều đó không thể xẩy ra, vì vậy, giả thiết về việc O có thể thực hiện thông suốt giao thức xác nhận để được mạo nhận danh tính là A là không đúng; sơ đồ xưng danh được chứng minh là *an toàn*.

Thí dụ: Giả sử TA chọn p =467, q =479, như vậy n =223693, TA cũng chọn thêm b=503.

Giả sử A chọn số bí mật u=101576, và tính

$$v=(101576^{-1})^{503} \bmod 223693$$
$$= 89888.$$

TA tạo chữ ký $s=sig_{TA}(ID(A), v)$ và cấp cho A chứng chỉ

$$C(A) = (ID(A), v, s).$$

Giả thiết A muốn xưng danh với B, A chọn k=187485, và gửi cho B giá trị γ =$187485^{503} \bmod 223693$ =24412. B dùng thuật toán kiểm thử ver_{TA} để thử điều kiện $ver_{TA}(ID(A), v, s)$ = đúng, sau đó gửi đến A câu hỏi r = 375. A sẽ trả lời lại bằng

$$y=187485.101576^{375} \bmod 223693$$
$$= 93725.$$

B thử điều kiện $\gamma \equiv v^r y^b \pmod n$, trong trường hợp này là

$$24412 \equiv 89888^{375}. 93725^{503} \pmod{223693},$$

đồng dư thức đó đúng. Vậy B xác nhận danh tính của A.

Bây giờ ta lại giả thiết là O biết được hai số r_1=401, r_2=375 và các số tương ứng y_1=103386 và y_2=93725. O biết rằng

$$v^{401}.103386^b \equiv v^{375}. 93725^b \pmod n.$$

O sẽ tính

$$t=(r_1 - r_2)^{-1} \bmod b = (401-375)^{-1} \bmod 503 = 445,$$

sau đó tính được

$$l = \frac{(r_1 - r_2)t - 1}{b} = \frac{(401-375)445 - 1}{503} = 23.$$

Cuối cùng, O sẽ tìm được giá trị bí mật u là

$$u = (y_1 / y_2)^t v^l \bmod n$$
$$= (103386/93725)^{445}.89888^{23} \bmod 223693$$
$$= 101576,$$

là số bí mật của A.

Chú ý: Sơ đồ xưng danh Guillou-Quisquater, cũng như các sơ đồ Schnorr và Okamoto trước đó, đều cần có chứng chỉ của TA cho mỗi người tham gia. Ta có thể thay đổi chút ít để biến sơ đồ xưng danh đó thành một **sơ đồ xưng danh dựa vào danh tính** mà không cần có chứng chỉ như sau: Sơ đồ dùng một hàm băm công khai h, và thay cho việc cấp chứng chỉ C(A) cho người tham gia A, TA sẽ cấp cho A danh tính $ID(A)$ cùng một số u được tính bởi công thức

$$u=(h(ID(A))^{-1})^a \bmod n .$$

(a là một số mũ bí mật của TA). Số u được A giữ riêng cho mình. Khi A cần xưng danh với B, A và B cùng thực hiện một giao thức xác nhận danh tính sau đây:

1. A chọn một số ngẫu nhiên k, $0 \le k \le n-1$, và tính

$$\gamma = k^b \bmod n,$$

rồi gửi ID(A) và γ cho B.

2. B tính $v = h(ID(A))$; chọn một số ngẫu nhiên r ($0 \le r \le 1$) và gửi r cho A.

3. A tính $y = ku^r \bmod n$ và gửi y cho B.

4. B thử điều kiện $\gamma \equiv v^r y^b \pmod{n}$ để xác nhận danh tính của A.

Khi xưng danh theo giao thức nói trên với B, A chỉ cần biết giá trị u là một giá trị được tính bởi TA (và chỉ TA tính được giá trị đó). O không thể giả mạo danh tính của A vì O không biết giá trị u.

6.5. Giao thức Feige-Fiat-Shamir.

Giao thức xưng danh Feige-Fiat-Shamir mà ta sẽ giới thiệu trong tiết này thường được xem là một giao thức điển hình, trong đó một chủ thể tự xưng danh bằng cách chứng minh là mình biết một bí mật với việc dùng một kiểu chứng minh mà ta sẽ gọi là *chứng minh không lộ tri thức* (zero-knowledge proof), tức là trong chứng minh đó không tiết lộ bất cứ một thông tin dù nhỏ nào liên quan đến giá trị bí mật của chủ thể xưng danh. Ở đây,thuật ngữ "tri thức" chỉ được dùng với một nghĩa rất hạn chế để nói về việc **biết** một bí mật của một chủ thể, mà cái biết này thường khi chỉ là biết một bit (0 hoặc 1, đúng hoặc sai), không lộ tri thức là không tiết lộ cái biết về một bit đó. Trong tiết sau ta sẽ đề cập đến các "chứng minh không lộ tri thức" với một nghĩa rộng hơn, khi đó "tri thức" sẽ có nghĩa là biết chứng minh của một bài toán, và chứng minh không lộ tri thức sẽ có nghĩa là thuyết phục một đối tác tin rằng mình biết cách chứng minh của bài toán đó, và ngoài việc bị thuyết phục đó ra thì đối tác không khai thác được bất cứ thông tin gì khác để có thể lặp lại chứng minh đó cả.

Bây giờ ta trở lại với việc trình bày giao thức xưng danh Feige-Fiat-Shamir.

Ở bước chuẩn bị, trung tâm được uỷ thác (TA) công bố một môđuyn chung $n = pq$ cho mọi người tham gia, sau khi đã chọn và giữ bí mật hai số nguyên tố lớn p và q, mỗi số này đều đồng dư với 3 theo mod4. Bài toán phân tích n thành thừa số được giả thiết là cực khó. Một số nguyên n như trên là số nguyên Blum, với -1 là một giả thặng dư bậc hai theo mod$n$ (tức là một bất thặng dư bậc hai có ký hiệu Jacobi bằng +1).

Mỗi người tham gia thực hiện các việc chuẩn bị như sau:

- Chọn k số nguyên ngẫu nhiên $s_1, s_2, \ldots, s_k$ trong tập $\{1, \ldots, n-1\}$, và k bit ngẫu nhiên $b_1, b_2, \ldots, b_k$.
- Tính $v_i = (-1)^{b_i} (s_i^2)^{-1} \bmod n$ với mọi $1 \le i \le k$.

- Mỗi chủ thể A đăng ký với TA khoá công khai (v_1,..., v_k ; n) của mình, và giữ cho riêng mình khoá bí mật (s_1 ,...,s_k).

Hoạt động của giao thức xưng danh sẽ gồm việc thực hiện t vòng hỏi-đáp sau đây; B sẽ chấp nhận danh tính của A nếu tất cả t vòng đó đều thành công. Giả thiết B có khoá công khai của A. Mỗi vòng gồm các bước :

(a) A chọn số nguyên ngẫu nhiên r ($1 \le r \le n-1$), và một bit ngẫu nhiên b, tính $x = (-1)^b.r^2 \bmod n$; và gửi x cho B như một *bằng chứng*.

(b) B gửi cho A một vectơ gồm k bit ngẫu nhiên (e_1,..., e_k) như một *câu hỏi* hay *lời thách đố*.

(c) A tính và gửi cho B $y = r.\prod_{j=1}^{k} s_j^{e_j} \bmod n$, như câu *trả lời*.

(d) B tính $z = y^2.\prod_{j=1}^{k} v_j^{e_j} \bmod n$, và *thử điều kiện* $z = \pm x$ và $z \neq 0$.

Chú ý rằng trong giao thức trên đây,các số k và t là các tham số an toàn như sẽ được giải thích trong một đoạn sau.

Thí dụ : Giả sử trung tâm TA chọn p =683 và q =811, và công bố $n = pq$ = 553913. Chọn các tham số k =3, t =1.

Giả sử A chọn s_1 =157, s_2 =43215, s_3 =4646, và 3 bit b_1=1, b_2=0, b_3=1. Tính ra v_1=441845, v_2=338402, v_3=124423.

Khoá công khai của A là (441845, 338402, 124423; 553913), khoá bí mật là (157, 43215, 4646).

Giao thức xưng danh của A có thể được thực hiện như sau:

a) A chọn r =1279, b =1, tính được x =25898, và gửi cho B,
b) B ra lời thách đố (e_1, e_2, e_3)=(0,0,1).
c) A trả lời lại bằng $y = rs_3 \bmod n$ = 403104.
d) B tính $z = y^2 v_3 \bmod n$ = 25898 và thử đúng $z = +x$ và $z \neq 0$. Do đó B chấp nhận danh tính của A.

Đối với giao thức Feige-Fiat-Shamir, người ta chứng minh được rằng khả năng thành công của việc mạo xưng danh tính có xác suất nhiều lắm là 2^{-kt} , do đó nếu chọn k và t sao cho kt =20 chẳng hạn thì xác suất đó là khoảng 1 phần triệu, và nếu kt =40 thì xác suất đó là khoảng 1 phần triệu triệu, có thể coi là không thể xẩy ra. Tính an toàn của giao thức dựa trên độ khó của bài toán khai căn bậc hai theo môđuyn là một hợp số lớn khó phân tích thành thừa số. Giao thức cũng có tính chất là một chứng minh không lộ tri thức theo nghĩa là nhờ biết khoá bí mật mà A thực hiện việc trả lời trong các vòng hỏi-đáp một cách trôi chảy, nhưng toàn bộ các trả lời của A không để lộ bất kỳ một chút bí mật nào để người khác (kể cả B) có thể khai thác nhằm phát hiện (khoá) bí mật của A.

6.6. Phép chứng minh không lộ tri thức.

(zero-knowledge proof)

Như đã giới thiệu trong phần mở đầu 6.1, bài toán xưng danh và xác nhận danh tính đóng một vai trò có ý nghĩa to lớn trong mọi hoạt động giao dịch của xã hội. Để việc xưng danh được an toàn, một yêu cầu quan trọng là cần chống được việc mạo xưng danh tính của người khác trong giao dịch. Khi việc giao dịch được điện tử hoá một cách rộng rãi, yêu cầu an toàn đặt ra nhiều vấn đề cần được giải quyết bằng những giải pháp khoa học. Những giải pháp đơn giản và thô sơ như trình tên tuổi, mật hiệu (password),... không còn an toàn, vì khó giữ được bí mật làm cho người khác có thể dễ dàng bắt chước để mạo xưng. Trong các phần trên của chương này, ta đã trình bày một số sơ đồ xưng danh dựa vào các giao thức hỏi-đáp, người kiểm thử đưa ra các câu hỏi, và người xưng danh trả lời, dựa trên các trả lời đó người kiểm thử hoặc đưa thêm những câu hỏi mới, hoặc chấp nhận (hay bác bỏ) danh tính của người xưng danh. Phần lớn các giao thức hỏi-đáp trong các sơ đồ xưng danh đó đều có ít nhiều tính chất của một chứng minh không lộ tri thức, dù tri thức mà ta đề cập đến chỉ là việc biết hay không biết một bí mật (của khoá xưng danh). Khái niệm *chứng minh không lộ tri thức* ban đầu xuất phát từ việc nghiên cứu các sơ đồ xưng danh, về sau đã được mở rộng cho nhiều loại bài toán khác.

Các bài toán mà ta sẽ tìm kiếm cho chúng những "chứng minh không lộ tri thức" thường là những bài toán quyết định, đó là những bài toán được xác định bởi một tập dữ liệu Σ và một tính chất Π, và nội dung của bài toán là xét xem với mỗi $x \in \Sigma$, x có tính chất Π hay không. Một số lớp các bài toán quyết định như vậy đã được xét đến khi ta nghiên cứu về độ phức tạp tính toán trong chương II. Tham gia vào một giao thức chứng minh gồm có hai người: một là người chứng minh (ký hiệu là P-prover) và một là người kiểm thử (ký hiệu V- verifier). Giao thức gồm các câu hỏi-đáp giữa V và P, thường là V đưa ra các câu hỏi hay thách đố, và V đưa ra các câu trả lời. Giả thử P biết chắc chắn rằng x có tính chất Π, P có thể dùng một giao thức chứng minh để thuyết phục V tin rằng x có tính chất Π, và một giao thức chứng minh được gọi là không lộ tri thức, nếu ngoài việc thuyết phục được V tin là x có tính chất Π ra, P không để lộ bất cứ một thông tin nào có thể giúp người khác (kể cả V) dùng để chứng minh x có tính chất Π. Trước khi đưa ra được các định nghĩa toán học về các khái niệm đó, ta hãy xét một thí dụ về một bài toán quen thuộc là *bài toán đẳng cấu graph*, với tập dữ liệu Σ là tập các cặp graph (G_1, G_2), và nội dung bài toán là câu hỏi: hai graph G_1 và G_2 có đẳng cấu với nhau không. Trong lý

thuyết về độ phức tạp tính toán, bài toán này có một vai trò đặc biệt, vì là một bài toán chưa biết có thuật toán nào với thời gian đa thức giải nó hay không, nhưng cũng chưa có chứng minh nào chứng tỏ nó là $\mathcal{NP}$-đầy đủ .

Dưới đây là sơ đồ tương tác chứng minh không lô tri thức của bài toán đẳng cấu graph:

Giả sử cho hai graph G_1 và G_2 có tập đỉnh {1, 2,...,*n*}. Giả sử P biết G_1 và G_2 đẳng cấu với nhau (chẳng hạn do biết một hoán vị σ trên tập {1, 2,...,*n*} sao cho G_1 là ảnh của G_2 qua hoán vị đó).
Sơ đồ tương tác chứng minh "G_1 và G_2 đẳng cấu" gồm m vòng hỏi-đáp, mỗi vòng có 4 bước sau đây:

1. P chọn một hoán vị ngẫu nhiên π của {1, 2,...,*n*}, lập graph H là ảnh của G_1 qua hoán vị π, và gửi H cho V.

2. V chọn số ngẫu nhiên $i \in \{1, 2\}$ và gửi nó cho P.

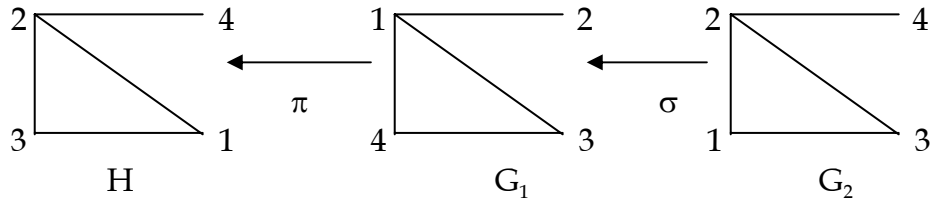
3. P tính một hoán vị ρ trên {1, 2,...,*n*} sao cho H là ảnh của G_i qua ρ (cụ thể, nếu i=1 thì lấy $\rho = \pi$, nếu i=2 thì lấy $\rho = \pi . \sigma$), rồi gửi ρ cho V.

4. V thử xem H có là ảnh của G_i qua ρ hay không.

V sẽ chấp nhận chứng minh của P nếu V thử đúng điều kiện 4 ở tất cả m vòng hỏi-đáp đó.

Thí dụ: Ta minh hoạ hoạt động của giao thức tương tác để chứng minh sự đẳng cấu của hai graph bằng thí dụ dưới đây:

Giả sử $G_1 = (V, E_1)$ và $G_2 = (V,E_2)$ là hai graph với tập đỉnh V={1, 2, 3, 4} và các tập cạnh E_1 ={12,13,14,34}, E_2={12,13,23,24}. Giả sử P biết G_2 đẳng cấu với G_1 qua hoán vị σ = {4 1 3 2}.



Một vòng của giao thức có thể xẩy ra như sau:

1. P chọn ngẫu nhiên hoán vị π = {2 4 1 3}. Graph H sẽ có tập cạnh {12,13,23,24}, là ảnh của G_1 qua π. P gửi H cho V.

2. V chọn i=2 và gửi cho P như một câu hỏi.

3. P thử thấy hoán vị $\rho = \pi . \sigma$ ={3 2 1 4} ánh xạ G_2 thành H và do đó gửi ρ cho V.

4. V thử đúng H là ảnh của G_2 qua hoán vị ρ. Ta kết luận vòng hỏi-đáp này đã thành công.

Toàn bộ giao thức gồm có $m = \log_2 n$ vòng.

Như vậy, nếu G_1 đẳng cấu với G_2 (hay chính xác hơn, nếu A biết G_1 đẳng cấu với G_2) và mọi qui định được tôn trọng, thì giao thức thành công, và xác suất của việc V chấp nhận chứng minh đó là 1. Đó là tính *đầy đủ* của giao thức.

Mặt khác, nếu G_1 và G_2 không đẳng cấu với nhau, thì cách duy nhất để P lừa V chấp nhận theo giao thức là ở mỗi vòng hỏi-đáp, P đoán trước đúng được câu hỏi (số i) mà V sẽ đưa ra ở bước 2, và do đó ở bước 1, P chọn ngẫu nhiên một hoán vị π và gửi cho V graph H là ảnh của G_i qua π, rồi ở bước 3 để trả lời câu hỏi (là số i) của V, P sẽ đáp lại bằng phép hoán vị $\rho = \pi$. Rõ ràng là V chấp nhận câu trả lời đó là đúng, và vòng hỏi-đáp đó thành công. Như vậy, P đã lừa được V một vòng, và xác suất thành công đó bằng xác suất P đoán trước đúng câu hỏi mà V sẽ đưa ra, tức là không lớn hơn 1/2. Vậy nếu G_1 và G_2 không đẳng cấu với nhau thì khả năng V bị lừa mà tin rằng G_1 và G_2 đẳng cấu là có xác suất không qúa $2^{-m} = 2^{-\log n} = 1/n$, một giá trị không đáng kể có thể bỏ qua vì n rất lớn. Điều đó cũng nói rằng nếu P không biết G_1 và G_2 đẳng cấu với nhau thì P cũng không thể lợi dụng giao thức đó mà lừa V rằng P biết G_1 và G_2 đẳng cấu. Đó là tính *đúng đắn* của giao thức.

Bây giờ ta nói đến tính *không lộ tri thức* của giao thức nói trên. Ta thấy rằng thực hiện mỗi vòng hỏi-đáp của giao thức, tất cả những gì mà P đưa đến cho V là một bản sao H đẳng cấu với G_1 và G_2, và một hoán vị ρ thực hiện sự đẳng cấu từ G_1 tới H hoặc từ G_2 tới H (nhưng không phải cả hai !). Từ các thông tin đó không đủ để V thiết lập được ngay một phép đẳng cấu của G_1 và G_2 (ta chú ý hoán vị ρ mà P chuyển cho V là $\rho = \pi$ hoặc $\rho = \pi . \sigma$, từ đó không dễ gì tìm được σ). Một cách trực giác, điều đó chứng tỏ là giao thức không lộ tri thức. Để có một định nghĩa toán học cho khái niệm *không lộ tri thức*, ta xét kỹ hơn lập luận trên đây.

Ta hãy xem qua một chứng minh tương tác như trên P và Vđể lại những thông tin gì. Ngoài thông tin về hai graph G_1 và G_2, ở mỗi vòng hỏi-đáp, P và V đã trao đổi các thông tin về một graph H, một câu hỏi i , và một trả lời ρ. Như vậy, ta có thể định nghĩa một *bản ghi* T của một chứng minh tương tác là

$$T = ((G_1 , G_2); (H_1, i_1, \rho_1) ;...; (H_m, i_m , \rho_m)).$$

Thông tin về một chứng minh tương tác được chứa đựng đầy đủ trong một bản ghi T. Bây giờ ta chú ý rằng một bản ghi cũng có thể được tạo ra một cách giả mạo. Thực vậy, ta có thể chọn ngẫu nhiên một số $i \in \{1, 2\}$, một hoán vị ρ, sau đó tính H là ảnh đẳng cấu của

G_i qua ρ. Thực hiện m lần như vậy, ta được m bộ ba (H,i,ρ), và cùng với (G_1, G_2) ta sẽ tạo được một bản ghi *giả mạo*, vì đó không phải là một bản ghi trung thực theo việc thực hiện *thực* một chứng minh đúng theo giao thức tương tác, nhưng không có cách nào để phân biệt một giao thức hợp thức với một giao thức gồm các bản ghi giả mạo. Thuật toán tạo ra cac bản ghi giả mạo được gọi là một *mô phỏng*. Bây giờ ta đã có thể đưa ra một định nghĩa cho khái niệm *không lộ tri thức* như sau:

Giả sử có một hệ chứng minh tương tác đối với bài toán quyết định Π, và một mô phỏng S[1], và x là một dữ liệu của bài toán có trả lời "đúng" đối với câu hỏi Π. Ký hiệu $\mathcal{T}(x)$ là tập tất cả các bản ghi hợp thức có thể có, và $\mathcal{F}(x)$ là tập hợp tất cả các bản ghi giả mạo có thể sinh ra bởi S. Giả thiết rằng $\mathcal{T}(x) = \mathcal{F}(x)$. Với mỗi $T \in \mathcal{T}(x)$ ký hiệu $p_{\mathcal{T}}(T)$ là xác suất của việc T là bản ghi sinh ra từ một chứng minh tương tác, và $p_{\mathcal{F}}(T)$ là xác suất của việc T là một bản ghi giả mạo sinh ra bởi mô phỏng S. Nếu $p_{\mathcal{T}}(T) = p_{\mathcal{F}}(T)$ với mọi $T \in \mathcal{T}(x)$, tức là các phân bố xác suất trên $\mathcal{T}(x)$ và $\mathcal{F}(x)$ là trùng nhau, thì ta nói rằng hệ chứng minh tương tác của ta là *không lộ tri thức hoàn hảo* (perfect zero-knowledge) đối với V.

Đối với bài toán đẳng cấu hai graph và với sơ đồ chứng minh tương tác kể trên, người ta chứng minh được rằng hai phân bố xác suất trên $\mathcal{T}(x)$ và $\mathcal{F}(x)$ trùng nhau, do đó, với định nghĩa của khái niệm không lộ tri thức hoàn hảo, ta có thể kết luận : *Đối với bài toán đẳng cấu hai graph, có một sơ đồ tương tác chứng minh không lộ tri thức hoàn hảo.*

Bây giờ ta giới thiệu thêm dưới đây một sơ đồ tương tác chứng minh không lộ tri thức đối với bài toán thặng dư bậc hai, là một bài toán $\mathcal{NP}$-đầy đủ.

Cho một số nguyên n là tích của hai số nguyên tố lớn p và q được giữ bí mật. Giả thiết P biết x là một thặng dư bậc hai theo mod$n$, và u là một căn bậc hai của nó (tức $u^2 \equiv x \pmod{n}$).Sơ đồ chứng minh tương tác gồm m vòng, mỗi vòng gồm 4 bước sau đây:

1. P chọn ngẫu nhiên một số $v \in Z_n^*$, tính $y = v^2 \bmod n$, và gửi y cho V.

2. V chọn ngẫu nhiên một số $i \in \{0, 1\}$ và gửi cho P.

[1] Thông thường người ta giả thiết là người kiểm thử V, cũng như bộ mô phỏng V, đều là các thuật toán có khả năng tính toán trong thời gian đa thức.

3. P tính $z = u^i v \bmod n$, và gửi z cho V.

4. V thử điều kiện $z^2 \equiv x^i y \pmod{n}$.

Nếu qua m vòng, V đều thử đúng điều kiện trên thì V chấp nhận chứng minh của P rằng x là thặng dư bậc hai theo mod$n$.

Giao thức chứng minh tương tác này cũng có các tính chất đầy đủ, đúng đắn, và là không lộ tri thức, nhưng chưa phải là không lộ tri thức hoàn hảo. Việc nghiên cứu các sơ đồ tương tác chứng minh không lộ tri thức là một chủ đề được nhiều người quan tâm trong vài thập niên vừa qua, và đã thu được nhiều kết quả lý thú, trong đó lý thú nhất có lẽ là các kết quả liên quan đến các bài toán $\mathcal{NP}$-đầy đủ. Người ta đã chứng tỏ rằng không có các chứng minh không lộ tri thức hoàn hảo đối với các bài toán $\mathcal{NP}$-đầy đủ; tuy nhiên, nếu không đòi hỏi chặt chẽ điều kiện "không lộ tri thức hoàn hảo", mà chỉ đòi hỏi một điều kiện nhẹ hơn chút ít về "không lộ tri thức tính toán" (computational zero-knowledge), thì người ta chứng minh được rằng đối với nhiều bài toán $\mathcal{NP}$-đầy đủ như bài toán thặng dư bậc hai theo mod$n$ ở trên hay bài toán tô ba mầu một graph là có thể xây dựng tương ứng các sơ đồ tương tác chứng minh không lộ tri thức tính toán. Rồi từ đó, do mọi bài toán trong lớp $\mathcal{NP}$ đều có thể qui dẫn trong thời gian đa thức về một bài toán $\mathcal{NP}$-đầy đủ, chẳng hạn bài toán tô ba màu một graph, nên có thể chứng minh được là đối với *mọi bài toán trong lớp $\mathcal{NP}$ đều có một sơ đồ tương tác chứng minh không lộ tri thức (tính toán).*

Khái niệm không lộ tri thức tính toán chỉ khác khái niệm không lộ tri thức hoàn hảo ở một điểm là nếu trong định nghĩa của "không lộ tri thức hoàn hảo" ta đòi hỏi hai phân bố xác suất trên $\mathcal{T}(x)$ và $\mathcal{F}(x)$ trùng nhau, thì đối với khái niệm "không lộ tri thức tính toán", ta chỉ đòi hỏi hai phân bố xác suất đó là "không phân biệt đượC" theo một nghĩa tương tự như "không ε-phân biệt đượC" mà ta đã xét đến trong mục 4.6.1, chương IV.

CHƯƠNG VII

Vấn đề phân phối khoá và thoả thuận khoá

7.1. Quản trị khoá trong các mạng truyền tin.

Trong các chương trước, ta đã làm quen với các phương pháp lập mật mã và các bài toán quan trọng khác liên quan đến việc truyền tin bảo mật trên các mạng truyền tin công cộng nói chung. Ta cũng đã thấy rằng các hệ mật mã khoá công khai có nhiều ưu việt hơn các hệ mật mã khoá đối xứng trong việc làm nền tảng cho các giải pháp an toàn thông tin, và đặc biệt nếu đối với các hệ mật mã khoá đối xứng việc thực hiện đòi hỏi những kênh bí mật để chuyển khoá hoặc trao đổi khoá giữa các đối tác, thì về nguyên tắc, đối với các hệ mật mã khoá công khai, không cần có những kênh bí mật như vậy, vì các khoá công khai có thể được truyền hoặc trao đổi cho nhau một cách công khai qua các kênh truyền tin công cộng. Tuy nhiên, trên thực tế, để bảo đảm cho các hoạt động thông tin được thật sự an toàn, không phải bất cứ thông tin nào về các khoá công khai của một hệ mật mã, của một thuật toán kiểm thử chữ ký, của một giao thức xác nhận thông báo hay xác nhận danh tính, v.v... cũng phát công khai một cách tràn lan trên mạng công cộng, mà dẫu là công khai nhưng người ta cũng mong muốn là những ai cần biết thì mới nên biết mà thôi. Do đó, dẫu là dùng các hệ có khoá công khai, người ta cũng muốn có những giao thức thực hiện việc trao đổi khoá giữa những đối tác thực sự có nhu cầu giao lưu thông tin với nhau, kể cả trao đổi khoá công khai. Việc trao đổi khoá giữa các chủ thể trong một cộng đồng nào đó có thể được thiết lập một cách tự do giữa bất cứ hai người nào khi có nhu cầu trao đổi thông tin, hoặc có thể được thiết lập một cách tương đối lâu dài trong một thời hạn nào đó trong cả cộng đồng với sự điều phối của một cơ quan được uỷ thác (mà ta ký hiệu là TA-trusted authority). Việc trao đổi khoá trong trường hợp thứ nhất ta gọi đơn giản là *thoả thuận khoá*, còn trong trường hợp thứ hai ta gọi là *phân phối khoá* , TA là nơi thực hiện việc phân phối, cũng tức là nơi quản trị khoá. Việc thoả thuận khoá nói chung không cần có sự tham gia của một TA nào và chỉ có thể xẩy ra khi

các hệ bảo mật mà ta sử dụng là hệ có khoá công khai, còn việc phân phối khoá thì có thể xẩy ra đối với các trường hợp sử dụng các hệ khoá đối xứng cũng như các hệ có khoá công khai. Việc phân phối khoá với vai trò quản trị khoá của một TA là một việc bình thường, đã tồn tại từ rất lâu trước khi có các hệ mật mã khoá công khai. Ta sẽ bắt đầu với việc giới thiệu một vài hệ phân phối khoá như vậy, rồi tiếp sau sẽ giới thiệu một số hệ phân phối hoặc trao đổi khoá khi dùng các sơ đồ an toàn và bảo mật có khoá công khai.

7. 2. Một số hệ phân phối khoá.

7. 2.1. Sơ đồ phân phối khoá Blom.

Giả sử ta có một mạng gồm có n người dùng, và mỗi người dùng đó đều có nhu cầu trao đổi thông tin bí mật với mọi người trong mạng. Giả sử sơ đồ mật mã được sử dụng là một sơ đồ mật mã khoá đối xứng (chẳng hạn, DES). Toàn bộ mạng cần có $\frac{n(n-1)}{2}$ khoá khác nhau cho chừng ấy cặp người dùng khác nhau trong mạng. Một cơ quan được uỷ thác TA quản lý chừng ấy khoá và phải chuyển cho mỗi người dùng n -1 khoá chung với n -1 người còn lại trong mạng, như vậy TA phải truyền bằng những kênh bí mật tất cả là n (n -1) lượt khoá đến cho tất cả n người dùng.

Blom (1985) đề nghị một sơ đồ phân phối khoá, mà sau đây ta gọi là sơ đồ Blom, trong trường hợp đơn giản nhất được mô tả như sau:

TA chọn một số nguyên tố $p \geq n$, và chọn cho mỗi người dùng A một số $r_A \in Z_p$. Số p và các số r_A được công bố công khai.

Sau đó, TA chọn ba số ngẫu nhiên $a,b,c \in Z_p$, và lập đa thức

$$f(x,y) = a + b(x+y) + cxy \bmod p.$$

Với mỗi người dùng A, TA tính $g_A(x) = f(x, r_A) = a_A + b_A x \bmod p$, trong đó $a_A = a + br_A \bmod p$, $b_A = b + cr_A \bmod p$. TA chuyển bí mật cặp số (a_A, b_A) cho A; như vậy, A biết $g_A(x) = a_A + b_A x$. So với việc TA phải truyền bí mật n (n -1) lượt khoá kể trên thì với sơ đồ Blom, TA chỉ phải truyền n lượt các cặp số (a_A, b_A) mà thôi.

Sau khi đã thực hiện xong các công việc chuẩn bị đó, bây giờ nếu hai người dùng A và B muốn tạo khoá chung để truyền tin bằng mật mã cho nhau, thì khoá chung $K_{A,B}$ đó sẽ là :

$$K_{A,B} = g_A(r_B) = g_B(r_A) = f(r_A, r_B),$$

mà mỗi người A và B tính được bằng những thông tin mình đã có.

Như vậy, theo sơ đồ phân phối này, TA phân phối cho mỗi người dùng một phần bí mật của khoá, hai người dùng bất kỳ phối hợp phần bí mật của riêng mình với phần công khai của người kia để cùng tạo nên khoá bí mật chung cho hai người. Sơ đồ này là *an toàn* theo nghĩa sau đây: *Bất kỳ một người thứ ba C nào (kể cả C là một người tham gia trong mạng) có thể phát hiện được khoá bí mật riêng của hai người A và B.* Thực vậy, dù C có là người tham gia trong mạng đi nữa, thì cái mà C biết nhiều lắm là hai số a_C, b_C do TA cấp cho. Ta chứng minh rằng với những gì mà C biết thì bất kỳ giá trị $l \in Z_p$ nào cũng có thể được chấp nhận là $K_{A,B}$. Những gì mà C biết, kể cả việc chấp nhận $l = K_{A,B}$, được thể hiện thành

$$\begin{aligned} a + b(r_A + r_B) + cr_A r_B &= l \\ a + br_C &= a_C \\ b + cr_C &= b_C \end{aligned}$$

Hệ thống phương trình đó, nếu xem a,b,c là ẩn số, có định thức các hệ số ở vế phải là

$$\begin{vmatrix} 1 & r_A + r_B & r_A r_B \\ 1 & r_C & 0 \\ 0 & 1 & r_C \end{vmatrix} = (r_C - r_A)(r_C - r_B),$$

theo giả thiết chọn các số r, định thức đó khác 0, do đó hệ phương trình luôn có nghiệm (a,b,c), tức việc chấp nhận l là giá trị của $K_{A,B}$ là hoàn toàn có thể. Bất kỳ giá trị $l \in Z_p$ nào cũng có thể được C chấp nhận là $K_{A,B}$, điều đó đồng nghĩa với việc C không biết $K_{A,B}$ là số nào!

Tuy nhiên, nếu có hai người tham gia C và D, khác A,B, liên minh với nhau để phát hiện $K_{A,B}$, thì lại rất dễ dàng, vì cả C và D biết

$$\begin{aligned} a + br_C &= a_C \\ b + cr_C &= b_C \\ a + br_D &= a_D \\ b + cr_D &= b_D \end{aligned}$$

Bốn phương trình đó đủ để xác định (a,b,c), từ đó tìm được $K_{A,B}$.

Ta có thể mở rộng sơ đồ Blom nói trên để được một sơ đồ Blom tổng quát, trong đó mọi khoá chung $K_{A,B}$ của hai người dùng A và B là bí mật hoàn toàn đối với bất kỳ liên minh nào gồm k người ngoài A và B, nhưng không còn là bí mật đối với mọi liên minh gồm k +1 người tham gia trong mạng. Muốn vậy, ta chỉ cần

thay đa thức $f(x,y)$ nói trên bằng một đa thức đối xứng bậc $2k$ sau đây :

$$f(x,y) = \sum_{i=0}^{k}\sum_{j=0}^{k} a_{ij}x^i y^j \bmod p,$$

trong đó $a_{ij} \in Z_p, 0 \le i, j \le k, a_{ij} = a_{ji}$ với mọi i, j.

7.2.2. Hệ phân phối khoá Kerberos.

Kerberos là tên của một hệ dịch vụ phân phối (hay cấp phát) khoá phiên (session key) cho từng phiên truyền tin bảo mật theo yêu cầu của người dùng trong một mạng truyền tin. Hệ mật mã được sử dụng thường là hệ có khoá đối xứng, chẳng hạn DES.

Để thực hiện hệ này, trước hết, cơ quan được uỷ thác (hay trung tâm điều phối) TA cần chia sẻ một khoá DES bí mật K_A với mỗi thành viên A trong mạng. Sau đó, mỗi lần A có nhu cầu truyền tin bảo mật với một thành viên khác B thì yêu cầu TA cấp một khoá phiên cho cả A và B. Việc cấp phát đó sẽ được thực hiện bằng một giao thức phân phối khoá như sau:

1. TA chọn ngẫu nhiên một khoá phiên K, xác định một tem thời gian T và một thời gian sống L (như thế có nghĩa là khoá phiên K có giá trị sử dụng trong khoảng thời gian từ T đến $T+L$).

2. TA tính

$$m_1 = e_{K_A}(K, ID(B), T, L),$$
$$m_2 = e_{K_B}(K, ID(A), T, L).$$

và gửi (m_1, m_2) đến A.

3. A dùng hàm giải mã d_{K_A} cho m_1 để thu được K, T, L, ID(B). Sau đó tính

$$m_3 = e_K(ID(A), T),$$

và gửi (m_3, m_2) cho B.

4. B dùng các hàm giải mã d_{K_B} cho m_2 và d_K cho m_3 để thu được K, T, L, ID(A) và ID(A), T. Nếu thử thấy hai giá trị của ID(A) và của T trùng nhau, thì B tính tiếp

$$m_4 = e_K(T+1)$$

và gửi m_4 cho A.

5. A dùng hàm giải mã d_K cho m_4, và thử xem kết quả thu được có đúng là $T+1$ hay không.

Trong giao thức kể trên, các ký hiệu ID(A) và ID(B) là chỉ cho danh tính của A và của B, các thông tin đó là công khai.

Hoàn thành giao thức gồm 5 bước nói trên, TA (cùng với A và B) đã thực hiện xong việc cấp phát một khoá phiên K cho hai người dùng A và B để truyền tin mật mã cho nhau. Tất cả các việc trao đổi thông tin của giao thức đó đều được thực hiện trên các kênh công cộng, dù khoá K vẫn là bí mật, chỉ A, B (và TA) là được biết mà thôi. Ngoài việc cấp phát khoá, giao thức đó còn thực hiện được việc xác nhận khoá: B và A đều tin chắc được rằng đối tác của mình đã thực sự có khoá K do kết quả của việc thực hiện các phép thử ở bước 4 và 5; thêm nữa, cả A và B còn biết được thời hạn có hiệu lực của khoá.

Phân phối khoá bí mật theo giao thức Kerberos là có độ tin cậy cao, tuy nhiên trong thực tế, việc sử dụng nó cũng đòi hỏi tốn nhiều thời gian, nên ngày nay cũng chỉ được dùng trong những trường hợp hạn chế.

7. 2.3. Hệ phân phối khoá Diffie-Hellman.

Hệ phân phối khoá Diffie-Hellman không đòi hỏi TA phải biết và chuyển bất kỳ thông tin bí mật nào về khoá của các người tham gia trong mạng để họ thiết lập được khoá chung bí mật cho việc truyền tin với nhau.

Trong một hệ phân phối khoá Diffie-Hellman, TA chỉ việc chọn một số nguyên tố lớn p và một phần tử nguyên thuỷ α theo mod$p$, sao cho bài toán tính $\log_\alpha$ trong Z_p^* là rất khó. Các số p và α được công bố công khai cho mọi người tham gia trong mạng. Ngoài ra, TA có một sơ đồ chữ ký với thuật toán ký (bí mật) sig_{TA} và thuật toán kiểm thử (công khai) ver_{TA}.

Một thành viên bất kỳ A với danh tính ID(A) tuỳ ý chọn một số $a_A (0 \le a_A \le p-2)$ và tính $b_A = \alpha^{a_A} \bmod p$. A giữ bí mật a_A và đăng ký các thông tin $(ID(\mathrm{A}), b_A)$ với TA. TA cấp cho A chứng chỉ

$$\mathrm{C(A)} = (ID(\mathrm{A}), b_A, sig_{\mathrm{TA}}(ID(\mathrm{A}), b_A)).$$

Các chứng chỉ của các thành viên trong mạng có thể được lưu giữ trong một cơ sở dữ liệu công khai, hoặc uỷ thác cho TA lưu giữ và cung cấp công khai cho các thành viên mỗi khi cần đến.

Khi hai thành viên A và B trong mạng cần có một khoá bí mật chung để truyền tin bảo mật cho nhau, thì A dùng thông tin công khai b_B có trong C(B) kết hợp với số bí mật của mình là a_A để tạo nên khoá

$$K_{A,B} = b_B^{a_A} \bmod p = \alpha^{a_B a_A} \bmod p.$$

Khoá chung đó B cũng tạo ra được từ các thông tin công khai b_A của A và số bí mật a_B của mình:

$$K_{A,B} = b_A^{a_B} \bmod p = \alpha^{a_A a_B} \bmod p.$$

Để bảo đảm được các thông tin về b_B và b_A là chính xác, A và B có thể dùng thuật toán ver_{TA} để kiểm thử chữ ký xác nhận của TA trong các chứng chỉ C(B) và C(A) tương ứng.

Độ an toàn của hệ phân phối khoá Diffie-Hellman được bảo đảm bởi điều sau đây: Biết b_A và b_B để tính $K_{\mathrm{A,B}}$ chính là bài toán Diffie-Hellman mà ta đã đề cập tới trong mục 4.1, chương IV: biết $\alpha^a \bmod p$ và $\alpha^b \bmod p$, tính $\alpha^{ab} \bmod p$. Đây là một bài toán khó tương đương bài toán tính lôgarit rời rạc hay bài toán phá mật mã ElGamal.

7.3. Trao đổi khoá và thoả thuận khoá.

7.3.1. Giao thức trao đổi khoá Diffie-Hellman.

Hệ phân phối khoá Diffie-Hellman nói trong mục trước có thể dễ dàng biến đổi thành một giao thức trao đổi (hay thoả thuận) khoá trực tiếp giữa các người sử dụng mà không cần có sự can thiệp của một TA làm nhiệm vụ điều hành hoặc phân phối khoá. Một nhóm bất kỳ người sử dụng có thể thoả thuận cùng dùng chung một số nguyên tố lớn p và một phần tử nguyên thuỷ α theo mod$p$, hai người bất kỳ trong nhóm A và B mỗi khi muốn truyền tin bảo mật cho nhau có thể cùng thực hiện giao thức say đây để trao đổi khoá:

1. A chọn ngẫu nhiên số a_{A} ($0 \le a_{\mathrm{A}} \le p-2$), giữ bí mật a_{A}, tính $b_A = \alpha^{a_A} \bmod p$ và gửi b_{A} cho B.
2. Tương tự, B chọn ngẫu nhiên số a_{B} ($0 \le a_{\mathrm{B}} \le p-2$), giữ bí mật a_{B}, tính $b_B = \alpha^{a_B} \bmod p$ và gửi b_{B} cho B.
3. A và B cùng tính được khoá chung

$$K_{A,B} = b_B^{a_A} \bmod p = b_A^{a_B} \bmod p \,(= \alpha^{a_A a_B} \bmod p).$$

Giao thức trao đổi khoá Diffie-Hellman có các tính chất sau:

1. *Giao thức là an toàn đối với việc tấn công thụ động*, nghĩa là một người thứ ba, dù biết b_{A} và b_{B} sẽ khó mà biết được $K_{\mathrm{A,B}}$.

Ta biết rằng bài toán "biết b_{A} và b_{B} tìm $K_{\mathrm{A,B}}$" chính là bài toán Diffie-Hellman, và trong mục 7.2.3 ta có nói rằng bài toán đó tương

đương với bài toán phá mật mã ElGamal. Bây giờ ta chứng minh điều này. Phép mật mã ElGamal với khoá $K = (p,\alpha,a,\beta)$, trong đó $\beta = \alpha^a \bmod p$, cho ta từ một bản rõ x và một số ngẫu nhiên $k \in Z_{p-1}$ lập được mật mã $e_K(x,k) = (y_1, y_2)$,

trong đó $y_1 = \alpha^k \bmod p,\ y_2 = x\beta^k \bmod p.$

Và phép giải mã được cho bởi

$$d_K(y_1, y_2) = y_2(y_1^a)^{-1} \bmod p.$$

Giả sử ta có thuật toán $\mathcal{A}$ giải bài toán Diffie-Hellman. Ta sẽ dùng $\mathcal{A}$ để phá mã ElGamal như sau: Cho mật mã (y_1, y_2). Trước hết, dùng $\mathcal{A}$ cho $y_1 = \alpha^k \bmod p$ và $\beta = \alpha^a \bmod p$, ta được

$$\mathcal{A}(y_1,\beta) = \alpha^{ka} = \beta^k \bmod p,$$

và sau đó ta thu được bản rõ x từ β^k và y_2 như sau :

$$x = y_2(\beta^k)^{-1} \bmod p.$$

Ngược lại, giả sử có thuật toán $\mathcal{B}$ phá mã ElGamal, tức

$$\mathcal{B}(p,\alpha,\beta,y_1,y_2) = x = y_2(y_1^a)^{-1} \bmod p\,.$$

Áp dụng $\mathcal{B}$ cho $\beta = b_A, y_1 = b_B, y_2 = 1$, ta được

$$\mathcal{B}(p,\alpha,b_A,b_B,1)^{-1} = (1.(b_B^{a_A})^{-1})^{-1} = \alpha^{a_A a_B} \bmod p,$$

tức là giải được bài toán Diffie-Hellman.

2. *Giao thức là không an toàn đối với việc tấn công chủ động bằng cách đánh tráo giữa đường*, nghĩa là một người thứ ba C có thể đánh tráo các thông tin trao đổi giữa A và B, chẳng hạn, C thay α^{a_A} mà A định gửi cho B bởi $\alpha^{a'_A}$,và thay α^{a_B} mà B định gửi cho A bởi $\alpha^{a'_B}$, như vậy, sau khi thực hiện giao thức trao đổi khoá, A đã lập một khoá chung $\alpha^{a_A a'_B}$ với C mà vẫn tưởng là với B, đồng thời B đã lập một khoá chung $\alpha^{a'_A a_B}$ với C mà vẫn tưởng là với A; C có thể giải mã mọi thông báo mà A tưởng nhầm là mình gửi đến B, cũng như mọi thông báo mà B tưởng nhầm là mình gửi đến A !

Một cách khắc phục kiểu tấn công chủ động nói trên là làm sao để A và B có thể kiểm thử để xác nhận tính đúng đắn của các khoá công khai b_A và b_B .Đưa vào giao thức trao đổi khoá Diffie-Hellman thêm vai trò điều phối của một TA để được một hệ phân phối khoá Diffie-Hellman như ở mục 7.2.3 là một cách khắc phục như vậy. Trong hệ phân phối khoá Diffie-Hellman, sự can thiệp của TA là rất yếu, thực ra TA chỉ làm mỗi một việc là cấp chứng chỉ xác nhận khoá công khai cho từng người dùng chứ không đòi hỏi biết thêm bất cứ một bí mật nào của người dùng. Tuy nhiên, nếu chưa

thoả mãn với vai trò hạn chế đó của TA, thì có thể cho TA một vai trò xác nhận yếu hơn, không liên quan gì đến khoá, chẳng hạn như xác nhận thuật toán kiểm thử chữ kỹ của người dùng, còn bản thân các thông tin về khoá (cả bí mật và công khai) thì do các người dùng trao đổi trực tiếp với nhau. Với cách khắc phục có vai trò rất hạn chế đó của TA, ta được giao thức sau đây:

7.3.2. Giao thức trao đổi khoá DH có chứng chỉ xác nhận.

Mỗi người dùng A có một danh tính ID(A) và một sơ đồ chữ ký với thuật toán ký sig_{A} và thuật toán kiểm thử ver_{A}. TA cũng có một vai trò xác nhận, nhưng không phải xác nhận bất kỳ thông tin nào liên quan đến việc tạo khoá mật mã của người dùng (dù là khoá bí mật hay là khoá công khai), mà chỉ là xác nhận một thông tin ít quan hệ khác như thuật toán kiểm thử chữ ký của người dùng. Còn bản thân các thông tin liên quan đến việc tạo khoá mật mã thì các người dùng sẽ trao đổi trực tiếp với nhau. TA cũng có một sơ đồ chữ ký của mình, gồm một thuật toán ký sig_{TA} và một thuật toán kiểm thử (công khai) ver_{TA}. Chứng chỉ mà TA cấp cho mỗi người dùng A sẽ là

$$\mathrm{C(A)} = (ID(\mathrm{A}),\ ver_{\mathrm{A}}\ ,\ sig_{\mathrm{TA}}(ID(\mathrm{A}),\ ver_{\mathrm{A}})).$$

Rõ ràng trong chứng chỉ đó TA không xác nhận bất kỳ điều gì liên quan đến việc tạo khoá của A cả. Việc trao đổi khoá giữa hai người dùng A và B được thực hiện theo giao thức sau đây:

1.A chọn ngẫu nhiên số $a_A(0 \le a_A \le p-2)$, tính $b_A = \alpha^{a_A} \bmod p$, và gửi b_{A} cho B.

2. B chọn ngẫu nhiên số $a_B(0 \le a_B \le p-2)$, tính $b_B = \alpha^{a_B} \bmod p$, tính tiếp

$$K = b_A^{a_B} \bmod p,$$

$$y_B = sig_B(b_B, b_A),$$

và gửi (C(B),b_{B}, y_{B}) cho A.

3. A tính

$$K = b_B^{a_A} \bmod p,$$

dùng ver_{B} để kiểm thử y_{B} , dùng ver_{TA} để kiểm thử C(B), sau đó tính

$$y_{\mathrm{A}} = sig_{\mathrm{A}}(b_{\mathrm{A}}\ ,\ b_{\mathrm{B}}\ ),$$

và gửi (C(A), y_{A}) cho B.

4.B dùng ver_{A} để kiểm thử y_{A} ,và dùng ver_{TA} để kiểm thử C(A).

Nếu tất cả các bước đó được thực hiện và các phép kiểm thử đều cho kết quả đúng đắn, thì giao thức kết thúc, và cả A và B đều có được khoá chung K. Do việc dùng các thuật toán kiểm thử nên A biết chắc giá trị b_{B} là của B và B biết chắc giá trị b_{A} là của A, loại

trừ khả năng một người C nào khác đánh tráo các giá trị đó giữa đường.

7.3.3. Giao thức trao đổi khoá Matsumoto-Takashima-Imai.

Giao thức trình bày trong mục trên cần dùng ba lần chuyển tin qua lại để thiết lập một khoá chung. Các tác giả Nhật Matsumoto, Takashima và Imai đề nghị một cải tiến để chỉ dùng một giao thức gồm hai lần chuyển tin (một từ A đến B và một từ B đến A) để thoả thuận khoá như sau:

Ta giả thử rằng trước khi thực hiện giao thức, TA đã ký cấp chứng chỉ cho mỗi người dùng A theo cách làm ở mục 7.2.3:

$$C(A) = (ID(A), b_A, sig_{TA}(ID(A), b_A)),$$

và thuật toán kiểm thử chữ ký ver_{TA} của TA là công khai. Trong giao thức này, các b_A không trực tiếp tạo nên các khoá mật mã cho truyền tin, mà với mỗi phiên truyền tin bảo mật, khoá phiên (sesion key) sẽ được tạo ra cho từng phiên theo giao thức.

Giao thức trao đổi khoá phiên MTI gồm ba bước (trong đó có hai lần chuyển tin) như sau:

1. A chọn ngẫu nhiên số $r_A (0 \le r_A \le p-2)$, tính $s_A = \alpha^{r_A} \bmod p$, và gửi (C(A), s_A) cho B.

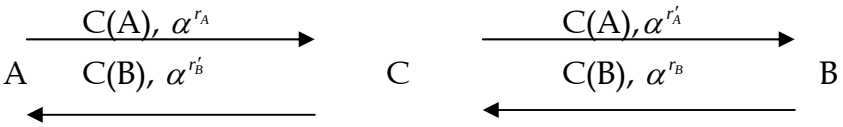
2. B chọn ngẫu nhiên số $r_B (0 \le r_B \le p-2)$, tính $s_B = \alpha^{r_B} \bmod p$, và gửi (C(B),$s_B$) cho A.

3. A tính $K = s_B^{a_A} . b_B^{r_A} \bmod p$, với giá trị b_B thu được từ C(B),

B tính $K = s_A^{a_B} . b_A^{r_B} \bmod p$, với giá trị b_A thu được từ C(A).

Hai cách tính đó đều cho cùng một giá trị $K = \alpha^{r_A a_B + r_B a_A} \bmod p$.

Giao thức này cũng có khả năng giữ bí mật khoá K như đối với giao thức Diffie-Hellman trước sự tấn công thụ động. Tuy nhiên, vì không có chứng chỉ đối với các giá trị s_A , s_B nên vẫn có nguy cơ của sự tấn công tích cực bằng việc đánh tráo giữa đường bởi một C nào đó theo kiểu sau đây:



Đáng lẽ A gửi đến B (C(A),s_A) thì C đánh tráo bằng cách nhận (C(A),s_A) và gửi đến B $(C(A), s'_A)$, với $s'_A = \alpha^{r'_A} \bmod p$, và ngược lại,

đáng lẽ B gửi đến A (C(B), s_B) thì C đánh tráo bằng cách nhận (C(B), s_B) và gửi đến A $(C(B), s'_B)$, với $s'_B = \alpha^{r'_B} \bmod p$. Khi đó, A tính được khoá

$$K_1 = \alpha^{r_A a_B + r'_B a_A} \bmod p,$$

và B tính được khoá

$$K_2 = \alpha^{r'_A a_B + r_B a_A} \bmod p.$$

Hai giá trị K_1 và K_2 này khác nhau, nên không giúp A và B truyền tin được cho nhau, nhưng C không có khả năng tính được giá trị nào trong hai giá trị đó (vì không biết a_A và a_B), nên khác với giao thức Diffie-Hellman ở mục 7.2.3, ở đây C chỉ có thể phá rối, chứ không thể đánh cắp thông tin được.

7.3.4. Giao thức Girault trao đổi khoá không chứng chỉ.

Giao thức Girault được đề xuất năm 1991. Trong giao thức này, người sử dụng A không cần dùng chứng chỉ C(A), mà thay bằng một *khoá công khai tự chứng thực* , được cấp trước bởi một TA. Phương pháp này sử dụng kết hợp các đặc tính của các bài toán RSA và lôgarit rời rạc.

Giả thử n là tíc của hai số nguyên tố lớn p và q, $n = p.q$, p và q có dạng $p = 2p_1+1$, $q = 2q_1+1$, trong đó p_1 và q_1 cũng là các số nguyên tố. Nhóm nhân Z_n^* đẳng cấu với tích $Z_p^* \times Z_q^*$. Cấp cao nhất của một phần tử trong Z_n^* là bội chung bé nhất của p -1 và q -1, tức là bằng $2p_1q_1$. Giả sử α là một phần tử cấp $2p_1q_1$ của Z_n^*. Nhóm cyclic sinh bởi α được ký hiệu là G, bài toán tính lôgarit rời rạc theo cơ số α trong G được giả thiết là rất khó.

Các số n và α là công khai. Chỉ TA biết p ,q . TA chọn số mũ công khai e , với gcd(e, $\phi(n)$)=1,và giữ bí mật $d = e^{-1} \bmod \phi(n)$.

Mỗi người dùng A có một danh tính ID(A), chọn ngẫu nhiên một số $a_A \in$ G, giữ bí mật a_A và tính $b_A = \alpha^{a_A} \bmod n$, rồi gửi a_A ,b_A cho TA. TA thử lại điều kiện $b_A = \alpha^{a_A} \bmod n$, rồi cấp cho A một *khoá công khai tự chứng thực* $p_A = (b_A - ID(A))^d \bmod n$. Trong khoá công khai p_A không có thông tin về a_A , nhưng TA cần biết a_A để thử điều kiện $b_A = \alpha^{a_A} \bmod n$.

Giao thức Girault trao đổi khoá giữa hai người dùng A và B được thực hiện bởi các bước sau đây:

1. A chọn ngẫu nhiên $r_A \in$G, tính $s_A = \alpha^{r_A} \bmod n$, và gửi cho B ($ID$(A),$p_A$, s_A).

2. B chọn ngẫu nhiên $r_B \in G$, tính $s_B = \alpha^{r_B} \bmod n$, và gửi cho A ($ID$(B), p_B, s_B).

3. A tính khoá $K = s_B^{a_A}(p_B^e + ID(V))^{r_A} \bmod n$,

B tính khoá $K = s_A^{a_B}(p_A^e + ID(A))^{r_B} \bmod n$.

Cả hai giá trị đó của K đều bằng nhau và bằng

$$K = \alpha^{r_A a_B + r_B a_A} \bmod n.$$

Bằng các lập luận như trong mục trước, ta dễ thấy rằng một người thứ ba C khó mà tạo ra các thông tin giả mạo để gửi đến A hoặc B, nếu tấn công bằng cách đánh tráo giữa đường thì có thể phá rối để ngăn cản A và B tạo lập khoá chung, nhưng không thể đánh cắp thông tin trao đổi giữa A và B.

Còn lại một vấn đề: Tại sao TA cần biết a_A và thử điều kiện $b_A = \alpha^{a_A} \bmod n$ trước khi cấp p_A cho A? Ta giả thử rằng TA không biết a_A và cấp $p_A = (b_A - ID(A))^d \bmod n$ cho A, và thử xem có thể xảy ra chuyện gì?

Một người thứ ba C có thể chọn một giá trị rởm a'_A, và tính $b'_A = \alpha^{a'_A} \bmod n$, rồi tính $b'_C = b'_A - ID(A) - ID(C)$, và đưa $(ID(C), b'_C)$ cho TA. TA sẽ cấp cho C một "khoá công khai tự chứng thực"

$$p'_C = (b'_C - ID(C))^d \bmod n.$$

Vì $b'_C - ID(C) = b'_A - ID(A)$, nên thực tế C đã được cấp

$$p'_C = p'_A = (b'_A - ID(A))^d \bmod n.$$

Bây giờ giả sử A và B thực hiện giao thức trao đổi khoá, và C xen vào ở giữa, như vậy, A gửi đến B $(ID(A), p_A, \alpha^{r_A} \bmod n)$, nhưng do bị C đánh tráo nên B lại nhận được $(ID(A), p'_A, \alpha^{r'_A} \bmod n)$, do đó B và C tính được cùng một khoá

$$K' = \alpha^{r'_A a_B + r_B a'_A} \bmod n = s_B^{a'_A}(p_B^e + ID(B))^{r'_A} \bmod n,$$

còn A tính được khoá

$$K = \alpha^{r_A a_B + r_B a_A} \bmod n.$$

B và C có cùng một khoá khác với khoá của A, nhưng B vẫn nghĩ rằng mình có chung khoá với A. Vì thế, C có thể giải mã mọi thông báo mà B gửi cho A, tức đánh cắp các thông tin từ B đến A. Việc TA biết a_A và thử điều kiện $b_A = \alpha^{a_A} \bmod n$ trước khi cấp p_A cho A là để loại trừ khả năng đánh tráo như vậy của một kẻ tấn công C.

CHÚ DẪN VỀ SÁCH THAM KHẢO

Sách báo về Khoa học mật mã tuy mới được công khai xuất bản từ khoảng ba thập niên gần đây, nhưng do nhu cầu nghiên cứu và ứng dụng rất lớn nên đã phát triển rất nhanh chóng, trong đó có cả những tài liệu giáo khoa do các trường Đại học xuất bản cũng như công trình nghiên cứu đăng tải trên các tạp chí khoa học và các tập công trình của các hội nghị khoa học quốc tế hàng năm về Mật mã. Đó là nguồn tài liệu hết sức phong phú và quí giá cho tất cả những ai quan tâm đến việc học tập và nghiên cứu về khoa học mật mã. Tập giáo trình này được biên soạn chủ yếu dựa vào một số sách chuyên khảo đã trở thành giáo khoa cho nhiều trường Đại học trên thế giới, được xuất bản trong những năm gần đây:

1. Douglas R. Stinson. *Cryptography. Theory and Practice*, CRC Press,1995.

2.A.J. Menezes, P.C. van Oorschot, S.A. Vanstone. *Handbook of Applied Cryptography,* CRC Press, 1997.

3.Bruce Schneier. *Applied Cryptography. Protocols, Algorithms and Source Code in C.* John Wiley &Son,Inc, 1996.

4. S. Goldwasser, M. Bellare. *Lecture Notes on Cryptography.* MIT Laboratory of Computer Science, 2001.

5.J.Seberry, J. Pieprzyk. *Cryptography. An introduction to Computer Security.* Prentice Hall, 1989.

6.Vitor Shoup. *A computational Introduction to Number Theory and Algebra*, New York University, 2003.