MỤC LỤC

LỜI CẢM ƠN

- Để hoàn thành đồ án này, trước hết, em xin gửi lời cảm ơn và biết ơn sâu sắc tới thầy giáo Trần Ngọc Thái, người đã tận tình hướng dẫn, chỉ bảo và giúp đỡ em trong suốt thời gian nghiên cứu và hoàn thành đồ án.

- Em xin chân thành cảm ơn tới các thầy cô trong khoa Công Nghệ Thông Tin cũng như các thầy cô trong trường Đại học dân lập Hải Phòng, những người đã tận tình giảng dậy, và tạo điều kiện cho em trong suốt quá trình học tập và nghiên cứu tại trường.

-Cuối cùng, em xin cảm ơn gia đình, bạn bè, người thân đã luôn ở bên động viên và là nguồn cổ vũ lớn lao, là động lực trong suốt quá trình học tập và nghiên cứu.

-Mặc dù em đã cố gắng hoàn thành đồ án trong phạm vi và khả năng có thể. Tuy nhiên sẽ không tránh khỏi những điều thiếu sót. Em rất mong nhận được sự cảm thông và tận tình chỉ bảo của quý thầy cô và toàn thể các bạn.

Một lần nữa em xin chân thành cảm ơn !

Hải Phòng, ngày ... tháng ... năm 2011

Sinh viên

MỞ ĐẦU

Mục đích:

- Hệ thống lại các kiến thức cơ bản về mật mã

-Tìm hiểu vềmã hóa đối xứng.

- Nghiên cứu về chữ ký điện tử và một số mô hình ứng dụng chữ ký điện tử.

- Xây dựng chương trình chữ ký điện tử bằng ngôn ngữ C#.

Ý nghĩa:

Luận văn gồm phần mở đầu, kết luận và 4 chương với các nội dung chính sau:

-Chương 1: Cơ sở toán học của mật mã

- Chương 2: Tổng quan về mật mã học

-Chương 3: Chữ ký điện tử

-Chương 4: Mô phỏng chữ ký điện tử

CHƯƠNG 1: CƠ SỞ TOÁN HỌC CỦA MẬT MÃ HỌC

1.1 Số nguyên tố và số nguyên tố cùng nhau

- Sốnguyên tốlàsố nguyên dương lớn hơn 1chỉchiahết cho1và chínhnó.

Ví dụ:2,3,5,7,11,…lànhữngsốnguyên tố.

- Hệmật mãthườngsửdụngcácsốnguyên tố ít nhấtlàlớnhơn10^{150}.

- Haisốmvàđượcgọilànguyêntốcùngnhaunếuướcsốchung lớnnhấtcủachúngbằng1.Kýhiệu:gcd(m, n)=1.

Vídụ:11và13 lànguyêntốcùngnhau.

Định lý số nguyên tố: Với mọi n>=2 đều có thể phân tích thành lũy thừa cơ số nguyên tố n = $p_1^{e1}p_2^{e2}p_3^{e3}$... , với p_i : số nguyên tố, $e_i \in Z^+$

Hệ quả: Giả sử a = $p_1^{e1}.p_2^{e2}p_3^{e3}\ldots p_k^{ek}$

$b = p_1^{f1}.p_2^{f2}.p_3^{f3}...p_k^{fk}$

thì gcd(a,b) = $p_1^{\min(e1,f1)}.p_2^{\min(e2,f2)}\ldots p_k^{\min(ek,fk)}$

lcm(a,b) = $p_1^{\max(e1,f1)}.p_2^{\max(e2,f2)}\ldots p_k^{\max(ek,fk)}$

Ví dụ: a = 4864=2^8.19 và b = 3458 =2.7.13.19

ta được : gcd(a,b)=2.19 và lcm(a,b)= 2^8.19.7.13

1.2 Khái niệm đồng dư

Cho n là một số nguyên dương. Nếu a và b là hai số nguyên ,khí đó a được gọi là đồng dư với b theo modulo n, được viết a≡b(mod n) nếu n | (a-b) và n được gọi là modulo của đồng dư.

Ví dụ 24≡9(mod 5),17≡5(mod 3)

Tính chất:

- Nếu a ≡ b(mod n),nếu và chỉ nếu a và b đều trả số dư như nhau khi đem chia chúng cho n.
- Nếu a ≡ a (mod n)(tính phản xạ)
- Nếu a ≡ b (mod n) thì b ≡ a (mod n)
- Nếu a ≡ b (mod n), b ≡ c (mod n) thì a ≡ c (mod n)

- Nếu $a \equiv a_1 \pmod n$ và $b \equiv b_1 \pmod n$ thì $a+b=(a_1+b_1) \pmod n$ và $a.b \equiv a_1 b_1 \pmod n$

1.3 Định nghĩa hàm phi Euler

Với n≥1 chúng ta gọi φ(n) là tập các số nguyên tố cùng nhau với n nằm trong khoảng [1,n].

Tính chất:

- Nếu p là số nguyên tố thì **φ**(p) = p – 1
- Nếu gcd(n.m)=1 thì **φ(**m.n**)=φ**(m).(n)
- Nếu $n=p_1^{e1}.p_2^{e2}\ldots p_k^{ek}$,dạng khai triển chính tắc của n thì
- $(n)=n(1-1/p_1)(1-1/p_2)\ldots(1-1/p_k)$

***Ví dụ :*φ(11)=11-1=10**

1.4 Thuật toán Euclide

Thuật toán: Tìm UCLN của hai số .

INPUT: Hai số nguyên không âm a và b, sao cho a≥b

OUTPUT: UCLN của a, b.

1.Trong khi $b \neq 0$ thực hiện

Đặt r← a mod b, a←b, b←r

2.Kết_qủa(a)

Ví dụ :Tính gcd(4864,3458)=38

4864=1.3458+1406

3458=2.1406+646

1406=2.646+114

646=5.114+76

114=1.76+38

76=2.38+0.

Thuật toán Euclidean có thể được mở rộng để không chỉ tính được ước số chung d của hai số nguyên a và b,mà còn có thể tính được hai số nguyên x,y thỏa mãn ax+by=d.

1.5 Thuật toán Euclidean mở rộng

INPUT :Hai số nguyên không âm a và b , a≥b

OUTPUT: d= UCLN(a,b) và các số nguyên x và y thỏa mãn ax + by = d

(1) Nếu b = 0 thì đặt d ←a, y ← 0, Kết_quả(d,x,y)

(2) Đặt $x_2 \leftarrow 1$, $x_1 \leftarrow 0$, $y_2 \leftarrow 0$, $y_1 \leftarrow 1$.

(3) Trong khi còn b > 0,thực hiện:

(3.1) q = [a/b], r ← a – qb, x ← $x_2 – qx_1$, y ←$y_2 – qy_1$

(3.2) a ← b, b ← r, $x_2 \leftarrow x_1$, $x_1 \leftarrow x$, $y_2 \leftarrow y_1$, $y_1 \leftarrow y$

(4) Đặt d ←a, x ←x_2, y ← y_2 ,Kết_quả(d,x,y).

Đánh giá độ phức tạp: Thuật toán Euclide mở rộng có độ phức tạp về thời gian :O((lg n)2).

1.6 Không gian Z_n và Z^*_n

1.6.1 Không gian Z_n

Là tập hợp các số nguyên {0,1,2,...,n-1}.Các phép toán trong Z_n như cộng,trừ,nhân,chia đều được thực hiện theo module n.

Ví dụ:Z_{21}={0,1,2,3,…,20}

1.6.2 Không gian Z^*_n

Là tập hợp các số nguyên a∈Z_n,nguyên tố cùng n . Tức là:Z^*_n={a∈Z_n ⎪ gcd(n,a)=1},(n) là số phần tử của Z^*_n.

Nếu là một số nguyên tố thì :Z^*_n={a€ Z_n ⎪ 1≤a≤n-1}

Ví dụ: Z_3={0,1,2} thì Z^*_3={1,2} vì gcd(1,3)=1 và gcd(2,3)=1.

1.7 Định nghĩa cấp của một số a∈ Z^*

Cho α∈ Z^*_n,khi đó cấp của a,kí hiệu ord(a) là số nguyên dương nhỏ nhất sao cho a^t≡1(mod n)trong Z^*_n.

1.8 Tập thặng dư bậc hai theo modulo

Cho a∈ Z^*_n,a được gọi là thặng dư bậc hai theo modulo n nếu tồn tại một x∈ Z^*_n sao cho x^2≡a(mod n) và nếu không tồn tại x như vậy thì a được gọi là bất thặng

dư bậc hai theo modulo n.Tập hợp các thặng dư bậc hai được ký hiệu là Q_n và tập các bất thặng dư bậc hai ký hiệu là $\overline{Qn}$.

1.9Phần tử nghịch đảo

Cho a $\in$ Z_n,số nghịch đảo của a theo modulo n là một số nguyên x$\in$ Z_n ,nếu a.x≡1 (mod n).Nếu tồn tại x như vậy thì nó là duy nhất và a được gọi là khả nghịch,nghịch đảo của a được ký hiệu là a^{-1}.

Tính chất : a $\in$ Z_n,a là khả nghịch khi và chỉ khi gcd(a,n)=1.

Ví dụ:Các phần tử khả nghịch trong Z_9 là 1,2,4,5,7 và 8.

Cho ví dụ ,4^{-1}=7 vì 4.7≡1(mod 9)

***Thuật toán tính nghịch đảo của Z_n**

INPUT: a $\in$ Z_n.

OUTPUT:a^{-1} mod n,nếu tồn tại

1.Sử dụng thuật toán Euclidean mở rộng,tìm x và y để ax+ny=d,trong đó,thì gcd(a,n).

2.Nếu d>1 thì a^{-1} không tồn tại .Ngược lại kết quả(x)

1.10 Lý thuyết độ phức tạp

Một chương trình máy tính thường được cài đặt dựa trên một thuật toán đúng để giải quyết bài toán hay vấn đề. Tuy nhiên, ngay cả khi thuật toán đúng, chương trình vẫn có thể không sử dụng được đối với một dữ liệu đầu vào nào đó vì thời gian để cho ra kết quả là quá lâu hoặc sử dụng quá nhiều bộ nhớ (vượt quá khả năng đáp ứng của máy tính).

Khi tiến hành phân tích thuật toán nghĩa là chúng ta tìm ra một đánh giá về thời gian và "không gian" cần thiết để thực hiện thuật toán. Không gian ở đây được hiểu là các yêu cầu về bộ nhớ, thiết bị lưu trữ, ... của máy tính để thuật toán có thể làm việc. Việc xem xét về không gian của thuật toán phụ thuộc phần lớn vào cách tổ chức dữ liệu của thuật toán. Trong phần này, khi nói đến độ phức tạp của thuật toán, chúng ta chỉ đề cập đến những đánh giá về mặt thời gian mà thôi.

Phân tích thuật toán là một công việc rất khó khăn, đòi hỏi phải có những hiểu biết sâu sắc về thuật toán và nhiều kiến thức toán học khác. Đây là công việc mà không phải bất cứ người nào cũng làm được. Rất may mắn là các nhà toán học đã phân tích cho chúng ta độ phức tạp của hầu hết các thuật toán cơ sở (sắp xếp, tìm kiếm,

các thuật toán số học, ...). Chính vì vậy, nhiệm vụ còn lại của chúng ta là hiểu được các khái niệm liên quan đến độ phức tạp của thuật toán.

Đánh giá về thời gian của thuật toán không phải là xác định thời gian tuyệt đối (chạy thuật toán mất bao nhiêu giây, bao nhiêu phút,...) để thực hiện thuật toán mà là xác định mối liên quan giữa dữ liệu đầu vào (input) của thuật toán và chi phí (số thao tác, số phép tính cộng,trừ, nhân, chia, rút căn,...) để thực hiện thuật toán. Sở dĩ người ta không quan tâm đến thời gian tuyệt đối của thuật toán vì yếu tố này phụ thuộc vào tốc độ của máy tính, mà các máy tính khác nhau thì có tốc độ rất khác nhau. Một cách tổng quát, chi phí *thực hiện thuật toán là một hàm số phụ thuộc vào dữ liệu đầu vào* :

$$T = f(input)$$

Tuy vậy**,** khi phân tích thuật toán người ta thường chỉ chú ý đến mối liên quan giữa *độ lớn của dữ liệu đầu vào và chi phí.* Trong các thuật toán, độ lớn của dữ liệu đầu vàothường được thể hiện bằng một *con số nguyên n.* Chẳng hạn : *sắp xếp* ***n*** *con số nguyên, tìm con số lớn nhất trong n số, tính điểm trung bình của* ***n*** *học sinh, ...* Lúc này, người ta thể hiện chi phí thực hiện thuật toán bằng một hàm số phụ thuộc vào n :

$$T = f(n)$$

Việc xây dựng một hàm T tổng quát như trên trong mọi trường hợp của thuật toán là một việc rất khó khăn, nhiều lúc không thể thực hiện được. Chính vì vậy mà người ta chỉ xây dựng hàm T cho một số trường hợp đáng chú ý nhất của thuật toán, thường là trường hợp *tốt nhất* và *xấu nhất.* Để đánh giá trường hợp tốt nhất và xấu nhất người ta dựa vào định nghĩa sau:

$f(n) = O(g(n))$ và nói $f(n)$ có cấp cao nhất là $g(n)$ khi tồn tại hằng số C và k sao cho $|f(n)| \leq C.g(n)$ với mọi $n > k$

Tuy chi phí của thuật toán trong trường hợp tốt nhất và xấu nhất có thể nói lên nhiều điều nhưng vẫn chưa đưa ra được một hình dung tốt nhất về độ phức tạp của thuật toán. Để có thể hình dung chính xác về độ phức tạp của thuật toán, ta xét đến một yếu tố khác là *độ* ***tăng*** *của chi phí khi độ lớn n của dữ liệu đầu vào* ***tăng.***

Một cách tổng quát, nếu hàm chi phí của thuật toán (xét trong một trường hợp nào đó) bị chặn bởi O(f(n)) thì ta nói rằng thuật toán có độ phức tạp là O(f(n)) trong trường hợp đó. Như vậy, thuật toán tìm số lớn nhất có độ phức tạp trong trường hợp

tốt nhất và xấu nhất đều là O(n). Người ta gọi các thuật toán có độ phức tạp O(n) là các thuật toán có độ phức tạp tuyến tính.

Sau đây là một số "thước đo" độ phức tạp của thuật toán được sử dụng rộng rãi. Các độ phức tạp được sắp xếp theo thứ tự tăng dần. Nghĩa là một bài toán có độ phức tạp O(nk) sẽ phức tạp hơn bài toán có độ phức tạp O(n) hoặc O(logn).

Tên gọi	Ký hiệu
Độ phức tạp hằng	O(C)
Độ phức tạp logarith	$O(\log_e n)$
Độ phức tạp tuyến tính	O(n)
Độ phức tạp nlogn	$O(n.\log_e n)$
Độ phức tạp đa thức	$O(n^k)$
Độ phức tạp lũy thừa	$O(a^n)$
Độ phức tạp giai thừa	O(n!)

CHƯƠNG 2: TỔNG QUAN VỀ MẬT MÃ HỌC

2.1 Lịch sử phát triển của mật mã

Mật mã học là một ngành có lịch sử từ hàng nghìn năm nay. Trong phần lớn thời gian phát triển của mình (ngoại trừ vài thập kỷ trở lại đây), lịch sử mật mã học chính là lịch sử của những phương pháp mật mã học cổ điển - các phương pháp mật mã hóa với bút và giấy, đôi khi có hỗ trợ từ những dụng cụ cơ khí đơn giản. Vào đầu thế kỷ XX, sự xuất hiện của các cơ cấu cơ khí và điện cơ, chẳng hạn như máy Enigma, đã cung cấp những cơ chế phức tạp và hiệu quả hơn cho việc mật mã hóa. Sự ra đời và phát triển mạnh mẽ của ngành điện tử và máy tính trong những thập kỷ gần đây đã tạo điều kiện để mật mã học phát triển nhảy vọt lên một tầm cao mới.Sự phát triển của mật mã học luôn luôn đi kèm với sự phát triển của các kỹ thuật phá mã (hay thám mã). Các phát hiện và ứng dụng của các kỹ thuật phá mã trong một số trường hợp đã có ảnh hưởng đáng kể đến các sự kiện lịch sử. Một vài sự kiện đáng ghi nhớ bao gồm việc phát hiện ra bức điện Zimmermann khiến Hoa Kỳ tham gia Thế chiến 1 và việc phá mã thành công hệ thống mật mã của Đức Quốc xã góp phần làm đẩy nhanh thời điểm kết thúc thế chiến II.Cho tới đầu thập kỷ 1970, các kỹ thuật liên quan tới mật mã học hầu như chỉ nằm trong tay các chính phủ. Hai sự kiện đã khiến cho mật mã học trở nên thích hợp cho mọi người đó là sự xuất hiện của tiêu chuẩn mật mã hóa DES và sự ra đời của các kỹ thuật mật mã hóa khóa công khai.

2.1.1 Mật mã học cổ điển

Những bằng chứng sớm nhất về sử dụng mật mã học là các chữ tượng hình không tiêu chuẩn tìm thấy trên các bức tượng Ai Cập cổ đại (cách đây khoảng 4500). Những ký hiệu tỏ ra không phải để phục vụ mục đích truyền thông tin bí mật mà có vẻ như là nhằm mục đích gợi nên những điều thần bí, trí tò mò hoặc thậm chí để tạo sự thích thú cho người xem. Ngoài ra còn rất nhiều ví dụ khác về những ứng dụng của mật mã học hoặc là những điều tương tự. Muộn hơn, các học giả về tiếng Hebrew có sử dụng một phương pháp mã hóa thay thế bảng chữ cái đơn giản chẳng hạn như mật mã hóa Atbash (khoảng năm 500 đến năm 600). Mật mã học từ lâu đã được sử dụng trong các tác phẩm tôn giáo để che giấu thông tin với chính quyền hoặc nền văn hóa thống trị. Ví dụ tiêu biểu nhất là "số chỉ kẻ thù của Chúa" (tiếng Anh: *Number of the Beast*) xuất hiện trong kinh Tân Ước của Cơ đốc giáo. Ở đây, số 666 có thể là cách mã hóa để chỉ đến Đế chế La Mã hoặc là đến hoàng đế Nero

của đế chế này. Việc không đề cập trực tiếp sẽ đỡ gây rắc rối khi cuốn sách bị chính quyền chú ý. Đối với Cơ đốc giáo chính thống thì việc che dấu này kết thúc khi Constantine cải đạo và chấp nhận đạo Cơ đốc là tôn giáo chính thống của đế chế.Người Hy Lạp cổ đại cũng được biết đến là đã sử dụng các kỹ thuật mật mã (chẳng hạn như mật mã scytale). Cũng có những bằng chứng rõ ràng chứng tỏ người La Mã nắm được các kỹ thuật mật mã (mật mã Caesar và các biến thể). Thậm chí đã có những đề cập đến một cuốn sách nói về mật mã trong quân đội La Mã tuy nhiên cuốn sách này đã thất truyền.Tại Ấn Độ, mật mã học cũng khá nổi tiếng. Trong cuốn sách Kama Sutra, mật mã học được xem là cách những người yêu nhau trao đổi thông tin mà không bị phát hiện.

2.1.2 Thời trung cổ

Nguyên do xuất phát có thể là từ việc phân tích bản kinh Qur'an, do nhu cầu tôn giáo, mà kỹ thuật phân tích tần suất đã được phát minh để phá vỡ các hệ thống mật mã đơn ký tự vào khoảng năm 1000. Đây chính là kỹ thuật phá mã cơ bản nhất được sử dụng, mãi cho tới tận thời điểm của thế chiến thứ II. Về nguyên tắc, mọi kỹ thuật mật mã đều không chống lại được kỹ thuật phân tích mã (*cryptanalytic technique*) này cho tới khi kỹ thuật mật mã đa ký tự được Alberti sáng tạo (năm 1465).Mật mã học ngày càng trở nên quan trọng dưới tác động của những thay đổi, cạnh tranh trong chính trị và tôn giáo. Chẳng hạn tại châu Âu, trong và sau thời kỳ Phục hưng, các công dân của các thành bang thuộc Ý, gồm cả các thành bang thuộc giáo phận và Công giáo La Mã, đã sử dụng và phát triển rộng rãi các kỹ thuật mật mã. Tuy nhiên rất ít trong số này tiếp thu được công trình của Alberti (các công trình của họ không phản ảnh sự hiểu biết hoặc tri thức về kỹ thuật tân tiến của Alberti) và do đó hầu như tất cả những người phát triển và sử dụng các hệ thống này đều quá lạc quan về độ an toàn. Điều này hầu như vẫn còn đúng cho tới tận hiện nay, nhiều nhà phát triển không xác định được điểm yếu của hệ thống. Do thiếu hiểu biết cho nên các đánh giá dựa trên suy đoán và hy vọng là phổ biến.Mật mã học, phân tích mã học và sự phản bội của nhân viên tình báo, của người đưa thư, đều xuất hiện trong âm mưu Babington diễn ra dưới triều đại của nữ hoàng Elizabeth I dẫn đến kết cục xử tử nữ hoàng Mary I của Scotland. Một thông điệp được mã hóa từ thời "người dưới mặt nạ sắt" (*Man in the Iron Mask*) (được giải mã vào khoảng 1900 bởi Étienne Bazeries) cho biết một số thông tin về số phận của tù nhân này (đáng tiếc thay là những thông tin này cũng chưa được rõ ràng cho lắm). Mật mã học và những lạm dụng của nó, cũng là những phần tử liên quan đến mưu đồ dẫn tới việc xử tử Mata Hari và âm mưu quỷ quyệt dẫn đến trò hề trong việc kết

án Dreyfus và bỏ tù hai người đầu thế kỷ 20. May mắn thay, những nhà mật mã học (*cryptographer*) cũng nhúng tay vào việc phơi bày mưu đồ dẫn đến các khúc mắc của Dreyfus ,Mata Hari, ngược lại, đã bị bắn chết.Ngoài các nước ở Trung Đông và châu Âu, mật mã học hầu như không được phát triển. Tại Nhật Bản, mãi cho tới 1510, mật mã học vẫn chưa được sử dụng và các kỹ thuật tiên tiến chỉ được biết đến sau khi nước này mở cửa với phương Tây (thập kỷ 1860).

2.1.3 Mật mã học từ năm 1800 đến Thế chiến II

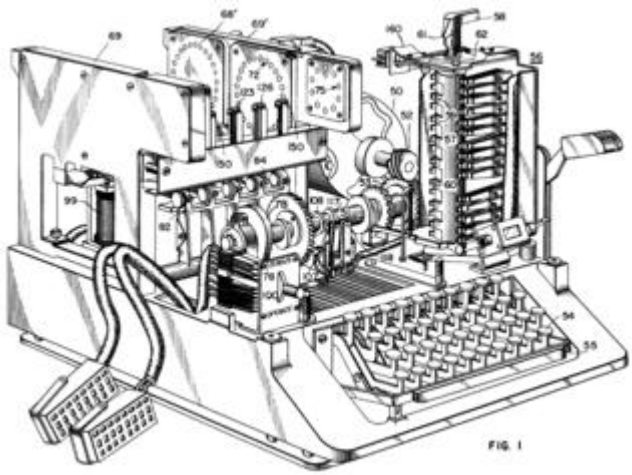
Tuy mật mã học có một lịch sử dài và phức tạp, mãi cho đến thế kỷ 19 nó mới được phát triển một cách có hệ thống, không chỉ còn là những tiếp cận nhất thời, vô tổ chức. Những ví dụ về phân tích mã bao gồm công trình của Charles Babbage trong kỷ nguyên của Chiến tranh Krim (*Crimean War*) về toán phân tích mật mã đơn ký tự. Công trình của ông, tuy hơi muộn màng, đã được Friedrich Kasiski, người Phổ, khôi phục và công bố. Tại thời điểm này, để hiểu được mật mã học, người ta thường phải dựa vào những kinh nghiệm từng trải (*rules of thumb*), xin xem thêm các bài viết về mật mã học của Auguste Kerckhoffs cuối thế kỷ 19. Trong thập niên 1840, Edgar Allan Poe đã xây dựng một số phương pháp có hệ thống để giải mật mã. Cụ thể là, ông đã bày tỏ khả năng của mình trong tờ báo hàng tuần *Alexander's Weekly (Express) Messenger* ở Philadelphia, mời mọi người đệ trình các phương pháp mã hóa của họ, và ông là người đứng ra giải. Sự thành công của ông gây chấn động với công chúng trong vài tháng. Sau này ông có viết một luận văn về các phương pháp mật mã hóa và chúng trở thành những công cụ rất có lợi, được áp dụng vào việc giải mã của Đức trong Thế chiến II.Trong thời gian trước và tới thời điểm của Thế chiến II, nhiều phương pháp toán học đã hình thành (đáng chú ý là ứng dụng của William F. Friedman dùng kỹ thuật thống kê để phân tích và kiến tạo mật mã, và thành công bước đầu của Marian Rejewski trong việc bẻ gãy mật mã của hệ thống Enigma của Quân đội Đức). Sau Thế chiến II trở đi, cả hai ngành, mật mã học và phân tích mã ngày càng sử dụng nhiều các cơ sở toán học. Tuy thế, chỉ đến khi máy tính và các phương tiện truyền thông Internet trở nên phổ biến, người ta mới có thể mang tính hữu dụng của mật mã học vào trong những thói quen sử dụng hằng ngày của mọi người, thay vì chỉ được dùng bởi các chính quyền quốc gia hay các hoạt động kinh doanh lớn trước đó.

2.1.4 Mật mã học trong Thế chiến II

Trong thế chiến II, các hệ thống mật mã cơ khí và cơ điện tử được sử dụng rộng rãi mặc dù các hệ thống thủ công vẫn được dùng tại những nơi không đủ điều

kiện. Các kỹ thuật phân tích mật mã đã có những đột phá trong thời kỳ này, tất cả đều diễn ra trong bí mật. Cho đến gần đây, các thông tin này mới dần được tiết lộ do thời kỳ giữ bí mật 50 năm của chính phủ Anh đã kết thúc, các bản lưu của Hoa Kỳ dần được công bố cùng với sự xuất hiện của các bài báo và hồi ký có liên quan.Người Đức đã sử dụng rộng rãi một hệ thống máy rôto cơ điện tử, dưới nhiều hình thức khác nhau, có tên gọi là máy Enigma. Vào tháng 12 năm 1932, Marian Rejewski, một nhà toán học tại Cục mật mã Ba Lan (tiếng Ba Lan: *Biuro Szyfrów*) đã dựng lại hệ thống này dựa trên toán học và một số thông tin có được từ các tài liệu do đại úy Gustave Bertrand của tình báo quân sự Pháp cung cấp. Đây có thể coi là đột phá lớn nhất trong lịch sử phân tích mật mã trong suốt một nghìn năm trở lại. Rejewski cùng với các đồng sự của mình là Jerzy Różycki và Henryk Zygalski đã tiếp tục nghiên cứu và bắt nhịp với những tiến hóa trong các thành phần của hệ thống cũng như các thủ tục mật mã hóa. Cùng với những tiến triển của tình hình chính trị, nguồn tài chính của Ba Lan trở nên cạn kiệt và nguy cơ của cuộc chiến tranh trở nên gần kề, vào ngày 25 tháng 7 năm 1939 tại Warszawa, cục mật mã Ba Lan, dưới chỉ đạo của bộ tham mưu, đã trao cho đại diện tình báo Pháp và Anh những thông tin bí mật về hệ thống Enigma.Ngay sau khi Thế chiến II bắt đầu (ngày 1 tháng 9 năm 1939), các thành viên chủ chốt của cục mật mã Ba Lan được sơ tán về phía tây nam và đến ngày 17 tháng 9, khi quân đội Liên Xô tiến vào Ba Lan, thì họ lại được chuyển sang Romania. Từ đây, họ tới Paris (Pháp). Tại PC Bruno, ở gần Paris, họ tiếp tục phân tích Enigma và hợp tác với các nhà mật mã học của Anh tại Bletchley Park lúc này đã tiến bộ kịp thời. Những người Anh, trong đó bao gồm những tên tuổi lớn của ngành mật mã học như Gordon Welchaman và Alan Turing, người sáng lập khái niệm khoa học điện toán hiện đại, đã góp công lớn trong việc phát triển các kỹ thuật phá mã hệ thống máy Enigma.Ngày 19 tháng 4 năm 1945, các tướng lĩnh cấp cao của Anh được chỉ thị không được tiết lộ tin tức rằng mã Enigma đã bị phá, bởi vì như vậy nó sẽ tạo điều kiện cho kẻ thù bị đánh bại cơ sở để nói rằng họ đã "không bị đánh bại một cách sòng phẳng" (*were not well and fairly beaten*). Các nhà mật mã học của Hải quân Mỹ (với sự hợp tác của các nhà mật mã học Anh và Hà Lan sau 1940) đã xâm nhập được vào một số hệ thống mật mã của Hải quân Nhật. Việc xâm nhập vào hệ thống JN-25 trong số chúng đã mang lại chiến thắng vẻ vang cho Mỹ trong trận Midway. SIS, một nhóm trong quân đội Mỹ, đã thành công trong việc xâm nhập hệ thống mật mã ngoại giao tối mật của Nhật (một máy cơ điện dùng "bộ chuyển mạch dịch bước" (*stepping switch*) được người Mỹ gọi là Purple) ngay cả trước khi thế chiến II bắt đầu. Người Mỹ đặt tên cho những bí mật mà học tìm được từ việc thám mã, cóthể đặc biệt là từ việc phá mã

máy Purple, với cái tên "Magic". Người Anh sau này đặt tên cho những bí mật mà họ tìm ra trong việc thám mã, đặc biệt là từ luồng thông điệp được mã hóa bởi các máy Enigma, là "Ultra". Cái tên Anh trước đó của Ultra là *Boniface*.Quân đội Đức cũng cho triển khai một số thử nghiệm cơ học sử dụng thuật toán mật mã dùng một lần (*one-time pad*). BletchleyPark gọi chúng là mã Fish và ông Max Newman cùng đồng nghiệp của mình đã thiết kế ra một máy tính điện tử số khả lập trình (*programmable digital electronic computer*) đầu tiên là máy Colossus để giúp việc thám mã của họ. Bộ ngoại giao Đức bắt đầu sử dụng thuật toán mật mã dùng một lần vào năm 1919 một số luồng giao thông của nó đã bị người ta đọc được trong Thế chiến II, một phần do kết quả của việc khám phá ra một số tài liệu chủ chốt tại Nam Mỹ, do sự bất cẩn của những người đưa thư của Đức không hủy thông điệp một cách cẩn thận.Bộ ngoại giao của Nhật cũng cục bộ xây dựng một hệ thống dựa trên nguyên lý của "bộ điện cơ chuyển mạch dịch bước" (được Mỹ gọi là Purple) và đồng thời cũng sử dụng một số máy tương tự để trang bị cho một số tòa đại sứ Nhật Bản. Một trong số chúng được người Mỹ gọi là "Máy-M" (*M-machine*) và một cái nữa được gọi là "Red". Tất cả những máy này đều ít nhiều đã bị phía Đồng Minh phá mã.SIGABA được miêu tả trong Bằng sáng chế của Mỹ 6.175.625, đệ trình năm 1944 song mãi đến năm 2001 mới được phát hành.



Các máy mật mã mà phe Đồng minh sử dụng trong thế chiến II, bao gồm cả máy TypeX của Anh và máy SIGABA của Mỹ, đều là những thiết kế cơ điện dùng rôto trên tinh thần tương tự như máy Enigma, song với nhiều nâng cấp lớn. Không có hệ thống nào bị phá mã trong quá trình của cuộc chiến tranh. Người Ba Lan sử dụng

máy Lacida, song do tính thiếu an ninh, máy không tiếp tục được dùng. Các phân đội trên mặt trận chỉ sử dụng máy M-209 và các máy thuộc họ M-94 ít bảo an hơn. Đầu tiên, các nhân viên mật vụ trong Cơ quan đặc vụ của Anh (*Special Operations Executive* - SOE) sử dụng "mật mã thơ" (các bài thơ mà họ ghi nhớ là những chìa khóa), song ở những thời kỳ sau trong cuộc chiến, họ bắt đầu chuyển sang dùng các hình thức của mật mã dùng một lần (*one-time pad*).

2.1.5 Mật mã học hiện đại

Nhiều người cho rằng kỷ nguyên của mật mã học hiện đại được bắt đầu với Claude Shannon, người được coi là cha đẻ của mật mã toán học. Năm 1949 ông đã công bố bài Lý thuyết về truyền thông trong các hệ thống bảo mật (*Communication Theory of Secrecy Systems*) trên tập san *Bell System Technical Journal* - Tập san kỹ thuật của hệ thống Bell - và một thời gian ngắn sau đó, trong cuốn *Mathematical Theory of Communication* - Lý thuyết toán học trong truyền thông - cùng với tác giả Warren Weaver. Những công trình này cùng với những công trình nghiên cứu khác của ông về lý thuyết về tin học và truyền thông (*information and communication theory*) đã thiết lập một nền tảng lý thuyết cơ bản cho mật mã học và thám mã học. Với ảnh hưởng đó mật mã học hầu như bị thâu tóm bởi các cơ quan truyền thông mật của chính phủ, chẳng hạn như NSA và biến mất khỏi tầm hiểu biết của công chúng. Rất ít các công trình được tiếp tục công bố, cho đến thời kỳ giữa thập niên 1970, khi mọi sự được thay đổi.

Thời kỳ giữa thập niên kỷ 1970 được chứng kiến hai tiến bộ công chính lớn (*công khai*). Đầu tiên là sự công bố đề xuất Tiêu chuẩn mật mã hóa dữ liệu (*Data Encryption Standard*) trong "Công báo Liên bang" (*Federal Register*) ở nước Mỹ vào ngày 17 tháng 3 năm 1975. Với đề cử của Cục Tiêu chuẩn Quốc gia (*National Bureau of Standards* - NBS) (hiện là NIST), bản đề xuất DES được công ty IBM (*International Business Machines*) đệ trình trở thành một trong những cố gắng trong việc xây dựng các công cụ tiện ích cho thương mại, như cho các nhà băng và cho các tổ chức tài chính lớn. Sau những chỉ đạo và thay đổi của NSA, vào năm 1977, nó đã được chấp thuận và được phát hành dưới cái tên Bản Công bố về Tiêu chuẩn Xử lý Thông tin của Liên bang (*Federal Information Processing Standard Publication* - FIPS) (phiên bản hiện nay là FIPS 46-3). DES là phương thức mật mã công khai đầu tiên được một cơ quan quốc gia như NSA "tôn sùng". Sự phát hành bản đặc tả của nó bởi NBS đã khuyến khích sự quan tâm chú ý của công chúng cũng như của các tổ chức nghiên cứu về mật mã học.

Năm 2001, DES đã chính thức được thay thế bởi AES (viết tắt của *Advanced Encryption Standard - Tiêu chuẩn mã hóa tiên tiến*) khi NIST công bố phiên bản FIPS 197. Sau một cuộc thi tổ chức công khai, NIST đã chọn Rijndael, do hai nhà mật mã người Bỉ đệ trình, và nó trở thành AES. Hiện nay DES và một số biến thể của nó (như Tam phần DES (*Triple DES*),xin xem thêm trong phiên bản FIPS 46-3), vẫn còn được sử dụng, do trước đây nó đã được gắn liền với nhiều tiêu chuẩn của quốc gia và của các tổ chức. Với chiều dài khoá chỉ là 56-bit, nó đã được chứng minh là không đủ sức chống lại những tấn công kiểu vét cạn (*brute force attack - tấn công dùng bạo lực*). Một trong những cuộc tấn công kiểu này được thực hiện bởi nhóm "nhân quyền cyber" (*cyber civil-rights group*) tên là Tổ chức tiền tuyến điện tử (*Electronic Frontier Foundation*) vào năm 1997 và đã phá mã thành công trong 56 tiếng đồng hồ -- câu chuyện này được nhắc đến trong cuốn *Cracking DES* (Phá vỡ DES), được xuất bản bởi "O'Reilly and Associates". Do kết quả này mà hiện nay việc sử dụng phương pháp mật mã hóa DES nguyên dạng, có thể được khẳng định một cách không nghi ngờ, là một việc làm mạo hiểm, không an toàn và những thông điệp ở dưới sự bảo vệ của những hệ thống mã hóa trước đây dùng DES, cũng như tất cả các thông điệp được truyền gửi từ năm 1976 trở đi sử dụng DES, đều ở trong tình trạng rất đáng lo ngại. Bất chấp chất lượng vốn có của nó, một số sự kiện xảy ra trong năm 1976, đặc biệt là sự kiện công khai nhất của Whitfield Diffie, chỉ ra rằng chiều dài khóa mà DES sử dụng (56-bit) là một khóa quá nhỏ. Đã có một số nghi ngờ xuất hiện nói rằng một số các tổ chức của chính phủ, ngay tại thời điểm hồi bấy giờ, cũng đã có đủ công suất máy tính để phá mã các thông điệp dùng DES rõ ràng là những cơ quan khác cũng đã có khả năng để thực hiện việc này rồi.Tiến triển thứ hai, vào năm 1976, có lẽ còn đột phá hơn nữa, vì tiến triển này đã thay đổi nền tảng cơ bản trong cách làm việc của các hệ thống mật mã hóa. Đó chính là công bố của bài viết phương hướng mới trong mật mã học (*New Directions in Cryptography*) của Whitfield Diffie và Martin Hellman. Bài viết giới thiệu một phương pháp hoàn toàn mới về cách thức phân phối các khóa mật mã. Đây là một bước tiến khá xa trong việc giải quyết một vấn đề cơ bản trong mật mã học, vấn đề phân phối khóa và nó được gọi là trao đổi khóa Diffie-Hellman (*Diffie-Hellman key exchange*). Bài viết còn kích thích sự phát triển gần như tức thời của một lớp các thuật toán mật mã hóa mới, các thuật toán chìa khóa bất đối xứng (*asymmetric key algorithms*).Trước thời kỳ này, hầu hết các thuật toán mật mã hóa hiện đại đều là những thuật toán khóa đối xứng (*symmetric key algorithms*), trong đó cả người gửi và người nhận phải dùng chung một khóa, tức khóa dùng trong thuật toán mật mã và cả hai người đều phải giữ bí mật về khóa này. Tất cả các

máy điện cơ dùng trong thế chiến II, kể cả mã Caesar và mã Atbashvà về bản chất mà nói, kể cả hầu hết các hệ thống mã được dùng trong suốt quá trình lịch sử nữa đều thuộc về loại này. Đương nhiên, khóa của một mã chính là sách mã (*codebook*) và là cái cũng phải được phân phối và giữ gìn một cách bí mật tương tự.

Do nhu cầu an ninh, khóa cho mỗi một hệ thống như vậy nhất thiết phải được trao đổi giữa các bên giao thông liên lạc bằng một phương thức an toàn nào đấy, trước khi họ sử dụng hệ thống (thuật ngữ thường được dùng là 'thông qua một kênh an toàn'), ví dụ như bằng việc sử dụng một người đưa thư đáng tin cậy với một cặp tài liệu được khóa vào cổ tay bằng một cặp khóa tay, hoặc bằng cuộc gặp gỡ mặt đối mặt, hay bằng một con chim bồ câu đưa thư trung thành... Vấn đề này chưa bao giờ được xem là dễ thực hiện và nó nhanh chóng trở nên một việc gần như không thể quản lý được khi số lượng người tham gia tăng lên, hay khi người ta không còn các kênh an toàn để trao đổi khóa nữa hoặc lúc họ phải liên tục thay đổi các chìa khóa - một thói quen nên thực hiện trong khi làm việc với mật mã. Cụ thể là mỗi một cặp truyền thông cần phải có một khóa riêng nếu, theo như thiết kế của hệ thống mật mã, không một người thứ ba nào, kể cả khi người ấy là một người dùng, được phép giải mã các thông điệp. Một hệ thống thuộc loại này được gọi là một hệ thống dùng chìa khóa mật hoặc một hệ thống mật mã hóa dùng khóa đối xứng. Hệ thống trao đổi khóa Diffie-Hellman (cùng những phiên bản được nâng cấp kế tiếp hay các biến thể của nó) tạo điều kiện cho các hoạt động này trong các hệ thống trở nên dễ dàng hơn rất nhiều, đồng thời cũng an toàn hơn, hơn tất cả những gì có thể làm trước đây.

Ngược lại, đối với mật mã hóa dùng khóa bất đối xứng, người ta phải có một cặp khóa có quan hệ toán học để dùng trong thuật toán, một dùng để mã hóa và một dùng để giải mã. Một số những thuật toán này, song không phải tất cả, có thêm đặc tính là một trong các khóa có thể được công bố công khai trong khi cái kia không thể nào (ít nhất bằng những phương pháp hiện có) được suy ra từ khóa 'công khai'. Trong các hệ thống này, khóa còn lại phải được giữ bí mật và nó thường được gọi bằng một cái tên, hơi có vẻ lộn xộn, là khóa 'cá nhân' (*private key*) hay *khóa bí mật*. Một thuật toán thuộc loại này được gọi là một hệ thống 'khóa công khai' hay hệ thống khóa bất đối xứng. Đối với những hệ thống dùng các thuật toán này, mỗi người nhận chỉ cần có một cặp chìa khóa mà thôi (bất chấp số người gửi là bao nhiêu đi chăng nữa). Trong 2 khóa, một khóa luôn được giữ bí mật và một được công bố công khai nên không cần phải dùng đến một kênh an toàn để trao đổi khóa. Chỉ cần đảm bảo khóa bí mật không bị lộ thì an ninh của hệ thống vẫn được đảm bảo và có thể sử dụng cặp khóa trong một thời gian dài. Đặc tính đáng ngạc nhiên

này của các thuật toán tạo khả năng, cũng như tính khả thi, cho phép việc triển khai các hệ thống mật mã có chất lượng cao một cách rộng rãi và ai cũng có thể sử dụng chúng được.

Các thuật toán mật mã khóa bất đối xứng dựa trên một lớp các bài toán gọi là hàm một chiều (one-way functions). Các hàm này có đặc tính là rất dễ dàng thực hiện theo chiều xuôi nhưng lại rất khó (về khối lượng tính toán) để thực hiện theo chiều ngược lại. Một ví dụ kinh điển cho lớp bài toán này là hàm nhân hai số nguyên tố rất lớn. Ta có thể tính tích số của 2 số nguyên tố này một cách khá dễ dàng nhưng nếu chỉ cho biết tích số thì rất khó để tìm ra 2 thừa số ban đầu. Do những đặc tính của hàm một chiều, hầu hết các khóa có thể lại là những khóa yếu và chỉ còn lại một phần nhỏ có thể dùng để làm khóa. Vì thế, các thuật toán khóa bất đối xứng đòi hỏi độ dài khóa lớn hơn rất nhiều so với các thuật toán khóa đối xứng để đạt được độ an toàn tương đương. Ngoài ra, việc thực hiện thuật toán khóa bất đối xứng đòi hỏi khối lượng tính toán lớn hơn nhiều lần so với thuật toán khóa đối xứng. Bên cạnh đó, đối với các hệ thống khóa đối xứng, việc tạo ra một khóa ngẫu nhiên để làm khóa phiên chỉ dùng trong một phiên giao dịch là khá dễ dàng. Vì thế, trong thực tế người ta thường dùng kết hợp,hệ thống mật mã khóa bất đối xứng được dùng để trao đổi khóa phiên còn hệ thống mật mã khóa đối xứng dùng khóa phiên có được để trao đổi các bản tin thực sự.

Mật mã học dùng khóa bất đối xứng, tức trao đổi khóa Diffie-Hellman, và những thuật toán nổi tiếng dùng khóa công khai / khóa bí mật (ví dụ như cái mà người ta vẫn thường gọi là thuật toán RSA), tất cả hình như đã được xây dựng một cách độc lập tại một cơ quan tình báo của Anh, trước thời điểm công bố của Diffie and Hellman vào năm 1976. Sở chỉ huy giao thông liên lạc của chính phủ (*Government Communications Headquarters - GCHQ*) - Cơ quan tình báo Anh Quốc - có xuất bản một số tài liệu quả quyết rằng chính họ đã xây dựng mật mã học dùng khóa công khai, trước khi bài viết của Diffie và Hellman được công bố. Nhiều tài liệu mật do GCHQ viết trong quá trình những năm 1960 và 1970, là những bài cuối cùng cũng dẫn đến một số kế hoạch đại bộ phận tương tự như phương pháp mật mã hóa RSA và phương pháp trao đổi chìa khóa Diffie-Hellman vào năm 1973 và 1974. Một số tài liệu này hiện được phát hành và những nhà sáng chế (James H. Ellis, Clifford Cocks và Malcolm Williamson) cũng đã cho công bố (một số) công trình của họ.

2.2 Một số thuật ngữ sử dụng trong hệ mật mã

Sender/Receiver: Người gửi/Người nhận dữ liệu.

Văn bản (Plaintext -Cleartext): Thông tin trước khi được mã hoá. Đây là dữ liệu ban đầu ở dạng rõ. Thông tin gốc được ghi bằng hình ảnh âm thanh, chữ số, chữ viết…mọi tín hiệu đều có thể được số hóa thành các xâu ký tự số.
Ciphertext: Thông tin, dữ liệu đã được mã hoá ở dạng mờ.
Khóa (key): Thành phần quan trọng trong việc mã hoá và giải mã.Khóa là đại lượng bí mật, biến thiên trong một hệ mật. Khóa nhất định phải là bí mật. Khóa nhất định phải là đại lượng biến thiên. Tuy nhiên, có thể có trường hợp đại lượng biến thiên trong hệ mật không phải là khóa. Ví dụ: vector khởi tạo (IV = Initial Vector) ở chế độ CBC, OFB và CFB của mã khối.

CryptoGraphic Algorithm: Là các thuật toán được sử dụng trong việc mã hoá hoặc giải mã thông tin

Hệ mã (CryptoSystem hay còn gọi là hệ thống mã): Hệ thống mã hoá bao gồm thuật toán mã hoá, khoá, Plaintext,Ciphertext.

Kỹ thuật mật mã (cryptology) là môn khoa học bao gồm hai lĩnh vực: **mật mã** (crytography) và **mã thám** (cryptoanalysis).

Mật mã (cryptography) là lĩnh vực khoa học về các phương pháp biến đổi thông tin nhằm mục đích bảo vệ thông tin khỏi sự truy cập của những người không có thẩm quyền.

Mã thám (cryptoanalysis) là lĩnh vực khoa học chuyên nghiên cứu, tìm kiếm yếu điểm của các hệ mật để từ đó đưa ra phương pháp tấn công các hệ mật đó. **Mật mã** và **mã thám** là hai lĩnh vực đối lập nhau nhưng gắn bó mật thiết với nhau. Không thể xây dựng một hệ mật tốt nếu không hiểu biết sâu về mã thám. Mã thám chỉ ra yếu điểm của hệ mật. Yếu điểm này có thể được sử dụng để tấn công hệ mật này nhưng cũng có thể được sử dụng để cái tiến hệ mật cho tốt hơn. Nếu người xây dựng hệ mật không có hiểu biết rộng về mã thám, không kiểm tra độ an toàn của hệ mật trước các phương pháp tấn công thì hệ mật của anh ta có thể tỏ ra kém an toàn trước một phương pháp tấn công nào đó mà anh ta chưa biết. Tuy nhiên, không ai có thể khẳng định là có những phương pháp thám mã nào đã được biết đến. Đặc nhiệm của các nước luôn giữ bí mật những kết quả thu được trong lĩnh vực mã thám kể cả phương pháp thám mã và kết qủa của việc thám mã.

Sơ đồ mật mã là tập hợp các thuật toán mã hóa, giả mã, kiểm tra sự toàn vẹn và các chức năng khác của một hệ mật.

Giao thức mật mã là tập hợp các quy tắc, thủ tục quy định cách thức sử dụng sơ đồ mật mã trong một hệ mật. Có thể thấy rằng "giao thức mật mã" và "sơ đồ mật mã" không đi liền với nhau. Có thể có nhiều giao thức khác mật mã khác nhau quy định các cách thức sử dụng khác nhau của cùng một sơ đồ mật mã nào đó.

Lập mã (Encrypt) là việc biến văn bản nguồn thành văn bản mã.

Giải mã (Decrypt) là việc đưa văn bản đã mã hóa trở thành dạng văn bản nguồn.

Định mã (encode/decode) là việc xác định ra phép tương ứng giữa các chữ và số

- Tốc độ mã được đặc trưng bởi số lượng phép tính (N) cần thực hiện để mã hóa (giải mã) một đơn vị thông tin. Cần hiểu rằng **tốc độ mã** chỉ phụ thuộc vào bản thân hệ mã chứ không phụ thuộc vào đặc tính của thiết bị triển triển khai nó (tốc độ máy tính, máy mã...).

Độ an toàn của hệ mã đặc trưng cho khả năng của hệ mã chống lại sự thám mã; nó được đo bằng số lượng phép tính đơn giản cần thực hiện để thám hệ mã đó trong điều kiện sử dụng thuật toán (phương pháp) thám tốt nhất. Cần phải nói thêm rằng có thể xây dựng những hệ mật với độ an tòan bằng vô cùng (tức là không thể thám được về mặt lý thuyết). Tuy nhiên các hệ mật này không thuận tiện cho việc sử dụng, đòi hỏi chi phí cao. Vì thế, trên thực tế, người ta sử dụng những hệ mật có giới hạn đối với độ an toàn. Do đó bất kỳ hệ mật nào cũng có thể bị thám trong thời gian nào đó (ví dụ như sau... 500 năm chẳng hạn).

Khả năng chống nhiễu của mã là khả năng chống lại sự phát tán lỗi trong bản tin sau khi giải mã, nếu trước đó xảy ra lỗi với bản mã trong quá trình bản mã được truyền từ người gửi đến người nhận. Có 3 loại lỗi là:

- lỗi thay thế ký tự: một ký tự bị thay đổi thành một ký tự khác.
  Ví dụ:abcd → atcd
- lỗi chèn ký tự: một ký tự được chèn vào chuỗi ký tự được truyền đi.
  Ví dụ: abcd → azbcd
- lỗi mất ký tự: một ký tự trong chuỗi bị mất.
  Ví dụ: abcd → abd.

Như vậy khái niệm "khả năng chống nhiễu" trong mật mã được hiểu khác hẳn so với khái niệm này trong lĩnh vực truyền tin. Trong truyền tin "khả năng chống nhiễu" là một trong những đặc trưng của "mã chống nhiễu" (noise combating code) - khả năng phát hiện và sửa lỗi của mã chống nhiễu. Ví dụ: mã (7,4) của Hemming có thể phát hiện 2 lỗi và sửa 1 lỗi trong khối 7 bits (4 bits thông tin có ích và 3 bits dùng để kiểm tra và sửa lỗi).

Mã dòng (Stream cipher) là việc tiến hành mã hóa liên tục trên từng ký tự hay từng bit.

Mã khối (Block cipher) là việc tiến hành mã trên từng khối văn bản.

Mục đích của mã hóa là che dấu thông tin trước khi truyền trên kênh truyền. Có nhiều phương pháp mật mã khác nhau, tuy vậy tất cả chúng có hai phép toán thực hiện trong mật mã là phép "mã hóa" và "giải mã". Có thể biểu thị phép mã hóa và phép toán giải mã như các hàm của hai biến số, hoặc có thể như một thuật toán, có nghĩa là một thủ tục đối xứng để tính kết quả khi giá trị các tham số đã cho.Bản tin rõ ở đây là tập hợp các dữ liệu trước khi thực hiện mã hóa. Kết quả của phép mã hóa là bản tin đã được mã hóa. Viêc giải mã bản tin đã được mã hóa sẽ thu được bản tin rõ ban đầu. Có biểu thức "bản tin rõ" và "bản tin đã mã hóa" đều có liên quan đến một mật mã cụ thể. Các chữ cái viết hoa D (Decipherment) và E (Encipherment) là ký hiệu cho các hàm giải mã và mã hóa tương ứng. Ký hiệu x là là bản tin và y là bản tin đã mã hóa thì biểu thức toán học của phép mã hóa là:

$$y= E_k(x)$$

và của phép giải mã là:

$$x=D_k(y)$$

Trong đó tham số phụ k là khóa mã.

Khóa mã là một đặc tính quan trọng của thuật toán mật mã.Về nguyên lý nếu hàm y=E(x) không có một khóa mã nào, thì cũng có thể che dấu được giá trị của x

Tập hợp các giá trị của khóa k được gọi là "không gian các khóa". Trong một mật mã nào đó, nếu khóa mã có 20 số thập phân sẽ cho không gian các khóa là 10^{20}. Nếu khóa nào đó có 50 số nhị phân thì không gian các khóa sẽ là 2^{50}. Nếu khóa là một hoán vị của 26 chữ cái A,B,C…Z thì không gian các khóa sẽ là 26!

Kí hiệu chung: P là thông tin ban đầu, trước khi mã hoá. E() là thuật toán mã hoá. D() là thuật toán giải mã. C là thông tin mã hoá. K là khoá. Chúng ta biểu diễn quá trình mã hoá và giải mã như sau:

Quá trình mã hoá được mô tả bằng công thức: $E_k(P)=C$

Quá trình giải mã được mô tả bằng công thức: $D_k(C)=P$

2.3 Định nghĩa mật mã học

Đối tượng cơ bản của mật mã là tạo ra khả năng liên lạc trên một kênh không mật cho hai người sử dụng (tạm gọi là Alice và Bob) sao cho đối phương (Oscar) không thể hiểu được thông tin truyền đi. Kênh này có thể là một đường dây điện thoại hoặc một mạng máy tính. Thông tin mà Alice muốn gửi cho Bob (bản rõ) có thể là bản tiếng anh, các dữ liệu bằng số hoặc bất kì tài liệu nào có cấu trúc tùy ý. Alice sẽ mã hóa bản rõ bằng một khóa đã được xác định trước và gửi bản mã kết quả trên kênh. Osar có bản mã thu trộm được trên kênh song không thể xác định nội dung của bản rõ, nhưng Bob (người đã biết khóa mã) có thể giải mã và thu được bản rõ.

Ta sẽ mô tả hình thức hóa nội dung bằng cách dùng khái niệm toán học như sau:

Một hệ mật mã là một bộ 5 thành phần (P,C,K,E,D) thỏa mãn các tính chất sau:

1.P là một tập hữu hạn các bản rõ có thể

2.C là một tập hữu hạn các bản mã có thể

3.K(không gian khóa) là tập hữu hạn các khóa có thể

4.Đối với mỗi $k\in K$ có một quy tắc mã e_k: $P\rightarrow C$ và một quy tắc giải mã tương ứng $d_k\in D$. Mỗi e_k:$P\rightarrow C$ và d_k :$C\rightarrow P$ là những hàm

$$D_k(e_k(x))=x \text{ với mọi bản rõ } x\in P$$

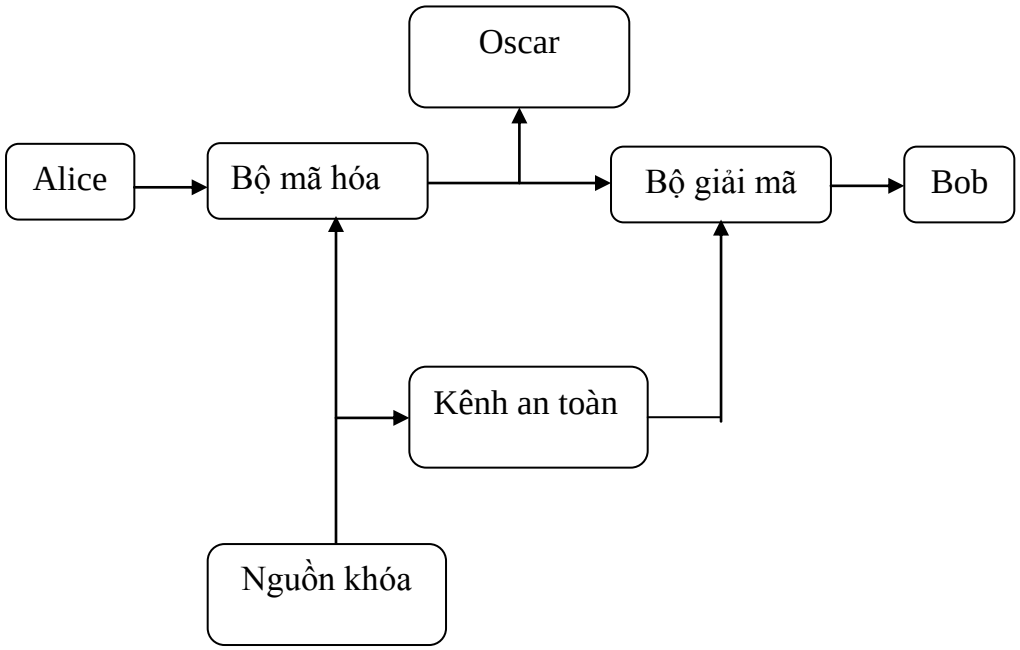
Trong tính chất 4 là tính chất chủ yếu ở đây. Nội dung của nó là nếu một bản rõ x được mã hóa bằng e_k và bản mã nhận được sau đó được giải mã bằng d_k thì ta phải thu được bản rõ ban đầu x. Alice và Bob sẽ áp dụng thủ tục sau khi dùng hệ mật khóa riêng. Trước tiên họ chọn một khóa ngẫu nhiên $k \in K$. Điều này được thực hiện khi họ ở cùng một chỗ và không bị Oscar theo dõi hoặc họ có một kênh mật trong trường hợp họ ở xa nhau. Sau đó giả sử Alice muốn gửi một thông báo cho Bob trên một kênh không mật và ta xem thông báo này là một chuỗi:

$$x = x_1,x_2 ,. . .,x_n$$

với số nguyên $n\geq 1$ nào đó. Ở đây mỗi ký hiệu của mỗi bản rõ $x_i\in P$, $1\leq i \leq n$. Mỗi x_i sẽ được mã hóa bằng quy tắc mã e_k với khóa k xác định trước đó.Bởi vậy Alice sẽ tính $y_i =e_k(x_i)$, $1\leq i \leq n$ và chuỗi bản nhận được:

$$y = y_1, y_2, \ldots, y_n$$

sẽ được gửi trên kênh. Khi Bob nhận được $y = y_1, y_2, \ldots, y_n$ anh ta sẽ giải mã bằng hàm giải mã d_k và thu được bản rõ gốc $x_1, x_2, \ldots, x_n$. Hình 1.1. là một ví dụ về một kênh liên lạc.



Rõ ràng trong trường hợp này hàm mã hóa phải là hàm đơn ánh (tức là ánh xạ 1-1), nếu không việc giải mã sẽ không thực hiện được một cách tường minh. Ví dụ:

$$y = e_k(x_1) = e_k(x_2)$$

trong đó $x_1 \neq x_2$, thì Bob sẽ không có cách nào biết liệu sẽ phải giải mã thành x_1 hay x_2. Chú ý rằng nếu $P = C$ thì mỗi hàm mã hóa ize="2". Bản quyền Công ty Phát tập các bản mã và tập các bản rõ là đồng nhất thì mỗi một hàm mã sẽ là một sự sắp xếp lại (hay hoán vị) các phần tử của tập này.

2.4 Phân loại hệ mật mã học

Lịch sử của mật mã học chính là lịch sử của phương pháp mật mã học cổ điển- phương pháp mã hóa bút và giấy. Sau này dựa trên nền tảng của mật mã học cổ điển đã xuất hiện phương pháp mã hóa mới. Chính vì vậy mật mã học được phân chia thành mật mã học cổ điển và mật mã học hiện đại.

2.4.1 Mật mã cổ điển (cái này ngày nay vẫn hay dùng trong trò chơi tìm mật thư)

Dựa vào kiểu của phép biến đối trong hệ mật mã cổ điển, người ta chia hệ mật mã làm 2 nhóm: mã thay thế (substitution cipher) và mã hoán vị (permutation/ transposition cipher).

Substitution: thay thế – phương pháp mã hóa trong đó từng kí tự (hoặc từng nhóm kí tự) của văn bản ban đầu(bản rõ - Plaintext) được thay thế bằng một (hay một nhóm) kí tự khác để tạo ra bản mờ (Ciphertext). Bên nhận chỉ cần đảo ngược trình tự thay thế trên Ciphertext để có được Plaintext ban đầu.

Transposition: hoán vị - Bên cạnh phương pháp mã hoá thay thế thì trong mã hoá cổ điển có một phương pháp khác nữa cũng nổi tiếng không kém, đó chính là mã hoá hoán vị. Nếu như trong phương pháp mã hoá thay thế, các kí tự trong Plaintext được thay thế hoàn toàn bằng các kí tự trong Ciphertext, thì trong phương pháp mã hoá hoán vị, các kí tự trong Plaintext vẫn được giữ nguyên, chúng chỉ được sắp xếp lại vị trí để tạo ra Ciphertext. Tức là các kí tự trong Plaintext hoàn toàn không bị thay đổi bằng kí tự khác.Cụ thể phương pháp hoán vị là phương pháp mã hóa trong đó các kí tự trong văn bản ban đầu chỉ thay đổi vị trí cho nhau còn bản thân các kí tự không hề bị biến đổi.

Ví dụ đơn giản nhất: mã hóa bản rõ bằng cách đảo ngược thứ tự các ký tự của nó. Giả sử bản rõ của bạn có độ dài N ký tự. Bạn sẽ hoán đổi vị trí ký tự thứ 1 và ký tự N, ký tự 2 và ký tự N-1,…Phức tạp hơn một chút, hoán vị không phải toàn bộ bản rõ mà chia nó ra các đoạn với độ dài L và thực hiện phép hoán vị theo từng đoạn.Khi đó L sẽ là khóa của bạn! Mặt khác L có thể nhận giá trị tuyệt đối (2,3,4…) hoặc giá trị tương đối (1/2,1/3,1/4…của N).

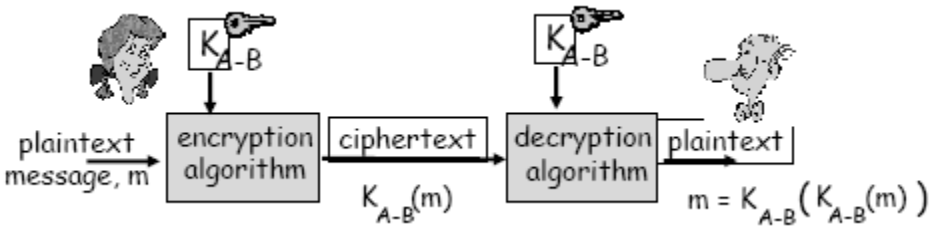
Vào khoảng thể kỷ V-IV trước Công nguyên, người ta đã nghĩ ra "thiết bị mã hóa". Đó là một ống hình trụ với bán hình R. Để mã hóa, người ta quấn băng giấy (nhỏ, dài như giấy dùng trong điện tín) quanh ống hình trụ này và viết nội dung cần mã hóa lên giấy theo chiều dọc của ống. Sau khi gỡ băng giấy khỏi ống thì nội dung sẽ được che dấu. Muoons giải mã thì phải cuốn băng giấy lên ống cùng có bán kính R.Bán kính R chính là khóa trong hệ mật này.

2.4.2 Mật mã hiện đại

Symmetric cryptography:mã hóa đối xứng, tức là cả hai quá trình mã hóa và giải mã đều dùng một chìa khóa. Để đảm bảo tính an toàn, chìa khóa này phải được giữ bí mật. Vì thế các thuật toán loại này còn có tên gọi khác là *secret key cryptography (hay private key cryptography),* tức là thuật toán mã hóa dùng chìa khóa riêng (hay bí mật). Các thuật toán loại này lý tưởng cho mục đích mã hóa dữ

liệu của cá nhân hay tổ chức đơn lẻ nhưng bộc lộ hạn chế khi thông tin đó phải được chia sẻ với một bên thứ hai.

Giả sử nếu Alice chỉ gửi thông điệp đã mã hóa cho Bob mà không hề báo trước về thuật toán sử dụng, Bob sẽ chẳng hiểu Alice muốn nói gì. Vì thế bắt buộc Alice phải thông báo cho Bob về chìa khóa và thuật toán sử dụng tại một thời điểm nào đó trước đấy. Alice có thể làm điều này một cách trực tiếp (mặt đối mặt) hay gián tiếp (gửi qua email, tin nhắn...). Điều này dẫn tới khả năng bị người thứ ba xem trộm chìa khóa và có thể giải mã được thông điệp Alice mã hóa gửi cho Bob.



Hình 1.Thuật toán mã hóa đối xứng

Bob và Alice có cùng một khóa K_{A-B}. Khóa này được xây dựng sao cho:

$m = K_{A-B}(K_{A-B}(m))$.

Trên thực tế, đối với các hệ mật đối xứng, khoá K luôn chịu sự biến đổi trước mỗi pha mã hóa và giải mã. Kết quả của sự biến đổi này ở pha giải mã K_d sẽ khác với kết quả biến đổi ở pha mã hóa K_e.Nếu coi K_e và K_d lần lượt là khóa mã hóa và khóa giải mã thì sẽ có khóa giải mã không trùng với khóa mã hóa. Tuy nhiên nếu biết được khóa K_e thì có thể dễ dàng tính được K_d và ngược lại. Vậy nên có một định nghĩa rộng hơn cho mã đối xứng là: "Mã đối xứng là nhóm mã trong đó khóa dùng để giải mã K_d có thể dễ dàng tính được từ khóa dùng để mã hóa K_e".

Trong hệ thống mã hoá đối xứng, trước khi truyền dữ liệu, 2 bên gửi và nhận phải thoả thuận về khoá dùng chung cho quá trình mã hoá và giải mã. Sau đó, bên gửi sẽ mã hoá bản rõ (Plaintext) bằng cách sử dụng khoá bí mật này và gửi thông điệp đã mã hoá cho bên nhận. Bên nhận sau khi nhận được thông điệp đã mã hoá sẽ sử dụng chính khoá bí mật mà hai bên thoả thuận để giải mã và lấy lại bản rõ (Plaintext). Trong quá trình tiến hành trao đổi thông tin giữa bên gửi và bên nhận thông qua việc sử dụng phương pháp mã hoá đối xứng, thì thành phần quan trọng nhất cần phải được giữ bí mật chính là khoá. Việc trao đổi, thoả thuận về thuật toán được sử dụng trong việc mã hoá có thể tiến hành một cách công khai, nhưng bước thoả thuận về khoá trong việc mã hoá và giải mã phải tiến hành bí mật. Chúng ta có thể

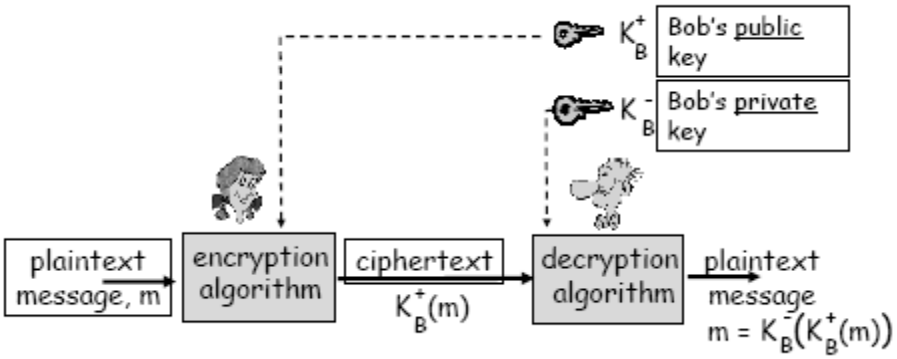
thấy rằng thuật toán mã hoá đối xứng sẽ rất có lợi khi được áp dụng trong các cơ quan hay tổ chức đơn lẻ. Nhưng nếu cần phải trao đổi thông tin với một bên thứ ba thì việc đảm bảo tính bí mật của khoá phải được đặt lên hàng đầu.

Mã hóa đối xứng có thể phân thành hai nhóm phụ:

- Block ciphers**:** thuật toán khối – trong đó từng khối dữ liệu trong văn bản ban đầu được thay thế bằng một khối dữ liệu khác có cùng độ dài. Độ dài mỗi khối gọi là block size, thường được tính bằng đơn vị bit. Ví dụ thuật toán 3-Way có kích thước khối bằng 96 bit. Một số thuật toán khối thông dụng là:DES, 3DES, RC5, RC6, 3-Way, CAST, Camelia, Blowfish, MARS, Serpent, Twofish, GOST...
- Stream ciphers**:** thuật toán dòng – trong đó dữ liệu đầu vào được mã hóa từng bit một. Các thuật toán dòng có tốc độ nhanh hơn các thuật toán khối, được dùng khi khối lượng dữ liệu cần mã hóa chưa được biết trước, ví dụ trong kết nối không dây. Có thể coi thuật toán dòng là thuật toán khối với kích thước mỗi khối là 1 bit. Một số thuật toán dòng thông dụng: RC4, A5/1, A5/2, Chameleon

Asymmetric cryptography: mã hóa bất đối xứng, sử dụng một cặp chìa khóa có liên quan với nhau về mặt toán học, một chìa công khai dùng để mã hoá (public key) và một chìa bí mật dùng để giải mã (private key). Một thông điệp sau khi được mã hóa bởi chìa công khai sẽ chỉ có thể được giải mã với chìa bí mật tương ứng. Do các thuật toán loại này sử dụng một chìa khóa công khai (không bí mật) nên còn có tên gọi khác là *public-key cryptography (thuật toán mã hóa dùng chìa khóa công khai).* Một số thuật toán bất đối xứng thông dụng là : RSA, Elliptic Curve, ElGamal, Diffie Hellman...

Quay lại với Alice và Bob, nếu Alice muốn gửi một thông điệp bí mật tới Bob, cô ta sẽ tìm chìa khoá công khai của Bob. Sau khi kiểm tra chắc chắn chìa khóa đó chính là của Bob chứ không của ai khác (thông qua chứng chỉ điện tử – digital certificate), Alice dùng nó để mã hóa thông điệp của mình và gửi tới Bob. Khi Bob nhận được bức thông điệp đã mã hóa anh ta sẽ dùng chìa bí mật của mình để giải mã nó. Nếu giải mã thành công thì bức thông điệp đó đúng là dành cho Bob. Alice và Bob trong trường hợp này có thể là hai người chưa từng quen biết. Một hệ thống như vậy cho phép hai người thực hiện được giao dịch trong khi không chia sẻ trước một thông tin bí mật nào cả.



Hình 2.Thuật toán mã hóa bất đối xứng

Trong ví dụ trên ta thấy khóa public và khóa private phải đáp ứng : $m = K_B^-(K_B^+(m))$ và từ khóa public K_B^+ người ta không thể tìm ra được khóa private.

Mã hoá khoá công khai ra đời để giải quyết vấn đề về quản lý và phân phối khoá của các phương pháp mã hoá đối xứng. Quá trình truyền và sử dụng mã hoá khoá công khai được thực hiện như sau:

- Bên gửi yêu cầu cung cấp hoặc tự tìm khoá công khai của bên nhận trên một server chịu trách nhiệm quản lý khoá.
- Sau đó hai bên thống nhất thuật toán dùng để mã hoá dữ liệu, bên gửi sử dụng khoá công khai của bên nhận cùng với thuật toán đã thống nhất để mã hoá thông tin được gửi đi.
- Khi nhận được thông tin đã mã hoá, bên nhận sử dụng khoá bí mật của mình để giải mã và lấy ra thông tin ban đầu.

Vậy là với sự ra đời của Mã hoá công khai thì khoá được quản lý một cách linh hoạt và hiệu quả hơn. Người sử dụng chỉ cần bảo vệ Private key. Tuy nhiên nhược điểm của Mã hoá khoá công khai nằm ở tốc độ thực hiện, nó chậm hơn rất nhiều so với mã hoá đối xứng. Do đó, người ta thường kết hợp hai hệ thống mã hoá khoá đối xứng và công khai lại với nhau và được gọi là Hybrid Cryptosystems. Một số thuật toán mã hoá công khai nổi tiếng: Diffle-Hellman, RSA,…

Trên thực tế hệ thống mã hoá khoá công khai có hạn chế về tốc độ chậm nên chưa thể thay thế hệ thống mã hoá khoá bí mật được, nó ít được sử dụng để mã hoá dữ liệu mà thường dùng để mã hoá khoá. Hệ thống mã hoá khoá lai ra đời là sự kết hợp giữa tốc độ và tính an toàn của hai hệ thống mã hoá ở trên. Vì vậy người ta thường sử dụng một hệ thống lai tạp trong đó dữ liệu được mã hóa bởi một thuật toán đối

xứng, chỉ có chìa dùng để thực hiện việc mã hóa này mới được mã hóa bằng thuật toán bất đối xứng. Hay nói một cách khác là người ta dùng thuật toán bất đối xứng để chia sẻ chìa khóa bí mật rồi sau đó dùng thuật toán đối xứng với chìa khóa bí mật trên để truyền thông tin.

Chúng ta có thể hình dung được hoạt động của hệ thống mã hoá này như sau:

- Bên gửi tạo ra một khoá bí mật dùng để mã hoá dữ liệu. Khoá này còn được gọi là Session Key.

- Sau đó, Session Key này lại được mã hoá bằng khoá công khai của bên nhận dữ liệu.

- Tiếp theo dữ liệu mã hoá cùng với Session Key đã mã hoá được gửi đi tới bên nhận.

- Lúc này bên nhận dùng khoá riêng để giải mã Session Key và có được Session Key ban đầu.

-Dùng Session Key sau khi giải mã để giải mã dữ liệu. Như vậy, hệ thống mã hoá khoá lai đã tận dụng tốt được các điểm mạnh của hai hệ thống mã hoá ở trên đó là: tốc độ và tính an toàn. Điều này sẽ làm hạn chế bớt khả năng giải mã của tin tặc.

Cần lưu ý rằng trên đây, chúng ta đã nhắc đến hai khái niệm có tính chất tương đối là "dễ" và "khó". Người ta quy ước rằng nếu thuật toán có độ phức tạp không vượt quá độ phức tạp đa thức thì bài toán được coi là dễ; còn lớn hơn thì bài toán được coi là khó.

2.5 Hệ mật mã cổ điển

2.5.1 Hệ mã Caesar

Hệ mã Caesar được xác định trên Z_{26} (do có 26 chữ cái trên bảng chữ cái tiếng Anh) mặc dù có thể xác định nó trên Z_n với modulus n tùy ý.Dễ dàng thấy rằng, mã dịch vòng sẽ tạo nên một hệ mật như đã xác định ở trên, tức là $D_k(E_k(x)) = x$ với $\forall x \in Z_{26}$.

Định nghĩa:Một hệ mật gồm bộ 5 (*P,C,K,E,D*). Giả sử $P = C = K = Z_{26}$ với $0 \leq k \leq 25$, định nghĩa:

$E_k(x)=x+k \bmod 26$

Và $D_k(x)=y-k \bmod 26$ $(x,y \in Z_{26})$

Nhận xét: Trong trường hợp k=3, hệ mật thường được gọi là mã Caesar đã từng được Julius Caesar sử dụng.Ta sẽ sử dụng mã dịch vòng (với modulo 26) để mã hóa một văn bản tiếng Anh thông thường bằng cách thiết lập sự tương ứng giữa các ký tự và các thặng dư theo modulo 26 như sau: A↔0, B↔1,….,Z↔25.

A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
N	**O**	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	**X**	**Y**	**Z**
13	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**

Ví dụ:

Giả sử khóa cho mã dịch vòng k=11 và bản rõ là: *wewillmeetatmidnight*

Trước tiên biến đổi bản rõ thành dãy các số nguyên nhờ dùng phép tương ứng trên.

Ta có:

22 4 22 8 11 11 12 4 4 19

0 19 12 8 3 13 8 6 7 19

Sau đó cộng 11 vào mỗi giá trị rồi rút gọn tổng theo modulo 26

7 15 7 19 22 22 23 15 15 4

11 4 23 19 14 24 19 17 18 4

Cuối cùng biến đổi dãy số nguyên này thành các ký tự thu được bản mã sau:

HPHTWWXPPELEXTOYTRSE

Để giả mã bản mã này, trước tiên, Bob sẽ biến đổi bản mã thành dãy các số nguyên rồi trừ đi giá trị cho 11 (rút gọn modulo 26) và cuối cùng biến đổi lại dãy này thành các ký tự

2.5.2 Hệ mã Affinne

Mã tuyến tính Affinne là bộ 5 *(P,C,K,E,D)* thỏa mãn:

Cho P=C=Z_{26} và giả sử P={(a,b) $\in Z_{26}$ x Z_{26}:UCLN(a,26)=1}

Với k=(a,b)$\in K$, ta định nghĩa:

$$E_k(x)=ax+b \bmod 26$$

$$\text{Và } D_k(y)=a^{-1}(y-b) \bmod 26, \qquad x,y\in Z_{26}$$

Với bất kỳ $y\in Z_{26}$, ta muốn có đồng nhất thức sau:

$$ax+b\equiv y \pmod{26}$$

phải có nghiệm x duy nhất.

Đồng dư thức<=>$ax\equiv y-b \pmod{26}$

vì y thay đổi trên Z_{26} nên y-b cũng thay đổi trên Z_{26} nên ta chỉ cần nghiên cứu phương trình đồng dư:

$$ax\equiv y \pmod{26} \ (y\in Z_{26})$$

ta biết rằng phương trình này có một nghiệm duy nhất đối với mỗi y khi và chỉ khi UCLN(a,26)=1.

Chứng minh:Trước tiên ta giả sử rằng, UCLN(a,26)=d>1. Khi đó, đồng dư thức $ax\equiv 0 \pmod{26}$ sẽ có ít nhất hai nghiệm phân biệt trong Z_{26} là x=0 và x=26/d. Trong trường hợp này, $E(x)=ax+b \pmod{26}$ không phải là một hàm đơn ánh và bởi vậy nó không thể là hàm mã hóa hợp lệ.

Ví dụ do UCLN(4,26)=2 nên 4x+7 không là hàm mã hóa hợp lệ: x và x+13 sẽ mã hóa thành cùng một giá trị đối với bất kỳ $x\in Z_{26}$.

Giả thiết UCLN(a,26)=1.Giả sử với x_1 và x_2 nào đó thỏa mãn:

$$ax_1\equiv ax_2 \pmod{26}$$

Khi đó:

$$a(x_1 - x_2) \equiv 0 \pmod{26}$$

bởi vậy $26 \mid a(x_1 - x_2)$

Nếu UCLN(a,b)=1 và a | bc thì a |c. Vì $26 \mid a(x_1 - x_2)$ và UCLN(a,26)=1 nên ta có:

$$26 \mid (x_1 - x_2)$$

Tức là:

$$x_1\equiv x_2 \pmod{26}$$

Nếu UCLN(a,26)=1 thì một đồng dư thức dạng $ax \equiv y \pmod{26}$ chỉ có nhiều nhất một nghiệm trong Z_{26}.Dó đó, nếu ta cho x thay đổi trên Z_{26} thì ax mod 26 sẽ nhận được 26 giá trị khác nhau theo modulo 26 và đồng dư thức $ax \equiv y \pmod{26}$ chỉ có nghiệm duy nhất.

Ví dụ:

Giả sử k=(7,3).Ta có $7^{-1} \bmod 26 = 15$.Hàm mã hóa là:

$$E_k(x)=7x+3$$

Hàm giải mã tương ứng là:

$$D_k(x)=15(y-3) \bmod 26=15y-19$$

ở đây tất cả các phép toán đều thực hiện trên Z_{26}. Ta sẽ kiểm tra liệu $D_k(E_k(x))=x$ với $x \in Z_{26}$ không? Dùng các tính toán trên Z_{26}, ta có

$$D_k(E_k(x)) = D_k(7x+3)$$
$$= 15(7x+3)-19$$
$$=x+45-19$$
$$=x$$

Để minh họa, ta hãy mã hóa bản rõ "hot". Trước tiên biến đổi các chữ h,o,t thành các thặng dư theo modulo 26. Ta được các số tương ứng là: 7, 14 và 19.Bây giờ mã hóa:

$7 \times 7 + 3 \bmod 26 = 52 \bmod 26 = 0$

$7 \times 14 + 3 \bmod 26 = 101 \bmod 26 = 23$

$7 \times 19 + 3 \bmod 26 = 136 \bmod 26 = 6$

Bây giờ 3 ký tự của bản mã là 0, 23 và 6 tương ứng với xâu ký tự AXG.

Giải mã: từ xâu ký tự của bản mã chuyển thành số nguyên trong bảng chữ cái tiếng Anh (26 chữ cái), ta được các số tương ứng 0, 23, 6

$D_k(0)=15 \times 0 - 19 \bmod 26 = 7$

$D_k(23)=15 \times 23 - 19 \bmod 26 = 14$

$D_k(6)=15 \times 6 - 19 \bmod 26 = 19$

Bây giờ 3 ký tự của bản rõ: h, o, t.

2.5.3 Hệ mã Vigenère

Trong cả hai hệ mã dịch chuyển và mã tuyến tính(một khi khóa đã được chọn) mỗi ký tự sẽ được ánh xạ vào một ký tự duy nhất. Vì lý do đó, các hệ mật còn lại được gọi là hệ thay thế đơn biểu. Bây giờ tôi sẽ trình bày một hệ mật không phải là bộ chữ đơn, đó là hệ mã Vigenère nổi tiếng. Mật mã này lấy tên của Blaise de Vigenère sống vào thế kỷ XVI.

Sử dụng phép tương ứng A$\Leftrightarrow$ 0, B $\Leftrightarrow$ 1, ….,Z$\Leftrightarrow$25 mô tả trên, ta có thể gắn cho mỗi khóa k với một chuỗi ký tự có độ dài m được gọi là từ khóa.Mật mã V sẽ mã hóa đồng thời m ký tự: mỗi phần tử của bản rõ tương đương với m ký tự

Ví dụ:

Giả sử m=6 và từ khóa là CIPHER. Từ khóa này tương ứng với dãy số k=(2,8,15,4,17).Giả sử bản rõ là xâu

thiscryptosystemisnotsecure

Định nghĩa:

Cho m là một số dương cố định nào đó. Cho P=C=K=$(Z_{26})^m$. Với khóa K=(k_1, k_2 ,…,k_m) ta xác định:

$$E_K(x_1, x_2, \ldots, x_m) = (x_1+k_1, x_2+k_2, \ldots, x_m+k_m)$$

và

$$D_K(y_1, y_2, \ldots, y_m) = (y_1-k_1, y_2-k_2, \ldots, y_m-k_m)$$

Trong đó tất cả các phép toán được thực hiện trong Z_{26}

Ta sẽ biến đổi các phần tử của bản rõ thành các thặng dư theo modulo 26, viết chúng thành các nhóm 6 rồi cộng với từ khóa theo modulo như sau:

19	7	8	18	2	17	24	15	19	14	18	24
2	8	15	7	4	17	2	8	15	7	4	17
21	15	23	25	6	8	0	23	8	21	22	15
18	19	4	12	8	18	13	14	19	18	4	2
2	8	15	7	4	17	2	8	15	7	4	17

20	1	19	19	12	9	15	22	8	15	8	19
	20	17	4								
	2	8	15								
	22	25	19								

Bởi vậy, dãy ký tự tương ứng của xâu bản mã sẽ là:

V P X Z G I A X I V W P U B T T M J P W I Z I T W Z T

Để giải mã ta có thể dùng cùng từ khóa nhưng thay cho cộng, ta trừ nó theo modulo 26

Ta thấy rằng các từ khóa có thể với số độ dài m trong mật mã Vigenère là 26^m, bởi vậy, thậm chí với các giá trị m khá nhỏ, phương pháp tìm kiếm vét cạn cũng yêu cầu thời gian khá lớn. Ví dụ, nếu m=5 thì khôn gian khóa cũng có kích thước lớn hơn $1{,}1 \times 10^7$. Lượng khóa này đã đủ lớn ngăn ngừa việc tìm khóa bằng tay

Trong hệ mật Vigenère có từ khóa độ dài m, mỗi ký tự có thể được ánh xạ vào trong m ký tự có thể có (giả sử rằng từ khóa chứa m ký tự phân biệt).Một hệ mật như vậy được gọi là hệ mật thay thê đa kiểu (poly alphabetic). Nói chung, việc thám mã hệ thay thế đa kiểu sẽ khó khăn hơn so việc thám mã hệ đơn kiểu.

2.5.4 Hệ mật Hill

Trong phần này sẽ mô tả một hệ mật thay thế đa kiểu khác được gọi là mật mã Hill. Mật mã này do Lester S.Hill đưa ra năm 1929. Giả sử m là một số nguyên, đặt $P = C = (Z_{26})^m$. Ý tưởng ở đây là lấy tổ hợp tuyến tính của m ký tự trong một phần tử của bản rõ để tạo ra m ký tự ở một phần tử của bản mã.

Định nghĩa: Mật mã Hill là bộ 5*(P, C, K, E, D)*. Cho m là một số nguyên dương cố định. Cho $P = C = (Z_{26})^m$ và cho

$$K=\{\text{các ma trận khả nghịch cấp } m \times m \text{ trên } Z_{26}\}$$

Với một khóa $K \in K$ ta xác định

$$E_K(x) = xK$$

và $$D_K(y) = yK^{-1}$$

tất cả các phép toán được thực hiện trong Z_{26}

Ví dụ:

Giả sử khóa:

$$K = \begin{bmatrix} 11 & 8 \\ 3 & 7 \end{bmatrix}$$

Ta có:

$$K^{-1} = \begin{bmatrix} 7 & 18 \\ 23 & 11 \end{bmatrix}$$

Giả sử cần mã hóa bản rõ "July". Ta có hai phần tử của bản rõ để mã hóa:(9,20)(ứng với Ju) và (11,24)(ứng với ly).

Ta có:

$$(9,20)\begin{bmatrix} 11 & 8 \\ 3 & 7 \end{bmatrix} = (99+60,\ 72+140) = (3,4)$$

Và

$$(11,24)\begin{bmatrix} 11 & 8 \\ 3 & 7 \end{bmatrix} = (121+72,\ 88+168) = (11,22)$$

Bởi vậy bản mã của july là DELW. Để giải mã Bob sẽ tính

$$(3,4)\begin{bmatrix} 7 & 18 \\ 23 & 11 \end{bmatrix} = (9,20)$$

Và

$$(11,22\begin{bmatrix} 7 & 18 \\ 23 & 11 \end{bmatrix} = (11,24)$$

Như vậy Bob đã nhận được bản đúng

Cho tới lúc này ta đã chỉ ra rằng có thể thực hiện phép giải mã nếu K có một nghịch đảo. Trên thực tế, để phép giải mã là có thể thực hiện được, điều kiện cần là K phải có nghịch đảo. (Điều này dễ dàng rút ra từ đại số tuyến tính sơ cấp).

2.5.5 Hệ mật Playfair

Phép thay thế *n*-gram:thay vì thay thế đối với các kí tự, người ta có thể thay thế cho từng cụm 2 kí tự (gọi là digram) hoặc cho từng cụm 3 kí tự (gọi là trigram) và tổng quát cho từng cụm *n* kí tự (gọi là *n*-gram). Nếu bảng chữ cái Σ gồm 26 kí tự

tiếng Anh thì phép thay thế *n*-gram sẽ có khoá là một hoán vị của 26^n *n*-gram khác nhau. Trong trường hợp digram thì hoán vị gồm 262 digram và có thể biểu diễn tốt nhất bằng một dãy 2 chiều 26 × 26 trong đó các hàng biểu diễn kí hiệu đầu tiên, các cột biểu diễn kí hiệu thứ hai, nội dung của các ô biểu diễn chuỗi thay thế. Ví dụ bảng 2 chiều sau biểu thị AA được thay bằng EG, AB được thay bằng RS, BA được thay bằng BO, BB được thay bằng SC,…

	A	B	…
A	EG	RS	
B	BO	SC	

Đây là một sơ đồ dựa trên sự thay thế digram trong đó khoá là một hình vuông kích thước 5 × 5 chứa một sự sắp xếp nào đó của 25 kí tự của bảng chữ cái (không tính kí tự J vì sự xuất hiện ít của nó và có thể thay nó bằng I). Giả sử chúng ta có ma trận khoá như sau:

B Y D G Z

W S F U P

L A R K X

C O I V E

Q N M H T

Sự thay thế sẽ được thực hiện như sau. Chẳng hạn nếu digram cần thay thế là AV thì trong hình chữ nhật có A, V là hai đỉnh chéo nhau thay A bằng đỉnh kề của nó theo đường thẳng đứng chính là O và tương tự thay V bằng đỉnh kề của nó theo đường thẳng đứng chính là K.

Tương tự nếu digram cần thay thế là VN thì chuỗi thay thế là HO. Nếu các kí tự của digram nằm trên hàng ngang thì chuỗi thay thế là các kí tự bên phải của chúng. Chẳng hạn nếu digram là WU thì chuỗi thay thế là SP, nếu digram là FP thì chuỗi thay thế là UW, nếu digram là XR thì chuỗi thay thế là LK. Tương tự nếu các kí tự của digram nằm trên hàng dọc thì chuỗi thay thế là các kí tự bên dưới của chúng. Chẳng hạn nếu digram là SO thì chuỗi thay thế là AN, nếu digram là MR thì chuỗi thay thế là DI, nếu digram là GH thì chuỗi thay thế là UG. Trong trường hợp digram là một cặp kí tự giống nhau chẳng hạn OO hoặc là một kí tự được đi kèm một

khoảng trắng chẳng hạn B□ thì có nhiều cách xử lý, cách đơn giản nhất là giữ nguyên không biến đổi digram này.

2.6 Hệ mật mã công khai

2.6.1 Giới thiệu mật mã với khóa công khai

2.6.1.1 Lịch sử

Mật mã hóa khóa côngkhai là một dạng mật mã hóa cho phép người sử dụng trao đổi các thông tin mật mà không cần phải trao đổi các khóa chung bí mật trước đó. Điều này được thực hiện bằng cách sử dụng một cặp khóa có quan hệ toán học với nhau là khóa công khai và khóa cá nhân (hay khóa bí mật).

Thuật ngữ mật mã hóa khóa bất đối xứng thường được dùng đồng nghĩa với mật mã hóa khóa công khai mặc dù hai khái niệm không hoàn toàn tương đương. Có những thuật toán mật mã khóa bất đối xứng không có tính chất khóa công khai và bí mật như đề cập ở trên mà cả hai khóa (cho mã hóa và giải mã) đều cần phải giữ bí mật.

Trong mật mã hóa khóa công khai, khóa cá nhân phải được giữ bí mật trong khi khóa công khai được phổ biến công khai. Trong 2 khóa, một dùng để mã hóa và khóa còn lại dùng để giải mã. Điều quan trọng đối với hệ thống là không thể tìm ra khóa bí mật nếu chỉ biết khóa công khai.

Hệ thống mật mã hóa khóa công khai có thể sử dụng với các mục đích:

- Mã hóa: giữ bí mật thông tin và chỉ có người có khóa bí mật mới giải mã được.
- Tạo chữ ký số: cho phép kiểm tra một văn bản có phải đã được tạo với một khóa bí mật nào đó hay không.
- Thỏa thuận khóa: cho phép thiết lập khóa dùng để trao đổi thông tin mật giữa 2 bên.

Thông thường, các kỹ thuật mật mã hóa khóa công khai đòi hỏi khối lượng tính toán nhiều hơn các kỹ thuật mã hóa khóa đối xứng nhưng những lợi điểm mà chúng mang lại khiến cho chúng được áp dụng trong nhiều ứng dụng.

Trong hầu hết lịch sử mật mã học, khóa dùng trong các quá trình mã hóa và giải mã phải được giữ bí mật và cần được trao đổi bằng một phương pháp an toàn khác (không dùng mật mã) như gặp nhau trực tiếp hay thông qua một người đưa thư tin cậy. Vì vậy quá trình phân phối khóa trong thực tế gặp rất nhiều khó khăn, đặc biệt là khi số lượng người sử dụng rất lớn. Mật mã hóa khóa công khai đã giải quyết

được vấn đề này vì nó cho phép người dùng gửi thông tin mật trên đường truyền không an toàn mà không cần thỏa thuận khóa từ trước.

Năm 1874, William Stanley Jevons xuất bản một cuốn sách mô tả mối quan hệ giữa các hàm một chiều với mật mã học đồng thời đi sâu vào bài toán phân tích ra thừa số nguyên tố (sử dụng trong thuật toán RSA). Tháng 7 năm 1996, một nhà nghiên cứuđã bình luận về cuốn sách trên như sau:

Trong cuốn *The Principles of Science: A Treatise on Logic and Scientific Method* được xuất bản năm 1890, William S. Jevons đã phát hiện nhiều phép toán rất dễ thực hiện theo một chiều nhưng rất khó theo chiều ngược lại. Một ví dụ đã chứng tỏ mã hóa rất dễ dàng trong khi giải mã thì không. Vẫn trong phần nói trên ở chương 7 (Giới thiệu về phép tính ngược) tác giả đề cập đến nguyên lý: ta có thể dễ dàng nhân các số tự nhiên nhưng phân tích kết quả ra thừa số nguyên tố thì không hề đơn giản. Đây chính là nguyên tắc cơ bản của thuật toán **mật mã hóa khóa công khai** RSA mặc dù tác giả không phải là người phát minh ra mật mã hóa khóa công khai.thông qua một kênh thông tin không an toàn. Kỹ thuật thỏa thuận khóa của Merkle có tên là hệ thống câu đố Merkle.

Thuật toán đầu tiên cũng được Rivest, Shamir và Adleman tìm ra vào năm 1977 tại MIT. Công trình này được công bố vào năm 1978 và thuật toán được đặt tên là RSA. RSA sử dụng phép toán tính hàm mũ môđun (môđun được tính bằng tích số của 2 số nguyên tố lớn) để mã hóa và giải mã cũng như tạo [chữ ký số]. An toàn của thuật toán được đảm bảo với điều kiện là không tồn tại kỹ thuật hiệu quả để phân tích một số rất lớn thành thừa số nguyên tố.Kể từ thập kỷ 1970, đã có rất nhiều thuật toán mã hóa, tạo chữ ký số, thỏa thuận khóa.. được phát triển. Các thuật toán như ElGamal (mật mã) do Netscape phát triển hay DSA do NSA và NIST cũng dựa trên các bài toán lôgarit rời rạc tương tự như RSA. Vào giữa thập kỷ 1980, Neal Koblitz bắt đầu cho một dòng thuật toán mới: mật mã đường cong elliptic và cũng tạo ra nhiều thuật toán tương tự. Mặc dù cơ sở toán học của dòng thuật toán này phức tạp hơn nhưng lại giúp làm giảm khối lượng tính toán đặc biệt khi khóa có độ dài lớn.

2.6.1.2 Lý thuyết mật mã công khai

Khái niệm về mật mã khóa công khai đã tạo ra sự cố gắng để giải quyết hai vấn đề khó khăn nhất trong mật mã khóa quy ước, đó là sự phân bố khóa và chữ ký số:

- Trong mã quy ước sự phân bố khóa yêu cầu hoặc là hai người truyền thông cùng tham gia một khóa mà bằng cách nào đó đã được phân bố tới họ hoặc sử dụng chung một trung tâm phân bố khóa.
- Nếu việc sử dụng mật mã đã trở nên phổ biến, không chỉ trong quân đội mà còn trong thương mại và những mục đích cá nhân thì những đoạn tin và tài liệu điện tử sẽ cần những chữ ký tương đương đã sử dụng trong các tài liệu giấy. Tức là, một phương pháp có thể được nghĩ ra có quy định làm hài lòng tất cả những người tham gia khi mà một đoạn tin số được gửi bởi một cá nhân đặc biệt hay không

Trong sơ đồ mã hóa quy ước, các khóa được dùng cho mã hóa và giải mã một đoạn tin là giống nhau. Đây là một điều kiện không cần thiết, nó có thể phát triển giải thuật mã hóa dựa trên một khóa cho mã hóa và một khóa khác cho giải mã.

Các bước cần thiết trong quá trình mã hóa công khai:

- Mỗi hệ thống cuối trong mạng tạo ra một cặp khóa để dùng cho mã hóa và giải mã đoạn tin mà nó sẽ nhận.
- Mỗi hệ thống công bố rộng rãi khóa mã hóa bằng cách đặt khóa vào một thanh ghi hay một file công khai, khóa còn lại được giữ riêng
- Nếu A muốn gửi một đoạn tin tới B thì A mã hóa đoạn tin bằng khóa công khai của B
- Khi B nhận đoạn tin mã hóa, nó có thể giải mã bằng khóa bí mật của mình. Không một người nào khác có thể giải mã đoạn tin này bởi vì chỉ có mình B biết khóa bí mật đó thôi .

Việc các tiếp cận này, tất cả những người tham gia có thể truy xuất khóa công khai. Khóa bí mật được tạo bởi từng cá nhân, vì vậy không bao giờ được phân bố. Ở bất kỳ thời điểm nào, hệ thống cũng có thể chuyển đổi cặp khóa để đảm bảo tính bí mật.

Bảng sau tóm tắt một số khía cạnh quan trọng về mã hóa quy ước và mã hóa công khai : để phân biệt được hai loại chúng ta tổng quát hóa liên hệ khóa sử dụng trong mã hóa quy ước là khóa bí mật, hai khóa sử dụng trong mã hóa công khai là khóa công khai và khóa bí mật.

Mã hóa quy ước	**Mã hóa công khai**

* Yêu cầu - Thuật giải tương tự cho mã hóa và giải mã. - Người gửi và người nhận phải tham gia cùng thuật giải và cùng khóa * Tính bảo mật - Khóa phải được bí mật - Không thể hay ít nhất không có tính thực tế để giải mã đoạn tin nếu thông tin khác có sẵn - Kiến thức về thuật giải cộng với các mẫu về mật mã không đủ để xác định khóa	* Yêu cầu - Một thuật giải cho mã hóa và một thuật giải cho giải mã - Người gửi và người nhận, mỗi người phải có cặp khóa riêng của mình * Tính bảo mật - Một trong hai khóa phải được giữ bí mật - Không thể hay ít nhất không có tính thực tế để giải mã đoạn tín nếu thông tin khác không có sẵn - Kiến thức về thuật giải cộng với một trong các khóa, cộng với các mẫu về mật mã không đủ để xác định khóa

2.6.1.3 Những yếu điểm, hạn chế của mật mã với khóa công khai

Tồn tại khả năng một người nào đó có thể tìm ra được khóa bí mật. Không giống với hệ thống mật mã sử dụng một lần (one-time pad) hoặc tương đương, chưa có thuật toán mã hóa khóa bất đối xứng nào được chứng minh là an toàn trước các tấn công dựa trên bản chất toán học của thuật toán. Khả năng một mối quan hệ nào đó giữa 2 khóa hay điểm yếu của thuật toán dẫn tới cho phép giải mã không cần tới khóa hay chỉ cần khóa mã hóa vẫn chưa được loại trừ. An toàn của các thuật toán này đều dựa trên các ước lượng về khối lượng tính toán để giải các bài toán gắn với chúng. Các ước lượng này lại luôn thay đổi tùy thuộc khả năng của máy tính và các phát hiện toán học mới.

Mặc dù vậy, độ an toàn của các thuật toán mật mã hóa khóa công khai cũng tương đối đảm bảo. Nếu thời gian để phá một mã (bằng phương pháp duyệt toàn bộ) được ước lượng là 1000 năm thì thuật toán này hoàn toàn có thể dùng để mã hóa các thông tin về thẻ tín dụng - Rõ ràng là thời gian phá mã lớn hơn nhiều lần thời gian tồn tại của thẻ (vài năm).

Nhiều điểm yếu của một số thuật toán mật mã hóa khóa bất đối xứng đã được tìm ra trong quá khứ. Thuật toán *đóng gói ba lô* là một ví dụ. Nó chỉ được xem là không an toàn khi một dạng tấn công không lường trước bị phát hiện. Gần đây, một số

dạng tấn công đã đơn giản hóa việc tìm khóa giải mã dựa trên việc đo đạc chính xác thời gian mà một hệ thống phần cứng thực hiện mã hóa. Vì vậy, việc sử dụng mã hóa khóa bất đối xứng không thể đảm bảo an toàn tuyệt đối. Đây là một lĩnh vực đang được tích cực nghiên cứu để tìm ra những dạng tấn công mới.

Một điểm yếu tiềm tàng trong việc sử dụng khóa bất đối xứng là khả năng bị tấn công dạng kẻ tấn công đứng giữa (man in the middle attack): kẻ tấn công lợi dụng việc phân phối khóa công khai để thay đổi khóa công khai. Sau khi đã giả mạo được khóa công khai, kẻ tấn công đứng ở giữa 2 bên để nhận các gói tin, giải mã rồi lại mã hóa với khóa đúng và gửi đến nơi nhận để tránh bị phát hiện. Dạng tấn công kiểu này có thể phòng ngừa bằng các phương pháp trao đổi khóa an toàn nhằm đảm bảo nhận thực người gửi và toàn vẹn thông tin. Một điều cần lưu ý là khi các chính phủ quan tâm đến dạng tấn công này: họ có thể thuyết phục (hay bắt buộc) nhà cung cấp chứng thực số xác nhận một khóa giả mạo và có thể đọc các thông tin mã hóa.

2.6.1.4 Ứng dụng của mật mã

a.Bảo mật

Ứng dụng rõ ràng nhất của mật mã hóa khóa công khai là bảo mật: một văn bản được mã hóa bằng khóa công khai của một người sử dụng thì chỉ có thể giải mã với khóa bí mật của người đó.

Phần mềm PGP miễn phí chỉ được sử dụng cho người dùng cá nhân với mục đích phi thương mại, có thể tải về tại địa chỉ :
http://www.pgp.com/products/freeware.html

b.Chứng thực

Các thuật toán tạo chữ ký số khóa công khai có thể dùng để nhận thực. Một người sử dụng có thể mã hóa văn bản với khóa bí mật của mình. Nếu một người khác có thể giải mã với khóa công khai của người gửi thì có thể tin rằng văn bản thực sự xuất phát từ người gắn với khóa công khai đó. Dùng chữ ký số cho email và mã hóa email khi gửi đi thông qua nhà cung cấp chứng chỉ số làm trọng tài điều khiển

Nhà chứng chỉ số của nhà cung cấp Thawte(www.thawte.com) cho phép bạn có thể đăng ký cho mình một tài khoản Personal Email Certificate hoàn toàn miễn phí tại đây để thực hiện giao dịch khi gởi và nhận mail

(http://www.thawte.com/secure-email/personal-email-certificates/index.htm)

c.Ứng dụng trong thương mại điện tử

Nhiều đơn vị, tổ chức ở Việt Nam đã đang xây dựng mạng máy tính có quy mô lớn phục vụ cho công việc kinh doanh của mình: mạng chứng khoán, mạng ngân hàng, mạng bán vé tàu xe, kê khai và nộp thuế qua mạng….

Công ty phần mềm và Truyền thông VASC đã chính thức ký kết hợp đồng "ứng dụng chứng chỉ số trong giao dịch ngân hàng điện tử" với ngân hàng cổ phần thương mại Á Châu (ACB) từ ngày 30/9/2003, cho phép khách hàng ACB sẽ giao dịch trực tuyến trên mạng với chữ ký điện tử do VASC cấp.

Mạng giao dịch chứng khoán VCBS (http://www.vebs.vn) : mở tài khoản ngân hàng cho phép giao dịch trực tiếp qua sàn, báo giá cổ phiếu, cho phép đặt lệnh mua bán cổ phần chỉ bằng thao tác click chuột. Mạng ngân hàng VCB, EAB (http://www.vietcombank.com.vn, http://ebanking.dongabank.com.vn) cho phép xem số dư, chuyển khoản cho tài khoản khác cùng hệ thống từ 20-500 triệu đồng mỗi ngày, bản kê chi tiết giao dịch của tài khoản trên Internet.

Hệ thống bán vé qua mạng của ngành hàng không (http://www.pacificairline.com.vn), đường sắt (http://www.vr.com.vn) đã triển khai 1/2007, mua bán trực tuyến (http://www.ebay.vn).

Chi cục thuế thành phố Hồ Chí Minh (http://www.hcmtax.gov.vn) đang thử nghiệm cho phép doanh nghiệp đăng ký tự in hóa đơn theo mẫu, tự kê khai báo cáo thuế, khấu trừ thuế qua mạng…

Nếu như có được một cơ chế bảo mật tốt, đảm bảo xác thực rõ ràng giữa các bên tham gia vào hệ thống thì chắc chắn rằng những vấn đề liên quan đến mạng máy tính nêu trên chỉ còn là vấn đề thời gian.

2.7 Hệ mật RSA

Trong mật mã học, **RSA** là một thuật toán mật mã hóa khóa công khai. Đây là thuật toán đầu tiên phù hợp với việc tạo ra chữ ký điện tử đồng thời với việc mã hóa. Nó đánh dấu một sự tiến bộ vượt bậc của lĩnh vực mật mã học trong việc sử dụng khóa công cộng. RSA đang được sử dụng phổ biến trong thương mại điện tử và được cho là đảm bảo an toàn với điều kiện độ dài khóa đủ lớn.

2.7.1 Lịch sử

Thuật toán được Ron Rivest, Adi Shamir và Len Adleman mô tả lần đầu tiên vào năm 1977 tại Học viện Công nghệ Massachusetts (MIT). Tên của thuật toán lấy từ 3 chữ cái đầu của tên 3 tác giả.Trước đó, vào năm 1973, Clifford Cocks, một nhà toán học người Anh làm việc tại GCHQ, đã mô tả một thuật toán tương tự. Với khả

năng tính toán tại thời điểm đó thì thuật toán này không khả thi và chưa bao giờ được thực nghiệm. Tuy nhiên, phát minh này chỉ được công bố vào năm 1997 vì được xếp vào loại tuyệt mật.

Thuật toán RSA được MIT đăng ký bằng sáng chế tại Hoa Kỳ vào năm 1983 (Số đăng ký 4,405,829). Bằng sáng chế này hết hạn vào ngày 21 tháng 9 năm 2000. Tuy nhiên, do thuật toán đã được công bố trước khi có đăng ký bảo hộ nên sự bảo hộ hầu như không có giá trị bên ngoài Hoa Kỳ. Ngoài ra, nếu như công trình của Clifford Cocks đã được công bố trước đó thì bằng sáng chế RSA đã không thể được đăng ký.

2.7.2 Mô tả thuật toán

Thuật toán RSA có hai khóa: khóa công khai (hay khóa công cộng) và khóa bí mật (hay khóa cá nhân). Mỗi khóa là những số cố định sử dụng trong quá trình mã hóa và giải mã. Khóa công khai được công bố rộng rãi cho mọi người và được dùng để mã hóa. Những thông tin được mã hóa bằng khóa công khai chỉ có thể được giải mã bằng khóa bí mật tương ứng. Nói cách khác, mọi người đều có thể mã hóa nhưng chỉ có người biết khóa cá nhân (bí mật) mới có thể giải mã được.

Ta có thể mô phỏng trực quan một hệ mật mã khoá công khai như sau : Bob muốn gửi cho Alice một thông tin mật mà Bob muốn duy nhất Alice có thể đọc được. Để làm được điều này, Alice gửi cho Bob một chiếc hộp có khóa đã mở sẵn và giữ lại chìa khóa. Bob nhận chiếc hộp, cho vào đó một tờ giấy viết thư bình thường và khóa lại (như loại khoá thông thường chỉ cần sập chốt lại, sau khi sập chốt khóa ngay cả Bob cũng không thể mở lại được-không đọc lại hay sửa thông tin trong thư được nữa). Sau đó Bob gửi chiếc hộp lại cho Alice. Alice mở hộp với chìa khóa của mình và đọc thông tin trong thư. Trong ví dụ này, chiếc hộp với khóa mở đóng vai trò khóa công khai, chiếc chìa khóa chính là khóa bí mật.

a. Tạo khóa

Giả sử Alice và Bob cần trao đổi thông tin bí mật thông qua một kênh không an toàn (ví dụ như Internet). Với thuật toán RSA, Alice đầu tiên cần tạo ra cho mình cặp khóa gồm khóa công khai và khóa bí mật theo các bước sau:

1. Chọn 2 số nguyên tố lớn p và q với $p \neq q$, lựa chọn ngẫu nhiên và độc lập.
2. Tính: $n = pq$
3. Tính: giá trị hàm số Ơle $\phi(n) = (p-1)(q-1)$.

4. Chọn một số tự nhiên *e* sao cho $1 < e < \phi(n)$ và là số nguyên tố cùng nhau với $\phi(n)$.
5. Tính: *d* sao cho $de \equiv 1 \pmod{\phi(n)}$.

Một số lưu ý:

- Các số nguyên tố thường được chọn bằng phương pháp thử xác suất.
- Các bước 4 và 5 có thể được thực hiện bằng giải thuật Euclid mở rộng (xem thêm: số học môđun).
- Bước 5 có thể viết cách khác: Tìm số tự nhiên x sao cho $d = \frac{x(p-1)(q-1)+1}{e}$ cũng là số tự nhiên. Khi đó sử dụng giá trị $d \mod (p-1)(q-1)$.
- Từ bước 3, PKCS#1 v2.1 sử dụng $\lambda = LCM(p-1, q-1)$ thay cho $\phi = (p-1)(q-1)$).

Khóa công khai bao gồm:

- *n*, môđun.
- *e*, số mũ công khai (cũng gọi là *số mũ mã hóa*).

Khóa bí mật bao gồm:

- *n*, môđun, xuất hiện cả trong khóa công khai và khóa bí mật, và
- *d*, số mũ bí mật (cũng gọi là *số mũ giải mã*).

Một dạng khác của khóa bí mật bao gồm:

- *p* and *q*, hai số nguyên tố chọn ban đầu,
- *d mod (p-1)* và *d mod (q-1)* (thường được gọi là *dmp1* và *dmq1*),
- *(1/q) mod p* (thường được gọi là *iqmp*)

Dạng này cho phép thực hiện giải mã và ký nhanh hơn với việc sử dụng định lý số dư Trung Quốc (tiếng Anh: *Chinese Remainder Theorem* - CRT). Ở dạng này, tất cả thành phần của khóa bí mật phải được giữ bí mật.

Alice gửi khóa công khai cho Bob, và giữ bí mật khóa cá nhân của mình. Ở đây, *p* và *q* giữ vai trò rất quan trọng. Chúng là các phân tố của n và cho phép tính *d* khi

biết *e*. Nếu không sử dụng dạng sau của khóa bí mật (dạng CRT) thì *p* và *q* sẽ được xóa ngay sau khi thực hiện xong quá trình tạo khóa.

b. Mã hóa

Giả sử Bob muốn gửi đoạn thông tin *M* cho Alice. Đầu tiên Bob chuyển *M* thành một số *m*<*n* theo một hàm có thể đảo ngược (từ *m* có thể xác định lại *M*) được thỏa thuận trước. Quá trình này được mô tả ở phần sau

Lúc này Bob có *m* và biết *n* cũng như *e* do Alice gửi. Bob sẽ tính *c* là bản mã hóa của *m* theo công thức:

$$c = m^e \mod n$$

Hàm trên có thể tính dễ dàng sử dụng phương pháp tính hàm mũ (theo môđun) bằng thuật toán bình phương và nhân. Cuối cùng Bob gửi *c* cho Alice.

c. Giải mã

Alice nhận *c* từ Bob và biết khóa bí mật *d*. Alice có thể tìm được *m* từ *c* theo công thức sau:

$$m = c^d \mod n$$

Biết *m*, Alice tìm lại *M* theo phương pháp đã thỏa thuận trước. Quá trình giải mã hoạt động vì ta có

$$c^d \equiv (m^e)^d \equiv m^{ed} \pmod{n}.$$

Do $ed \equiv 1 \pmod{p-1}$ và $ed \equiv 1 \pmod{q-1}$, (*theo Định lý Fermat nhỏ*) nên:

$$m^{ed} \equiv m \pmod{p}$$

và

$$m^{ed} \equiv m \pmod{q}$$

Do *p* và *q* là hai số nguyên tố cùng nhau nên ta có:

$$m^{ed} \equiv m \pmod{pq}.$$

hay:

$$c^d \equiv m \pmod{n}.$$

Ví dụ:

Sau đây là một ví dụ với những số cụ thể. Ở đây chúng ta sử dụng những số nhỏ để tiện tính toán còn trong thực tế phải dùng các số có giá trị đủ lớn.

Lấy:

$P = 61$ — số nguyên tố thứ nhất (giữ bí mật hoặc hủy sau khi tạo khóa)

$Q = 53$ — số nguyên tố thứ hai (giữ bí mật hoặc hủy sau khi tạo khóa)

$N = pq = 3233$ — môđun (công bố công khai)

$E = 17$ — số mũ công khai

$D = 2753$ — số mũ bí mật

Khóa công khai là cặp (e, n). Khóa bí mật là d. Hàm mã hóa là:

$$\text{encrypt}(m) = m^e \bmod n = m^{17} \bmod 3233$$

với m là văn bản rõ. Hàm giải mã là:

$$\text{decrypt}(c) = c^d \bmod n = c^{2753} \bmod 3233$$

với c là văn bản mã.

Để mã hóa văn bản có giá trị 123, ta thực hiện phép tính:

$$\text{encrypt}(123) = 123^{17} \bmod 3233 = 855$$

Để giải mã văn bản có giá trị 855, ta thực hiện phép tính:

$$\text{decrypt}(855) = 855^{2753} \bmod 3233 = 123$$

Cả hai phép tính trên đều có thể được thực hiện hiệu quả nhờ giải thuật bình phương và nhân.

2.7.3 Tốc độ mã hóa RSA

Tốc độ và hiệu quả của nhiều phần mềm thương mại có sẵn và công cụ phần cứng của RSA đang gia tăng một cách nhanh chóng. Việc Pentium 90Mhz, bộ toolkit BSAFE 3.0 của cơ quan bảo mật dữ liệu RSA đạt tốc độ tính khóa bí mật là 21,6 Kbps với khóa 512 bit và 7,4 Kbps với khóa 1024 bit. Phần cứng RSA nhanh

nhất đạy 300 Kbps với khóa 512 bit, nếu được xử lý song song thì đạt 600 Kbps với khóa 512 bit và 185 Kbps với khóa 970 bit.

So sánh với giải thuật DES và các giải thuật mã khối khác thì RSA chậm hơn: về phần mềm DES nhanh hơn RSA 100 lần, về phần cứng DES nhanh hơn RSA từ 1000 tới 10000 lần tùy thuộc công cụ (implementation) sử dụng

(thông tin này được lấy từ http://www.rsa.com)

Kích thước của khóa trong RSA:

Hiệu quả của một hệ thống mật mã khóa bất đối xứng phụ thuộc vào *độ khó* (lý thuyết hoặc tính toán) của một vấn đề toán học nào đó chẳng hạn như bài toán phân tích ra thừa số nguyên tố. Giải các bài toán này thường mất nhiều thời gian nhưng thông thường vẫn nhanh hơn là thử lần lượt từng khóa theo kiểu duyệt toàn bộ. Vì thế, khóa dùng trong các hệ thống này cần phải dài hơn trong các hệ thống mật mã khóa đối xứng. Tại thời điểm năm 2002, độ dài 1024 bít được xem là giá trị tối thiểu cho hệ thống sử dụng thuật toán RSA.

Năm 2003, công ty RSA Security cho rằng khóa RSA 1024 bít có độ an toàn tương đương với khóa 80 bít, khóa RSA 2048 bít tương đương với khóa 112 bít và khóa RSA 3072 bít tương đương với khóa 128 bít của hệ thống mật mã khóa đối xứng. Họ cũng đánh giá rằng, khóa 1024 bít có thể bị phá vỡ trong khoảng từ 2006 tới 2010 và khóa 2048 bít sẽ an toàn tới 2030. Các khóa 3072 bít cần được sử dụng trong trường hợp thông tin cần giữ bí mật sau 2030. Các hướng dẫn về quản lý khóa của NIST cũng gợi ý rằng khóa RSA 15360 bít có độ an toàn tương đương với khóa đối xứng 256 bít.

Một dạng khác của thuật toán mật mã hóa khóa bất đối xứng, mật mã đường cong elliptic (ECC), tỏ ra an toàn với khóa ngắn hơn khá nhiều so với các thuật toán khác. Hướng dẫn của NIST cho rằng khóa của ECC chỉ cần dài gấp đôi khóa của hệ thống khóa đối xứng. Giả định này đúng trong trường hợp không có những đột phá trong việc giải các bài toán mà ECC đangsử dụng. Một văn bản mã hóa bằng ECC với khóa 109 bít đã bị phá vỡ bằng cách tấn công duyệt toàn bộ.

Tùy thuộc vào kích thước bảo mật của mỗi người và thời gian sống của khóa mà khóa có chiều dài thích hợp

- loại Export 512 bit
- loại Person 768 bit
- loại Commercial 1024 bit

- loại Militery 2048 bit

Chu kỳ sống của khóa phụ thuộc vào

- việc đăng ký và tạo khóa
- việc phân bố khóa
- việc kích hoạt và không kích hoạt khóa
- việc thay thế hoặc cập nhật khóa
- việc hủy bỏ khóa
- việc kết thúc khóa bao gồm sự phá hoại hoặc sự lưu trữ

2.7.4 Độ an toàn của RSA

Độ an toàn của hệ thống RSA dựa trên 2 vấn đề của toán học: bài toán phân tích ra thừa số nguyên tố các số nguyên lớn và bài toán RSA. Nếu 2 bài toán trên là khó (không tìm được thuật toán hiệu quả để giải chúng) thì không thể thực hiện được việc phá mã toàn bộ đối với RSA. Phá mã một phần phải được ngăn chặn bằng các phương pháp chuyển đổi bản rõ an toàn.

Bài toán RSA là bài toán tính căn bậc e môđun n (với n là hợp số): tìm số m sao cho $m^e = c$ mod n, trong đó (e, n) chính là khóa công khai và c là bản mã. Hiện nay phương pháp triển vọng nhất giải bài toán này là phân tích n ra thừa số nguyên tố. Khi thực hiện được điều này, kẻ tấn công sẽ tìm ra số mũ bí mật d từ khóa công khai và có thể giải mã theo đúng quy trình của thuật toán. Nếu kẻ tấn công tìm được 2 số nguyên tố p và q sao cho: $n = pq$ thì có thể dễ dàng tìm được giá trị $(p-1)(q-1)$ và qua đó xác định d từ e. Chưa có một phương pháp nào được tìm ra trên máy tính để giải bài toán này trong thời gian đa thức (*polynomial-time*). Tuy nhiên người ta cũng chưa chứng minh được điều ngược lại (sự không tồn tại của thuật toán).

Tại thời điểm năm 2005, số lớn nhất có thể được phân tích ra thừa số nguyên tố có độ dài 663 bít với phương pháp phân tán trong khi khóa của RSA có độ dài từ 1024 tới 2048 bít. Một số chuyên gia cho rằng khóa 1024 bít có thể sớm bị phá vỡ (cũng có nhiều người phản đối việc này). Với khóa 4096 bít thì hầu như không có khả năng bị phá vỡ trong tương lai gần. Do đó, người ta thường cho rằng RSA đảm bảo an toàn với điều kiện n được chọn đủ lớn. Nếu n có độ dài 256 bít hoặc ngắn hơn, nó có thể bị phân tích trong vài giờ với máy tính cá nhân dùng các phần mềm có sẵn. Nếu n có độ dài 512 bít, nó có thể bị phân tích bởi vài trăm máy tính tại thời điểm năm 1999. Một thiết bị lý thuyết có tên là TWIRL do Shamir và Tromer mô tả

năm 2003 đã đặt ra câu hỏi về độ an toàn của khóa 1024 bít. Vì vậy hiện nay người ta khuyến cáo sử dụng khóa có độ dài tối thiểu 2048 bít.

Năm 1993, Peter Shor công bố thuật toán Shor chỉ ra rằng: máy tính lượng tử (trên lý thuyết) có thể giải bài toán phân tích ra thừa số trong thời gian đa thức. Tuy nhiên, máy tính lượng tử vẫn chưa thể phát triển được tới mức độ này trong nhiều năm nữa.

Vì khóa là khóa công khai nên người giải mã thường dựa vào cặp khóa này để tìm cặp khóa bí mật. Điều quan trọng là dựa vào n để tính hai thừa số p,q của n từ đó tính được d. Có nhiều giải thuật như thế, đàu tiên ta xét trường hợp đơn giản nhất là người giải mã biết được $\phi(n)$. Khi đó tính p,q đưa về việc giải hai phương trình sau:

$$n = p.\ q$$

$$\phi(n) = (p - 1)(q - 1)$$

Thay q= n/p ta được phương trình bậc hai:

$$p^2 - (n - \phi(n) + 1)p + n = 0$$

Hai nghiệm của phương trình bậc hai sẽ là p,q. tuy nhiên vấn đề có được $\phi(n)$ còn khó hơn tính hai thừa số nhiều

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

Có một vài điều cần lưu ý khi chọn các số p,q để tránh rơi vào trường hợp tích hợp của pq bị phân tích nhanh nhờ những thuật toán đặc biệt: p và q cần chọn sao cho p-1 và q-1 không chỉ có toàn ước nguyên tố nhỏ. Ngoài ra, UCLN(p-1,q-1) phải là số nhỏ, p và q phải có chữ số trong khai triển thập phân khác nhau không nhiều.

Một nhận định chung là tất cả các cuộc tấn công giải mã đều mang mục đích không tốt. Tính bảo mật của RSA chủ yếu dựa vào việc giữ bí mật khóa giải mã hay giữ bí mật các thừa số p,q của n. Ta thử xét một vài phương thức tấn công điển hình của kẻ địch nhằm giải mã trong thuật toán này(nhằm xâm phạm tới các yếu tố bí mật đó).

- Trường hợp 1: Chúng ta xét đến trường hợp khi kẻ địch nào đó biết được modulo n, khóa công khai K_B và bản tin mã hóa C, khi đó kẻ địch sẽ tìm ra bản tin gốc (plaintext) như thế nào. Để làm được điều đó kẻ địch thường tấn công vào hệ thống mật mã bằng hai phương thức sau đây:

- phương thức thứ nhất: Trước tiên dựa vào phân tích thừa sô modulo n. Tếp theo sau chúng sẽ tìm cách tính otans ra hai thừa số p,q và có khả năng thành công khi đó sẽ tính được $\phi(n)=(p-1)(q-1)$ và khóa bí mật K_B. Ta thấy n cần phải là tích của hai số nguyên tố, vì nếu n là tích của hai số nguyên tố thì thuật toán phân tích thừa số đơn giản cần tối đa $n^{1/2}$ bước, bởi vì có một số nguyên tố nhỏ hơn $n^{1/2}$. Mặt khác, nêu sn là tích của n số nguyên tố thì thuật phân tích thừa sô đơn giản cần $n^{1/n}$ bước.
- Phương thức thứ hai: phương thức tấn công thứ hai vào hệ mã hóa RSA là có thể khởi đầu bằng cách giải quyết trường hợp thích hợp của bàn toán logarit rời rạc. Trường hợp này kẻ địch đã có trong tay bản mã C và khóa công khai K_B tức là cặp (K_B,C)

- Trường hợp 2: Chúng ta xét trường hợp khi kẻ địch biết được modul n và $\phi(n)$, khi đó kẻ địch sẽ tìm ra bản gốc (plaintext) bằng cách sau:

Biết $\phi(n)$ thì có thể tính p,q theo hệ phương trình:

$$pq=n,\ (p-1)(q-1)=\phi(n)$$

do đó p,q là hai nghiệm của phương trình bậc hai:

$$p^2-(n-\phi(n)+1)p+n=0$$

Ví dụ n=84773093 và biết $\phi(n)$ =84754668. Giải phương trình bậc hai tương ứng ta sẽ được hai nghiệm p=9539, q=8887.

2.7.5 Sự che dấu thông tin trong hệ thống RSA

Hệ thống RSA có một đặc điểm đặc trưng là thông tin không phải luôn luôn được che dấu. Giả sử người gửi có e= 17, n=35. Nếu anh ta muốn gửi bất cứ data nào thuộc tập sau:

{1,6,7,8,13,14,15,20,21,22,27,28,29,34}

Thì mọi mật mã cũng chính là data ban đầu. Nghĩa là: $M=M^e \bmod n$. Còn khi p=109, q=97, e= 865 thì hệ thống hoàn toàn không có sự che dấu thông tin bởi vì: $M=M^e \bmod (109*97)$ với mọi M

Với mỗi modul n, không che dấu được ít nhất 9 message:

$$M=M^e \bmod n(1)$$

Hay $M=M^e \bmod p$ và $M=M^e \bmod q(2)$

Với mỗi e,(2) có ít nhất 3 giải pháp thuộc tập {-1,0,1}. Do đó tất cả message thỏa (1) là: {M=[M(mod p), M(mod q)] | M(mod p), M(mod q)∈{-1,0,1}}

Để xác định chính xác số message không được che dấu (không bị thay đổi sau khi mã hóa) ta sử dụng định lý sau: "Nếu các message được mã hóa trong hệ thống RSA được xác định bởi số modul n=pq(p,q, là số nguyên tố) và khóa công khai e thì có: m=[1+UCLN(e-1,p-1)][1+UCLN(e-1,q-1)] message không bị che dấu ".

Một số lưu ý khi sử dụng hệ mật mã RSA:

Mọi người đều biết "điểm mạnh " của hệ mã với chìa khóa công khai RSA là dựa trên "điểm yếu" của máy tính trong việc phân tích một số nguyên (đủ lớn) ra các thừa số nguyên tố. Với thời gian hơn 20 năm tồn tại trên via trò một hệ mã công kahi thông dụng nhất, RSA dã đương đầu với các kiểu tấn công đủ loại của giới thám mã chuyên nghiệp. Kết quả hơn 20 năm "công phá" hệ mã RSA của các nhà thám mã đã được tóm lược trong bài báo của Dan Boneh với tiêu đề "Hai mươi năm tấn công hệ mã RSA" (đăng trong tờ Notice ò the AMS, tháng 2-1999), trong đó cho thấy rõ RSA có thể bị "bẻ" khi người ta không biết dùng nó một cách "bài bản". Khi chìa khóa lập mã hoặc giải mã là một số nguyên tố nhỏ thì người ta có những giải pháp "bẻ" RSA một cách không mấy khó khăn. Thêm vào đó, không phải mọi hợp số lớn đều khó phân tích(kể cả khi nó là tích của 2 số nguyên tố rất lớn), cho nên việc chọn các số nguyên tố p,q phải rất thận trọng.

Gần đây người ta có đề cập đến khả năng phá hệ mã RSA bằng các máy tính đặc biệt với các con chip đặc chủng(chuyên dụng cho việc phân tích số) hoặc dùng thuật toán song song. Mặc dù hứa hẹn những tiến bộ vượt bậc nhưng khả năng này vẫn chưa trở thành hiện thực trong tương lai gần, nhất là chuẩn của RSA được nâng cao thêm một bậc

Trong các hệ mã RSA, một bản tin có thể được mã hóa trong thời gian tuyến tính

Đối với các bản tin dài, độ dài của các số được dùng cho các khóa có thể được coi như là hằng. Tương tự như vậy, nâng một số lên lũy thừa được thực hiện trong thời gian hằng, các số không được phép dài hơn một độ dài hằng. Thực ra tham số này che dấu nhiều chi tiết cài đặt có liên quan đến việc tính toán với các con số dài, chi phí của các phép toán thực sự là một yếu tố ngăn cản sự phổ biến ứng dụng của phương pháp này. Phần quan trọng nhất của việc tính toán có liên quan đến việc mã hóa bản tin. Nhưng chắc chắn là sẽ không có hệ mã hóa nào hết nếu không tính được các khóa của chúng là các số lớn.

- Các khóa cho hệ mã hóa RSA có thể được tạo ra mà không phải tính toán quá nhiều.

Một lần nữa ta nói đến các phương pháp kiểm tra số nguyên tố. Mỗi số nguyên tố lớn có thể được phát sinh băng cách đầu tiên tạo ra một số ngẫu nhiên lớn, sau đó kiểm tra các số kế tiếp cho tới khi tìm được một số nguyên tố. Một phương pháp đơn giản thực hiện một phép tính trên một con số ngẫu nhiên, với xác suất ½ sẽ chứng minh rằng số được kiểm tra không phải nguyên tố. Bước cuối cùng tính p dựa vào thuật toán Euclid

Như phần trên đã trình bày trong hệ mã hóa công khai thì khóa giải mã (private key) k_B và các thừa số p,q là được giữ bí mật và sự thành công của phương pháp là tùy thuộc vào kẻ địch có khả năng tìm ra được giá trị của k_B hay không nếu cho trước n và K_B .Rất khó có thể tìm ra được k_B từ K_B, cần biết về p, q.Như vậy cần phân tích n ra thành thừa số để tính p,q. Nhưng việc tính phân tích ra thừa số là một việc làm tốn rất nhiều thời gian, với kỹ thuật hiện đại ngày nay thì cần tới hang triệu năm để phan tích một số có 200 chữ số ra thừa số.

Độ an toàn của thuật toán RSA dựa trên cơ sở những khó khăn của việc xác định các thừa số nguyên tố của một số lớn. Bảng dưới đây cho biết các thời gian dự đoán, giả sử rằng mỗi phép toán thực hiện trong một micro giây

Số các chữ số trong số được phân tích	Thời gian phân tích
50	4 giờ
75	104 giờ
100	74 năm
200	4.000.000 năm
300	$5x\ 10^{15}$ năm
500	$4x\ 10^{25}$ năm

2.8 Hệ mật Rabin

Hệ thống mã hoá Rabin: có thể xem như gần gũi với RSA, mặc dù nó có quá trình giải mã khác. Điều thú vị là sự phá mã của Rabin tương với việc phân tích thừa số.

Rabin sử dụng lũa thừa của 2 (hay bất kì một số tự nhiên nào) thay thế cho các số nguyên tố như trong RSA. Điều này dẫn tới 2 kết quả sau: Trước tiên, hệ thống mã hoá Rabin tương đương với việc phân tích thừa số, thứ 2 việc giải mã trở nên khó khăn hơn, ít ra là về cảm giác. Vấn đề tiếp theo là làm sao để biết đầu ra của tiến trình giải mã là đúng.

2.8.1 Mô tả giải thuật Rabin

a.Tạo khóa

Mỗi đầu tạo một khóa công khai và một khóa bí mật tương ứng theo các bước sau:

(1) Tạo hai số nguyên lớn, ngẫu nhiên và phân biệt p và q có kích thước xấp xỉ nhau

(2) Tính n=pq

(3) Khóa công khai là n, khóa bí mật là cặp số (p,q)

b. Mã hóa

A phải thực hiện các bước sau:

- Nhận khóa công khai của B: n.

- Biểu thị bản tin dưới dạng một số nguyên m nằm trong dải [0,n-1].
- Tính $c=m^2 \bmod n$.
- Gửi bản mã c cho B.

c. Giải mã

Để khôi phục bản rõ m từ c, B phải thực hiện các bước sau: Tìm 4 căn bậc hai của c mod n là m_1, m_2, m_3 hoặc m_4.

Thông báo cho người gửi là một trong 4 giá trị m_1, m_2, m_3 hoặc m_4 , bằng một cách nào đó B sẽ quyết định m là giá trị nào.

Ví dụ:

Tạo khóa:B chọn các số nguyên tố p=277 và q=331. B tính n=277*331= 91687.Khóa công khai của B là 91687. Khóa bí mật của A là cặp số (p=277, q=331)

- Mã hóa:Giả sử 6 bit cuối cùng của bản gốc được lặp lại trước khi thực hiện mã hóa. Việc thêm vào các bit thừa này nhằm giúp cho bên giải mã nhận biết được bản mã đúng

Để mã hóa bản tin 10 bit m=1001111001, A sẽ lặp lại 6 bit cuối cùng của m để có được bản tin 16 bit sau: m=1001111001111001, biểu diễn thập phân tương ứng là m=40569

Sau đó A tính $c = m^2 \bmod n = 40569^2 \bmod 91687 = 62111$ rồi gửi c cho B

- Giải mã:Để giải mã bản mã c, B tính bốn giá trị căn bậc hai của c mod n:

$m_1 = 69654$, $m_2 = 220033$, $m_3 = 40596$, $m_4 = 51118$

Biểu diễn nhị phân tương ứng của các số trên là :

$m_1 = 10001000000010110$, $m_2 = 101011000010001$,

$m_3 = 1001111001111001$, $m_4 = 1100011110101110$

vì chỉ có m_3 mới có độ thừa cần thiết nên B sẽ giải mã c bằng m_3 và khôi phục bản tin gốc là m = 1001111001

2.8.2 Đánh giá hiệu quả

Thuật giải mã hóa Rabin là một thuật toán cực kỳ nhanh vì nó chỉ cần thực hiện một phép bình phương modulo đơn giản. Trong khi đó, chẳng hạn với thuật toán RSA có e = 3 phải cần tới một phép nhân modulo và một phép bình phương

modulo. Thuật toán giải mã Rabin có chậm hơn thuật toán mã hóa, tuy nhiên về mặt tốc độ nó cũng tương đương với thuật toán giải mã RSA.

CHƯƠNG 3: CHỮ KÝ ĐIỆN TỬ

Hằng ngày, chúng ta vẫn thường hay dùng chữ ký để xác minh một vấn đề. Chẳng hạn như tên một bức điện nhận tiền từ ngân hàng, hay những hợp đồng ký kết mua bán, chuyển nhượng… Những chữ ký đó là chữ ký viết tay. Những yếu tố nào đã làm nên "sức thuyết phục" của nó ?

Về mặt lý tưởng:

- Chữ ký là bằng chứng thể hiện người ký có chủ định ký văn bản.
- Chữ ký thể hiện "chủ quyền ", nó làm cho người nhận văn bản biết rằng ai đích thị là người đã ký văn bản.
- Chữ ký không thể "tái sử dụng được" , tức là nó là phần của văn bản mà.
- không thể sao chép sang văn bản khác.
- Văn bản đã ký không thể thay đổi được.
- Chữ ký không thể giải mạo và cũng là thứ không thể chối bỏ.

Trong cuộc sống, mọi thứ không diễn ra theo đúng "mô hình lý tưởng" nêu trên, nhưng với khả năng kiểm định sát sao thì việc làm khác đi không phải là dễ. Chúng ta có lý do để mang mô hình này vào thế giới máy tính, nhưng có những khó khăn hiển nhiên: các dòng thông tin trên máy tính được sao chép một cách quá dễ dàng, hình ảnh của chữ ký tay của một người nào đó dù khó bắt chước tới đâu cũng dễ dàng sao chép từ văn bản này sang văn bản khác…

Để có các đặc tính như đã mô tả trên , giao thức ký trong thế giới điện tử cần tới sự hỗ trợ của công nghệ mã hóa. Đó là chữ ký điện tử(*electronic signature*).

Về căn bản, chữ ký điện tử cũng giống như chữ viết tay. Chúng ta dùng nó để xác nhận lời hứa hay cam kết của mình và sau đó không thể rút lại được. Chữ ký điện tử không đòi hỏi phải sử dụng giấy mực, nó gắn đặc điểm nhận dạng của người ký vào một bản cam kết nào đó.Có được một bản chứng nhận điện tử cũng giống như dùng bằng lái xe để xác nhận nhận dạng của mình. Bạn có thể thi lấy được bằng lái xe tại Hà Nội những nó lại cho phép bạn điều khiển phương tiện tại TP HCM. Tương tự như vậy, bản chứng nhận điện tử là vật để khẳng định nhận dạng của bạn trên Internet với những người chấp nhận nó.

Chữ ký điện tử (tiếng anh: *electronic signature*) là thông tin đi kèm theo dữ liệu (văn bản, hình ảnh, video...) nhằm mục đích xác định người chủ của dữ liệu đó.

Ngày nay khi sự phát triển của internet và công nghệ thông tin ngày càng cao. Đã cho phép chúng ta thực hiện những giao dịch điện tử thông qua internet nhưng tính linh hoạt của internet cũng tạo cơ hội cho "bên thứ ba" có thể thực hiện các hành động bất hợp pháp cụ thể là:

- Nghe trộm: Thông tin thì không bị thay đổi nhưng sự bí mật của nó thì không còn. Ví dụ: Thông tin về số thẻ tín dụng, thông tin về trao đổi giao dịch … cần bảo mật.

Giả mạo: Các thông tin trong khi truyền đi bị thay đổi hoặc thay thế trước khi đến với người nhận. Ví dụ: Đơn đặt hàng hay lý lịch cá nhân của một khách hàng …

- Mạo danh: Thông tin được gửi tới một cá nhân mạo nhận là người nhận hợp pháp theo hai hình thức. Hình thức thứ nhất là bắt trước, tức là một cá nhân có thể giả vờ như một người khác như dùng địa chỉ mail của một người khác hoặc giả mạo một tên miền của một trang web. Hình thức thứ hai là xuyên tạc, tức là một cá nhân hay một tổ chức có thể đưa ra những thông tin không đúng sự thật về họ như một trang web mạo nhận chuyên về kinh doanh trang thiết bị nội thất, nhưng thực tế lại là một trang chuyên ăn cắp mã thẻ tín dụng và không bao giờ gửi hàng cho khách.

Do vậy để đảm bảo an toàn trong thương mại điện tử và giao dịch điện tử cần có các hình thức bảo mật có hiệu quả nhất công nghệ phổ biến hiện nay được sử dụng là chữ ký điện tử, chữ ký số và chứng thực điện tử.

Chữ ký điện tử được sử dụng trong các giao dịch điện tử. Xuất phát từ thực tế, chữ ký điện tử cũng cần đảm bảo các chức năng: xác định được người chủ của một dữ liệu nào đó: văn bản, ảnh, video, ... dữ liệu đó có bị thay đổi hay không.

Hai khái niệm chữ ký số (*digital signature*) và chữ ký điện tử (*electronic signature*) thường được dùng thay thế cho nhau mặc dù chúng không hoàn toàn có cùng nghĩa. Chữ ký số chỉ là một tập con của chữ ký điện tử (chữ ký điện tử bao hàm chữ ký số)

3.1 Lịch sử ra đời của chữ ký điện tử

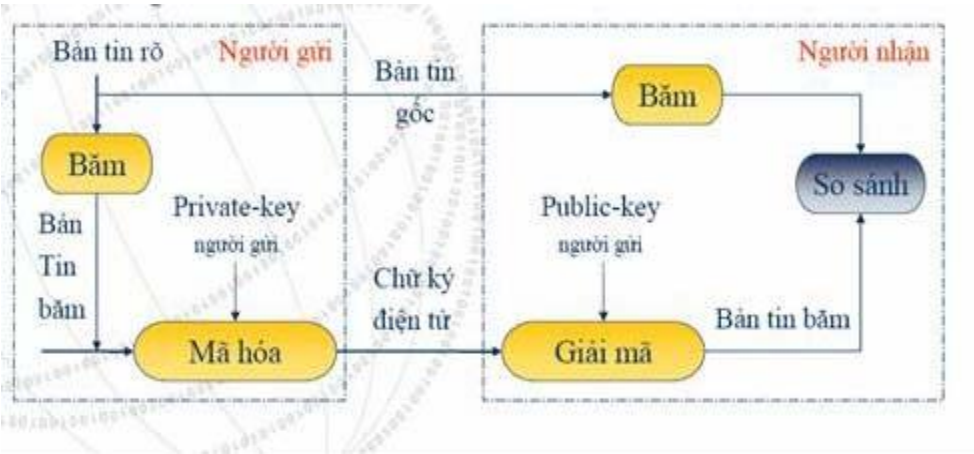
Con người đã sử dụng các hợp đồng dưới dạng điện tử từ hơn 100 năm nay với việc sử dụng mã Morse và điện tín. Vào năm 1889, tòa án tối cao bang New Hampshire (Hoa kỳ) đã phê chuẩn tính hiệu lực của chữ ký điện tử. Tuy nhiên, chỉ

với những phát triển của khoa học kỹ thuật gần đây thì chữ ký điện tử mới đi vào cuộc sống một cách rộng rãi .Vào thập kỷ 1980, các công ty và một số cá nhân bắt đầu sử dụng máy fax để truyền đi các tài liệu quan trọng. Mặc dù chữ ký trên các tài liệu này vẫn thể hiện trên giấy nhưng quá trình truyền và nhận chúng hoàn toàn dựa trên tín hiệu điện tử.Hiện nay, chữ ký điện tử có thể bao hàm các cam kết gửi bằng email, nhập các số định dạng cá nhân (PIN) vào các máy ATM, ký bằng bút điện tử với thiết bị màn hình cảm ứng tại các quầy tính tiền, chấp nhận các điều khoản người dùng (EULA-End User Lisence Agreement) khi cài đặt phần mềm máy tính, ký các hợp đồng điện tử online...

3.2 Khái niệm và mô hình chung của chữ ký điện tử

Chữ ký điện tử là đoạn dữ liệu gắn liền với văn bản gốc để chứng thực tác giả của văn bản và giúp người nhận kiểm tra tính toàn vẹn của văn bản gốc .

Chữ ký điện tử được tạo ra bằng cách áp dụng thuật toán băm một chiều trên văn bản gốc để tạo ra bản ra bản phân tích văn bản (message digest) hay còn gọi là fingerprint, sau đó mã hóa bằng private key tạo ra chữ ký số đính kèm với văn bản gốc để gửi đi. Khi nhận, văn bản được tách làm 2 phần, phần văn bản gốc được tính lại fingerprint để so sánh với fingerprint cũ cũng được phục hồi từ việc giải mã chữ ký số. Như vậy ta có thể xác định được thông điệp bị gửi không bị sửa đổi hay can thiệp trong quá trình gửi.



Mô hình chung cho chữ ký điện tử:

Đặc điểm của chữ ký điện tử rất đa dạng, có thể là một tên hoặc hình ảnh cá nhân kèm theo dữ liệu điện tử, một mã khoá bí mật, hay một dữ liệu sinh trắc học (chẳng hạn như hình ảnh mặt, dấu vân tay, hình ảnh mống mắt...) có khả năng xác thực người gửi.

Độ an toàn của từng dạng là khác nhau .

Quy trình thực hiện chữ ký điện tử:

Các bước mã hóa:

- Dùng giải thuật băm để thay đổi thông điệp cần truyền đi. Kết quả ta được một message digest. Dùng giải thuật **MD5** (Message Digest 5) ta được digest có chiều dài 128-bit, dùng giải thuật **SHA** (Secure Hash Algorithm) ta có chiều dài 160 bit.
- Sử dụng khóa private key của người gửi để mã hóa message digest thu được ở bước 1. Thông thường ở bước này ta dùng giải thuật rsa, kết quả thu được gọi là digital signature của message ban đầu .
- Gộp digital signature vào message ban đầu công việc này gọi là "ký nhận" vào message sau khi đã ký nhận vào message, mọi sự thay đổi trên message sẽ bị phát hiện trong giai đoạn kiểm tra ngoài ra, việc ký nhận này đảm bảo người nhận tin tưởng message này xuất phát từ người gửi chứ không phải ai khác.

1. Các bước kiểm tra:

- Dùng Public key của người gửi (khóa này được thông báo đến mọi người) để giải mã chữ ký số của message .
- Dùng giải thuật MD5 hoặc SHA băm message đính kèm .

So sánh kết quả thu được ở các bước trên . Nếu trùng nhau ta kết luận message này không bị thay đổi trong quá trình truyền và message này là của người gửi.

Chữ ký điện tử sử dụng mã khóa công khai:

Các nhà khoa học Diffie và Hellman đã đề xuất ra phương pháp Ký trên các văn bản điện tử sử dụng hệ mã khoá công khai theo ý tưởng :

- Người gửi ký văn bản sẽ gửi cho người nhận bằng cách mã hoá văn bản đó với mã khoá riêng Private Key của mình sau đó gửi cho người nhận.
- Người nhận sử dụng chìa khoá công khai của người gửi là Public Key để giải mã văn bản mã hoá nhận được.

Theo cách như vậy thì chữ ký điện tử đã đảm bảo được các tính năng của chữ ký viết tay:

- Khẳng định rằng văn bản đó là do người gửi có chủ định ký với khoá riêng của mình.
- Khẳng định chủ nhân của văn bản đó là người có chiếc khoá Private Key đi cùng cặp với Public Key dùng để giải mã văn bản mã hoá tương ứng.
- Chữ ký trên văn bản mã hoá là không thể tái sử dụng vì cho dù có biết Public Key thì cũng không tìm được ra Private Key tương ứng.
- Văn bản đã ký là không thể thay đổi vì nếu văn bản mã hoá đã được giải mã thì không thể mã hoá lại được vì không biết Private Key trước đó.
- Người ký văn bản không thể phủ nhận chữ ký của mình vì chỉ có mình anh ta biết chìa khoá bí mật để mã hoá văn bản đó.

Như vậy mỗi cá nhân khi tham gia vào hệ thống chữ ký điện tử cần phải được cung cấp một bộ khóa (Public key, Private key) dùng để định danh cá nhân đó bởi một tổ chức cơ quan có thẩm quyền và được công nhận trong phạm vi xử dụng.

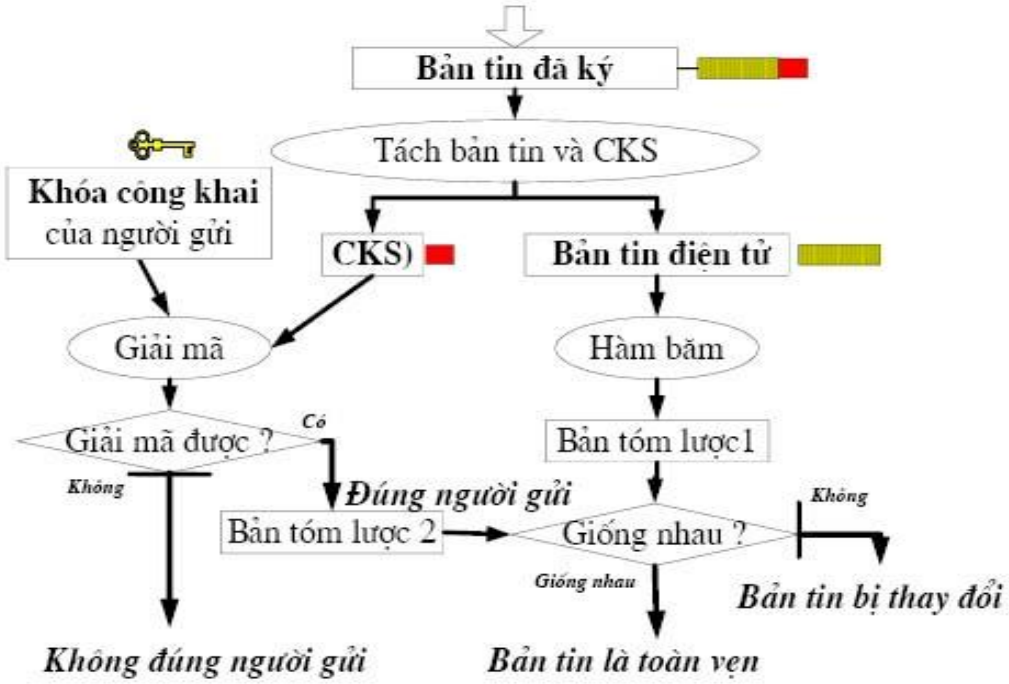
Chữ ký số:

Là hình thức chữ ký điện tử phổ dụng nhất hiện nay. Chữ ký số là một dạng đặc biệt của chữ ký điện tử sử dụng công nghệ khóa công khai PKI (**Public Key Infrastructure**). Trong đó mỗi người tham gia ký cần một cặp khóa bao gồm một khóa công khai và một khóa bí mật. Khóa bí mật dùng để tạo chữ ký số, khóa công khai dùng để thẩm định, xác thực chữ ký số .

Quy trình tạo và kiểm tra chữ ký số:

Tạo chữ ký số:

Quá trình thẩm định chữ ký số:



Tính chất của chữ kí số :

Khả năng nhận thực

Các hệ thống mật mã hóa khóa công khai cho phép mật mã hóa văn bản với khóa bí mật mà chỉ có người chủ của khóa biết. Để sử dụng chữ ký số thì văn bản không cần phải được mã hóa mà chỉ cần mã hóa hàm băm của văn bản đó (thường có độ dài cố định và ngắn hơn văn bản). Khi cần kiểm tra, bên nhận giải mã (với khóa công khai – public key) để lấy lại hàm băm và kiểm tra với hàm băm của văn bản nhận được. Nếu 2 giá trị này khớp nhau thì bên nhận có thể tin tưởng rằng văn bản xuất phát từ người sở hữu khóa bí mật. Tất nhiên là chúng ta không thể đảm bảo 100% là văn bản không bị giả mạo vì hệ thống vẫn có thể bị phá vỡ.

Vấn đề nhận thực đặc biệt quan trọng đối với các giao dịch tài chính. Chẳng hạn một chi nhánh ngân hàng gửi một gói tin về trung tâm dưới dạng *(a,b)*, trong đó *a* là số tài khoản và *b* là số tiền chuyển vào tài khoản đó. Một kẻ lừa đảo có thể gửi một số tiền nào đó để lấy nội dung gói tin và truyền lại gói tin thu được nhiều lần để thu lợi (tấn công truyền lại gói tin).

Tính toàn vẹn

Cả hai bên tham gia vào quá trình thông tin đều có thể tin tưởng là văn bản không bị sửa đổi trong khi truyền vì nếu văn bản bị thay đổi thì hàm băm cũng sẽ thay đổi và lập tức bị phát hiện. Quá trình mã hóa sẽ ẩn nội dung của gói tin đối với bên thứ 3 nhưng không ngăn cản được việc thay đổi nội dung của nó. Một ví dụ cho trường hợp này là tấn công đồng hình (homomorphism attack): tiếp tục ví dụ như ở trên, một kẻ lừa đảo gửi 1.000.000 đồng vào tài khoản của a, chặn gói tin *(a,b)* mà chi nhánh gửi về trung tâm rồi gửi gói tin (a,b^3) thay thế để lập tức trở thành triệu phú!

Tính không thể phủ nhận

Trong giao dịch, một bên có thể từ chối nhận một văn bản nào đó là do mình gửi. Để ngăn ngừa khả năng này, bên nhận có thể yêu cầu bên gửi phải gửi kèm chữ ký số với văn bản. Khi có tranh chấp, bên nhận sẽ dùng chữ ký này như một chứng cứ để bên thứ ba giải quyết. Tuy nhiên, khóa bí mật vẫn có thể bị lộ và tính không thể phủ nhận cũng không thể đạt được hoàn toàn.Vậy làm thế nào để đảm bảo các tính chất trên? Ở đây chúng ta sử dụng mã hóa để thực hiện việc tạo chữ kí điện tử. Một số thuật toán sau được sử dụng trong việc tạo ra chữ kí điện tử :

- Full Domain Hash, RSA-PSS,… dựa trên RSA
- DSA
- ECDSA
- ElGamal signature scheme

- Undeniable signature
- SHA (thông thường là SHA-1) với RSA

Ở đây chúng ta sẽ chỉ tìm hiểu chủ yếu về 2 loại mã hóa được dùng nhiều nhất là RSA và SHA (Secure Hash Alogrithm)

3.3 Hàm Băm

Chúng ta có thể thấy rằng các sơ đồ chữ ký chỉ cho phép ký các bức điện nhỏ. Ví dụ khi dùng DSS, bức điện 160 bit sẽ được ký bằng chữ ký dài 320 bit. Thực tế ta cần các bức điện dài hơn nhiều. Chẳng hạn một tài liệu về pháp luật có thể dài nhiều Megabyte.

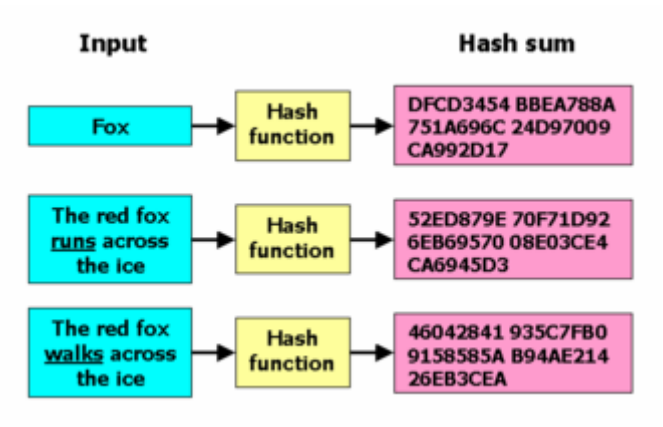
Một cách đơn giải để giải bài toán này là chặt các bức điện dài thành nhiều đoạn 160 bit, sau đó ký lên các đoạn đó độc lập nhau. Điều này cũng tương tự như mã một đoạn chuỗi dài bản rõ bằng cách mã ký tự rõ độc lập bằng cùng một bản khóa (ví dụ: chế độ ECB trong DES).

Biện pháp này có một số vấn đề trong việc tạo ra các chữ ký số. Trước hết với một bức điện dài, ta kết thúc bằng một chữ ký rất lớn (dài gấp đôi bức điện gốc trong trường hợp DSS). Nhược điểm khác là các sơ đồ chữ ký "an toàn" lại chậm vì chúng dùng các ký pháp số học phức tạp như sô mũ modulo. Tuy nhiên, vấn đề quan trọng hơn với phép toán này là bức điện đã ký có thể bị sắp xếp lại các đoạn khác nhau, hoặc một số đoạn trong chúng có thể bị loại bị loại bỏ và bức điện nhận được vẫn phải xác minh được. Ta cần bảo vệ sự nguyên vẹn của toàn bộ bức điện và điều này không thể thực hiện được bằng cách ký độc lập từng mẩu nhỏ của chúng.

Giải pháp cho tất cả các vấn đề này là dùng hàm Hash mã khóa công khai nhanh. Hàm này lấy một bức điện có độ dài tùy ý và tạo ra một bản tóm lược thông báo có kích thước quy định(160 bit nếu dùng DSS). Sau đó bản tóm lược thông báo để được ký.

Trong ngành mật mã học, một hàm băm mật mã học (tiếng Anh: *Cryptographic hash function*) là một hàm băm với một số tính chất bảo mật nhất định để phù hợp việc sử dụng trong nhiều ứng dụng bảo mật thông tin đa dạng, chẳng hạn như chứng thực (authentication) và kiểm tra tính nguyên vẹn của thông điệp (*message integrity*). Một hàm băm nhận đầu vào là một xâu ký tự dài (hay *thông điệp*) có độ dài tùy ý và tạo ra kết quả là một xâu ký tự có độ dài cố định, đôi khi được gọi là *tóm tắt thông điệp* (*message digest*) hoặc *chữ ký số* (*digital fingerprint*).

Trong nhiều chuẩn và ứng dụng, hai hàm băm thông dụng nhất là MD5 và SHA-1. Năm 2005, người ta đã tìm ra lỗi bảo mật của cả hai thuật toán trên.



Hoạt động của một hàm băm

Nói rộng, một hàm băm mật mã học phải hoạt động càng giống với một hàm ngẫu nhiên càng tốt, trong khi vẫn có tính chất đơn định và tính toán có hiệu quả.

Một hàm băm mật mã học được coi là không an toàn nếu một trong các việc sau là khả thi về mặt tính toán:

- Cho một tóm tắt (digest), tìm một thông điệp (chưa biết) khớp với tóm tắt đó
- Tìm các "xung đột băm" (*hash collision*), trong đó hai thông điệp khác nhau có tóm tắt trùng nhau.

Nếu có thể thực hiện một trong hai việc trên, một người có thể tấn công bằng cách dùng các cách trên để thay một thông điệp không được xác nhận (unauthorisez message) vào chỗ của một thông điệp được xác nhận.

Về lý tưởng, việc tìm hai thông điệp có tóm tắt rất giống nhau cũng nên không khả thi, người ta không muốn một kẻ tấn công có thể tìm hiểu được điều gì đó hữu ích về một thông điệp nếu biết tóm tắt.

Nguyên lý:Khi Bob muốn ký bức điện x, trước tiên anh ta xây dựng một bản tóm lược thông báo z = h(x) và sau đó tính $y = sig_K (z)$. Bob truyền cặp (x,y) trên kênh. Xét thấy có thể thực hiện xác minh (bởi ai đó) bằng cách trước hết khôi phục bản tóm lược thông báo z =h (x) bằng hàm h công khai và sau đó kiểm tra xem $ver_k (x,y)$ = true, hay không.

Bức điện : x độ dài tùy ý
↓
Bản tóm lược thông báo:z = h (x) 160 bit
↓
Chữ ký $y = sig_K(z)$ 320 bit

Chúng ta cần chú ý rằng, việc dùng hàm hash h không làm giảm sự an toàn của sơ đồ chữ ký vì nó là bản tóm lược thông báo được chữ ký không phải là bức điện. Điều cần thiết đối với h là cần thỏa mãn một số tính chất nào đó để tranh sự giả mạo.Kiểu tấn công thông thường là Oscar bắt đầu bằng một bức điện được ký hợp lệ (x,y), $y= sig_K(h(x))$,(Cặp (x, y) là bức điện bất kỳ được Bob ký trước đó). Sau đó anh ta tính z = h(x) và thử tìm $x \neq x'$ sao cho $h(x') = h(x)$. Nếu Oscar làm được như vậy,(x',y) sẽ là bức điện hợp lệ, tức một bức điện giả mạo. Để tránh kiểu tấn công này, h cần thỏa mãn tính không va chạm tức là bức điện x không thể tiến hành về mặt tính toán để tìm một bức điện $x \neq x'$ sao cho $h(x') = h(x)$

Một kiểu tấn công kiểu khác như sau: trước hết Oscar tìm hai bức điện $x \neq x'$ sao cho $h(x) = h(x')$. Sau đó Oscar đưa cho Bob thuyết phục Bob ký bản tóm lược thông báo h(x) để nhận được y. Khi đó (x',y) là thông báo giả mạo hợp lệ

Kiểu tấn công thứ 3: giả sử Oscar tính chữ ký trên bản tóm lược thông báo z ngẫu nhiên. Sau đó anh ta tìm x sao cho z=h(x). Nếu làm được như vậy thì (x,y) là bức điện giả mạo hợp lệ. Để tránh được tấn công này, h cần thỏa mã tính chất một chiều.

Bản tóm lược(giá trị của hàm băm) còn được gọi là đại diện văn bản (message digest). Một message digest có chiều dài cố định với các đặc điểm như sau:

- Giá trị trả lại của các hàm băm duy nhất đối với mỗi giá trị đầu vào. Bất kỳ sự thay đổi nào của dữ liệu vào cũng dẫn đến một kết quả sai
- Từ đại diện văn bản không thể suy ra dữ liệu gốc là gì, chính vì điều này người ta gọi là one-way như đã đề cập trong phần mã hóa khóa công khai, nó có thể sử dụng khóa bí mật của bạn cho việc mã hóa và khóa công khai cho việc giải mã. Cách sử dụng cặp khóa như vậy không được dùng khi có sự bí mật thông tin,mà chủ yếu nó dùng để "ký" cho dữ liệu. Thay cho việc đi mã hóa dữ liệu, các phần mềm ký tạo ra đại diện văn bản (message digest) của dữ liệu và sử dụng khóa bí mật để mã hóa đại diện đó. Hình dưới đây là mô hình đơn giản hóa việc chữ ký số được sử dụng như thế nào để kiểm tra tính toàn vẹn của dữ liệu được ký.

Trong hình trên có hai phần được gửi cho người nhận: dữ liệu gốc và chữ ký số. Để kiểm tra tính toàn vẹn của dữ liệu, người nhận trước tiên sử dụng khóa công khai của người ký để giải mã đại diện văn bản từ dữ liệu gốc và mới. Nếu không giống nhau tức là dữ liệu đã bị giả mạo, điều này cũng có thể xảy ra khi sử dụng hai khóa khóa công khai và khóa bí mật không tương ứng.

Nếu như hai đại diện văn bản giống nhau, người nhận có thể chắc chắn rằng khóa công khai được sử dụng để giải mã chữ ký số là tương ứng với khóa bí mật được sử dụng để giải mã chữ ký số. Để xác thực định danh của một đối tượng cũng cần phải

xác thực khóa công khai của đối tượng đó.Trong một vài trường hợp, chữ ký số được dánh giá là có thể thay thế chữ ký bằng tay. Chữ ký số chỉ có thể được đảm bảo khi khóa bí mật không bị lộ. Khi khóa bí mật bị lộ thì người sở hữu chữ ký không thể ngăn chặn được việc bị giả mạo chữ ký

3.4 Một số sơ đồ chữ ký điện tử

3.4.1 Sơ đồ chữ ký RSA(đề xuất năm 1978)

Có thể coi bài toán xác thực là bài toán "đối ngẫu" với bài toán bảo mật. Vì vậy, sử dụng ngược thuật toán RSA ta có thể có được một sơ đồ chữ ký số RSA như sau:

- Sinh khóa: chọn p,q là số nguyên tố lớn. Tính $n=p\times q$, $\phi(n)=(p-1)\times(q-1)$

Đặt **P** = **A** = Z_n**,** Chọn một số tự nhiên *e* sao cho **1 < e<Φ(N)** và là số nguyên tố cùng nhau với **Φ(N), K** = {(e,d) / ed≡ 1 (mod ϕ (n))}. (hay *hay d= (1 + i * Phi_N) / E)* với $i=\overline{1,n}$).

Với K=(n,e,d) ta có D=d là khóa bí mật, E=(n,e) là khóa công khai, m là bản tin cần ký

- Tạo chữ ký : với mỗi bộ khóa K=(n,e,d) định nghĩa

Chữ ký trên $m\in P$ là $S= Sig_D(m)= m^d \bmod n$, $S\in A$

- Kiểm tra chữ ký: $Ver_E(m,S)= TRUE \Leftrightarrow m= S^e \bmod n$
- Hoạt động của sơ đồ chữ ký RSA có thể mô tả như sau:

1. Trường hợp bản tin rõ m không cần bí mật(A ký bản tin m và gửi cho B,

B kiểm tra chữ ký của A)

Giả sử muốn gửi cho B bản tin rõ m có xác thực bằng chữ ký số của mình. Trước tiên A tính chữ ký số

$S_A = Sig_{DA}(m)= m^{dA} \bmod n_A$

Sau đó A gửi cho B bộ đôi (m, S_A) và kiểm tra xem điều kiện $m \equiv S^{eA}_A \bmod n_A$có thỏa mãn không. Nếu thỏa mãn, thì khi đó B khẳng định rằng $Ver_{EA}(m,S_A)$ nhận giá trị TRUE và chấp nhận chữ ký của A trên m

a. A ký bản tin rõ m để được chữ ký S_A. Sau đó A dùng khóa mã công khai E_B của B để lập bản mã $M= E_B(m, S_A)$ rồi gửi đến B. Khi nhận được bản mã M, B dùng khóa bí mật D_B của mình để giải mã cho M và thu được m, S_A. Tiếp đó dùng thuật toán kiểm tra Ver_{EA} để xác nhận chữ ký của A
b. Ví dụ sau đây sử dụng sơ đồ chữ ký RSA với thông điệp lớn
c. Sinh khóa
d. Thực thể A chọn số nguyên p=7927 và q=6997 và tính n=pq= 5546521 và $\phi= 7926\times6996=55450296$. A chọn a=5 và giải $ab=5b\equiv1 \pmod{55450296}$

được b=44360237. Khóa công khai của A là (n=55465219, a=5) và khóa riêng của A là b=44360237

e. Sinh chữ ký

f. Để ký một thông điệp m=31229978, A tính m_1'= h(m)= 31229978 và tính toán chữ ký s=m_1'b mod 312299784430237 mod 55465219 =30729435

g. Xác nhận chữ ký

h. B tính m_2'= s^a mod n= 30729435^5 mod 55465219 = 31229978. Cuối cùng B chấp nhận chữ ký vì m_2'= m_1'

Chú ý

So sánh giữa sơ đồ chữ ký RSA và sơ đồ mật mã RSA ta thấy có sự tương ứng. Việc Alice ký vào m tương ứng với việc mã hóa văn bản m. Thuật toán kiểm thử chính là việc sử dụng hàm giải mã như RSA để kiểm tra xem sau khi giải mã có đúng là văn bản trước khi ký không. Thuật toán kiểm thử là công khai, bất kỳ ai cũng có thể kiểm thử chữ ký

Như vậy việc ký chẳng qua là mã hóa, việc kiểm thử lại chính là việc giải mã. Văn bản m mã hóa trước khi gửi. Nhưng giữa việc ký và mã hóa có mối liên hệ gì không? Nên ký trước hay mã hóa trước

vấn đề giải mã

2. Giả sử người gửi Alice muốn gửi văn bản m cũng chữ ký S đến Bob, có 2 cách xử lý:

a. Ký trước, mã hóa sau.

Alice ký trước vào m bằng chữ ký S= $Sig_A(m)$, sau đó mã hóa m và S nhận được z =$e_A(m,S)$. Alice gửi z cho Bob

Nhận được z Bob giải mã z để được m, S.Tiếp theo kiểm tra chữ ký $Ver_B(m,S)$=True không?

b. Mã hóa trước, ký sau.

Alice mã hóa trước m bằng u=$e_A(m)$, sau đó ký vào u bằng chữ ký v=$Sig_A(u)$. Alice gửi (u,v) cho N. Nhận được (u,v) , Bob giải mã được m.Tiếp theo kiểm tra chữ ký $Ver_B(u,v)$= true?

1. Giả sử Oscar lấy trộm được thông tin trên đường truyền từ Alice đến Bob

trường hợp a, Oscar sẽ lấy được z. Trong trường hợp b, Oscar lấy được(u,v)

- Để tấn công văn bản m trong cả hai trường hợp, Oscar đều phải giải mã thông tin lấy được.
- Nếu muốn tấn công vào chữ ký, thay bằng chữ ký giả mạo thì xảy ra điều gì?

Trường hợp a, để có thể tấn công chữ ký S Oscar phải giải mã z, mới nhận được S

Trường hợp b, để có thể tấn công chữ ký v, Oscar đã sẵn có v, sau đó gửi (u,v') đến Bob

Oscar thay chữ ký v của Alice trên u, bằng chữ ký của Oscar là v'= $Sig_O(u)$, sau đó gửi (u,v') đến Bob.Khi nhận được v', Bob kiểm thử thấy sai, gửi phản hồi lại Alice.Alice có thể chứng minh chữ ký đó là giả mạo. Alice đưa chữ ký đúng cho Bob nhưng quá trình truyền tin sẽ bị chậm lại.

Như vậy trong trường hợp b, Oscar có thể giả mạo chữ ký mà không cần giải mã.

Vì thế có lời khuyên: hãy ký trước khi mã hóa cả chữ ký.

3.4.2 Sơ đồ chữ ký ElGama

Sơ đồ chữ ký ElGama được thiết kế với mục đích dành riêng cho chữ ký số, điểm mạnh của nó là cùng số nguyên tố p trong cùng một sơ đồ thì với R là ngẫu nhiên nên ta có thể có nhiều chữ ký số. Điều này có nghĩa là có nhiều chữ ký hợp lệ trên bức điện cho trước bất kỳ. Thuật toán xác minh phải có khả năng chấp nhận bất kỳ chữ ký hợp lệ nào khi xác thực chữ ký đó.

❖ Sơ đồ chữ ký ElGama

- Chọn p là một số nguyên tố khi đó Z_p là một trường và Z_p^* sẽ là một nhóm với phép nhân.
- Giả sử g là phần tử sinh của Z_p^*.
- Chọn ngẫu nhiên $r \in Z_p$ và tính $K= g^r \bmod p$

công khai K, p,g.

❖ Yếu tố xác thực hóa.

- A gửi m cho B với $m \in Z_p$
- Chọn ngẫu nhiên $R \in Z_p$ sao cho (R,p-1)=1
- Yếu tố xác thực hóa:$X=g^R$ và Y được xác định từ phương trình:

$$m=r*X+R*Y(\bmod p-1)$$

Khi gửi A sẽ gửi bộ (m,X,Y) cho B

❖ Xác thực:

B tính $Z=K^X * X^Y \pmod p$, nếu $Z=g^m$ là đúng, $Z \neq g^m$ là sai. Nếu chữ ký được thiết lập đúng thì xác minh sẽ thành công vì:

$$K^X * X^Y \equiv g^{rX} g^{RY} \pmod p$$

$$\equiv g^m \pmod p$$

B tính chữ ký bằng cách dùng cả giá trị mật r lẫn số ngẫu nhiên mật R(dùng để ký lên bức điện m). Việc xác minh có thể thực hiện duy nhất bằng thông tin công khai.

Ví dụ:

Với m=5, p=11 $\rightarrow$g=2

Chọn r=8 $\rightarrow$ $K=2^8= 25 \bmod 11=3$

Chọn R=9

- yếu tố xác thực hóa: $X=2^9= 3*2=6$. Từ phương trình $5= 8*6+9*Y \pmod{10}$

suy ra : $Y=(5-8*6)*9^{-1}\pmod{10}$

$=(55-48)*9\pmod{10}=3$

- thử xác thực

$Z=3^6 *6^3 \bmod 11=10$

$g^m= 2^5 \bmod 11=10$(đúng)

❖ Xét độ mật của sơ đồ chữ ký ElGama

Giả sử, Oscar thử giả mạo chữ ký trên bức điện m cho trước mà không biết r. Nếu Oscar chọn X và sau đó thử tìm giá trị Y tương ứng. Anh ta phải tính Logarithm rời rạc $Log_X g^m K^{-X}$. Mặt khác, nếu đầu tiên anh ta chọn Y và sau đó thử tìm X và thử giải phương trình:

$$K^X * X^Y \equiv g^m \pmod p$$

Đây là bài toán chưa có lời giải nào. Tuy nhiên, dường như nó chưa được gắn với bài toán đã nghiên cứu kỹ nào nên vẫn còn khả năng có cách nào đó để tính X,Y đồng thời để (Y,X) là một chữ ký. Hiện thời không ai tìm được cách giải song cũng không ai khẳng định rằng nó không thể giải được.

Nếu Oscar chọn X và Y và sau đó thử giải tìm m, anh ta sẽ phải đối mặt với bài toán Logarithm rời rạc. Vì thế Oscar không thể ký một bức điện ngẫu nhiên bằng biện pháp này. Tuy nhiên, có một số cách để Oscar có thể giả mạo chữ ký lên bức điện.

Sau đây là kiểu giả mạo mà Oscar có thể ký một bức điện ngẫu nhiên bằng việc chọn X, Y và m đồng thời

Giả thiết i và j là các số nguyên $0 \le i \le p -2$, $0 \le j \le p-2$ và UCLN(j,p-2)=1

Khi đó thực hiện các tính toán sau:

$$X=g^i K^j \bmod p$$

$$Y=-Xj^{-1} \bmod(p-1)$$

$$m=- Xij^{-1} \bmod(p-1)$$

trong đó j^{-1} được tính theo modulo (p-1) (UCLN(j, p-1)=1)

Ta nói rằng (X,Y) là chữ ký hợp lệ của m. Điều này được chứng minh qua việc kiểm tra điều kiện xác minh

$$K^X * X^Y \equiv g^m \pmod p$$

Sau đây là kiểu giả mạo thứ hai trong đó Oscar bắt đầu bức điện được B ký trước đây. Giả sử (X,Y) là chữ ký hợp lệ trên m. Khi đó Oscar có khả năng ký lên bức

điện khác nhau. Giả sử i, j, h là các số nguyên $0 \le i, j,h \le p-2$ và UCLN(hX-jY,p-1)=1. Ta thực hiện tính toán sau:

$$\lambda = X^h g^i K^j \bmod p$$

$$\mu = Y\lambda\,(hX - jY)^{-1} \bmod (p-1)$$

$$m' = \lambda(hm+iY)^{-1}(hX - jY)^{-1} \bmod (p-1),$$

trong đó $(hX - jY)^{-1}$ được tính theo modulo (p-1). Khi đó dễ dàng kiểm tra điều kiện xác minh.

$$K^{\lambda}\lambda^{\mu} \equiv g^{m'} \pmod p$$

Vì thế (λ, μ) là chữ ký hợp lệ của m'

Cả hai trường hợp trên đều tạo ra các chữ ký giả mạo hợp lệ song không xuất hiện khả năng đối phương giả mạo chữ ký trên bức điện có sự lựa chọn của chính họ mà không phải giải bài toán Logarithm rời rạc. Vì thế không có gì nguy hiểm về độ an toàn của sơ đồ chữ ký Elgamal

Cuối cùng ta sẽ nêu cách có thể phá được sơ đồ này nếu không áp dụng nó một cách cẩn thận. Trước hết, giá trị R ngẫu nhiên được dùng để tính chữ ký phải được gữ bí mật không được để lộ. Vì nếu R bị lộ, khá đơn giản để tính:

$$R=(m-RX)Y^{-1} \bmod (p-1)$$

Một khi r bị lộ thì hệ thống bị phá và Oscar có thể dễ dàng giả mạo chữ ký

Một kiểu dùng sai sơ đồ nữa là dùng cùng giá trị R để ký hai bức điện khác nhau. Điều này cũng tạo thuận lợi cho Oscar tính r và phá hệ thống. Sau đây là cách thực hiện. Giả sử (X, Y_1) là chữ ký trên m_1 và (X,Y_2) là chữ ký trên m_2. Khi đó

$$K^X X^{Y1} \equiv g^{m1} \pmod p$$

$$\text{Và } K^X X^{Y2} \equiv g^{m2} \pmod p$$

Như vậy:

$$g^{m1} g^{m2} \equiv X^{Y1Y2} \pmod p$$

tương đương với phương trình: $m_1 - m_2 \equiv R(Y_1 - Y_2) \pmod{p-1}$, giả sử

$d = UCLN(Y_1 - Y_2, p-1)$. Vì $d \mid (p-1)$ và $d \mid (Y_1 - Y_2)$ nên $d \mid (m_1 - m_2)$. Ta định nghĩa:

$$m' = (m_1 - m_2)/d$$

$$Y' = (Y_1 - Y_2)/d$$

$$p' = (p-1)/d$$

khi đó đồng dư thức trở thành

$$m' \equiv RY' \pmod{p'}$$

vì UCLN (Y',p')=1 nên ta có thể tính

$$\varepsilon = (Y')^{-1} \bmod p'$$

Khi đó giá trị R xác định theo modulo p’ sẽ là

$$R = m' \varepsilon \bmod p'$$

Phương trình này cho d giá trị có thể của R: $R = m'\varepsilon + ip' \pmod{p}$ với i nào đó, $0 \leq i \leq d-1$. Trong đó giá trị d có thể này có thể xác định được một giá trị đúng duy nhất qua việc kiểm tra điều kiện : $X \equiv g^R \pmod{p}$.

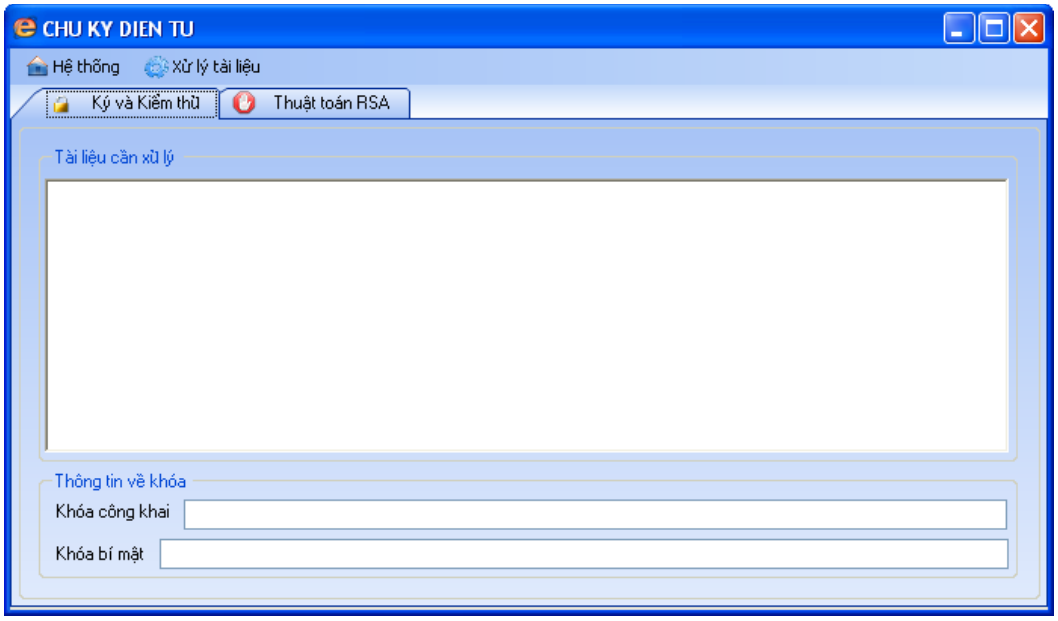
CHƯƠNG 4: MÔ PHỎNG CHỮ KÝ ĐIỆN TỬ

4.1 Cài đặtchương trình

Quy trình sử dụng chữ ký điện tử

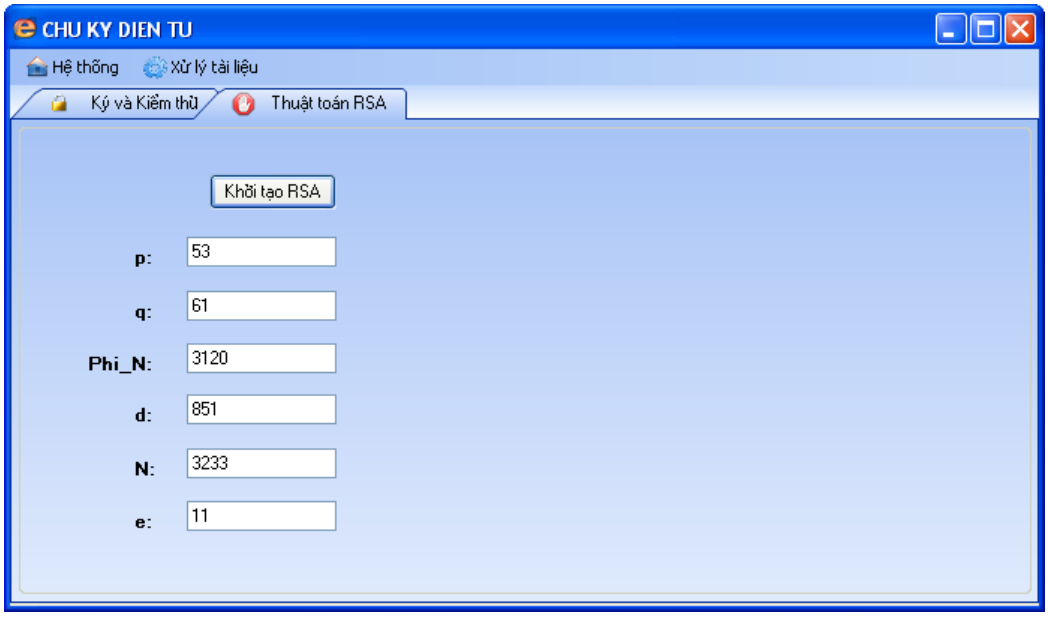
<table>
<tr><th>Quy trình</th><th>Người Gửi</th><th>Người Nhận</th></tr>
<tr><td rowspan="2">Bước 1:</td><td colspan="2">-Hệ thống khởi tạo giá trị khóa</td></tr>
<tr><td>-Lưu khóa bí mật</td><td>-Lưu khóa công khai</td></tr>
<tr><td>Bước 2:</td><td>-Sử dụng khóa bí mật ký trên văn bản
→ Chữ ký
-Gửi văn bản + chữ ký cho người nhận</td><td></td></tr>
<tr><td>Bước 3:</td><td></td><td>-Mở văn bản
-Kiểm tra chữ ký</td></tr>
</table>

Giao diện chương trình

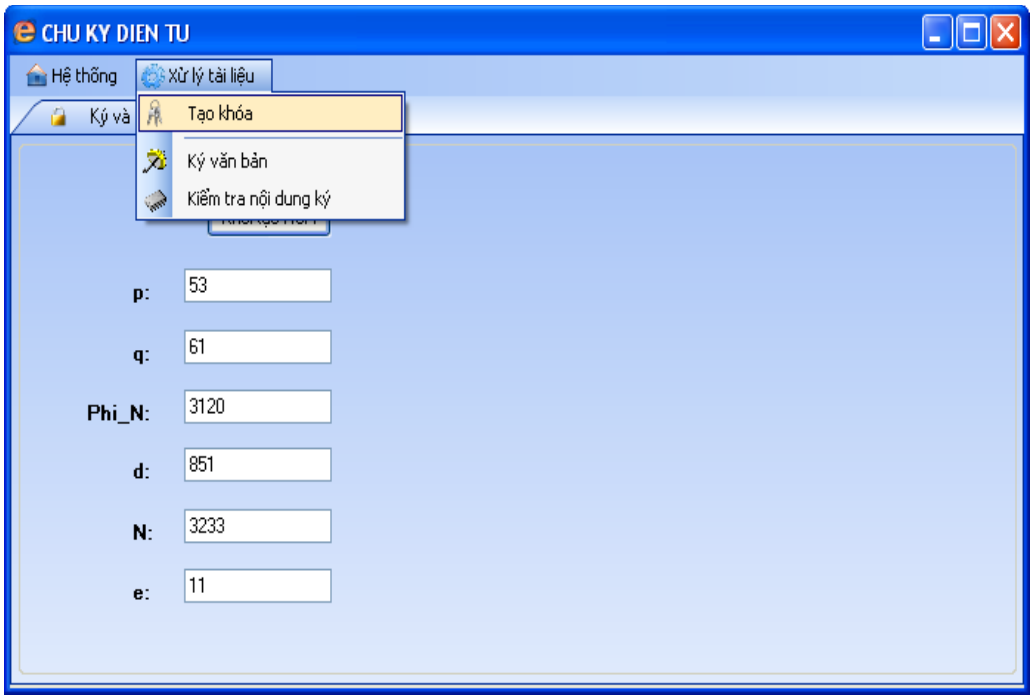


Bước 1:

Khởi tạo RSA

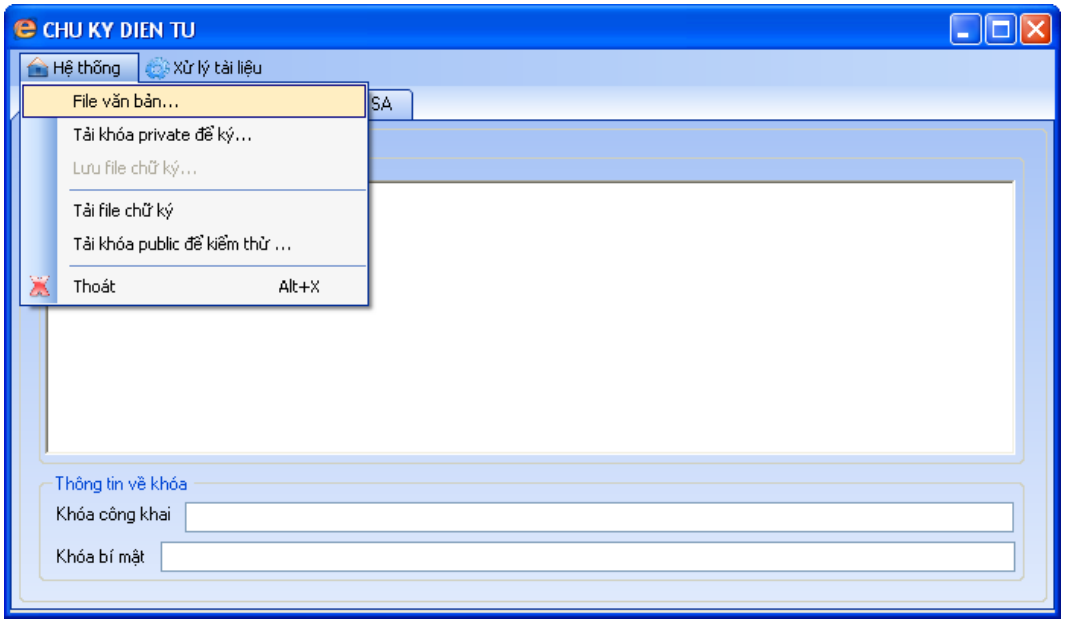


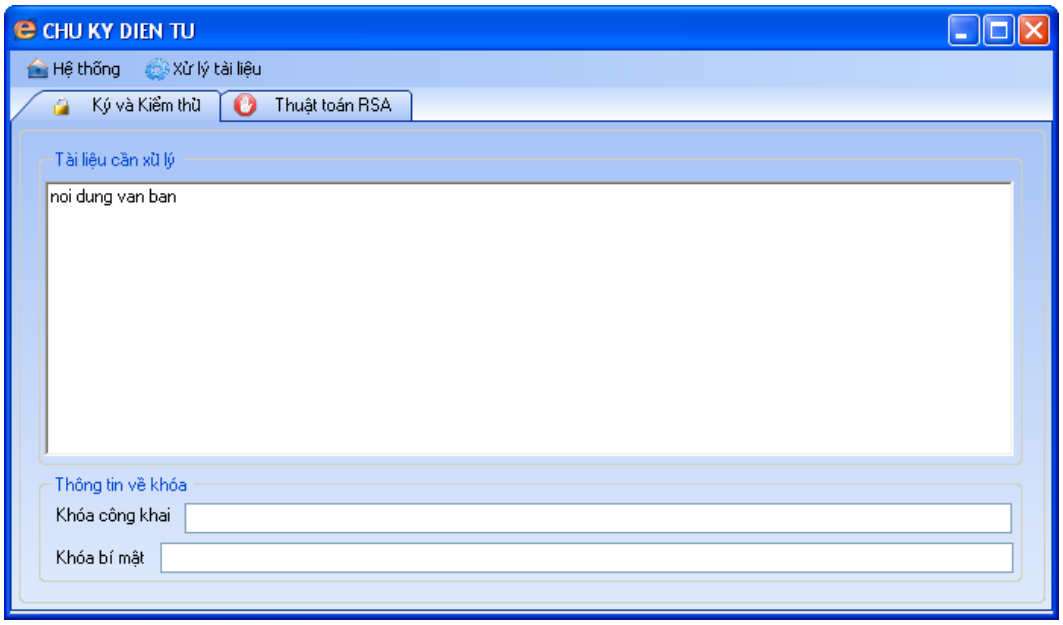
Tạo khóa



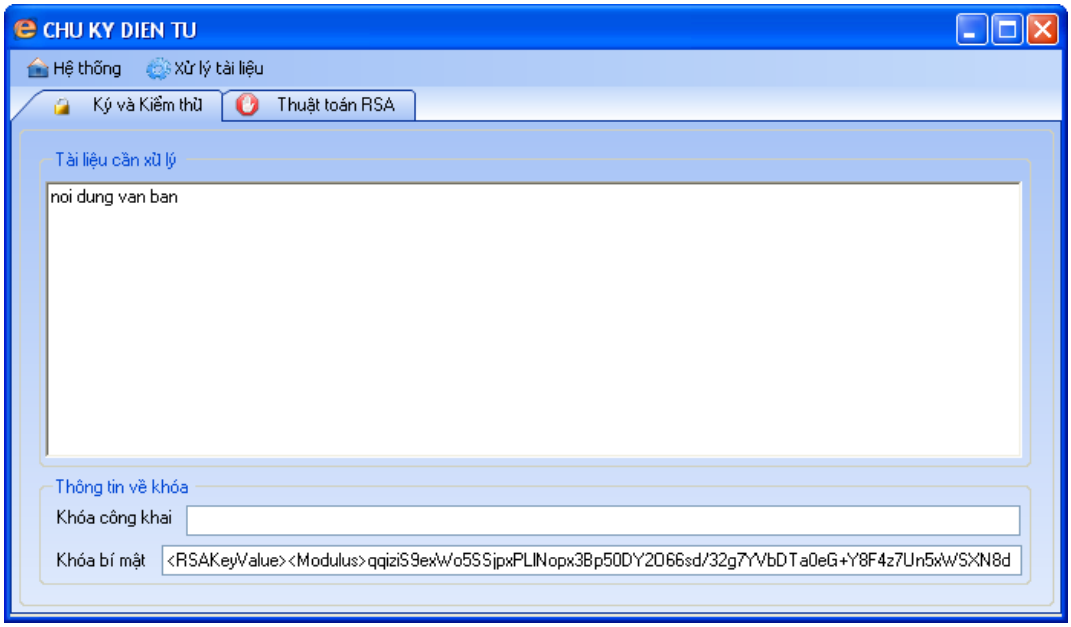
Bước 2:

Tải nội dung văn bản và khóa bí mật để ký

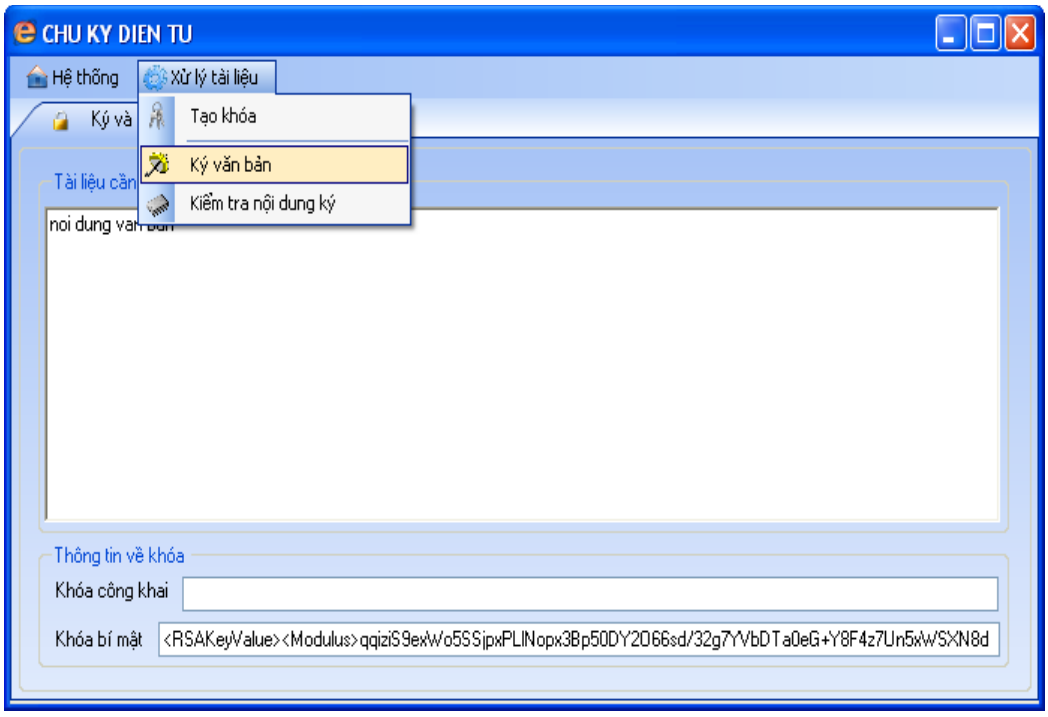




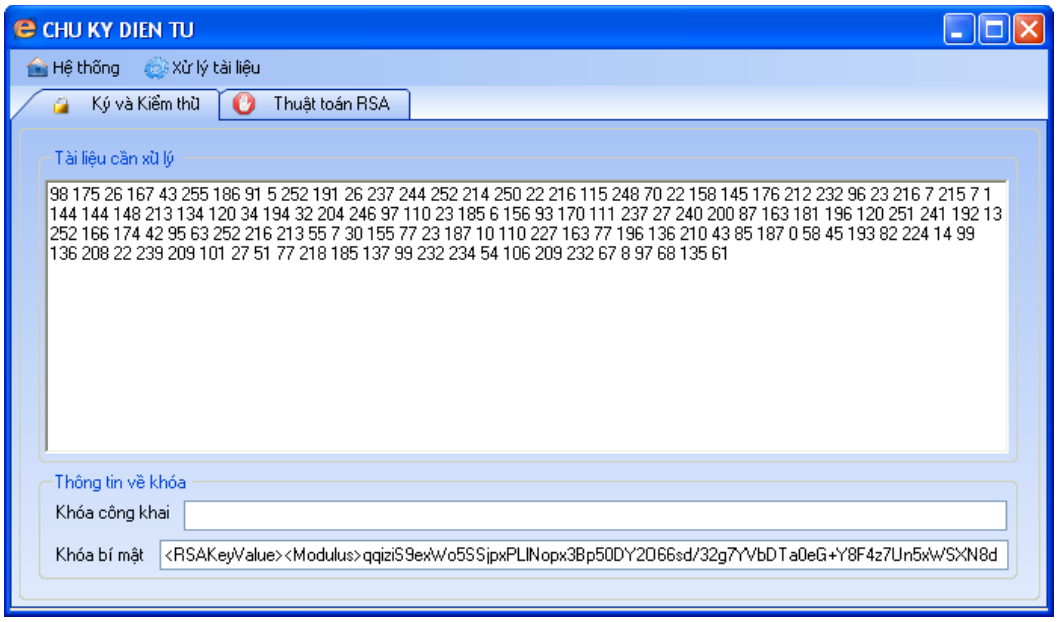
Tải khóa bí mật



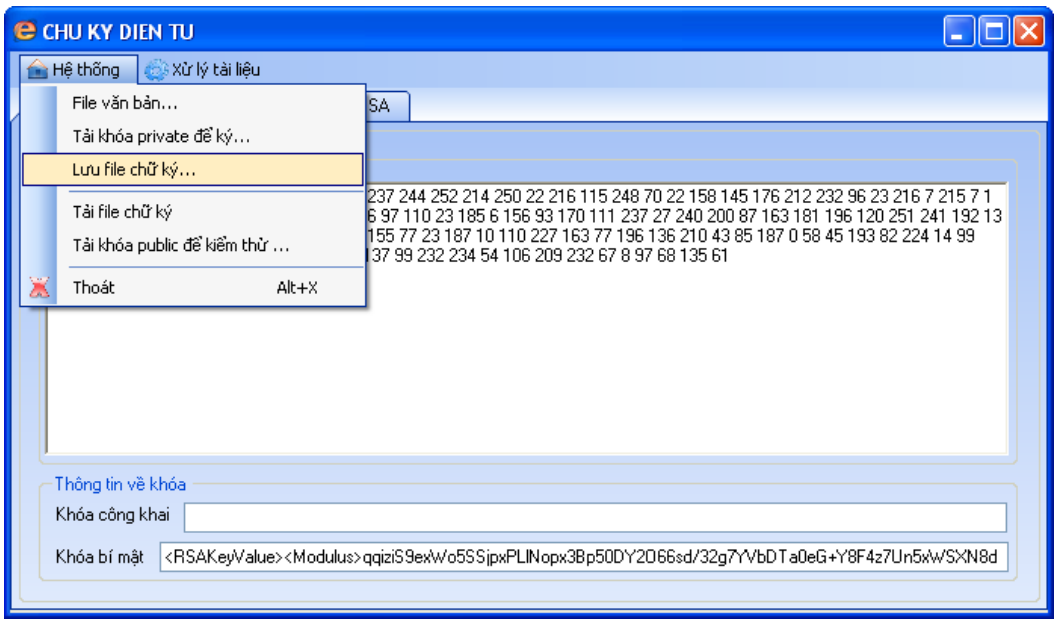
Ký văn bản


CHU KY DIEN TU
Hệ thống
Xử lý tài liệu
Ký và
Tạo khóa
Ký văn bản
Kiểm tra nội dung ký
Tài liệu cần
noi dung van ban
Thông tin về khóa
Khóa công khai
Khóa bí mật
<RSAKeyValue><Modulus>qqiziS9exWo5SSjpxPLINopx3Bp50DY2O66sd/32g7YVbDTa0eG+Y8F4z7Un5xWSXN8d

Chữ ký điện tử

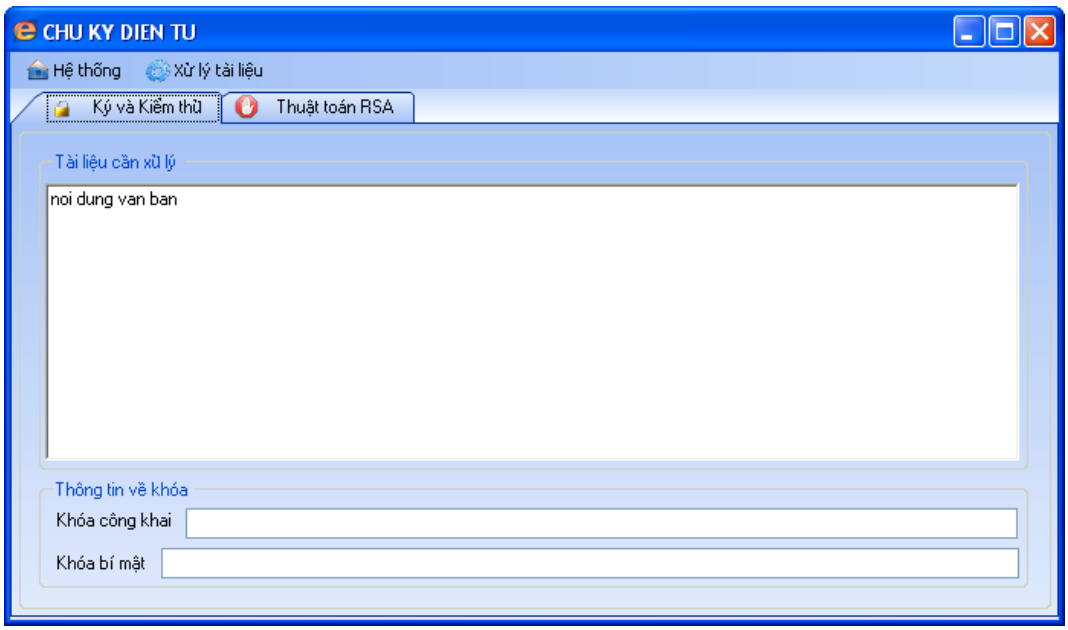


Lưu file chữ ký

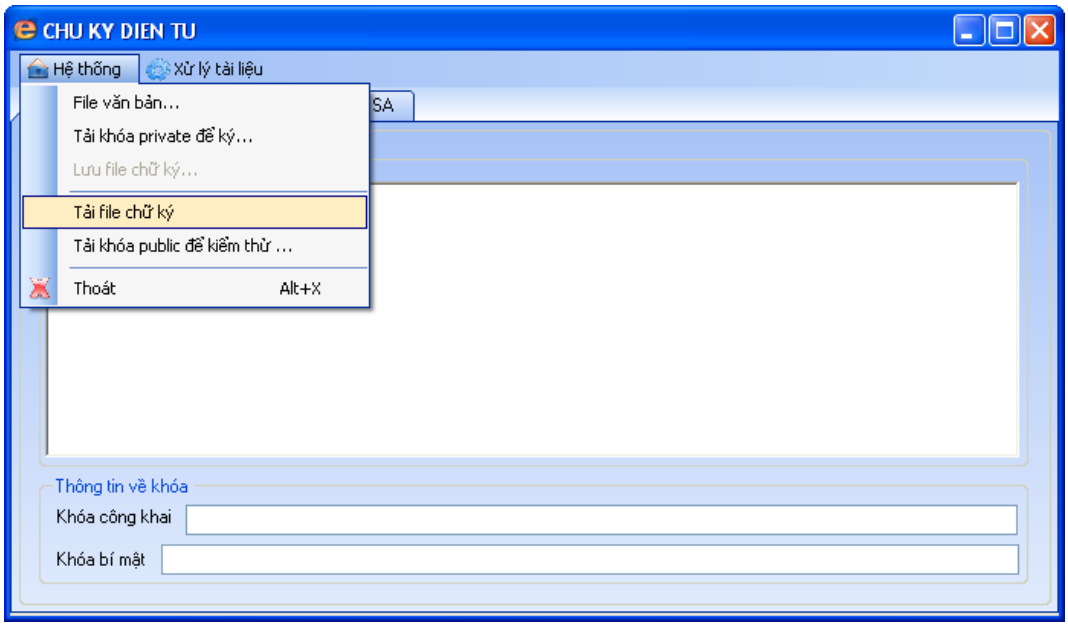


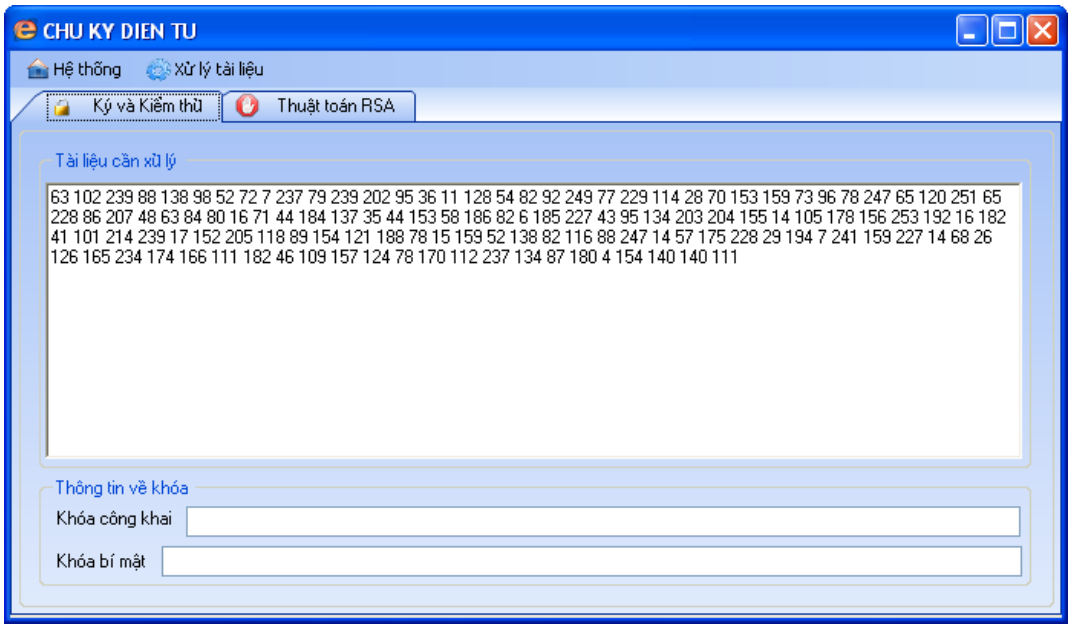
Bước 3:

Người nhận đọc văn bản

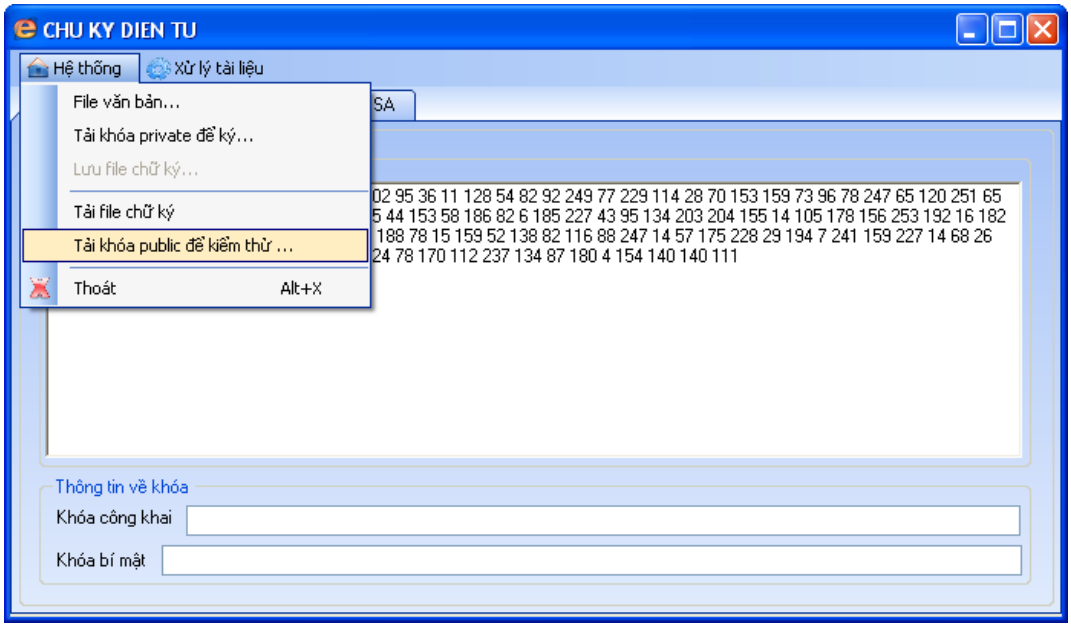


Kiểm tra file chữ ký

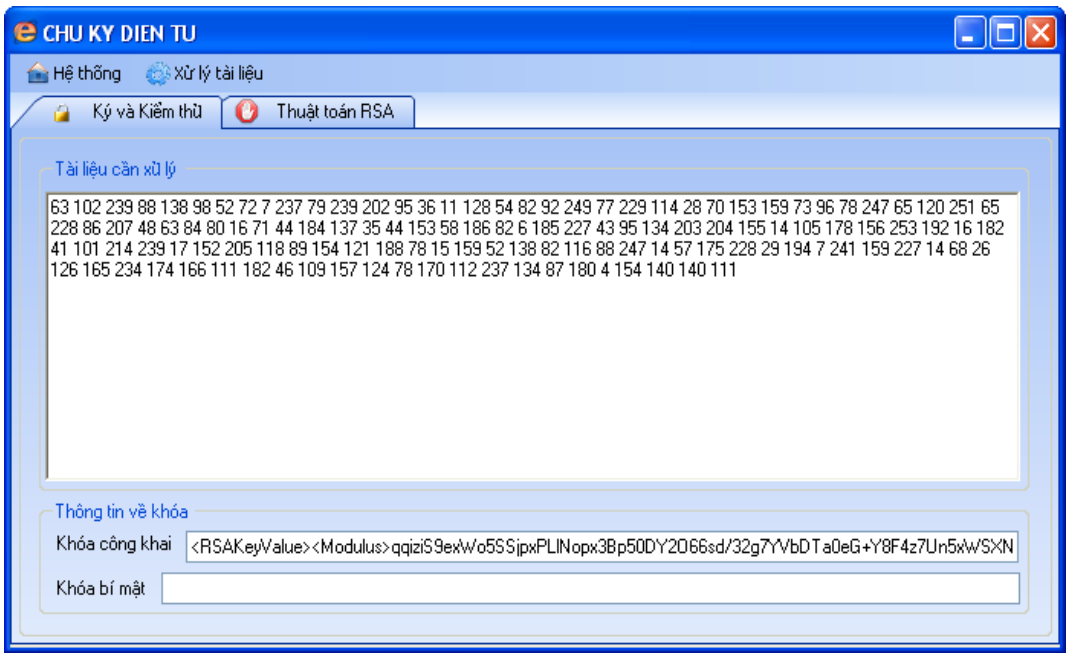




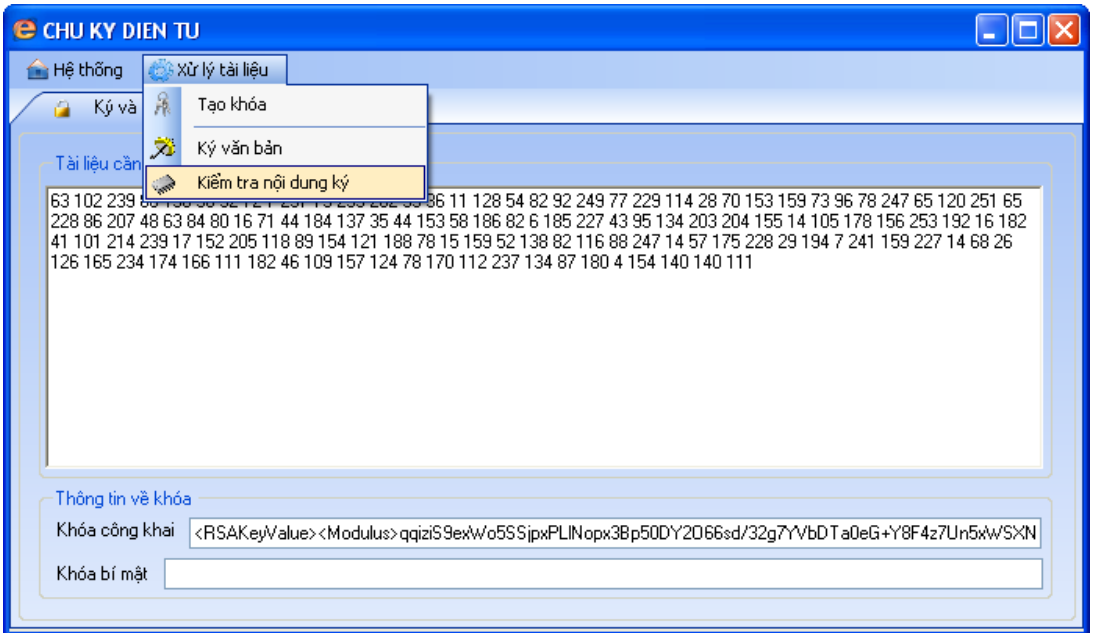
Dùng khóa công khai để kiểm thử



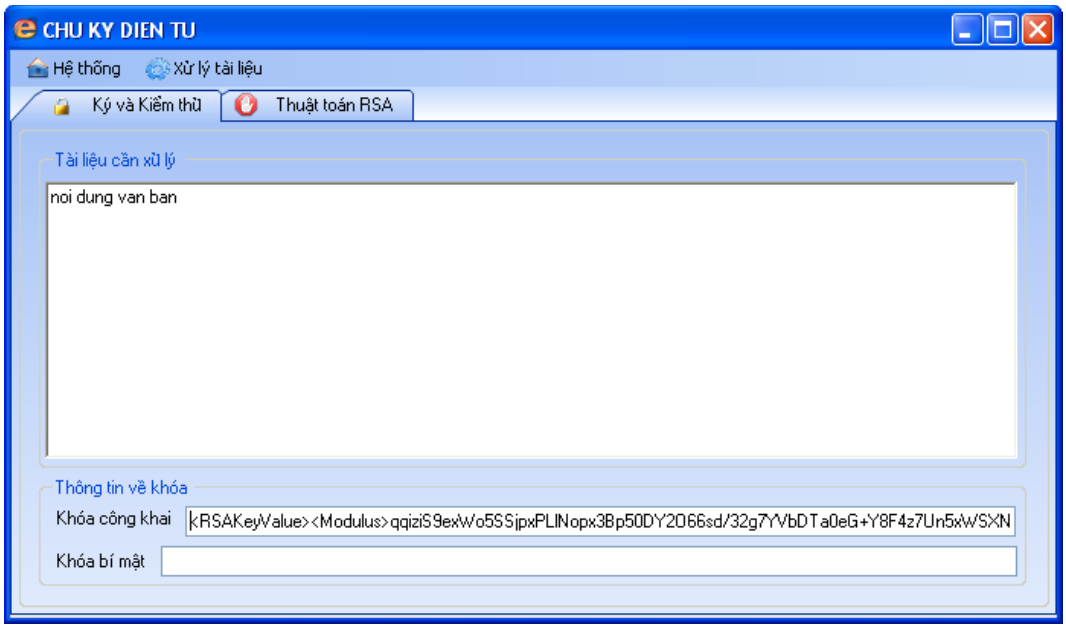
Tải khóa công khai để kiểm thử



Kiểm tra nội dung ký



So sánh với văn bản ban đầu



KẾT LUẬN

Ngày nay, cùng với sự phát triển của khoa học công nghệ hiện đại và Công nghệthông tin, ngành mật mã đã có những bước phát triển mạnh mẽ, đạt được nhiều kết quảlý thuyết sâu sắc và tạo cơ sở cho việc phát triển các giải pháp bảo mật, an toàn thong tin trong mọi lĩnh vực hoạt động của con người. Đặc biệt là những ưu điểm của chữ ký số. Chữ ký số được biết đến khi sự trao đổi thông tin ngày càng phổ biến trên các mạng truyền thông ở nơi mà chữ ký tay không thể phát huy tác dụng.Khi ứng dụng trên mạng máy tính càng trở lên phổ biến, thuận lợi và quan trọng thì yêu cầu về an toàn mạng, an ninh dữ liệu mạng ngày càng trở lên cấp bách và cần thiết. Nguồn tài nguyên mạng rất dễ bị đánh cắp hoặc phá hỏng nếu không có một cơ chế bảo mật cho chúng hoặc sử dụng những cơ chế bảo mật quá lỏng lẻo. Thông tin trên mạng, dù đang truyền hay được lưu trữ đều cần được bảo vệ. Các thông tin ấy phải được giữ bí mật. Cho phép người ta kiểm tra để tin tưởng rằng chúng không bị sửa đổi so với dạng nguyên thủy của mình và chúng đúng là của người nhận gửi nó cho ta. Trong báo cáo này em đã trình bày được kiến thức cơ bản về chữ ký điện tử, một số mô hình ứng dụng chữ ký điện tử và đã xây dựng được chương trình mô phỏng chữ ký điện tử dưới sự hướng dẫn tận tình của ThS.Trần Ngọc Thái. Tuy nhiên do trình độ bản thân có giới hạn và thời gian nghiên cứu chưa sâu nên báo cáo không thể trảnh khỏi thiếu sót. Vì vậy em rất mong nhận được sự đóng góp ý kiến của các thầy giáo cô giáo trong khoa, cũng như các thấy các cô giáo trong hội đồng phản biện để bài báo cáo tốt nghiệp của em được hoàn thiện hơn.

Em xin chân thành cảm ơn các thầy các cô!