CHƯƠNG 5: MÃ CÔNG KHAI VÀ CHỮ KÝ ĐIỆN TỬ

5.1. Các khái niệm mã hóa công khai

Nhược điểm khi dùng mã khóa đối xứng

Mã khóa đối xứng còn được gọi là mã khóa đơn hay mật. Ở đây chỉ dùng một khóa, dùng chung cả người nhận và người gửi. Khi khóa này được dùng, việc trao đổi thông tin về khóa sẽ được thỏa thuận trước. Liệu có cách nào trao đổi trực tiếp an toàn trên mạng mà không cần bên thứ ba không.

Người ta còn gọi đây là mã đối xứng, vì hai đối tác có vai trò như nhau. Do đó không bảo vệ người gửi khỏi việc người nhận giả mạo mẩu tin và tuyên bố là nó được gửi từ người gửi. Nghĩa là khi hai người dùng mã đối xứng, thì họ giữ được bí mật nội dung trao đổi, nhưng bản thân mẩu tin không mang thông tin xác thực được người gửi.

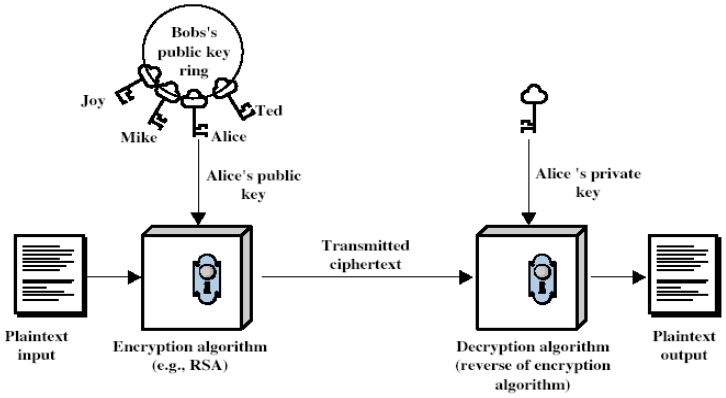
5.1.1. Khái niệm mã khóa công khai

Mã khóa công khai ra đời vào đầu những năm 1970. Có thể nói đây là bước tiến quan trọng nhất trong lịch sử 3000 năm mã hóa. Ở đây người ta sử dụng hai khóa: một khóa riêng và một khóa công khai. Hai khóa này khác nhau, không đối xứng với nhau, do đó mã khóa công khai, còn được gọi là mã không đối xứng. Người ta đã ứng dụng một cách thông minh các kết quả của lý thuyết số về các cặp bài toán thuận - dễ, ngược – khó.

Mã khóa công khai ra đời hỗ trợ thêm để giải quyết một số bài toán an ninh, chứ không phải thay thế mã khóa riêng. Cả hai loại mã cùng tồn tại, phát triển và bổ sung cho nhau.

Mã khóa công khai (hai khóa, không đối xứng) bao gồm việc sử dụng hai khóa:

- **Khóa công khai**, mọi người đều biết, được dùng để mã hóa mẩu tin hoặc kiểm chứng chữ ký.
- **Khóa riêng,** chỉ người sở hữu biết, đề giải mã bản tin hoặc để tạo chữ ký.
- Là không đối xứng vì những người mã hóa hoặc kiểm chứng chữ ký không thể giải mã hoặc tạo chữ ký.



Sơ đồ mã khoa công khai

Tại sao lại phải dùng mã khóa công khai

Người ta muốn giải quyết hai vấn đề sau về khóa nảy sinh trong thực tế:

- Phân phối khóa - làm sao có thể phân phối khóa an toàn mà không cần trung tâm phân phối khóa tin cậy?
- Chữ ký điện tử - làm sao có thể kiểm chứng được rằng mẩu tin gửi đến nguyên vẹn từ đúng người đứng tên gửi?

Nếu chỉ dùng khóa đối xứng, thì không có giải pháp cho hai bài toán trên. Mã khóa công khai được phát minh bởi hai nhà bác học Whitfield Diffie và Martin Hellman ở trường Đại học Stanford vào năm 1976. Tuy nhiên khái niệm ban đầu về nó đã được biết đến sớm hơn bởi cộng đồng các nhà khoa học.

5.1.2. Các đặc trưng của khóa công khai

Các thuật toán khóa công khai dùng hai khóa với các đặc trưng sau:

- Không có khả năng tính toán để tìm khóa giải mã nếu chỉ biết thuật toán mã và khóa dùng để mã.
- Có thể dễ dàng mã hóa hoặc giải mã mẩu tin nếu biết khóa tương ứng
- Trong một số sơ đồ: một khóa bất kỳ trong hai khóa có thể dùng để mã, còn khóa kia dùng để giải mã. Chúng có vai trò đối ngược nhau.

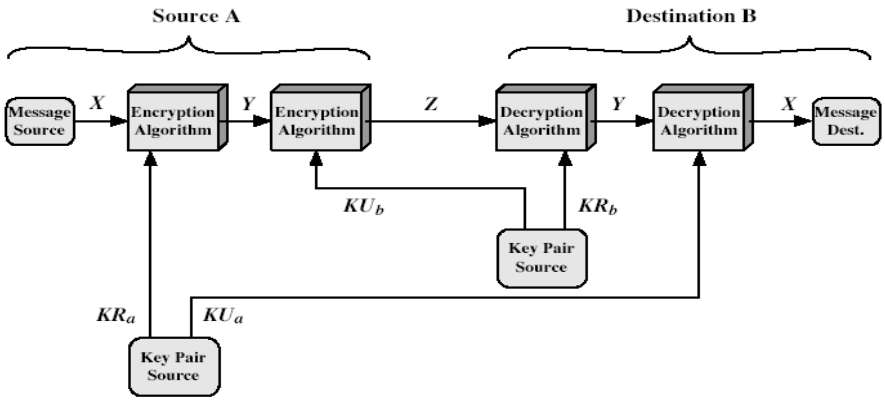


Figure 9.4 Public-Key Cryptosystem: Secrecy and Authentication

5.1.3. Ứng dụng khóa công khai

Có thể phân loại các ứng dụng của khóa công khai thành ba loại khác nhau:

- Mã/giải mã – cung cấp bảo mật. Đây là ứng dụng bảo mật truyền thống giống như ta vẫn thường dùng với khóa đối xứng.
- Chữ ký điện tử - cung cấp xác thực. Một trong các ứng dụng mới của khóa công khai mà khóa đối xứng không thể thực hiện được, đó là khóa công khai có đủ cơ sở để xác nhận người gửi và có thể là một lựa chọn để tạo chữ ký điện tử của người gửi.
- Dùng khóa công khai phân phối khóa mật giữa hai người sử dụng.

Một số thuật toán mã công khai phù hợp với mọi ứng dụng, còn một số khác chuyên dùng cho ứng dụng cụ thể.

5.1.4. Tính an toàn của các sơ đồ khóa công khai

Cũng giống như khóa riêng việc tìm kiếm vét cạn luôn luôn có thể, tức là khi biết một trong hai khóa và thuật toán mã hóa về nguyên tắc ta có thể dò tìm khóa thứ hai bằng cách tính toán các giá trị liên quan. Nói chung khối lượng cần tính toán là rất lớn do độ phức tạp của bài toán xác định khóa. Nếu khóa sử dụng là rất lớn cỡ hơn 512 bit, thì hầu như bài toán tìm khóa thứ hai là không khả thi, không thể thực hiện được trong thời gian có nghĩa, cho dù nguồn lực có thể rất lớn.

Tính an toàn dựa trên sự khác biệt đủ lớn giữa các bài toán dễ là mã/giải mã khi biết khóa và bài toán khó là thám mã khi không biết khóa tương ứng. Vì bài toán thám mã nằm trong lớp các bài toán khó tổng quát hơn đã được biết đến và về mặt lý thuyết đã được chứng minh là nó rất khó có thể thực hiện trên thực tế. Bởi vì nó đòi hỏi sử dụng số rất lớn, nên số phép toán cần thực hiện là rất nhiều. Đây là ý tưởng chính để tạo nên một mã công khai. Ta tìm kiếm các bài toán mà nếu biết thông tin mật nào đó được che giấu thì nó rất dễ thực hiện, còn nếu không thì nó thuộc lớp bài toán rất khó giải, hầu như không thể giải trên thực tế.

Mã công khai thường chậm hơn khá nhiều so với mã đối xứng, nên nó thường được dùng mã hóa những thông tin nhỏ quan trọng.

5.2. Mã công khai RSA

RSA là mã công khai được sáng tạo bởi Rivest, Shamir & Adleman ở MIT (Trường Đại học Công nghệ Massachusetts) vào năm 1977. RSA là mã công khai được biết đến nhiều nhất và sử dụng rộng rãi nhất hiện nay. Nó dựa trên các phép toán lũy thừa trong trường hữu hạn các số nguyên theo modulo nguyên tố. Cụ thể, mã hóa hay giải mã là các phép toán luỹ thừa theo modulo số rất lớn. Việc thám mã, tức là tìm khóa riêng khi biết khóa công khai, dựa trên bài toán khó là phân tích một số rất lớn đó ra thừa số nguyên tố. Nếu không có thông tin gì, thì ta phải lần lượt kiểm tra tính chia hết của số đó cho tất cả các số nguyên tố nhỏ hơn căn của nó. Đây là việc làm không khả thi.

Người ta chứng minh được rằng, phép lũy thừa cần $O((\log n)^3)$ phép toán, nên có thể coi lũy thừa là bài toán dễ. Cần chú ý rằng ở đây ta sử dụng các số rất lớn khoảng 1024 bit, tức là cỡ 10^{350}. Tính an toàn dựa vào độ khó của bài toán phân tích ra thừa số các số lớn. Bài toán phân tích ra thừa số yêu cầu $O(e^{\log n \log\log n})$ phép toán, đây là bài toán khó.

Mã công khai RSA gồm hai giai đoạn: khởi tạo khóa RSA và giai đoạn mã hóa/giải mã.

5.2.1. Khởi tạo khóa RSA

Mỗi người sử dụng A tạo một cặp khóa công khai – riêng như sau:

- Chọn ngẫu nhiên hai số nguyên tố lớn p và q khác nhau.
- Tính số N làm modulo của hệ thống: N = p.q
  - Ta đã biết $\Phi(N) = (p - 1)(q - 1)$
- Chọn ngẫu nhiên khóa mã e làm khóa công khai, sao cho $1 < e < \Phi N)$ và $\gcd(e,\Phi(N)) = 1$, tức là e và $\Phi(N)$ là hai số nguyên tố cùng nhau.
- Nghịch đảo của e theo modulo $\Phi(N)$ là khóa riêng d, vậy tìm d từ phương trình

(e.d) mod Φ(N) = 1, với 0< d < Φ(N) hay d = e^{-1} mod Φ(N)

Chú ý: vai trò của e và d có thể thay đổi cho nhau, tức là có thể lấy e làm khóa mật, khi đó tính d nghịch đảo của e làm khóa công khai.

- Người sử dụng A in khóa mã công khai: KU = {e,N} và thông báo cho mọi người biết.
- Người sử dụng A giữ bí mật khóa riêng: KR = {d,p,q}

5.2.2. Sử dụng RSA

- Để mã hóa mẩu tin M, người gửi B:
  - Lấy khóa công khai của người nhận A: KU = {e,N}
  - Mã hóa thông điệp M bằng khóa công khai của người nhận A:

    C = M^e mod N, trong đó 0 ≤ M < N
- Để giải mã bản mã, người sử dụng A:
  - Sử dụng khóa riêng KR = {d,p,q}
  - Giải mã thông điệp, tính

    M = C^d mod N
- Lưu ý rằng bản tin M < N, do đó khi cần chia khối bản rõ thành các khối nhỏ để thỏa mãn tính chất này.

5.2.3. Cơ sở của RSA

Ta sẽ chứng minh rằng C^d mod N = M.

Thật vậy, ta có e và Φ(N) nguyên tố cùng nhau, nên tồn tại số nghịch đảo d, tức là:

(e.d) mod Φ(N) = 1,

Do đó tồn tại số nguyên dương k nào đó, sao cho

e.d = 1 + k.Φ(N)

Suy ra

$C^d = (M^e)^d = M^{1+k.\Phi(N)} = M^1.(M^{\Phi(N)})^k$,

Vì dựa theo định lý Euler có thể chứng minh được rằng

$M^{\Phi(N)}$ mod N = 1

Nên

C^d Mod N = $M^1.(M^{\Phi(N)})^k$ mod N = M^1 mod N.$(M^{\Phi(N)}$ mod N$)^k$

= M.1 mod N = M

Ví dụ 1: Tạo khóa công khai cho người sử dụng A, sau đó người sử dụng B dùng khóa công khai của A mã hóa thông điệp gửi cho A và cuối cùng A sử dụng khóa riêng của mình để giải mã thông điệp.

Tạo bộ khóa công khai cho người sử dụng A

1. Chọn các số nguyên tố: p = 17 và q = 11.
2. Tính N = p.q, N = 17.11 = 187
3. Tính Φ(N) = (p –1).(q – 1) = 16.10 = 160

4. Chọn e: gcd(e,160) = 1; Lấy e = 7
5. Xác định d: (e.d) mod 160 = 1 và d < 160
   Giá trị cần tìm là d = 23, vì 23.7 = 161 = 10.160 + 1
6. In khóa công khai của người sử dụng A: KU={7,187}
7. Giữ khóa riêng bí mật riêng của A: KR={23,17,11}

B sử dụng khóa công khai của A để mã hóa mã thông điệp gửi cho A:

- Cho mẩu tin M = 88 (thỏa mã điều kiện 88 < 187)
- Mã C = 88^7 mod 187 = 11

Người sử dụng A dùng khóa riêng của mình để giải mã như sau:

$M = 11^{23} \bmod 187 = 88$

- Có thể dùng định lý phần dư Trung Hoa để giải mã cho nhanh như sau:
  - Tính 11^{23} mod 11 = 0
  - Tính 11^{23} mod 17 = $(-6)^{23}$ mod 17 = $(-6)^{16}(-6)^4(-6)^2(-6)$ mod 17 = 3
    Vì $(-6)^2$ mod 17 = 2, nên $(-6)^4$ mod 17 = 4, $(-6)^8$ mod 17 = -1; $(-6)^{16}$ mod 17 = 1
  - 11^{-1} mod 17 = $(-6)^{-1}$ mod 17 = 14 nên c_2 = 11.(11^{-1} mod 17) = 11.(14 mod 17) = 154
  - Vậy M = (3.154) mod 187 = 462 mod 187 = 88

Giải mã hiệu quả:

Như chúng ta thấy qua ví dụ trên, nếu biết N = p.q, thì ta có thể giải mã nhanh bằng cách sử dụng định lý phần dư Trung Hoa tính toán theo các modulo p và q. Sau đó kết hợp lại để tìm ra bản rõ. Vì ở đây người sử dụng sở hữu khóa riêng, nên biết được p và q, do đó có thể sử dụng kỹ thuật này. Nếu sử dụng định lý phần dư Trung Hoa để giải mã thì hiệu quả là nhanh gấp bốn lần so với giải mã tính trực tiếp.

Sinh khóa RSA

Người sử dụng RSA cần phải xác định ngẫu nhiên hai số nguyên tố rất lớn, thông thường khoảng 512 bit. Do đó việc sinh ra ngẫu nhiên p, q và kiểm tra xác suất tính nguyên tố của chúng có nhiều giải pháp khác nhau với độ tin cậy cao. Sau khi chọn được một khóa e hoặc d nguyên tố cùng nhau với Φ(N), dễ dàng tính được khóa kia chính là số nghịch đảo của nó qua thuật toán Euclide mở rộng.

5.2.4. An toàn của RSA

Trên thực tế có nhiều cách tấn công khác nhau đối với mã công khai RSA như sau:

Tìm kiếm khóa bằng phương pháp vét cạn, phương pháp này không khả thi với kích thước đủ lớn của các số hoặc tấn công bằng toán học dựa vào độ khó việc tính Φ(N) bằng cách phân tích N thành hai số nguyên tố p và q hoặc tìm cách tính trực tiếp Φ(N). Trong quá trình nghiên cứu việc thám mã người ta đề xuất kiểu tấn công thời gian trong khi giải mã, tức là căn cứ vào tốc độ mã hóa và giải mã các mẩu tin cho trước mà phán đoán các thông tin về khóa. Cuối cùng có những nghiên cứu tấn công RSA với điều kiện biết trước bản mã cho trước. Cụ thể như sau:

Bài toán phân tích

- Tấn công toán học có 3 dạng:
  - Phân tích N = p.q, sau đó tính Φ(N) và d;

- Tìm n trực tiếp Φ(N) và tính d;
- Tìm d trực tiếp.

- Hiện tại tin rằng tất cả các bài toán trên đều tương đương với bài toán phân tích một số ra thừa số.
  - Có các bước tiến chậm theo thời gian;
  - Hiện tại cho rằng RSA 1024 hoặc 2048 là an toàn.
- Tấn công thời gian:
  - Được phát triển vào giữa năm 1990;
  - Paul Kocher chỉ ra rằng kẻ thám mã có thể xác định được khóa riêng nếu theo dõi thời gian máy tính cần để giải mã các bản tin.
  - Tấn công thời gian không chỉ áp dụng cho RSA, mà cả với các hệ mã công khai khác.
  - Tấn công thời gian giống như kẻ cướp đoán số điện thoại bằng cách quan sát một người nào đó trong bao lâu chuyển quay điện thoại từ số này sang số khác.
- Tấn công bản mã chọn trước
  - RSA có điểm yếu với tấn công bản mã chọn trước;
  - Kẻ tấn công chọn bản mã và đoán bản rõ được giải mã;
  - Chọn bản mã cho việc khám phá RSA, cung cấp thông tin để thám mã;

5.3. Trao đổi khóa Diffie Hellman

5.3.1. Yêu cầu

Trao đổi khóa Diffie Hellman là sơ đồ khóa công khai đầu tiên được đề xuất bởi Diffie và Hellman năm 1976 cùng với khái niệm khóa công khai. Sau này được biết đến bởi James Ellis (Anh), người đã đề xuất bí mật năm 1970 mô hình tương tự. Đây là phương pháp thực tế trao đổi công khai các khóa mật đối xứng. Nó thúc đẩy việc nghiên cứu tìm ra các mã khóa công khai. Sơ đồ được sử dụng trong nhiều sản phẩm thương mại.

Trao đổi khóa Diffie Hellman là sơ đồ trao đổi khóa mật đối xứng dùng khóa công khai có các đặc trưng sau:

- Không thể dùng để trao đổi mẩu tin bất kỳ.
- Tuy nhiên nó có thể thiết lập khóa chung.
- Chỉ có hai đối tác biết đến.
- Giá trị khóa phụ thuộc vào các đối tác (và các thông tin về khóa công khai và khóa riêng của họ).
- Dựa trên phép toán lũy thừa trong trường hữu hạn (chẳng hạn, modulo theo số nguyên tố) là bài toán dễ.
- Độ an toàn dựa trên độ khó của bài toán tính logarit rời rạc (giống bài toán phân tích ra thừa số) là bài toán khó.

5.3.2. Khởi tạo Diffie Hellman

- Mọi người dùng thỏa thuận dùng tham số chung:

- Số nguyên tố rất lớn q
- α là căn nguyên thủy của mod q.

- Mỗi người dùng (A chẳng hạn) tạo khóa của mình:
  - Chọn một số x_A làm khóa mật (số) của A: $x_A < q$
  - Tính khóa công khai y_A của A: $y_A = \alpha^{x_A} \bmod q$.
  - Mỗi người dùng thông báo công khai khóa công khai y_A và giữ bí mật khóa riêng x_A của mình.

5.3.3. Trao đổi khóa Diffie Hellman

Sau bước khởi tạo khóa, giả sử hai người sử dụng A và B đã thông bào công khai cho nhau biết khóa công khai y_A và y_B. Khi đó mỗi người sử dụng A và B có thể tự tính được khóa phiên dùng chung như sau:

- Khóa phiên dùng chung cho hai người sử dụng A, B là K_{AB}

$$K_{AB} = \alpha^{x_A.x_B} \bmod q$$
$$= y_A^{x_B} \bmod q \text{ (mà B có thể tính)}$$
$$= y_B^{x_A} \bmod q \text{ (mà A có thể tính)}$$

- K_{AB} được sử dụng như khóa phiên để mã hóa và giải mã thông điệp trao đổi giữa A và B.
- A và B lần lượt trao đổi với nhau, họ có khóa chung K_{AB} cho đến khi họ chọn khóa mới.
- Kẻ thám mã cần biết khóa riêng của một trong hai người sử dụng, do đó phải giải bài toán logarit rời rạc, tìm x_A từ phương trình $y_A = \alpha^{x_A} \bmod q$. Đây là bài toán khó, nó là cơ sở an toàn cho thủ tục trao đổi khóa Diffie – Hellman.

Ví dụ 2: Hai người sử dụng Alice và Bob muốn trao đổi khóa phiên:

- Đồng ý chọn số nguyên tố: $q = 353$ và $\alpha = 3$
- Chọn các khóa mật ngẫu nhiên:

  A chọn $x_A = 97$; B chọn $x_B = 233$
- Tính các khóa công khai của A và B:

  $y_A = 3^{97} \bmod 353 = 40$ (Alice)

  $y_B = 3^{233} \bmod 353 = 248$ (Bob)
- Tính khóa phiên chung:

  $K_{AB} = y_B^{x_A} \bmod 353 = 248^{97} = 160$ (Alice)

  $K_{AB} = y_A^{x_B} \bmod 353 = 40^{233} = 160$ (Bob)

5.4. Xác thực mẩu tin

5.4.1. Xác thực thông điệp

Các khái niệm

Xác thực mẩu tin (thông điệp) liên quan đến các khía cạnh sau, khi truyền tin trên mạng:

- Bảo vệ tính toàn vẹn của mẩu tin: bảo vệ mẩu tin không bị thay đổi hoặc có các biện pháp phát hiện nếu mẩu tin bị thay đổi trên đường truyền.
- Kiểm chứng danh tính và nguồn gốc: xem xét mẩu tin có đúng do người xưng tên gửi không hay một kẻ mạo danh nào khác gửi.
- Không chối từ bản gốc: trong trường hợp cần thiết, bản thân mẩu tin chứa các thông tin chứng tỏ chỉ có người xưng danh gửi, không một ai khác có thể làm điều đó. Như vậy người gửi không thể từ chối hành động gửi, thời gian gửi và nội dung của mẩu tin.

Ngoài ra có thể xem xét bổ sung thêm các yêu cầu bảo mật như mã hóa. Với mong muốn đáp ứng các yêu cầu trên, có ba hàm lựa chọn sau đây được sử dụng:

- Mã mẩu tin bằng mã đối xứng hoặc mã công khai.
- Mã xác thực mẩu tin (MAC): dùng khóa và một hàm nén mẩu tin cần gửi để nhận được một đặc trưng đính kèm với mẩu tin và người gửi đó.
- Hàm hash (hàm băm) là hàm nén mẩu tin tạo thành "dấu vân tay" cho mẩu tin.

Các yêu cầu bảo mật khi truyền mẩu tin trên mạng.

Tìm các biện pháp cần thiết để chống đối lại các hành động phá hoại sau:

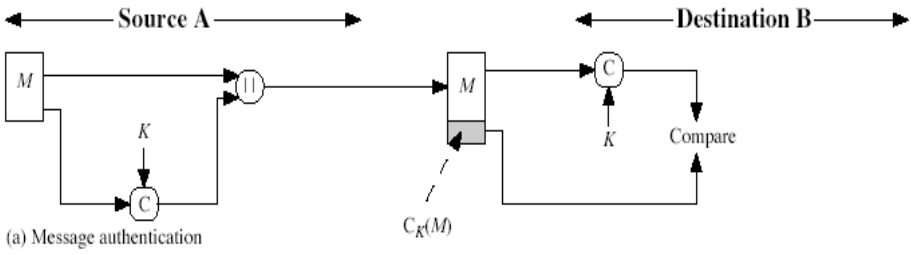
- Để lộ bí mật: đọc nội dung mẩu tin không được phép.
- Thám mã đường truyền: theo dõi hoặc phân tích đường truyền.
- Giả mạo: lấy danh nghĩa người khác để gửi tin.
- Sửa đổi nội dung: thay đổi, cắt xén, thêm bớt thông tin.
- Thay đổi trình tự các gói tin nhỏ của mẩu tin truyền.
- Sửa đổi thời gian: làm trì hoãn mẩu tin.
- Từ chối gốc: người gửi từ chối trách nhiệm là tác giả mẩu tin.
- Từ chối đích: người nhận phủ định sự tồn tại và đến đích của mẩu tin đã gửi.

Mã mẩu tin

- Mã mẩu tin bản thân đã cung cấp một phần tính xác thực, vì khóa được chia sẻ giữa người gửi và người nhận cũng như việc thay đổi nội dung cũng không dễ dàng thực hiện nếu không có khóa.
- Cụ thể nếu mã đối xứng được sử dụng, thì người nhận biết người gửi phải tạo ra mẩu tin, vì chỉ có người gửi và người nhận biết được khóa sử dụng.
- Người nhận có thể biết nội dung không bị sửa đổi, nếu mẩu tin có cấu trúc phù hợp, tính dư thừa và tổng kiểm tra để phát hiện bất cứ thay đổi nào.
- Nếu khóa công khai được sử dụng thì mã cung cấp không đủ độ tin cậy về người gửi, vì mọi người đều có thể biết khóa công khai của người nhận. Tuy nhiên nếu người gửi ký mẩu tin sử dụng khóa riêng của họ và sau đó mã với khóa công khai của người nhận, thì khi đó đảm bảo cả tính bảo mật và xác thực của mẩu tin. Cần phải bổ sung các biện pháp để phát hiện các mẩu tin đã bị làm hỏng. Việc sử dụng khóa riêng của người gửi kết hợp với khóa công khai của người nhận có nhiều ưu việt, nhưng với giá phải trả là chậm do dùng hai mã khóa công khai trên mẩu tin.

5.4.2. Mã xác thực mẩu tin (MAC – Message Authentication Code)

- Mã xác thực mẩu tin MAC sinh ra bởi một thuật toán mà tạo ra một khối thông tin nhỏ có kích thước cố định
  - Phụ thuộc vào cả mẩu tin và khóa nào đó ;
  - Giống như mã nhưng không cần phải giải mã.
- Bổ sung vào mẩu tin như chữ ký để gửi kèm theo làm bằng chứng xác thực.
- Người nhận thực hiện tính toán nào đó trên mẩu tin và kiểm tra xem nó có phù hợp với MAC đính kèm không.
- Tạo niềm tin rằng mẩu tin không bị thay đổi và đến từ người gửi.



Các mã xác thực mẩu tin MAC cung cấp sự tin cậy cho người nhận là mẩu tin không bị thay đổi và được gửi từ đích danh người gửi. Cũng có thể sử dụng mã xác thực MAC kèm theo với việc mã hóa để bảo mật. Nói chung người ta sử dụng các khóa riêng biệt cho mỗi MAC và có thể tính MAC trước hoặc sau mã hóa, tốt hơn là thực hiện MAC trước và mã hóa sau.

Sử dụng MAC có nhược điểm là MAC phụ thuộc vào cả mẩu tin và cả người gửi, nhưng đôi khi chỉ cần xác thực mẩu tin và thông tin xác thực đó chỉ phụ thuộc mẩu tin để lưu trữ làm bằng chứng cho tính toàn vẹn của nó. Và các hàm MAC không được phổ biến, chia sẻ dùng chung. Khi đó người ta sử dụng hàm Hash thay vì MAC. Cần lưu ý rằng MAC không phải là chữ ký điện tử, vì cả người gửi và người nhận đều biết thông tin về khóa.

Các tính chất của MAC

MAC là thông tin nén của mẩu tin kết hợp với khóa MAC = $C_K(M)$

- Nén bản tin M có độ dài tùy ý;
- Sử dụng khóa mật K;
- Tạo nên dấu xác thực có độ dài cố định;
- Là hàm nhiều - một, nghĩa là có nhiều bản tin khác nhau nhưng có cùng MAC. Tuy nhiên ta phải lựa chọn hàm MAC sao cho xác suất để các mẩu tin có ý nghĩa có MAC trùng nhau là rất nhỏ. Việc tìm được các mẩu tin như vậy là rất khó khăn.

Yêu cầu đối với MAC

Tùy thuộc vào kiểu tấn công mà MAC phải có các tính chất khác nhau để chống đối lại. Nhưng nói chung MAC phải thỏa mãn các điều sau:

- Biết mẩu tin và MAC, không thể tìm được mẩu tin khác có cùng MAC.
- Các MAC cần phải phân bố đều.

- MAC phải phụ thuộc như nhau vào tất cả các bit trong mẩu tin. Tức là khi thay đổi một bit thông tin nào đó, MAC sẽ có những thay đổi kéo theo.

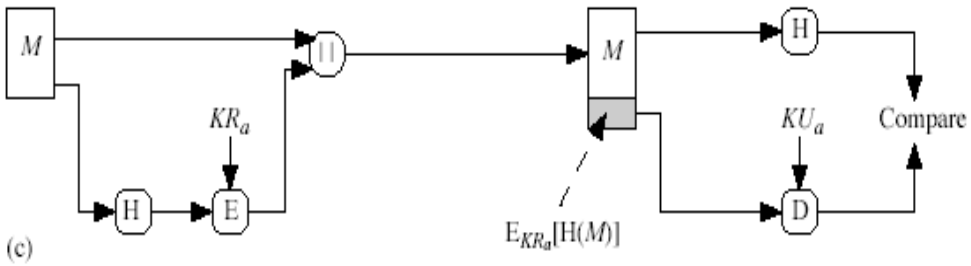
Sử dụng mã đối xứng cho MAC

- Có thể dùng mã khối với chế độ chuỗi móc nối bất kỳ và sử dụng khối cuối cùng của mã khối làm MAC của mẩu tin.
- *Thuật toán xác thực dữ liệu (DAA – Data Authentication Algorithm)* là MAC được sử dụng rộng rãi dựa trên chế độ DES-CBC, trong đó
  - Sử dụng véc tơ ban đầu IV = 0 và bộ đệm 0 của block cuối cùng.
  - Và mã mẩu tin sử dụng chuẩn mã dữ liệu DES trong chế độ CBC.
  - Gửi lấy block cuối cùng như là MAC của cả mẩu tin.
    - Hoặc M bit trái nhất ($16 \le M \le 64$) của khối cuối cùng.
    - Bây giờ MAC cuối cùng với kích thước 64 bit cũng là quá nhỏ để đảm bảo an toàn. Do đó người ta tìm cách tạo nên các MAC có kích thước lớn hơn.

5.5. Các hàm Hash (hay còn gọi là hàm băm)

5.5.1. Mục đích

- Nén mẩu tin bất kỳ về kích thước cố định. Và giả thiết là hàm hash là công khai và không dùng khóa. Hash chỉ phụ thuộc mẩu tin, còn MAC phụ thuộc thêm cả vào khóa.
- Hash được sử dụng để phát hiện thay đổi của mẩu tin. Hash có thể sử dụng nhiều cách khác nhau với mẩu tin, Hash thường được kết hợp dùng để tạo chữ ký trên mẩu tin.



5.5.2. Các tính chất của hàm Hash

- Hàm Hash tạo nên dấu vân tay (tức là thông tin đặc trưng) của một tệp, mẩu tin hay dữ liệu

$$h = H(M)$$

- Nén mẩu tin có kích thước tùy ý về dấu vân tay có kích thước cố định. Hàm Hash được giả thiết là công khai, mọi người đều biết cách sử dụng

Các yêu cầu của hàm Hash

- Có thể áp dụng cho mọi mẩu tin có kích thước tùy ý. Tuy nhiên phải tạo đầu ra h có kích thước cố định, thường là 128 bit đến 1024 bit.
- Dễ tính h = H(M) cho mọi mẩu tin M, hàm H tính toán nhanh, hiệu quả phụ thuộc chặt vào mẩu tin M và không tính toán ngược lại.

- Cho trước h không thể tìm được (rất khó) x sao cho H(x) = h. Tính chất này gọi là tính chất một chiều, chiều tìm nghịch ảnh rất khó khăn, tuy chiều tìm ảnh lại dễ dàng.
- Cho x không thể tìm được y sao cho H(y) = H(x). Đây là tính chất chống đỡ va chạm yếu, không tìm được mẩu tin có cùng Hash với mẩu tin đã cho.
- Và không thể tìm được x, y sao cho H(y) = H(x). Đây gọi là tính chất chống đỡ va chạm mạnh, đây là yêu cầu cao hơn tính chống đỡ va chạm yếu.

5.5.3. Các hàm hash đơn giản

Có một số đề xuất cho một số hàm hash đơn giản. Chẳng hạn biểu diễn mẩu tin dưới dạng bit, sau đó chia chúng thành các khối bit có kích thước bằng kích thước mong muốn của Hash. Rồi dựa trên phép toán XOR các bit thông tin ở cùng vị trí tương ứng của các khối, kết quả nhận được là Hash của cả mẩu tin. Hàm hash trên là không an toàn vì đối với mẩu tin bất kỳ có thể tìm được mẩu tin mà có cùng hàm hash. Cần phải có hàm mạnh hơn mà sẽ xét trong phần sau.

	bit 1	bit 2	• • •	bit *n*
block 1	b_{11}	b_{21}		b_{n1}
block 2	b_{12}	b_{22}		b_{n2}
	•	•	•	•
	•	•	•	•
	•	•	•	•
block *m*	b_{1m}	b_{2m}		b_{nm}
hash code	C_1	C_2		C_n

Figure 3.3 Simple Hash Function Using Bitwise XOR

Tấn công ngày sinh nhật

Có thể nghĩ hash 64 bit là an toàn, có nghĩa là khó tìm được bản tin có cùng hash. Nhưng không phải vậy vì *nghịch lý ngày sinh nhật như sau*: trong lớp có ít nhất bao nhiêu sinh viên, để xác suất có ít nhất hai sinh viên trùng ngày sinh nhật là lớn hơn 0,5. Theo lý thuyết xác suất thống kê gọi số sinh viên ít nhất trong lớp là k, khi đó xác suất q để không có 2 người nào trùng ngày sinh là tỷ số giữa cách chọn k ngày khác nhau trong 365 ngày trên số cách chọn k ngày bất kỳ trong 365 ngày. Vậy

$$q = C^k_{365} / 365^k$$

Do đó, xác suất p để có ít nhất 2 người trùng ngày sinh là

$$p = 1 - q = 1 - C^k_{365} / 365^k$$

Để p > 0,5, thì k > 22 hay k = 23, cụ thể khi đó p = 0,5073.

Khi chưa tính toán chi tiết chúng ta nghĩ là trong lớp phải có ít nhất khoảng 365/2 tức là 184 sinh viên. Nhưng trên thực tế con số đó ít hơn rất nhiều, chỉ cần 23 sinh viên, chính vì vậy ta gọi đây là nghịch lý ngày sinh nhật.

Điều đó muốn nói lên rằng, trong nhiều trường hợp xác suất để hai mẩu tin có cùng bản Hash là không nhỏ như chúng ta tưởng.

Tấn công ngày sinh nhật hoạt động như sau:

- Kẻ thám mã tạo ra $2^{m/2}$ biến thể của mẩu tin đúng, mà tất cả đều có bản chất ngữ nghĩa như nhau, với m ở đây là độ dài của bản mã hash.
- Kẻ thám mã cũng có thể tạo ra $2^{m/2}$ biến thể khác nhau của mẩu tin lừa dối, tức là có ngữ nghĩa ngược lại
- Hai tập tin được so sánh với nhau để tìm cặp có cùng bản hash (xác suất lớn hơn hoặc bằng 0,5 dựa vào nghịch lý ngày sinh nhật)
- Người dùng ký vào mẩu tin đúng, sau đó bị thay thế bằng mẩu tin giả mà cũng có chữ ký đúng.
- Kết luận là cần phải dùng MAC và Hash có kích thước lớn hơn nữa.

Mã khối như hàm Hash

- Có thể sử dụng mã khối như hàm Hash
  - Sử dụng $H_0 = 0$ và bộ đệm không cho khối cuối cùng;
  - Tính $H_i = E_{Mi} [H_{i-1}]$;
  - Và sử dụng khối cuối cùng như giá trị hàm hash;
  - Giống chế độ CBC, nhưng không có khóa;
- Hash kết quả quá nhỏ (64 bit)
  - Cả vì tấn công sinh nhật trực tiếp;
  - Cả tấn công ở trung gian;
- Các phương án khác cũng dễ bị tấn công.

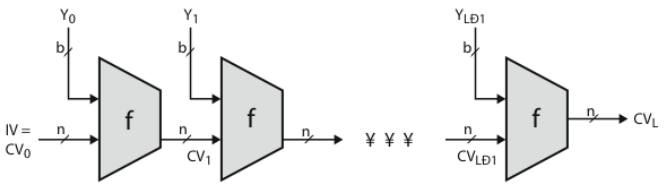
Kẻ thám mã cũng có thể tạo ra $2^{m/2}$ biến thể khác nhau của mẩu tin.

5.5.4. Tính an toàn của hàm Hash và MAC.

Giống như đối với mã khối, hàm hash cũng có tấn công vét cạn, cụ thể: Hash chống va chạm mạnh có giá $2^{m/2}$, có nghĩa là với m là độ dài mã hash, thì $2^{m/2}$ xác định sức mạnh của nó chống đối lại tấn công vét cạn. Ta cần lựa chọn m đủ lớn để việc duyệt tìm $2^{m/2}$ phương án là không khả thi. Có đề xuất Hash 128 bit cho MD5 phần cứng. Nhưng có thể tìm được va chạm sau 24 ngày. Do đó có thể coi là hash 128 bit có thể có lỗ hổng, không an toàn, tốt hơn dùng hash 160 bit.

Tấn công vét cạn trên MAC khó hơn, vì chúng đòi hỏi một cặp MAC của mẩu tin đã biết, do nó phụ thuộc thêm vào khóa. Có thể tấn công vào không gian khóa (như là tìm khóa) hoặc MAC. Độ dài ít nhất 128 bit MAC là cần thiết để đảm bảo an toàn.

Cấu trúc thuật toán Hash



5.5.5. Thuật toán Hash an toàn SHA (Secure Hash Algorithm)

SHA có nguồn gốc từ Viện chuẩn công nghệ quốc gia Hoa Kỳ - NIST & NSA vào năm 1993, sau đó được nâng cấp vào 1995 theo chuẩn US và chuẩn là FIPS 180-1 1995 và Internet RFC3174, được nhắc đến như SHA-1. Nó được sử dụng với sơ đồ chữ ký điện tử DSA (Digital Signature Algorithm).

Thuật toán là SHA dựa trên thiết kế MD4 với một số khác biệt tạo nên giá trị Hash 160 bit. Các kết quả nghiên cứu 2005 về an toàn của SHA-1 đề xuất sử dụng nó trong tương lai.

5.5.6. Thuật toán SHA-1

Mô tả thuật toán. Đầu vào của thuật toán là một thông điệp có chiều dài bất kỳ nhỏ hơn 2^{64} bit, SHA-1 cho ra kết quả là một thông điệp rút gọn có độ dài là 160 bit. Đầu vào được xử lý theo các khối 512 bit. Thuật toán SHA1 được thực hiện theo các bước sau:

- Bổ sung bộ đệm bít. Thông điệp được bổ sung sao cho độ dài của nó đồng dư với 448 mod 512.
- Bổ sung độ dài. Thêm khối 64 bit vào cuối thông điệp, nó biểu diễn độ dài thực của thông điệp.
- Khởi tạo bộ đệm. 160 bit bộ đệm được sử dụng để giữ giá trị trung gian và cuối cùng của bản băm, gồm năm thanh ghi 32 bit: A, B, C, D, E.
- Xử lý các khối dữ liệu 512 bit hay 16 từ 32 bit gồm bốn vòng, mỗi vòng 20 bước. Bốn vòng sử dụng bốn hàm logic f_1, f_2, f_3, f_4.

Mở rộng thông điệp

Thông điệp M được mở rộng trước khi thực hiện băm. Mục đích của việc mở rộng này là để đảm bảo cho thông điệp mở rộng có độ dài là bội số của 512. Thông điệp M được mở rộng trước khi thực hiện băm. Mục đích của việc mở rộng này là để đảm bảo cho thông điệp mở rộng có độ dài là bội số của 512.

Thông điệp M được mở rộng trước khi thực hiện băm. Mục đích của việc mở rộng này là để đảm bảo cho thông điệp mở rộng có độ dài là bội số của 512.

Giả sử độ dài của thông điệp là L bit. Thêm bit 1 vào cuối thông điệp, theo sau là k bit 0 (k là số dương không âm nhỏ nhất sao cho L + 1 + k = 448 (mod 512)) . Sau đó thêm khối 64 bit là biểu diễn nhị phân của L.

Bốn hàm logic với 20 bước trong mỗi vòng, f(t;B,C,D) được định nghĩa như sau:

f(t;B,C,D) = (B AND C) OR ((NOT B) AND D) $(0 \le t \le 19)$

f(t;B,C,D) = B XOR C XOR D $(20 \le t \le 39)$

f(t;B,C,D) = (B AND C) OR (B AND D) OR (C AND D) $(40 \le t \le 59)$

f(t;B,C,D) = B XOR C XOR D $(60 \le t \le 79)$.

Phân tích thông điệp mở rộng:

Sau khi thông điệp đã được mở rộng, thông điệp mở rộng được phân tích thành N khối 512 bit M(1), M(2),…, M(N). Trong đó 512 bit của khối dữ liệu đầu vào có thể được thể hiện bằng 16 từ 32 bit,

Khởi tạo giá trị băm:

Giá trị băm là một chuỗi bit có kích thước bằng kích thước của thông điệp băm gồm các từ ghép lại. Trong đó $H_j^{(i)}$ là từ j trong giá trị băm ở lần lặp i với $0 \le i \le N$ (số block có được sau khi chia văn bản được đệm) và $0 \le j \le$ (số từ trong giá trị băm - 1). Trước khi thực hiện giá trị băm, với mỗi thuật toán băm an toàn, giá trị băm ban đầu H(0) phải được thiết lập. Kích thước và số lượng từ trong H(0) tùy thuộc vào kích thước thông điệp băm rút gọn.

SHA-1 sử dụng dãy các hằng số K(0),…,K(79) có giá trị như sau:

K(t) = 5A827999 (0 <= t <= 19)

K(t) = 6ED9EBA1 (20 <= t <= 39)

K(t) = 8F1BBCDC (40 <= t <= 59)

K(t) = CA62C1D6 (60 <= t <= 79).

Thuật toán của bước tính giá trị băm SHA-1

SHA-1 được sử dụng để băm thông điệp M có độ dài L bit thoả mãn điều kiện $0 \le L \le 2^{64}$. Thuật toán sử dụng:

- Một bảng phân bố thông điệp gồm 80 từ 32 bit
- 5 biến 32 bit
- Một giá trị băm gồm 5 từ 32 bit

Kết quả của SHA-1 là một thông điệp băm rút gọn có độ dài 160 bit. Các từ của bảng phân bố thông điệp được ký hiệu W(0),W(1),…,W(79). 5 biến được ký hiệu là A, B, C, D, E. Các từ của giá trị băm ký hiệu $H_0^{(i)}$,$H_1^{(i)}$, $H_2^{(i)}$, $H_3^{(i)}$,$H_4^{(i)}$. H(0) giữ giá trị băm ban đầu và được thay thế bằng các giá trị băm thành công. H(i) sau mỗi khối thông điệp được xử lý và kết thúc bằng giá trị băm cuối cùng H(N).

Tính toán thông điệp băm

Định nghĩa: $S^n(X) = (X << n)$ or $(X >> 32-n)$.

X << n có nghĩa là loại bỏ từ trái sang phải n bit và thêm vào kết quả n số 0 vào bên phải. X >> n có nghĩa là loại bỏ từ phải qua trái n bit và thêm vào kết quả n số 0 vào bên trái.

Khởi tạo các giá trị của H:

H0 = 67452301 ; H1 = EFCDAB89

H2 = 98BADCFE ; H3 = 10325476

H4 = C3D2E1F0.

Chia khố 512 bit M(i) thành 16 từ W(0), W(1),…,W(15)

For t = 16 to 79

$W(t) = S^1(W(t-3)$ XOR W(t-8) XOR W(t-14) XOR W(t-16)).

Đặt A = H0 , B = H1, C = H2, D = H3, E = H4

For t = 0 to 79 do

$TEMP = S^5(A) + f(t;B,C,D) + E + W(t) + K(t)$;

$E = D;\ D = C;\ C = S^{30}(B);\ B = A;\ A = TEMP;$

Đặt H0 = H0 + A, H1 = H1 + B, H2 = H2 + C, H3 = H3 + D, H4 = H4+ E.

Sau khi tính toán được hết M(n), thông điệp băm rút gọn là một chuỗi 160 bit là biểu diễn của 5 từ: H0 H1 H2 H3 H4

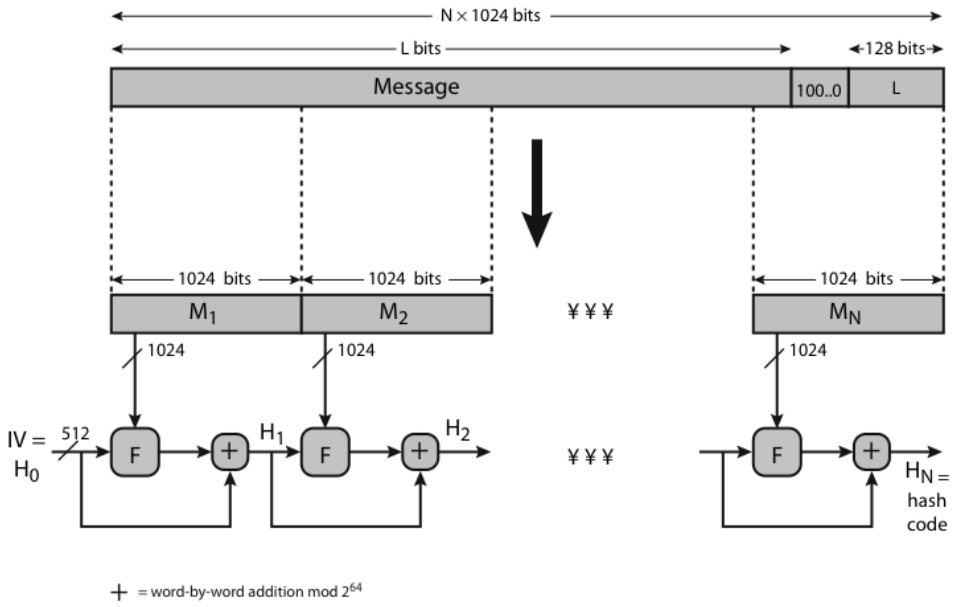
Đánh giá thuật toán

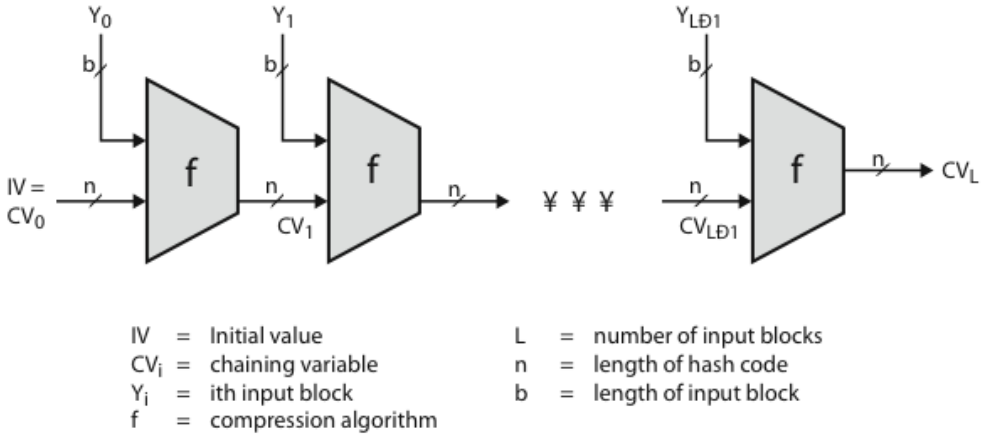
- SHA-1 được xem là an toàn đối với hiện tượng đụng độ vì rất khó tìm được hai thông điệp khác nhau có giá trị băm giống nhau
- SHA-1 được coi là chuẩn của việc bảo vệ các kênh liên lạc trực tuyến tồn tại trong 9 năm qua.
- SHA-1 được thiết kế cho bộ xử lý 32 bit, thế hệ sắp tới của máy tính dùng các bộ xử lý 64 bit mà SHA-1 không hiệu quả trên bộ xử lý này.
- Tháng 2 năm 2005 SHA-1 bị tấn công bởi ba chuyên gia người Trung Quốc. Thuật toán này đã bị giải mã thông qua phương pháp tính phân bổ.

Chuẩn Hash an toàn nâng cao

Viện chuẩn công nghệ quốc gia NIST xuất bản bản sửa FIPS 180-2 vào năm 2002, đề nghị bổ sung ba phiên bản mới của SHA: SHA-256, SHA-384, SHA-512. Các phiên bản trên được thiết kế tương thích với việc tăng độ an toàn được cung cấp bởi chuẩn mã nâng cao AES. Về cấu trúc và chi tiết giống SHA-1, suy ra việc phân tích cũng tương tự, nhưng mức độ an toàn cao hơn nhiều so với SHA-1.

Tổng quan SHA 512



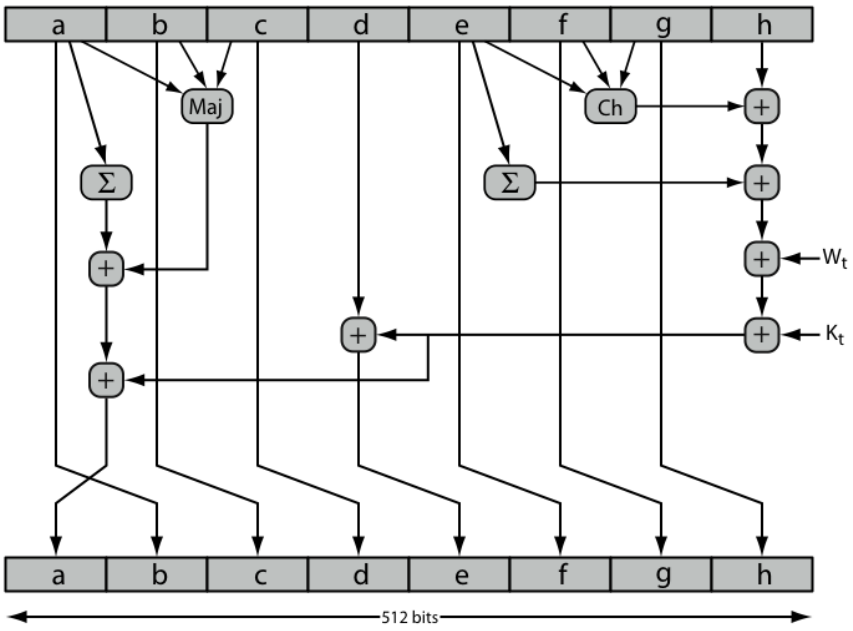


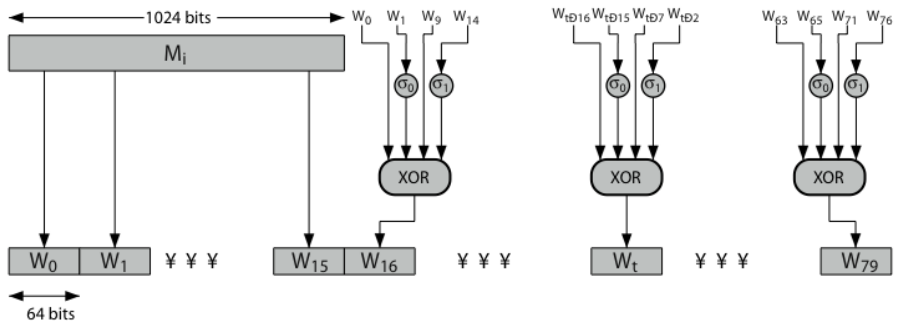
Hàm nén SHA-512

SHA-512 là trọng tâm của thuật toán. Ở đây xử lý mẩu tin với các khối 1024 bit và bao gồm 80 vòng:

- Cập nhật bộ đệm 512 bit;
- Sử dụng giá trị Wt 64 bit được lấy ra từ block hiện tại của mẩu tin;
- Và hằng số quay vòng dựa trên căn bậc ba của 80 số nguyên tố đầu tiên.

Hàm quay vòng của SHA-512





Hàm Hash có khóa giống như MAC

Khi đã có các hàm Hash tốt, chúng ta muốn có các mã xác thực mẩu tin MAC dựa trên các hàm Hash đó. Vì hàm Hash thông thường nhanh hơn và mã nguồn của hàm Hash được phổ biến rộng rãi hơn, nên việc sử dụng chúng tạo nên MAC sẽ hiệu quả hơn. Ta có thể coi MAC là Hash bao gồm cả khóa với mẩu tin, cụ thể được đề xuất như sau:

KeyedHash = Hash (Key | Message)

Trong trường hợp này có một số điểm yếu đã được tìm thấy. Chính vì muốn khắc phục các điểm yếu đó, một phương án kết hợp Hash để tạo nên MAC được phát triển là HMAC.

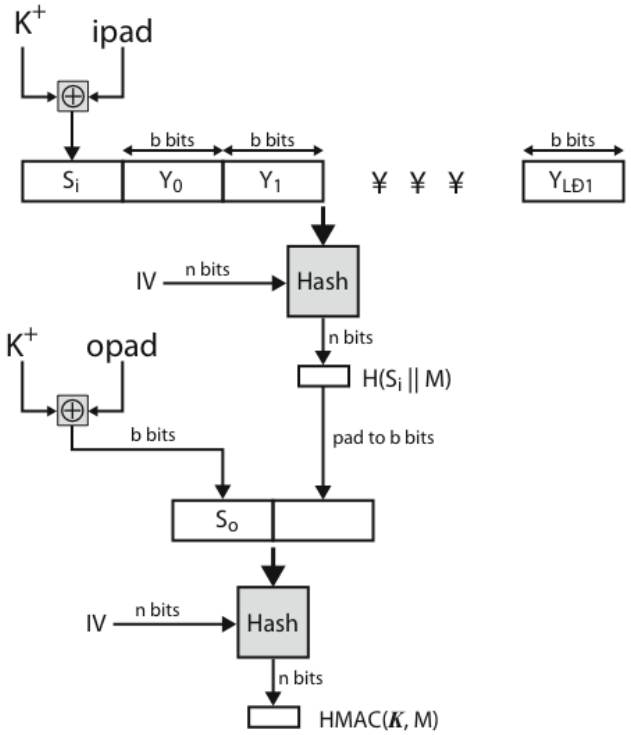
HMAC

HMAC được thiết kế theo chuẩn Internet RFC2104, sử dụng hàm Hash trên mẩu tin:

$$HMAC_K = Hash [(K^+ \text{ XOR opad}) || Hash [(K^+ \text{ XOR ipad}) || M)]]$$

trong đó K^+ là khóa đệm mở rộng của K và opad, ipad là các hằng bộ đệm đặc biệt, M là mẩu tin. Như vậy cần tính nhiều hơn 3 hàm Hash so với nếu bản tin đứng một mình. Bất cứ hàm Hash nào cũng có thể được sử dụng trong sơ đồ trên: MD5, SHA-1, RIPEMD-160 hay Whirlpool.

Tổng quan HMAC



An toàn HMAC

Sự an toàn được chứng minh liên quan đến thuật toán Hash nền trong sơ đồ trên. Tấn công HMAC yêu cầu phải hoặc:

- Tấn công vét cạn khóa đã sử dụng;
- Tấn công ngày sinh nhật (tuy cần quan sát số lượng rất lớn mẩu tin).

Có thể lựa chọn phán đoán hàm Hash được sử dụng dựa trên tốc độ và các ràng buộc an toàn.

5.6. CHỮ KÝ ĐIỆN TỬ

Chữ ký điện tử của một người sử dụng trên một mẩu tin tại một thời điểm xác định được xem như nén mẩu tin về một kích thước cố định và được xác thực. Nó cung cấp các khả năng để

- Kiểm chứng tác giả, ngày và giờ ký;
- Xác thực nội dung mẩu tin;
- Được kiểm chứng bởi bên thứ ba để chống từ chối.

Vì vậy việc tạo chữ ký điện tử bao gồm việc sử dụng một hàm băm bản tin và một hàm xác thực có một số khả năng bổ sung như nhận bản băm, thông tin mật của người ký, một số thông tin ngẫu nhiên đặc trưng cho thời điểm ký để tạo ra chữ ký điện tử đính kèm mẩu tin.

Đồng thời thuật toán cũng hỗ trợ người nhận kiểm tra chữ ký điện tử kèm với mẩu tin đó.

5.6.1. Các tính chất của chữ ký điện tử

- Cần phải phụ thuộc vào nội dung ký.

- Cần sử dụng thông tin đặc trưng duy nhất đối với người gửi.
  Để chống cả giả mạo và từ chối.
- Cần phải tương đối dễ dàng tạo ra.
- Dễ dàng đoán nhận và kiểm chứng.
- Không thể tính toán giả mạo được.
  - Với bản tin mới và chữ ký đã có.
  - Với chữ ký giả mạo cho 1 bản tin.
- Có thể lưu trữ chữ ký điện tử.

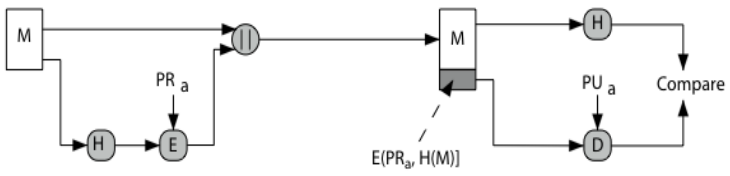
Chữ ký điện tử trực tiếp

- Bao gồm chỉ người gửi và người nhận;
- Người nhận được phép có khóa công khai của người nhận;
- Quan trọng là ký trước và sau đó mã hóa bản tin và chữ ký;
- Tính an toàn phụ thuộc vào khóa riêng của người gửi.

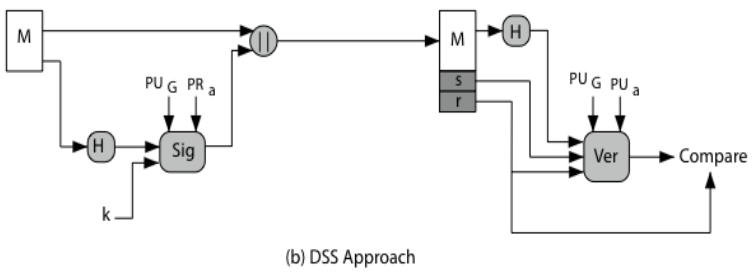
Chữ ký điện tử RSA: chẳng hạn khi A muốn tạo chữ ký RSA, A băm thông điệp, rồi mã hóa nó bằng khóa riêng của A và đính kèm với thông điệp gửi cho B. Người S muốn kiểm tra thông điệp có toàn vẹn không và có phải gửi từ A không, sẽ băm thông điệp nhận được, đồng thời giải mã bản băm đính kèm với thông điệp bằng khóa oông khai của A.. Và so sánh hai kết quả nhận được ở bước trước, nếu chúng trùng nhau, thì thông điệp không bị sửa và đúng là do A gửi.

Chữ ký điện tử có trọng tài

- Bao gồm cả việc sử dụng trọng tài A
  - Kiểm tra mọi mẩu tin được ký;
  - Sau đó điền ngày giờ và gửi cho người nhận.
- Đòi hỏi mức độ tin cậy hợp lý đối với trọng tài.
- Có thể cài đặt với thuật toán khóa riêng hoặc khóa công khai.
- Trọng tài có thể được xem hoặc không được xem mẩu tin.



(a) RSA Approach



(b) DSS Approach

5.6.2. Chuẩn chữ ký điện tử (DSS)

- Chính phủ Mỹ ủng hộ sơ đồ chữ ký điện tử FIPS 186.
- Sử dụng thuật toán hash SHA.
- DSS là chuẩn và DSA là thuật toán.
- Có phương án cải biên Elgamal và Schnorr.
- Tạo 320 bit chữ ký và độ an toàn 512-1024 bit an toàn.
- An toàn phụ thuộc vào độ khó của tính logarit rời rạc.

Sinh khóa DSA

- Chia sẻ giá trị khóa công khai tổng thể (p, q, g):
  - Số nguyên tố lớn $p = 2^L$, ở đó L= 512 đến 1024 bit và là bội của 64;
  - Chọn q là số nguyên tố 160 bit và là ước của p-1;
  - Chọn $g = h^{(p-1)/q}$, ở đó $h < p-1$, $h^{(p-1)/q} \pmod p > 1$;
- Người sử dụng chọn khóa riêng và tính khóa công khai
  - Chọn khóa riêng: $x < q$
  - Tính khóa công khai: $y = g^x \pmod p$

Tạo chữ ký DSA

Để ký mẩu tin M người gửi trước hết cần:

- Sinh khóa chữ ký ngẫu nhiên k: $k < p$, k phải là số ngẫu nhiên, được xóa sau khi dùng và không bao giờ dùng lại;
- Sau đó tính cặp chữ ký:

$$r = (g^k \pmod p) \pmod q$$

$$s = (k^{-1}(SHA(M) + x.r) \pmod q$$

- Gửi cặp chữ ký (r, s) cùng với bản tin M.

Kiểm chứng chữ ký DSA

- Nhận được bản tin M cùng với chữ ký (r, s)
- Để kiểm chứng chữ ký người nhận cần tính:

$$w = s^{-1} \pmod q$$

$$u1 = (SHA(M).w) \pmod q$$

$$u2 = (r.w) \pmod q$$

$$v = (g^{u1}.y^{u2} \pmod p) \pmod q$$

- Nếu v = r thì chữ ký đã được kiểm chứng

Ví dụ 3. *Tạo chữ ký điện tử:*

- Chọn p = 23, q = 11, h = 7
  $g = h^2 \bmod 23 = 3$
- Chọn x = 5, $y = 3^5 \bmod 23 = 13$
  k = 6, H(M) = 9
  $r = (g^k \pmod p) \pmod q$

r = (3^6 mod 23) mod 11 = (4^2) mod 11 = 5

s = (k^{-1}(H(M)+ x.r))(mod q)

s = (6^{-1}.(9 + 5.5))mod 11 = 2

- Chữ ký điện tử (5,2)

Kiểm tra chữ ký điện tử

- w = 2^{-1} mod 11 = 6
- u1 = (9.6) mod 11 = 10
- u2 = (5.6) mod 11 = 8
- v = (310.138 mod 23) mod 11 = (43.3.138 mod 23) mod 11
  = (16) mod 11 = 5
- v = 5 = r, chữ ký đúng

Như vậy, bản tin nhận được là toàn vẹn, không bị sửa và thư được gửi bởi đúng người như đã tuyên bố.

5.7. Các bài tập

5.7.1. Mã công khai RSA

- Chọn ngẫu nhiên 2 số nguyên tố p và q
- Tính: N = p.q; Φ(N) = (p - 1).(q - 1)
- Người dùng A chọn ngẫu nhiên khoá công khai (hoặc riêng) e: 1 < e < Φ(N), gcd(e, Φ(N)) = 1.
- Tìm khóa riêng (hoặc công khai) d của A: (e.d) mod Φ(N) = 1, 0 < d < Φ(N).
- Để mã hoá mẩu tin gửi cho A, người gửi B:
  - Tính C = M^e mod n, trong đó 0≤ M < n.
  - Để giải mã, người sở hữu khóa riêng:
  - Tính M = C^d mod n
- Để ký mẩu tin M gửi cho B, người gửi A mã bằng khóa riêng của mình:
  - Tính C = M^d mod n, trong đó 0 ≤ M < n.
  - Để kiểm tra chữ ký, người nhận giải mã bằng khóa công khai của người gửi:
  - Tính M = C^e mod n
- Cho p = 3; q = 11; khóa công khai e = 7; thông điệp M = 5.
  - N = 3.11 = 33; Φ(N) = 2.10 = 20;
  - d = e^{-1} mod Φ(N) = 7^{-1}mod 20 = 3, khóa riêng d = 3;
  - Mã: C = M^e mod n = 5^7mod 33 = (-8)(-2).5 mod33 = 14;
  - Giải mã: M = C^d mod n =14^3mod 33 = (-2).14 mod 33 = 5.
- Cho p = 5; q = 11; khoá riêng e = 3; thông điệp M = 9.
  - N = 5.11 = 55 ; Φ(N) = 4.10 = 40;
  - d = e^{-1} mod Φ(N) = 3^{-1} mod 40 = 27, khóa công khai d = 27;
  - Ký: C = M^e mod n = 9^3 mod 55 = 26.9 mod 55 = 14;
  - Kiểm tra chữ ký: M = C^d mod n = 14^{27} mod55 = (14^{16}.14^8.14^2.14) mod 55

= (36.16.31.14) mod 55 = (26(-6)) mod 55 = 9;

- $14^2 \bmod 55 = 31$, $14^4 \bmod 55 = 26$, $14^8 \bmod 55 = 16$, $14^{16} \bmod 55 = 36$.

- Cho p = 7; q = 11; khoá công khai e = 13; thông điệp M = 3.
  - N = 7.11 = 77; Φ(N) = 6.10 = 60;
  - Khóa riêng $d = e^{-1} \bmod \Phi(N) = 13^{-1} \bmod 60 = 37$;
  - Mã: $C = M^e \bmod n = 3^{13} \bmod 77 = (3^8 3^4 3) \bmod 77 = (4^2.4.3) \bmod 77 = 38$;
  - Giải mã: $M = C^d \bmod n = 38^{37} \bmod 77 = 3$.
- Có thể dùng định lý phần dư Trung Hoa để giải mã cho nhanh:
  - Tính $C^d \bmod 7 = 38^{37} \bmod 7 = 3^{37} \bmod 7 = 3^{36}.3 \bmod 7 = 3$;
  - Tính $C^d \bmod 11 = 38^{37} \bmod 11 = 5^{37} \bmod 11 = 5^{30}.5^7 \bmod 11 = 3$;
  - Tính $a_1 = 11^{-1} \bmod 7 = 4^{-1} \bmod 7 = 2$;
  - Tính $a_2 = 7^{-1} \bmod 11 = 8$;
  - $c_1 = 11.(11^{-1} \bmod 7) = 11.2 = 22$;
  - $c_2 = 7(7^{-1} \bmod 11) = 7.8 = 56$;
- Vậy $M = (a_1c_1 + a_2c_2) \bmod 77 = (3.22 + 3.56) \bmod 77 = 3$.

5.7.2. Trao đổi khóa DIFFIE - HELLMAN

- Mọi người dùng thỏa thuận dùng tham số chung:
  - Lấy số nguyên tố rất lớn q;
  - Chọn α là căn nguyên tố của q.
- Mỗi người dùng (A chẳng hạn) tạo khoá của mình:
  - Chọn một khoá mật (số) $x_A < q$;
  - Tính khoá công khai $y_A = \alpha^{x_A} \bmod q$
  - Mỗi người dùng thông báo công khai khóa của mình y_A
- Khóa bộ phận dùng chung cho hai người sử dụng A, B là K_{AB}
  - $K_{AB} = \alpha^{x_A.x_B} \bmod q$

    $= y_A^{x_B} \bmod q$ (mà B có thể tính)

    $= y_B^{x_A} \bmod q$ (mà A có thể tính)
- Hai người dùng A và B muốn trao đổi khoá phiên:
  - Đồng ý chọn số nguyên tố q = 11 và α = 2;
  - A chọn khoá riêng $x_A = 9$; B chọn khóa riêng $x_B = 3$;
  - Tính các khoá công khai:

    $y_A = \alpha^{x_A} \bmod q = 2^9 \bmod 11 = 6$

    $y_B = \alpha^{x_B} \bmod q = 2^3 \bmod 11 = 8$
- Tính khoá phiên chung:

$K_{AB} = y_B^{x_A} \bmod q = 8^9 \bmod 11 = 7$ (A)

$K_{AB} = y_A^{x_B} \bmod q = 6^3 \bmod 11 = 7$ (B)

- Hai người sử dụng A và B muốn trao đổi khoá phiên:
  - Đồng ý chọn số nguyên tố q = 13 và α = 6
  - A chọn khoá riêng $x_A = 5$; B chọn khóa riêng $x_B = 7$
  - Tính các khoá công khai:

$y_A = \alpha^{x_A} \bmod q = 6^5 \bmod 13 = 2$

$y_B = \alpha^{x_B} \bmod q = 6^7 \bmod 13 = 7$

- Tính khoá phiên chung:

$K_{AB} = y_B^{x_A} \bmod q = 7^5 \bmod 13 = 11$ (A)

$K_{AB} = y_A^{x_B} \bmod q = 2^7 \bmod 13 = 11$ (B)

5.7.3. Chữ ký điện tử DSA

Bài tập:

- Chọn p = 23, q = 11, h = 7

  chọn $g = h^{(p-1)/q} \pmod p$

  ở đó $h < p-1$; $h^{(p-1)/q} \pmod p > 1$

- $g = h^2 \bmod 23 = 3$
- Chọn x = 4, $y = 3^4 \bmod 23 = 12$

Tạo chữ ký điện tử

  - k = 5, H(M) = 8
  - $r = (g^k \pmod p) \pmod q$

    $r = (3^5 \bmod 23) \bmod 11 = (12.3 \bmod 23) \bmod 11 = 2$
  - $s = (k^{-1}(H(M) + x.r)) \pmod q$

    $s = (5^{-1}.(8 + 4.r)) \bmod 11 = (5^{-1}.(8 + 4.2)) \bmod 11 = 1$

- Chữ ký điện tử (r,s) = (2,1)

Kiểm tra chữ ký điện tử

$w = s^{-1} \pmod q$

$u1 = (H(M).w) \pmod q$

$u2 = (r.w) \pmod q$

$v = (g^{u1}.y^{u2} \pmod p) \pmod q$

  - $w = 1^{-1} \bmod 11 = 1$
  - $u_1 = 8.1 \bmod 11 = 8$

- $u_2 = 2.1 \bmod 11 = 2$
- $v = (3^8.12^2 \bmod 23) \bmod 11 = 2$
- v = r, chữ ký điện tử đúng

TÓM LƯỢC CUỐI BÀI

- Khái niệm mã công khai.
- Mã công khai RSA.
- Trao đổi khóa Diffie-Hellman.
- Xác thực thông điệp, mã xác thực MAC.
- Hàm băm an toàn SHA.
- Chữ ký điện tử DSA.

CÂU HỎI TRẮC NGHIỆM CUỐI BÀI

Câu 1: Xét khoá công khai, tìm kết luận sai trong các khẳng định sau
A. Khoá công khai thông báo cho mọi người biết
B. Người sử dụng phải giữ bí mật khoá riêng của mình
C. Tính an toàn dựa vào độ khó của bài toán cho khoá công khai tìm khoá riêng
D. Không có thuật toán tính được khoá riêng khi biết khoá công khai

Câu 2: Xét khoá công khai, tìm kết luận đúng trong các khẳng định sau
A. Khoá riêng thông báo cho mọi người biết
B. Người sử dụng phải giữ bí mật khoá riêng của mình
C. Tính an toàn dựa vào độ khó của bài toán cho khoá riêng tìm khoá công khai
D. Không có thuật toán tính được khoá riêng khi biết khoá công khai

Câu 3: Xét mã RSA, tìm kết luận sai trong các khẳng định sau
A. Độ an toàn dựa vào độ khó của bài toán phân tích 1 số ra thừa số
B. Tính an toàn dựa vào độ khó bài toán nhân hai số nguyên tố rất lớn
C. Dựa trên lũy thừa trường hữu hạn các số nguyên modulo nguyên tố
D. Sử dụng các số rất lớn 1024 bit

Câu 4: Xét mã RSA, tìm kết luận sai trong các khẳng định sau
A. Chỉ dùng mã các dữ liệu nhỏ
B. Kết hợp với mã đối xứng
C. Tính an toàn dựa vào độ khó bài toán logarit rời rạc
D. Kết hợp hàm hash tạo chữ ký điện tử

Câu 5: Trao đổi khoá Diffie Hellman là thủ tục giữa 2 người sử dụng để
A. Trao đổi khoá công khai
B. Trao đổi khoá mật bằng khoá công khai
C. Trao đổi xác nhận khoá công khai (gồm khoá công khai và danh tính)
D. Trao đổi khoá mật mới bằng khoá mật cũ

Câu 6: Sự an toàn của trao đổi khoá Diffie Hellman dựa trên
A. Việc trao đổi trên kênh riêng của 2 người sử dụng
B. Thông qua bên đối tác thứ ba tin cậy
C. Độ khó của bài toán logarit rời rạc
D. Độ mật của khoá dùng chung cũ

Câu 7: Tìm khẳng định sai trong các câu sau về mã xác thực bản tin
A. Mã xác thực là bản nén của một bản tin về kích thước cố định
B. Mã xác thực phụ thuộc vào bản tin và khoá
C. Mã xác thực có vai trò như chữ ký điện tử

D. Cả bên nhận và bên gửi đều biết thuật toán nén và khoá

Câu 8: Tìm khẳng định đúng trong các câu sau về mã xác thực bản tin

A. Mã xác thực có thể giải mã để nhận lại bản tin
B. Mã xác thực phụ thuộc vào bản tin và khoá
C. Mã xác thực có vai trò như chữ ký điện tử
D. Bên nhận không biết thuật toán mã xác thực và khoá

Câu 9: Tìm khẳng định sai trong các câu sau về hàm hash

A. Hash phụ thuộc vào bản tin và khoá
B. Hash là bản nén của một bản tin về kích thước cố định
C. Hash được coi là dấu vân tay xác định tính toàn vẹn của bản tin
D. Hash dùng kết hợp với khoá công khai tạo chữ ký điện tử trên bản tin

Câu 10: Tìm khẳng định đúng trong các câu sau về hàm hash

A. Hash phụ thuộc vào bản tin và khoá
B. Hash có vai trò như chữ ký điện tử trên bản tin
C. Hash được coi là dấu vân tay xác định tính toàn vẹn của bản tin
D. Có thể giải mã Hash để khôi phục lại bản tin

Câu 11: Tìm khẳng định đúng trong các câu sau về hàm hash

A. Hash phụ thuộc vào bản tin và khoá
B. Hash có vai trò như chữ ký điện tử trên bản tin
C. Dễ dàng tìm 2 bản tin có cùng Hash
D. Hash dùng kết hợp với khoá công khai tạo chữ ký điện tử trên bản tin

Câu 12: Tìm khẳng định sai về chữ ký điện tử trong các câu sau

A. Chữ ký điện tử phụ thuộc bản tin và người ký
B. Chữ ký điện tử xác nhận người gửi và nội dung gửi
C. Dùng chữ ký điện tử chống từ chối người gửi
D. Người nhận có thể tạo ra chữ ký điện tử của người gửi trên bản tin để so sánh

Câu 13: Tìm khẳng định đúng về chữ ký điện tử trong các câu sau

A. Chữ ký điện tử chỉ phụ thuộc vào người ký, không phụ thuộc bản tin
B. Chữ ký điện tử xác nhận người gửi và nội dung gửi
C. Dùng chữ ký điện tử chống từ chối người nhận
D. Người nhận có thể tạo ra chữ ký điện tử của người gửi trên bản tin để so sánh

Câu 14: Xét chuẩn chữ ký điện tử DSS, khẳng định nào là sai

A. Chọn bộ tham số (p, q, g) gồm 2 số nguyên tố và một căn nguyên tố
B. Mỗi người sử dụng chọn khoá riêng và tính khoá công khai
C. Khi gửi bản tin ký, chọn số ngẫu nhiên và tính hai thành phần chữ ký
D. Người nhận chỉ dùng một thành phần chữ ký tính thành phần kia rồi so sánh với thành phần thứ hai đính kèm

Câu 15: Chữ ký điện tử DSA, khẳng định nào là đúng

A. Chữ ký người gửi giống nhau trên mọi bản tin
B. Người nhận cũng có thể tạo chữ ký như người gửi
C. Khi gửi bản tin ký, chọn số ngẫu nhiên và tính hai thành phần chữ ký
D. Người nhận chỉ dùng một thành phần chữ ký tính thành phần kia rồi so sánh với thành phần thứ hai đính kèm

ĐÁP ÁN CÂU HỎI TRẮC NGHIỆM CUỐI BÀI

Câu 1: D, có thuật toán, nhưng tính rất khó, lâu
Câu 2: B, Người sử dụng phải giữ bí mật khoá riêng của mình
Câu 3: B, An toàn dựa vào bài toán khó phân tích ra tíach hai số nguyên tố rất lớn
Câu 4: C, Tính an toàn dựa vào độ khó bài toán câu 3
Câu 5: B, Ttrao đổi khoá mật bằng khoá công khai
Câu 6: C, Độ khó của bài toán logarit rời rạc

Câu 7: C, Mã xác thực không phải là chữ ký điện tử, cả hai người đều có thể tạo ra
Câu 8: B, Mã xác thực phụ thuộc vào bản tin và khoá
Câu 9: A, Hash chỉ phụ thuộc vào bản tin
Câu 10: C, Hash được coi là dấu vân tay xác định tính toàn vẹn của bản tin
Câu 11: D, Hash dùng kết hợp với khoá công khai tạo chữ ký điện tử trên bản tin
Câu 12: D, Người nhận có thể tạo ra chữ ký điện tử của người gửi trên bản tin để so sánh
Câu 13: B, Chữ ký điện tử xác nhận người gửi và nội dung gửi
Câu 14: D, Người nhận chỉ dùng một thành phần chữ ký tính thành phần kia rồi so sánh với thành phần thứ hai đính kèm
Câu 15: C, Khi gửi bản tin ký, chọn số ngẫu nhiên và tính hai thành phần chữ ký

THUẬT NGỮ TRONG BÀI

- Mã khoá đối xứng còn được gọi là mã khoá đơn hay mật. Ở đây chỉ dùng một khoá, dùng chung cả người nhận và người gửi. Khi khoá này được dùng, việc trao đổi thông tin về khoá sẽ được thỏa thuận trước.
- Mã khoá công khai ra đời vào đầu những năm 1970. Có thể nói đây là bước tiến quan trọng nhất trong lịch sử 3000 năm mã hoá. Ở đây người ta sử dụng 2 khoá: một khoá riêng và một khoá công khai. Hai khoá này khác nhau, không đối xứng với nhau, do đó mã khoá công khai, còn được gọi là mã không đối xứng.
- RSA là mã công khai được sáng tạo bởi Rivest, Shamir & Adleman ở MIT (Trường Đại học Công nghệ Massachusetts) vào năm 1977. RSA là mã công khai được biết đến nhiều nhất và sử dụng rộng rãi nhất hiện nay. Nó dựa trên các phép toán lũy thừa trong trường hữu hạn các số nguyên theo modulo nguyên tố.
- Trao đổi khoá Diffie Hellman là sơ đồ khoá công khai đầu tiên được đề xuất bởi Diffie và Hellman năm 1976 cùng với khái niệm khoá công khai. Đây là phương pháp thực tế trao đổi công khai các khoá mật đối xứng.
- Xác thực mẩu tin liên quan đến các khía cạnh sau khi truyền tin trên mạng
    - Bảo vệ tính toàn vẹn của mẩu tin: bảo vệ mẩu tin không bị thay đổi hoặc có các biện pháp phát hiện nếu mẩu tin bị thay đổi trên đường truyền.
    - Kiểm chứng danh tính và nguồn gốc: xem xét mẩu tin có đúng do người xưng tên gửi không hay một kẻ mạo danh nào khác gửi.
    - Không chối từ bản gốc: trong trường hợp cần thiết, bản thân mẩu tin chứa các thông tin chứng tỏ chỉ có người xưng danh gửi, không một ai khác có thể làm điều đó. Như vậy người gửi không thể từ chối hành động gửi, thời gian gửi và nội dung của mẩu tin.
- Mã xác thực thông điệp: Sinh ra bởi một thuật toán mà tạo ra một khối thông tin nhỏ có kích thước cố định: phụ thuộc vào cả mẩu tin và khoá nào đó, giống như mã nhưng không cần phải giải mã, bổ sung vào mẩu tin như chữ ký để gửi kèm theo làm bằng chứng xác thực.
- Hash – băm: nén mẩu tin bất kỳ về kích thước cố định. Giả thiết là hàm hash là công khai và không dùng khoá. Hash chỉ phụ thuộc mẩu tin. Hash được sử dụng để phát hiện thay đổi của mẩu tin. Hash có thể sử dụng nhiều cách khác nhau với mẩu tin, Hash thường được kết hợp dùng để tạo chữ ký trên mẩu tin.

- SHA: thuật toán băm an toàn (Secure Hash Algorithm). SHA có nguồn gốc từ Viện chuẩn công nghệ quốc gia Hoa kỳ - NIST & NSA vào năm 1993, sau đó được nâng cấp vào 1995 theo chuẩn US và chuẩn là FIPS 180-1 1995 và Internet RFC3174. Nó được sử dụng với sơ đồ chữ ký điện tử DSA.
- Chữ ký điện tử: được xem như mẩu tin có kích thước cố định được xác thực cung cấp các khả năng để kiểm chứng tác giả, ngày và giờ ký, xác thực nội dung mẩu tin, được kiểm chứng bởi bên thứ 3 để chống từ chối. Vì vậy bao gồm hàm xác thực và một số khả năng bổ sung
- DSS : Chuẩn chữ ký điện tử, được chính phủ Mỹ ủng hộ từ sơ đồ chữ ký điện tử FIPS 186. Sử dụng thuật toán hash SHA và thuật toán DSS. Tạo 320 bit chữ ký và độ an toàn 512-1024 bit, an toàn phụ thuộc vào độ khó của tính logarit rời rạc.

CÂU HỎI THƯỜNG GẶP

1. Mã công khai là gì? Khóa riêng và khóa công khai dành cho ai?
2. Tại sao biết khóa công khai lại không biết được khóa riêng? An toàn khóa công khai dựa vào đâu?
3. Nêu cách mã công khai dùng trong bảo mật mẩu tin? Tại sao chỉ dùng mã hóa thông điệp có kích thước nhỏ
4. Nêu cách khóa công khai xác thực người gửi? Và nếu muốn bảo mật cho người nhận thi cần phải làm gi?
5. Nêu các đặc trưng của RSA? an toàn dựa vào đâu? Mô tả sơ đồ sinh khóa RSA trên thực tế?
6. Nêu cách mã và giải mã RSA? Muốn tăng tốc độ cần phải làm gì?
7. Nêu các cách tấn công thám mã RSA
8. Thuật toán Diffie-Hellman dùng để làm gì? Mô tả việc sinh khóa công khai.
9. Nêu cách 2 người sử dụng tính khóa mật dùng chung? Độ an toán D-H dựa trên cơ sở nào?
10. Thế nào là xác thực thông điệp?
11. Có các cách nào xác thực thông điệp. Ưu, nhược dùng Mac
12. Định nghĩa Hash. Các tính chất cần có của hash.
13. Thế nào là nghịch lý ngày sinh nhật. Tấn công dựa vào nó như thế nào
14. Mô tả các thao tác của hàm băm SHA1?
15. Nêu định nghĩa chữ ký điện tử? Các loại chữ ký?
16. Môt tả quá trình sinh khóa, ký và kiểm tra chữ ký DSA?

TRẢ LỜI CÂU HỎI THƯỜNG GẶP

1. Mã công khai là mã dùng 2 khóa khác nhau cho 1 NSD; khóa công khai dành cho mọi người mã hoá gửi thông điệp cho người đó hoặc giải mã thông điệp từ người đó, mã riêng dành riêng cho người đó ký hoặc giải mã.
2. Do thuật toán tính khóa riêng từ khóa công khai là bài toán khó, đòi hỏi nhiều thời gian, độ an toàn dựa vào độ khó của bài toán này
3. NSD B dùng khóa công khai của A, mã hóa và gửi cho A. Mã mẩu tin ngắn, vì thời gian mã hóa và giải mã lâu.
4. NSD A mã hóa thông điệp bằng khóa riêng của mình, rồi gửi cho B. Nếu muốn chỉ B nhận được thì sau khi mã bằng khoá riêng của mình A mã hóa tiếp bằng khóa công khai của B.

5. RSA có kích thước khóa riêng và khóa công khai cỡ 512 đến 1024 bit. Độ an toàn dựa vào phân tích 1 số lớn cỡ 1024 thành 2 số cỡ 512 bit. Chọn hai số p, q nguyên tố cùng nhau. Chọn: e.d=1 mod ø(N) với 0≤d≤ ø(N)
6. Mã: tính $C = M^e \bmod n$, giải mã: tính $M = C^d \bmod n$. Muốn nhanh sử dụng Định lý phần dư Trung hoa.
7. Tấn công RSA có 3 dạng
    a. Phân tích n = p.q, sau đó tính ø(n) và d
    b. Tìm n trực tiếp và tính d
    c. Tìm d trực tiếp
8. Là sơ đồ trao đổi dùng khoá công khai, có thể thiết lập khoá chung dựa trên khóa riêng của hai đối tác. Chỉ có 2 đối tác tính được khóa chung đó,
9. Giá trị khoá phụ thuộc vào thông tin khoá công khai của đối tác và khoá riêng của mình. Độ an toàn dựa trên độ khó của bài toán tính logarit rời rạc
10. Xác thực mẩu tin liên quan đến
    a. Bảo vệ tính toàn vẹn của mẩu tin
    b. Kiểm chứng danh tính và nguồn gốc
    c. Không chối từ bản gốc
11. Có 2 hàm lựa chọn: Mã xác thực mẩu tin (MAC), Hàm hash. Sinh ra bởi một thuật toán mà tạo ra một khối nhỏ kích thước cố định, phụ thuộc vào cả mẩu tin và khoá nào đó, giống như mã nhưng không cần phép toán ngược lại. Thuật toán MAC ít công khai, phổ cập, nhiều khi không cần khóa
12. Hash: nén mẩu tin bất kỳ về kích thước cố định như nên dấu vân tay của mẩu tin, giả thiết là hàm hash là công khai và không dùng khoá
13. Nghịch lý ngày sinh nhật: trong lớp có ít nhất 23 sinh viên, để xác xuất có 2 bạn trùng ngày sinh nhật lớn hơn hoặc bằng 0.5. Tấn công ngày sinh nhật hoạt động như sau
    a. Kẻ thám mã tạo ra $2^{m/2}$ biến thể của mẩu tin đúng mà tất cả đều có bản chất ngữ nghĩa như nhau và tạo ra $2^{m/2}$ biến thể khác nhau của mẩu tin lừa dối
    b. Hai tập tin được so sánh với nhau để tìm cặp có cùng bản hash. Người dùng ký vào mẩu tin đúng, sau đó bị thay thế bằng mẩu tin giả mà cũng có chữ ký đúng.
14. Tạo nên giá trị Hash 160 bit. Xem bài giảng
15. Chữ ký điện tử cung cấp các khả năng để
    a. Kiểm chứng tác giả, ngày và giờ ký
    b. Xác thực nội dung mẩu tin
    c. Được kiểm chứng bởi bên thứ 3 để chống từ chối

    Có chữ ký trực tiếp và chữ ký có trọng tài
16. DSA tạo 320 bit chữ ký
    a. Với lựa chọn 512-1024 bit an toàn hơn
    b. Nhỏ và nhanh hơn RSA
    c. Có sơ đồ chữ ký điện tử - xem bài giảng
    d. An toàn phụ thuộc vào độ khó của tính logarit rời rạc

CÂU HỎI TỰ LUẬN

Câu 1. Mã công khai làm được điều gì mà mã đối xứng không làm được?

Câu 2. Tại sao nói mã đối xứng dành cho bảo mật thông điệp, còn mã công khai dành cho xác thực thông điệp?

Câu 3. Tại sao nói mã công khai dựa trên các cặp bài toán thuận - dễ, nghịch – khó trong lý thuyết số?

Câu 4. Nêu đặc trưng, quá trình sinh khóa, mã hóa và giải mã RSA.

Câu 5. Giải thích tại sao người sử dụng sở hữu khóa riêng có thể áp dụng định lý phần dư Trung Hoa để tăng tốc độ tính toán?

Câu 6. Nêu các cách thám mã RSA, qua đó đánh giá độ an toàn của nó.

Câu 7. Nêu khó khăn của bài toán phân phối khóa mật giữa hai người sử dụng. Diffie-Hellman đề xuất hướng giải quyết như thế nào?

Câu 8. Mô tả chi tiết thủ tục trao đổi khóa Diffie-Hellman. Nêu cơ sở tính an toàn của thủ tục đó.

Câu 9. Thế nào là xác thực thông điệp? Nêu một số ví dụ cụ thể về yêu cầu xác thực.

Câu 10. Có các phương pháp nào xác thực thông điệp? Nêu cách sử dụng MAC, nhược điểm của nó.

Câu 11. Thế nào là bản băm của một thông điệp? Nó cần có các tính chất gì và được dùng để xác thực như thế nào?

Câu 12. Nghịch lý ngày sinh nhật là gì? Thám mã dựa vào nghịch lý này như thế nào?

Câu 13. Mô tả thuật toán băm an toàn SHA1.

Câu 14. Thế nào là chữ ký điện tử cho thông điệp của một người nào đó? Nó cần có tính chất gì? Mô tả chữ ký điện tử sử dụng RSA.

Câu 15. Nêu cách tạo chữ ký điện tử DSA. Nó có phụ thuộc vào số ngẫu nhiên được chọn không?

Câu 16. Nêu cách người nhận kiểm tra chữ ký điện tử. Kết quả kiểm tra cho biết điều gì?

BÀI TẬP TRẮC NGHIỆM

1. Xét mã khóa công khai, tìm kết luận sai trong các khẳng định sau:
   A. Khóa công khai thông báo cho mọi người biết;
   B. Người sử dụng phải giữ bí mật khóa riêng của mình;
   C. Người sử dụng sở hữu khóa riêng cũng rất khó để tính được khóa công khai của mình;
   D. Tính an toàn dựa vào độ khó của bài toán cho khóa công khai tìm khóa riêng.

2. Xét mã RSA, tìm kết luận sai trong các khẳng định sau**:**
   A. Độ an toàn dựa vào độ khó của bài toán phân tích một số ra thừa số;
   B. Tính toán dựa trên phép toán lũy thừa theo modulo số nguyên;
   C. Tính an toàn dựa vào độ khó bài toán lũy thừa với số mũ lớn;
   D. Sử dụng các số rất lớn khoảng 512 đến 1024 bit.

3. Trao đổi khóa Diffie Hellman là thủ tục giữa hai người sử dụng để
   A. Trao đổi khóa công khai;
   B. Trao đổi giấy chứng nhận khóa công khai;
   C. Trao đổi khóa mật dùng chung giữa hai người sử dụng một cách công khai;
   D. Trao đổi khóa mật mới bằng khóa mật cũ.

4. Sự an toàn của trao đổi khóa Diffie Hellman dựa trên
   A. Việc trao đổi trên kênh riêng của hai người sử dụng;
   B. Thông qua bên đối tác thứ ba tin cậy;
   C. Độ mật của khóa dùng chung cũ;
   D. Độ khó của bài toán logarit rời rạc.

5. Tìm khẳng định sai trong các câu sau về mã xác thực bản tin
   A. Mã xác thực là bản nén của một bản tin về kích thước cố định;
   B. Mã xác thực phụ thuộc vào bản tin và khóa;
   C. Cả bên nhận và bên gửi đều biết thuật toán nén và khóa;
   D. Dễ dàng tìm được thông điệp có mã xác thực cho trước.

6. Tìm khẳng định đúng trong các câu sau về mã xác thực bản tin MAC
   A. Mã xác thực có thể giải mã để nhận lại bản tin;
   B. Mã xác thực có vai trò như chữ ký điện tử của người gửi;
   C. Mã xác thực có độ dài cố định phụ thuộc vào bản tin và khóa;
   D. Bên nhận không cần khóa có thể kiểm tra tính toàn vẹn của bản tin.

7. Tìm khẳng định sai trong các câu sau về hàm hash
   A. Hash là bản nén của một bản tin về kích thước cố định;
   B. Hash được coi là dấu vân tay đặc trưng xác định tính toàn vẹn của bản tin;
   C. Hash phụ thuộc vào bản tin và thông tin mật của người dùng;
   D. Hash dùng kết hợp với khóa công khai tạo chữ ký điện tử trên bản tin.

8. Tìm khẳng định đúng trong các câu sau về hàm hash
   A. Hash phụ thuộc vào bản tin và người gửi;
   B. Hash có vai trò như chữ ký điện tử trên bản tin;
   C. Có thể giải mã Hash để khôi phục lại bản tin;
   D. Hash được coi là dãy bit đặc trưng cho bản tin.

9. Tìm khẳng định đúng trong các câu sau về hàm hash
   A. Hash phụ thuộc vào bản tin và khóa;
   B. Hash có vai trò như chữ ký điện tử trên bản tin;
   C. Hash gửi kèm với bản tin để kiểm tra tính xác thực của nó;
   D. Dễ dàng tìm 2 bản tin có cùng Hash.

10. Thuật toán nào không phải là thuật toán hash
   a) MD5;
   b) DSA ;
   c) SHA-1 ;
   d) MD4.

11. Tìm khẳng định đúng về chữ ký điện tử trong các câu sau:
   A. Chữ ký điện tử chỉ phụ thuộc vào người ký, không phụ thuộc bản tin;
   B. Dùng chữ ký điện tử chống từ chối người nhận;
   C. Chữ ký điện tử xác nhận người gửi, nội dung gửi và thời gian gửi;
   D. Người nhận có thể tạo ra chữ ký điện tử của người gửi trên bản tin để kiểm tra.

12. Khẳng định nào là sai về chữ ký điện tử DSS:
   A. Chọn một số nguyên tố cỡ 512 bit và một nguyên tố khác cỡ 160 bit là ước của số trước bớt 1;
   B. Mỗi người sử dụng chọn khóa riêng và tính khóa công khai;
   C. Người nhận sử dụng khóa riêng của người gửi để kiểm tra chữ ký;
   D. Khi gửi bản tin ký, chọn số ngẫu nhiên và tính hai thành phần chữ ký.

13. Khẳng định nào là đúng về chữ ký điện tử DSS:
    A. Chữ ký người gửi giống nhau trên mọi bản tin;
    B. Người nhận cũng có thể tạo chữ ký của người gửi để kiểm tra;
    C. Dùng thuật toán RSA để tạo chữ ký;
    D. Khi gửi chữ ký kèm bản tin, chọn số ngẫu nhiên, dùng khóa riêng và bản băm tính hai thành phần chữ ký.

14. Cho p = 11, q = 13, khóa riêng của A là e = 71. Tính khóa công khai RSA của A
    A. d = 2;
    B. d = 3;
    C. d = 4;
    D. d = 5.

15. Cho p = 7, q = 13, khóa riêng của A là e = 23. Tính khóa công khai RSA của A
    A. d = 2;
    B. d = 4;
    C. d = 3;
    D. d = 5.

16. Cho p = 13, q = 11, khóa riêng của A là e = 71 và khóa công khai d = 2. Cho bản tin M = 67, tìm bản mã RSA bằng khóa công khai của A:
    A. C = 121;
    B. C = 129;
    C. C = 135;
    D. C = 27.

17. Cho q = 11 và $\alpha = 2$. Người sử dụng A chọn $x_A = 3$, B chọn $x_B = 2$. Khẳng định nào về trao đổi khóa Diffie-Hellman sau là đúng:
    A. Khóa công khai A, B là $y_A = 4$, $y_B = 8$, khóa mật chung $K_{AB} = 7$;
    B. Khóa công khai A, B là $y_A = 4$, $y_B = 4$, khóa mật chung $K_{AB} = 8$;
    C. Khóa công khai A, B là $y_A = 8$, $y_B = 4$, khóa mật chung $K_{AB} = 9$;
    D. Khóa công khai A, B là $y_A = 8$, $y_B = 8$, khóa mật chung $K_{AB} = 10$.

18. Cho q = 11 và $\alpha = 3$. Người sử dụng A chọn $x_A = 3$, B chọn $x_B = 2$. Khẳng định nào về trao đổi khóa Diffie-Hellman sau là đúng:
    A. Khóa công khai A, B là $y_A = 4$, $y_B = 8$, khóa mật chung $K_{AB} = 5$;
    B. Khóa công khai A, B là $y_A = 5$, $y_B = 9$, khóa mật chung $K_{AB} = 3$;
    C. Khóa công khai A, B là $y_A = 8$, $y_B = 4$, khóa mật chung $K_{AB} = 9$;
    D. Khóa công khai A, B là $y_A = 8$, $y_B = 8$, khóa mật chung $K_{AB} = 4$.

19. Chữ ký điện tử DSA: cho p = 23 và q = 11 và h = 5. Tính g. Bạn chọn một khóa riêng x = 7, rồi tính khóa công khai y và báo cho bạn thân của bạn. Bạn gửi bức thư có bản băm H(M) = 9 và chọn một số ngẫu nhiên k tùy ý, chẳng hạn k = 6, rồi ký và gửi cho bạn thân của bạn. Nêu cách bạn thân đó kiểm tra chữ ký.
    A. (r, s) = (7, 2);
    B. (r, s) = (7, 1);
    C. (r, s) = (5, 1);

D. (r, s) = (7, 3).

20. Chữ ký điện tử DSA: cho p = 47 và q = 23 và h = 7. Tính g. Bạn chọn một khóa riêng x = 13, rồi tính khóa công khai y và báo cho bạn thân của bạn. Bạn gửi bức thư có bản băm H(M) = 11 và chọn một số ngẫu nhiên k tuỳ ý, chẳng hạn k = 5, rồi ký và gửi cho bạn thân của bạn. Nêu cách bạn thân đó kiểm tra chữ ký.
    a) (r, s) = (8, 5);
    b) (r, s) = (8, 4);
    c) (r, s) = (7, 5);
    d) (r, s) = (7, 4).

BÀI TẬP ÔN TẬP

1. Cho hệ thống mã RSA với p = 3; q = 11; khóa công khai của người sử dụng A là e = 7:
    a. Tìm khóa riêng của A
    b. Tìm bản mã của thông điệp M = 5 do một người khác mã bằng khóa công khai của A
2. Cho hệ thống mã RSA với p = 3; q = 11; khóa riêng của người sử dụng A là d = 3: Sử dụng Định lý phần dư Trung Hoa giải mã C = 14 bằng khóa riêng của A.
3. Cho hệ thống mã RSA với p = 5; q = 11; khóa riêng của người sử dụng A là d = 7:
    a. Tìm khóa công khai của A
    b. Sử dụng Định lý phần dư Trung Hoa tìm mã của thông điệp M = 9 do A ký bằng khóa riêng của A.
4. Cho hệ thống mã RSA với p = 5; q = 11; khóa công khai của người sử dụng A là e = 27. Tìm bản rõ của bản mã C = 14 (mà A ký) do một người khác giải mã bằng khóa công khai của A.
5. Cho hệ thống mã RSA với p = 7; q = 11; khóa công khai của người sử dụng A là e = 13:
    a. Tìm khóa riêng của A
    b. Tìm bản mã của thông điệp M = 3 do một người khác mã bằng khóa công khai của A
6. Cho hệ thống mã RSA với p = 7; q = 11; khóa riêng của người sử dụng A là d = 37: Sử dụng Định lý phần dư Trung Hoa giải mã C = 14 (do người khác gửi cho A) bằng khóa riêng của A.
7. Trong hệ thống PKI, mỗi người sử dụng có ít nhất một khóa riêng và một khóa công khai được chứng thực. Trong hệ thống mã RSA với p = 7; q = 11; khóa riêng của người sử dụng A là d = 37.
    a. Tìm khóa công khai của A.
    b. Sử dụng Định lý phần dư Trung Hoa tìm chữ ký RSA của người sử dụng A cho bản băm thông điệp H(M) = 2 do A ký bằng khóa riêng.

c. Nêu cách người khác kiểm tra chữ ký RSA của A.

8. Trong hệ thống trao đổi công khai khóa dùng chung Diffie-Hellman giữa hai người sử dụng A và B, cho các tham số chung: số nguyên tố $q = 11$ và $\alpha = 2$; A chọn khoá riêng $x_A = 9$; B chọn khóa riêng $x_B = 3$. Tính

   a. Khóa công khai của A và B

   b. Nêu cách A tính khóa dung chung giữa A và B

   c. Nêu cách B tính khóa dung chung giữa B và A.

9. Trong hệ thống trao đổi công khai khóa dùng chung Diffie-Hellman giữa hai người sử dụng A và B, cho các tham số chung: số nguyên tố $q = 13$ và $\alpha = 6$; A chọn khoá riêng $x_A = 5$; B chọn khóa riêng $x_B = 7$. Tính

   a. Khóa công khai của A và B.

   b. Nêu cách A tính khóa dung chung giữa A và B.

   c. Nêu cách B tính khóa dung chung giữa B và A.

10. Trong hệ thống chữ ký điện tử DSS, chọn p = 23, q = 11, h = 7

    a. Tính tham số dùng chung g.

    b. Giả sử người sử dụng A có khóa riêng $X_A = 4$. Tính khóa công khai Y_A của A.

    c. Cho thông điệp M có bản băm H(M) = 8 và số ngẫu nhiên k = 5. Tìm chữ ký DSS của người sử dụng A trên M.

    d. Giả sử người sử dụng B nhận được thông điệp M có H(M) = 9 và chữ ký (r,s) = (2,1). Nêu cách B kiểm tra chữ ký của A.

    e. Giả sử người sử dụng B nhận được thông điệp M có H(M) = 8 và chữ ký (r,s) = (2,1). Nêu cách B kiểm tra chữ ký của A.

11. Cho p = 11; q = 13; A chọn khoá công khai 7, tính khóa riêng của A. Giả sử B sử dụng khoá công khai của A mã hoá bản tin M = 5. Tính bản mã và giải mã

A. $P_{RA} = 23$; C = 37; B. $P_{RA} = 103$; C = 47;

C. $P_{RA} = 53$; C = 27; D. $P_{RA} = 73$; C = 57;

12. Trao đổi khóa Diffie-Hellman: cho $q = 17$, $\alpha = 10$, $x_A = 7$, $x_B = 5$. Tính y_A ; y_B và khoá chung K_{AB}.

A. $y_A = 5$; $y_B = 4$; $K_{AB} = 11$; B. $y_A = 2$; $y_B = 7$; $K_{AB} = 9$;

C. $y_A = 5$; $y_B = 11$; $K_{AB} = 14$; D. $y_A = 5$; $y_B = 6$; $K_{AB} = 15$;

13. Cho p = 47 và q = 23 và h = 7. Tính g. Bạn chọn khoá riêng x = 13, rồi tính khoá công khai y. Bạn gửi bức thư có bản băm H(M) = 11 và chọn một số ngẫu nhiên k = 5, rồi ký. Nêu cách người nhận kiểm tra chữ ký. Sinh chữ ký

A. g = 3, y = 15, r = 7, s = 13; B. g = 2, y = 20, r = 10, s = 11;

C. g = 2, y = 20, r = 9, s = 15;
D. g = 2, y = 20, r = 10, s = 19;

14. Kiểm tra chữ ký trong câu 15

A. w = 17, u1 = 3, u2 = 7, v = 10;
B. w = 15, u1 = 4, u2 = 9, v = 11;
C. w = 17, u1 = 3, u2 = 9, v = 10;
C. w = 17, u1 = 3, u2 = 8, v = 10;