
KSTP.Ebook

Chương 4.
Hàm băm xác thực và chữ kí số

Nội dung

- Giới thiệu
- 4.1 Các hàm băm và tính toàn vẹn của dữ liệu
- 4.2 Trao đổi và thoả thuận khoá
- 4.3 Hệ mật dựa trên định danh
- 4.4 Các sơ đồ chữ kí số không nén
- 4.5 Các sơ đồ chữ kí số có nén

Giới thiệu

- **Một số khái niệm:**
  - 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.

Giới thiệu

- **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 như sau:
    - Để lộ bí mật: giữ bí mật nội dung mẩu tin, chỉ cho người có quyền biết.
    - Thám mã đường truyền: không cho theo dõi hoặc làm trì hoãn việc truyền tin.
    - 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.

Giới thiệu

- Sửa đổi thời gian: làm trì hoãn mẩu tin.
- Từ chối gốc: không cho phép người gửi từ chối trách nhiệm của tác giả mẩu tin.
- Từ chối đích: không cho phép người nhận phủ định sự tồn tại và đến đích của mẩu tin đã gửi.

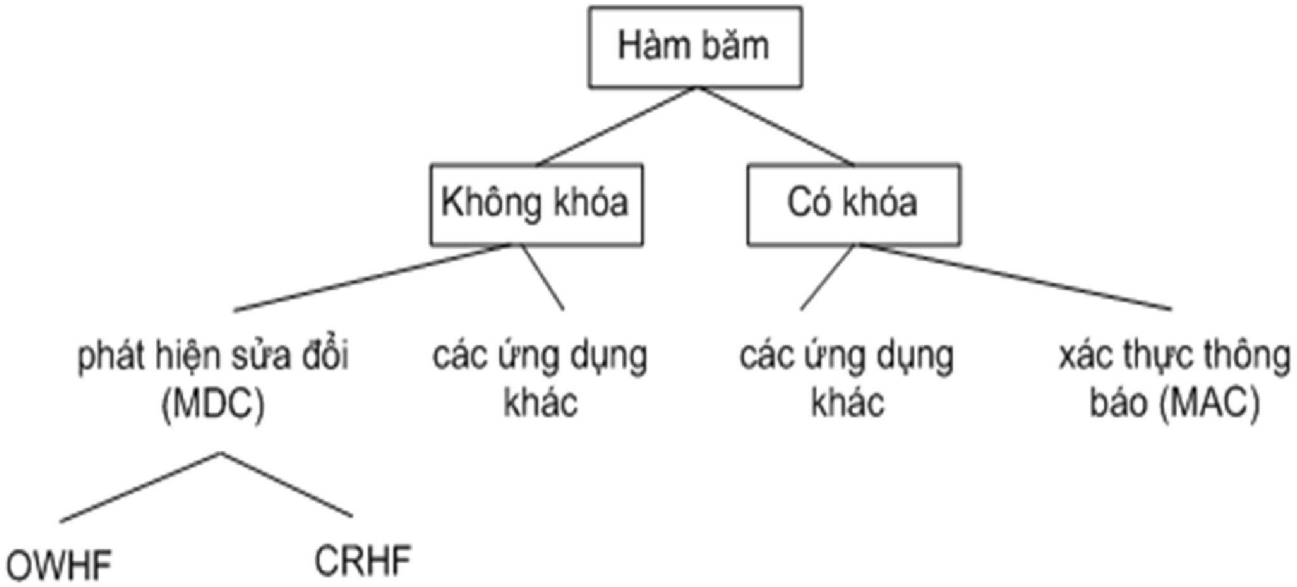
Giới thiệu

- Các hàm băm mật mã đóng vai trò quan trọng trong mật mã hiện đại:
  - Được dùng để xác thực tính nguyên vẹn dữ liệu
  - Được dùng trong quá trình tạo chữ kí số trong giao dịch điện tử.
- Các hàm băm lấy một thông báo đầu vào và tạo một đầu ra được xem như là:
  - Mã băm (hash code),
  - Kết quả băm (hash result),
  - Hoặc giá trị băm (hash value).

Giới thiệu

- Vai trò cơ bản của các hàm băm mật mã là một giá trị băm coi như ảnh đại diện thu gọn, đôi khi gọi là một dấu vết (imprint), vân tay số (digital fingerprint), hoặc tóm lược thông báo (message digest) của một xâu đầu vào, và có thể được dùng như là một định danh duy nhất với xâu đó.
- Các hàm băm thường được dùng cho toàn vẹn dữ liệu kết hợp với các lược đồ chữ kí số.
- Một lớp các hàm băm riêng được gọi là mã xác thực thông báo (MAC) cho phép xác thực thông báo bằng các kĩ thuật mã đối xứng.

Giới thiệu



Phân loại các hàm băm mật mã và ứng dụng

4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- **Giới thiệu hàm băm**
  - Việc sử dụng các hệ mật mã và các sơ đồ chữ ký số, thường là mã hóa và ký số trên từng bit của thông tin, sẽ tỷ lệ với thời gian để mã hóa và dung lượng của thông tin.
  - Thêm vào đó có thể xảy ra trường hợp: Với nhiều bức thông điệp đầu vào khác nhau, sử dụng hệ mật mã, sơ đồ ký số giống nhau (có thể khác nhau) thì cho ra kết quả bản mã, bản ký số giống nhau (ánh xạ N-1: nhiều – một). Điều này sẽ dẫn đến một số rắc rối về sau cho việc xác thực thông tin.

4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- Với các sơ đồ ký số, chỉ cho phép ký các bức thông điệp (thông tin) có kích thước nhỏ và sau khi ký, bản ký số có kích thước gấp đôi bản thông điệp gốc
  - Ví dụ: với sơ đồ chữ ký chuẩn DSS chỉ ký trên các bức thông điệp có kích thước 160 bit, bản ký số sẽ có kích thước 320 bit.
- Trong khi đó trên thực tế, ta cần phải ký các thông điệp có kích thước lớn hơn nhiều, chẳng hạn vài chục MB. Hơn nữa, dữ liệu truyền qua mạng không chỉ là bản thông điệp gốc, mà còn bao gồm cả bản ký số (có dung lượng gấp đôi dung lượng bản thông điệp gốc), để đáp ứng việc xác thực sau khi thông tin đến người nhận.

4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- Một cách đơn giản để giải bài toán (với thông điệp có kích thước vài chục MB) này là chia thông điệp thành nhiều đoạn 160 bit, sau đó ký lên các đoạn đó độc lập nhau. Nhưng biện pháp này có một số vấn đề trong việc tạo ra các chữ ký số:
  - Thứ nhất: với một thông điệp có kích thước ***a***, thì sau khi ký kích thước của chữ ký sẽ là 2***a*** (trong trường hợp sử dụng DSS).
  - Thứ hai: với các chữ ký "an toàn" thì tốc độ chậm vì chúng dùng nhiều phép tính số học phức tạp như số mũ modulo.
  - Thứ ba: vấn đề nghiêm trọng hơn đó là kết quả sau khi ký, nội dung của thông điệp có thể bị xáo trộn các đoạn với nhau, hoặc một số đoạn trong chúng có thể bị mất mát, trong khi người nhận cần phải xác minh lại thông điệp. Ta cần phải bảo vệ tính toàn vẹn của thông điệp

4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- Giải pháp cho các vấn đề vướng mắc đến chữ ký số là dùng "**hàm băm**" để trợ giúp cho việc ký số
- Các thuật toán băm với đầu vào là các bức thông điệp có dung lượng, kích thước tùy ý (vài KB đến vài chục MB thậm chí hơn nữa) – các bức thông điệp có thể là dạng văn bản, hình ảnh, âm thanh, file ứng dụng v.v… - và với các thuật toán băm: MD2, MD4, MD5, SHA cho các bản băm đầu ra có kích thước cố định: 128 bit với dòng MD, 160 bit với SHA.
- Như vậy, bức thông điệp kích thước tùy ý sau khi băm sẽ được thu gọn thành những bản băm – được gọi là các "**văn bản đại diện**" – có kích thước cố định (128 bit hoặc 160 bit).

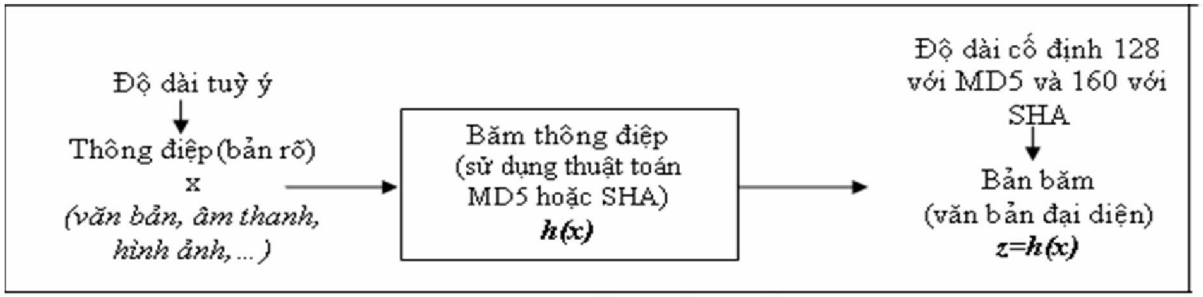
4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- Với mỗi thông điệp đầu vào chỉ có thể tính ra được một văn bản đại diện – giá trị băm tương ứng – duy nhất
- Hai thông điệp khác nhau chắc chắn có hai văn bản đại diện khác nhau. Khi đã có văn bản đại diện duy nhất cho bức thông điệp, áp dụng các sơ đồ chữ ký số ký trên văn bản đại diện đó

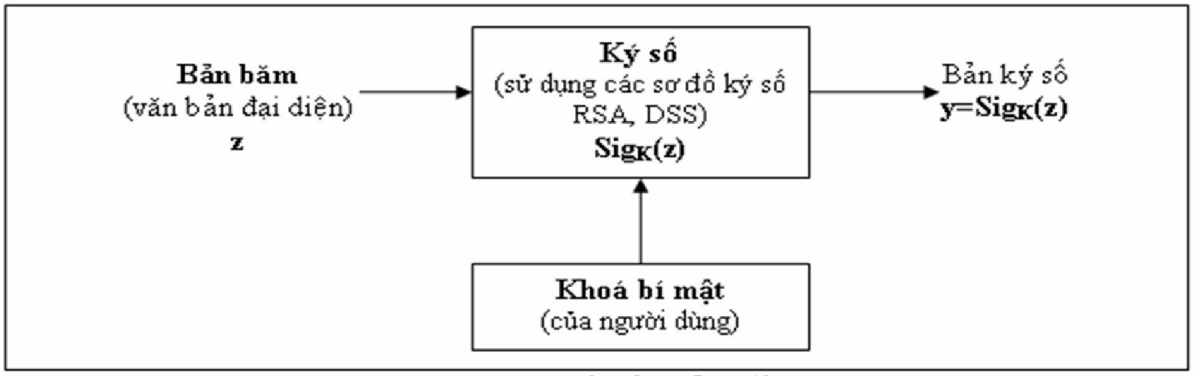
4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- Giả sử A muốn gửi cho B thông điệp x. A thực hiện các bước sau:
  - (1) A băm thông điệp $\boldsymbol{x}$, thu được bản đại diện $z = h(x)$ – có kích thước cố định 128 bit hoặc 160 bit.
  - (2) A ký số trên bản đại diện z, bằng khóa bí mật của mình, thu được bản ký số y = sig(z).
  - (3) A gửi (x, y) cho B.

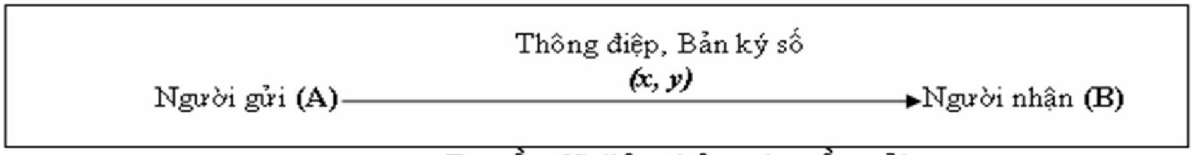
4.1 Các hàm băm và tính toàn vẹn của dữ liệu



(1) *Băm thông điệp.*



(2) *Ký trên bản băm.*

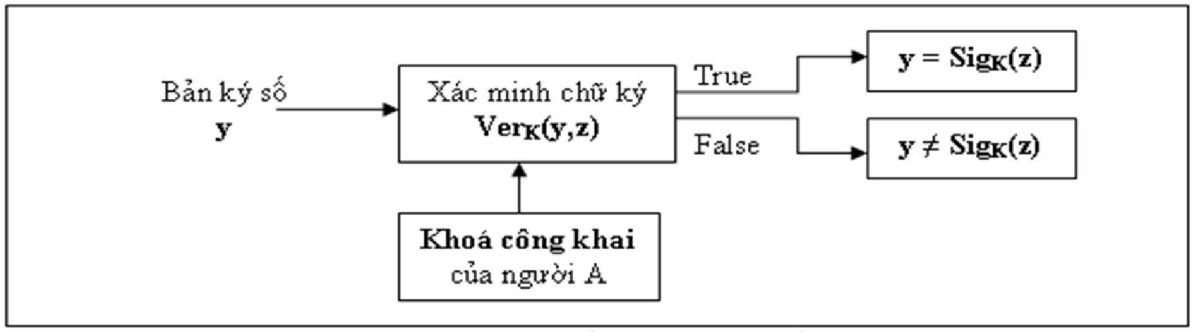


(3) *Truyền dữ liệu thông tin cần gửi.*

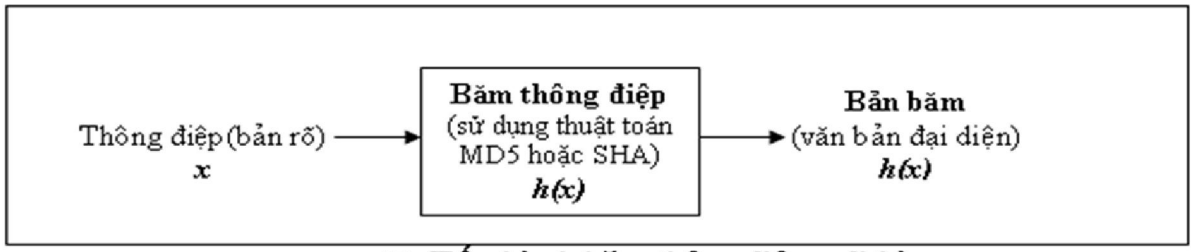
4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- Khi B nhận được (x, y). B thực hiện các bước sau:
  - (4) B kiểm tra chữ ký số để xác minh xem thông điệp mà mình nhận được có phải được gửi từ A hay không bằng cách giải mã chữ ký số y, bằng khóa công khai của A, được z.
  - (5) B dùng một thuật toán băm – tương ứng với thuật toán băm mà A dùng – để băm thông điệp x đi kèm, nhận được h(x).
  - (6) B so sánh 2 giá trị băm z và h(x), nếu giống nhau thì chắc chắn rằng thông điệp x – mà A muốn gửi cho B – còn nguyên vẹn, bên cạnh đó cũng xác thực được người gửi thông tin là ai.

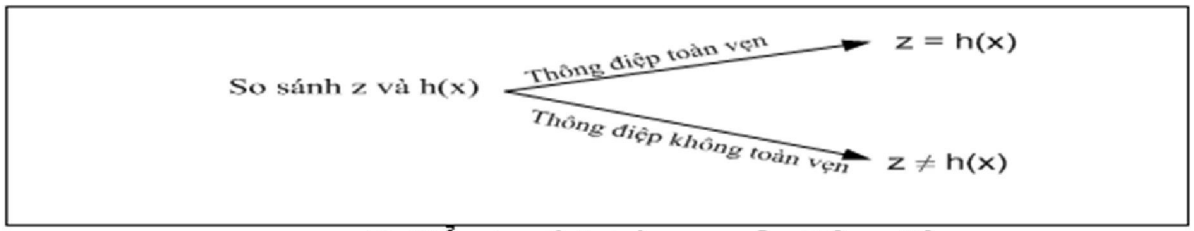
4.1 Các hàm băm và tính toàn vẹn của dữ liệu



(4) *Xác minh chữ ký*



(5) *Tiến hành băm thông điệp x đi kèm*



(6) *Kiểm tra tính toàn vẹn của thông điệp*

4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- Hàm băm đã trợ giúp cho các sơ đồ ký số nhằm giảm dung lượng của dữ liệu cần thiết để truyền qua mạng
  - Ví dụ: lúc này chỉ còn bao gồm dung lượng của bức thông điệp gốc và 256 bit (sử dụng MD) hay 320 bit (sử dụng SHA) của bức ký số được ký trên bản đại diện của thông điệp gốc, tương đương với việc giảm thời gian truyền tin qua mạng.
- Hàm băm thường kết hợp với chữ ký số để tạo ra một loại chữ ký điện tử vừa an toàn hơn (không thể cắt/dán) vừa có thể dùng để kiểm tra tính toàn vẹn của thông điệp.
- Hàm băm được ứng dụng rất mạnh trong vấn đề an toàn thông tin trên đường truyền. Các ứng dụng có sử dụng hàm băm không chỉ đảm bảo về mặt an toàn thông tin, mà còn tạo được lòng tin của người dùng vì họ có thể dễ dàng phát hiện được thông tin của mình có còn toàn vẹn hay không, họ biết rằng thông tin của mình chắc chắn được bí mật với phía các nhà cung cấp.

4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- **Định nghĩa hàm băm:**
  - Hàm băm là các thuật toán không sử dụng khóa để mã hóa (ở đây ta dùng thuật ngữ "băm" thay cho "mã hóa"), nó có nhiệm vụ "lọc" (băm) thông điệp được đưa vào theo một thuật toán ***h*** một chiều nào đó, rồi đưa ra một bản băm – văn bản đại diện – có kích thước cố định. Do đó người nhận không biết được nội dung hay độ dài ban đầu của thông điệp đã được băm bằng hàm băm.
  - Giá trị của hàm băm là duy nhất, và **không thể suy ngược** lại được nội dung thông điệp từ giá trị băm này.

4.1 Các hàm băm và tính toàn vẹn của dữ liệu

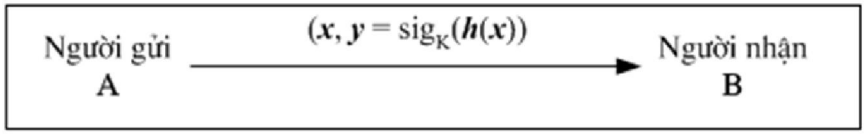
- **Đặc trưng:**
  - Hàm băm h là hàm băm một chiều (one-way hash) với các đặc tính sau:
    - Với thông điệp đầu vào x thu được bản băm $z = h(x)$ là duy nhất.
    - Nếu dữ liệu trong thông điệp x thay đổi hay bị xóa để thành thông điệp x' thì $h(x') \neq h(x)$. Cho dù chỉ là một sự thay đổi nhỏ hay chỉ là xóa đi 1 bit dữ liệu của thông điệp thì giá trị băm cũng vẫn thay đổi. Điều này có nghĩa là: hai thông điệp hoàn toàn khác nhau thì giá trị hàm băm cũng khác nhau.
    - Nội dung của thông điệp gốc không thể bị suy ra từ giá trị hàm băm. Nghĩa là: với thông điệp x thì dễ dàng tính được $z = h(x)$, nhưng lại không thể (thực chất là khó) suy ngược lại được x nếu chỉ biết giá trị hàm băm h

4.1 Các hàm băm và tính toàn vẹn của dữ liệu

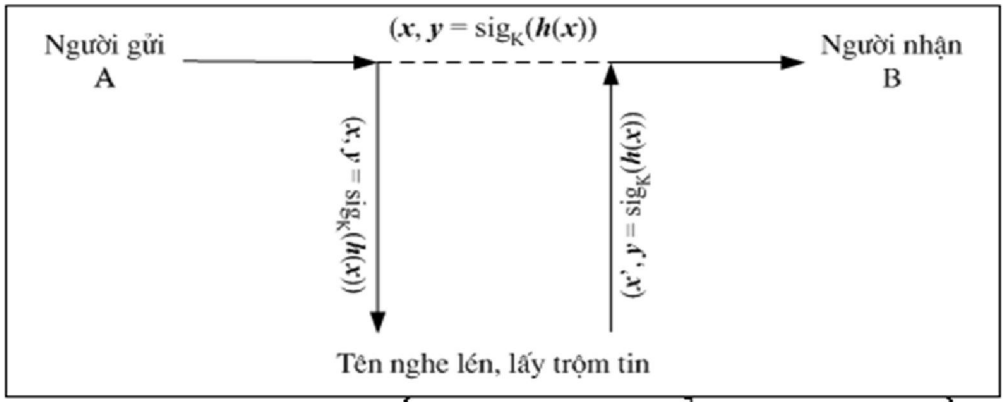
- **Tính chất:**
  - Việc đưa hàm băm *h* vào dùng trong sơ đồ chữ ký số không làm giảm sự an toàn của sơ đồ chữ ký số vì nó là bản tóm lược thông báo – bản đại diện cho thông điệp – được ký chứ không phải là thông điệp gốc. Điều cần thiết đối với *h* là cần thỏa mãn một số tính chất sau để tránh bị giả mạo:
    - Tính chất 1: Hàm băm ***h*** là không va chạm yếu.

4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- Ví dụ xét một kiểu tấn công sau:
  - Đáng lẽ: thông tin phải được truyền đúng từ A đến B



  - Nhưng: trên đường truyền, thông tin bị lấy trộm và bị thay đổi



4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- Người A gửi cho B (x, y) với y = sigK($h(x)$). Nhưng trên đường truyền, tin bị lấy trộm. Tên trộm, bằng cách nào đó tìm được một bản thông điệp x' có $h(x') = h(x)$ mà $x' \neq x$. Sau đó, hắn đưa x' thay thế x rồi truyền tiếp cho người B. Người B nhận được và vẫn xác thực được thông tin đúng đắn.
- Do đó, để tránh kiểu tấn công như trên, hàm h phải thỏa mãn tính không va chạm yếu: *Hàm băm h là không va chạm yếu nếu khi cho trước một bức điện x, không thể tiến hành về mặt tính toán để tìm ra một bức điện $x' \neq x$ mà $h(x') = h(x)$.*

4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- **Tính chất 2**: Hàm băm ***h*** là không va chạm mạnh
  - Xét một kiểu tấn công như sau: Đầu tiên, tên giả mạo tìm ra được hai bức thông điệp x' và x ($x' \neq x$) mà có $h(x') = h(x)$ (ta coi bức thông điệp x là hợp lệ, còn x' là giả mạo). Tiếp theo, hắn đưa cho ông A và thuyết phục ông này kí vào bản tóm lược $h(x)$ để nhận được y. Khi đó (x', y) là bức điện giả mạo nhưng hợp lệ.
  - Để tránh kiểu tấn công này, hàm h phải thỏa mãn tính không va chạm mạnh: *Hàm băm h là không va chạm mạnh nếu không có khả năng tính toán để tìm ra hai bức thông điệp x và x' mà* $x \neq x'$ *và* $h(x) = h(x')$.

4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- **Tính chất 3**: Hàm băm h là hàm một chiều:
  - Xét một kiểu tấn công như sau: Việc giả mạo các chữ ký trên bản tóm lược ***z*** thường xảy ta với các sơ đồ chữ ký số. Giả sử tên giả mạo tính chữ ký trên bản tóm lược ***z***, sau đó hắn tìm một bản thông điệp ***x'*** được tính ngược từ bản đại diện ***z***, ***z*** = ***h***(***x'***). Tên trộm thay thế bản thông điệp ***x*** hợp lệ bằng bản thông điệp ***x'*** giả mạo, nhưng lại có ***z*** = ***h***(***x'***). Và hắn ký số trên bản đại diện cho ***x'*** bằng đúng chữ ký hợp lệ. Nếu làm được như vậy thì (***x'***, ***y***) là bức điện giả mạo nhưng hợp lệ.
  - Để tránh được kiểu tấn công này, h cần thỏa mãn tính chất một chiều: *Hàm băm h là một chiều nếu khi cho trước một bản tóm lược thông báo z thì không thể thực hiện về mặt tính toán để tìm ra thông điệp ban đầu x sao cho h(x) = z .*

4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- **Hàm băm gồm 2 loại:**
  - **Hàm băm không có khóa:** các hàm băm dựa trên mật mã khối.
  - **Hàm băm có khóa (MAC)** dùng để xác thực thông báo

4.1 Các hàm băm và tính toàn vẹn của dữ liệu

Các hàm băm không có khóa

- **Định nghĩa 1:** Mật mã khối (n, r) là một mã khối xác định một hàm khả nghịch từ các bản rõ n bit sang các bản mã n bit bằng cách sử dụng một khoá r bit. Nếu E là một phép mã hoá như vậy thì $E_k(x)$ ký hiệu cho phép mã hoá x bằng khoá k.
- **Định nghĩa 2:** Cho h là một hàm băm có lặp được xây dựng từ một mật mã khối với hàm nén *f* thực hiện s phép mã hoá khối để xử lý từng khối bản tin n bit. Khi đó tốc độ của h là 1/s.

4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- **Tính chất:**
  - Tính chất nén
  - Tính dễ dàng tính toán
  - Tính khó tính toán nghịch ảnh
  - Khó tìm nghịch ảnh thứ hai: vớ x cho trước thì không có khả năng tìm x’ ≠ x sao cho: h(x) = h(x’)
  - Tính kháng va chạm: không có khả năng về tính toán để tìm hai đầu vào khác nhau bất kì x’ và x để h(x) = h(x’)
  - Hàm băm thỏa mãn tính chất trên được gọi là hàm băm mật mã hay hàm băm an toàn.

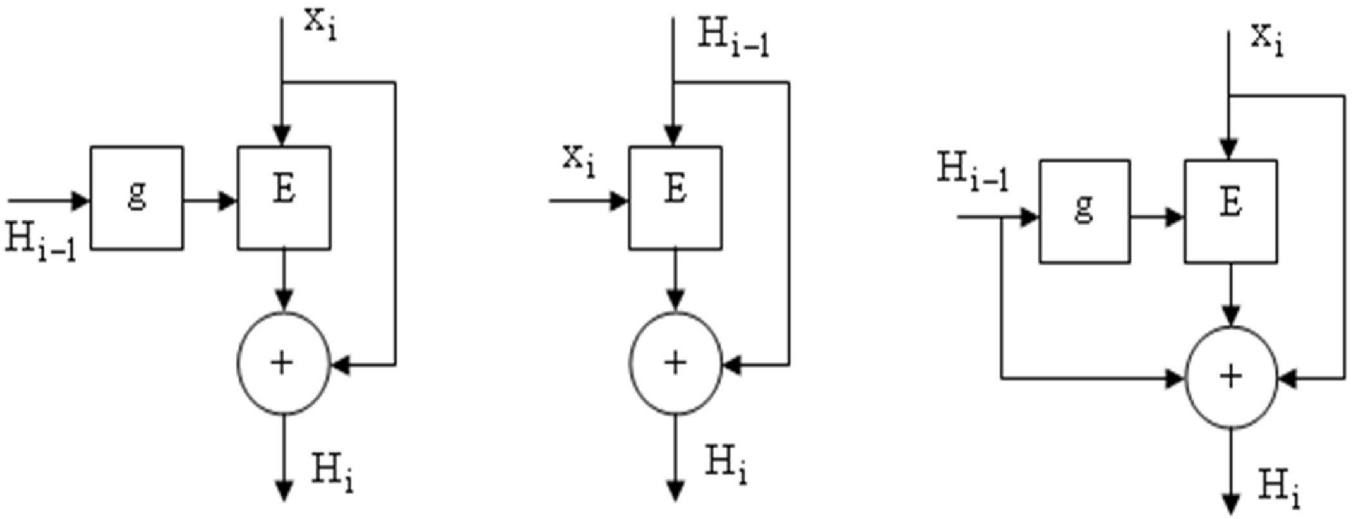
4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- **MDC (Manipulating detection codes): mã phát hiện sửa đổi**
  - Các hàm băm dòng MD: MD2, MD4, MD5 được Rivest đưa ra thu được kết quả đầu ra với độ dài là 128 bit. Hàm băm MD4 đưa ra vào năm 1990. Một năm sau phiên bản mạnh MD5 cũng được đưa ra thay thế cho MD4.
  - Mục đích của MDC là cung cấp một biểu diễn ảnh hoặc băm của thông báo, nó là lớp con của các hàm băm không có khóa. Các lớp đặc biệt của MDC là:
    - Các hàm băm một chiều (OWHF): là các hàm băm mà việc tìm một đầu vào để băm thành một giá trị băm được xác định trước là rất khó
    - Các hàm băm kháng va chạm (CRHF): là các hàm băm mà việc tìm hai đầu vào có cùng giá trị băm là khó

4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- **MDC độ dài đơn.**
  - Ba sơ đồ dưới đây có liên quan chặt chẽ với các hàm băm độ dài đơn, xây dựng trên các mật mã khối. Các sơ đồ này có sử dụng các thành phần được xác định trước như sau:
    - Một mật mã khối n bit khởi sinh E_k được tham số hoá bằng một khoá đối xứng k.
    - Một hàm g ánh xạ n bit vào thành khoá k sử dụng cho E (Nếu các khoá cho E cũng có độ dài n thì g có thể là hàm đồng nhất)
    - Một giá trị ban đầu cố định IV thích hợp để dùng với E.

4.1 Các hàm băm và tính toàn vẹn của dữ liệu



Matyas - Mayer - Oseas Davies - Mayer Miyaguchi - Preneel

4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- **Thuật toán băm Matyas - Meyer – Oseas**
  - Vào: Xâu bit n
  - Ra: Mã băm n bit của x
    - (1) Đầu vào x được phân chia thành các khối n bit và được độn nếu cần thiết nhằm tạo khối cuối cùng hoàn chỉnh. Ta được t khối n bit: $x_1\ x_2\ \ldots\ x_t$. Xác định trước một giá trị ban đầu n bit (kí hiệu IV)
    - (2) Đầu ra là H_t được xác định như sau:

$$H_0 = IV,\ H_i = E_g(x_i) \oplus x_i,\ 1 \le i \le t$$

4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- **Thuật toán băm Davies - Meyer**
  - Vào: Xâu bit n
  - Ra: Mã băm n bit của x
    - (1) Đầu vào x được phân chia thành các khối n bit và được độn nếu cần thiết nhằm tạo khối cuối cùng hoàn chỉnh. Ta được t khối n bit: $x_1\ x_2\ \ldots\ x_t$. Xác định trước một giá trị ban đầu n bit (kí hiệu IV)
    - (2) Đầu ra là H_t được xác định như sau:

$$H_0 = IV,\ H_i = E_{xi}(H_{i-1}) \oplus H_{i-1},\ 1 \le i \le t$$

4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- **Thuật toán băm Miyaguchi - Preneel**
  - Sơ đồ này tương tự như ở thuật toán M-M-O ngoại trừ H_{i-1} (đầu ra ở giai đoạn trước) được cộng mod 2 với tín hiệu ra ở giai đoạn hiện thời. Như vậy:

$$H_0 = IV,\ H_i = E_{g(Hi-1)}(x_i) \oplus x_i \oplus H_{i-1};\ 1 \le i \le t$$

4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- **MDC độ dài kép:**
  - MDC – 2 và MDC – 4 là các mã phát hiện sự sửa đổi yêu cầu tương ứng là 2 và 4 phép toán mã hoá khối trên mỗi khối đầu vào hàm băm. MDC-2 và MDC- 4 sử dụng các thành phần xác định như sau:
    - DES được dùng làm mật mã khối E_k có đầu vào/ ra 64 bit và được tham số hoá bằng khoá k 56 bit.
    - Hai hàm g và $\tilde{g}$ ánh xạ các giá trị 64 bit U thành các khoá DES 56 bit như sau:
      - Cho $U = u_1\ u_2 \ldots u_{64}$, xoá mọi bit thứ 8 bắt đầu từ u_8 và đặt các bit thứ 2 và thứ 3 về "10" đối với g và "01" đối với $\tilde{g}$
      - Điều này đảm bảo rằng chúng không phải là các khoá DES yếu hoặc nửa yếu. Đồng thời điều này cũng đảm bảo yêu cầu bảo mật là $g(IV) \neq \tilde{g}(IV)$

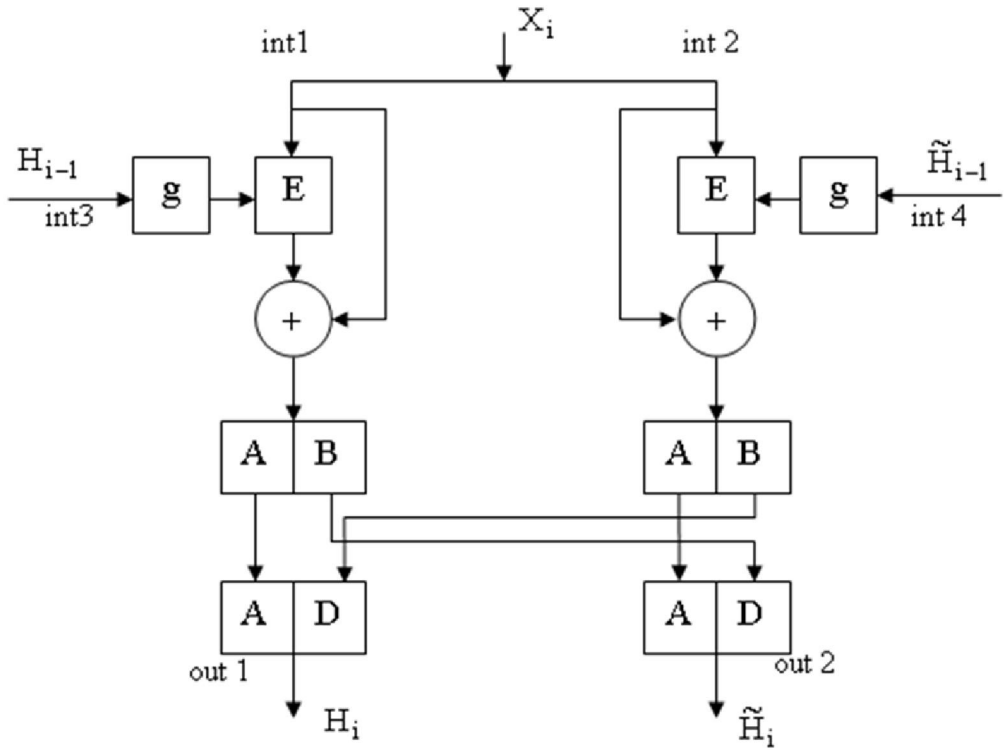
4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- **Thuật toán MD2**
  - VÀO: Xâu bit x có độ dài r = 64t, t ≥ 2
  - RA: Bản băm, đại diện cho thông điệp gốc, độ dài cố định 128 bit
  - **Mô tả thuật toán**
    - (1) Phân x thành các khối 64 bit x_i: $x_1, \ldots, x_t$
    - (2) Chọn IV và $\widetilde{IV}$ như sau:
      - $IV = 0x5252525252525252;$
      - $\widetilde{IV} = 0x2525252525252525$
    - (3) Ký hiệu $||$ là phép ghép và C_i^L, C_i^R là các nửa 32 bit phải và trái của C_i đầu ra $h(x) = H_t \| \widetilde{H}_t$ được xác định như sau (với $1 \le i \le t$):

$$H_0 = IV\ ,\ \ k_i = g(H_{i-1})\ ,\ C_i = E_{k_i}(x_i) \oplus x_i\ \ ,\ \ H_i = C_i^L \| \widetilde{C}_i^R$$

$$\widetilde{H}_0 = \widetilde{IV}\ ,\ \ \widetilde{k}_i = \widetilde{g}(\widetilde{H}_{i-1})\ ,\ \widetilde{C}_i = E_{\widetilde{k}_i}(x_i) \oplus x_i\ \ ,\ \ \widetilde{H}_i = \widetilde{C}_i^L \| C_i^R$$

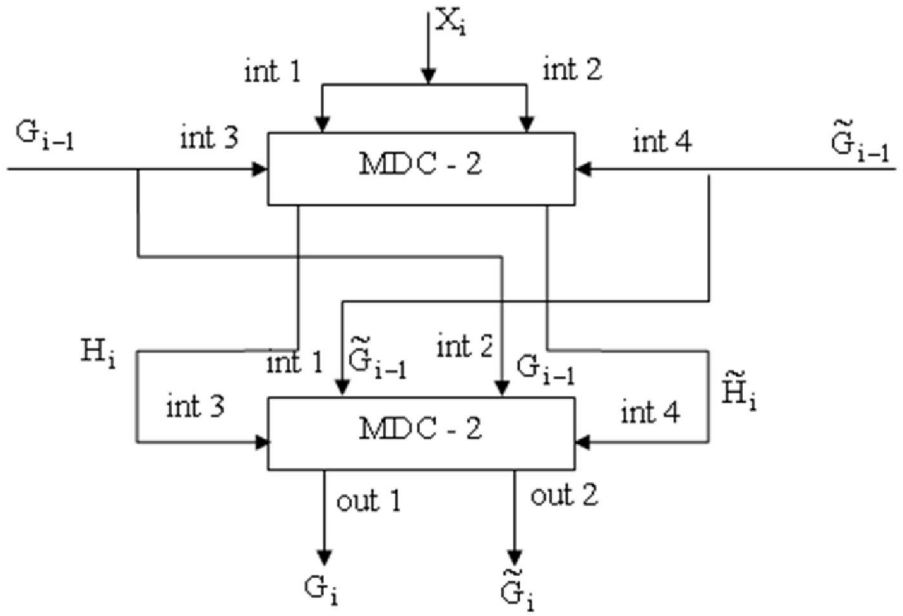
4.1 Các hàm băm và tính toàn vẹn của dữ liệu



Sơ đồ thuật toán MD2

4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- **Thuật toán MD4:**



Sơ đồ thuật toán MD4

4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- **Thuật toán MD5**
  - **Mô tả thuật toán**
    - Đầu vào: là một thông điệp có độ dài tuỳ ý
    - Đầu ra là một chuỗi có độ dài cố định là 128 bit.
    - Thuật toán được thiết kế để chạy trên các máy tính 32 bit.
  - **Thuật toán?**

4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- **Thuật toán MD5**
  - **Mô tả thuật toán**
    - Đầu vào: là một thông điệp có độ dài tuỳ ý
    - Đầu ra là một chuỗi có độ dài cố định là 128 bit.
    - Thuật toán được thiết kế để chạy trên các máy tính 32 bit.
  - **Thuật toán:**
    - Thông điệp đầu vào có độ dài b bit bất kỳ. Biểu diễn các bit dưới dạng như sau: m[0] m[1] m[2] ... m[b-1]
    - ***Bước1:*** Các bit gắn thêm : Thông điệp được mở rộng, thêm bit vào phía sau sao cho độ dài của nó (bit) đồng dư với 448 theo môđun 512. Nghĩa là thông điệp được mở rộng sao cho nó còn thiếu 64 bit nữa thì sẽ có một độ dài chia hết cho 512. Việc thêm bit này được thực hiện như sau:
      - Một bit '1' được thêm vào sau thông điệp
      - Sau đó các bit '0' được thêm vào để có một độ dài đồng dư với 448 môđun 512.

4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- ***Bước 2:*** Gắn thêm độ dài: Dạng biểu diễn 64 bit độ dài b của chuỗi ban đầu được thêm vào phía sau kết quả của bước 1.
- ***Bước 3:*** Khởi tạo bộ đệm MD: Một bộ đệm 4 từ (A,B,C,D) được dùng để tính mã số thông điệp. Ở đây mỗi A,B,C,D là một thanh ghi 32 bit. Những thanh ghi này được khởi tạo theo những giá trị hex sau :

  A=0x01234567
  B=0x89abcdef
  C=0xfedcba98
  D=0x76543210

- ***Bước 4 :*** Xử lý thông điệp theo từng khối 16 từ. Định nghĩa các hàm phụ, các hàm này nhận giá trị đầu vào là 3 từ 32 bit và tạo tạo ra một word 32 bit.

  F(X,Y,Z) = XY v not(X) Z
  G(X,Y,Z)= XZ v Y not(Z)
  H(X,Y,Z) = X xor Y xor Z
  I(X,Y,Z) = Y xor (X v not(Z)).

  Bước này sử dụng một bảng 64 giá trị T[1 .. 64] được tạo ra từ hàm sin. Gọi T là phần tử thứ i của bảng, thì T là phần nguyên của 4294967296*|sin(i)| , i được tính theo radian

4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- **Đánh giá thuật toán MD5**
  - Về tốc độ sinh ra chuỗi cốt yếu thì MD5 chậm hơn so với MD4 nhưng nó lại an toàn hơn rất nhiều so với MD4.
  - Thuật toán số hóa thông điệp MD5 khá đơn giản để thực hiện, cung cấp một giá trị băm của thông điệp với độ dài tuỳ ý.

4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- **Các mục tiêu của đối phương với các thuật toán MDC:**
  - Mục tiêu của đối phương muốn tấn công một MDC là như sau:
    - (a) Để tấn công một OWHF: cho trước giá trị băm y, tìm một tiền ảnh x sao cho y =h(x) hoặc một cặp (x, h(x)), tìm một tiền ảnh thứ hai x’ sao cho h(x) = h(x’)
    - (b) Để tấn công một CRHF: tìm hai đầu vào bất kì x, x’ sao cho h(x) = h(x’). Một CRHF phải được thiết kế để chống lại các “tấn công ngày sinh nhật”

4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- **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 2 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:

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

4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- 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.

4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- **Tấn công ngày sinh nhật hoạt động như thế nào?**

4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- **Các hàm băm có khóa (MAC):**
  - MAC là một lớp con của hàm băm có khóa. Mục đích của MAC là bảo đảm cả tài nguyên của thông báo và tính toàn vẹn của nó. Gồm các tính chất sau:
    - Dễ dàng tính toán: với h_k đã biết, giá trị k cho trước và một đầu vào x, $h_k(x)$ có thể được tính dễ dàng ($h_k(x)$ được gọi là giá trị MAC)
    - Nén: ánh xạ một đầu vào x có độ dài bit hữu hạn tuỳ ý tới một đầu ra $h_k(x)$ có độ dài bit n cố định.
    - Kháng tính toán: Với các cặp giá trị $(x_i, h(x_i))$ không có khả năng tính một cặp (x, h(x)) với $x \neq x_i$ (kể cả có khả năng $h_k(x) = h_k(x_i)$ với một i nào đó).

4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- Các hàm băm có khoá được sử dụng để xác thực thông báo và thường được gọi là *các thuật toán tạo mã xác thực thông báo (MAC).*
- MAC dựa trên các mật mã khối.
  - Thuật toán
    - VÀO: Dữ liệu x, mật mã khối E, khoá MAC bí mật k của E.
    - RA : n bit MAC trên x (n là độ dài khối của E)
      - (1) Độn và chia khối: Độn thêm các bit vào x nếu cần. Chia dữ liệu đã độn thành từng khối n bit : $x_1, \ldots, x_t$

4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- (2) Xử lý theo chế độ CBC. Ký hiệu E_k là phép mã hoá E với khoá k.Tính khối H_t như sau:

$$H_1 \leftarrow E_k(x_1)$$

$$H_i \leftarrow E_k(H_{i-1} \oplus x_i) \quad 2 \le i \le t$$

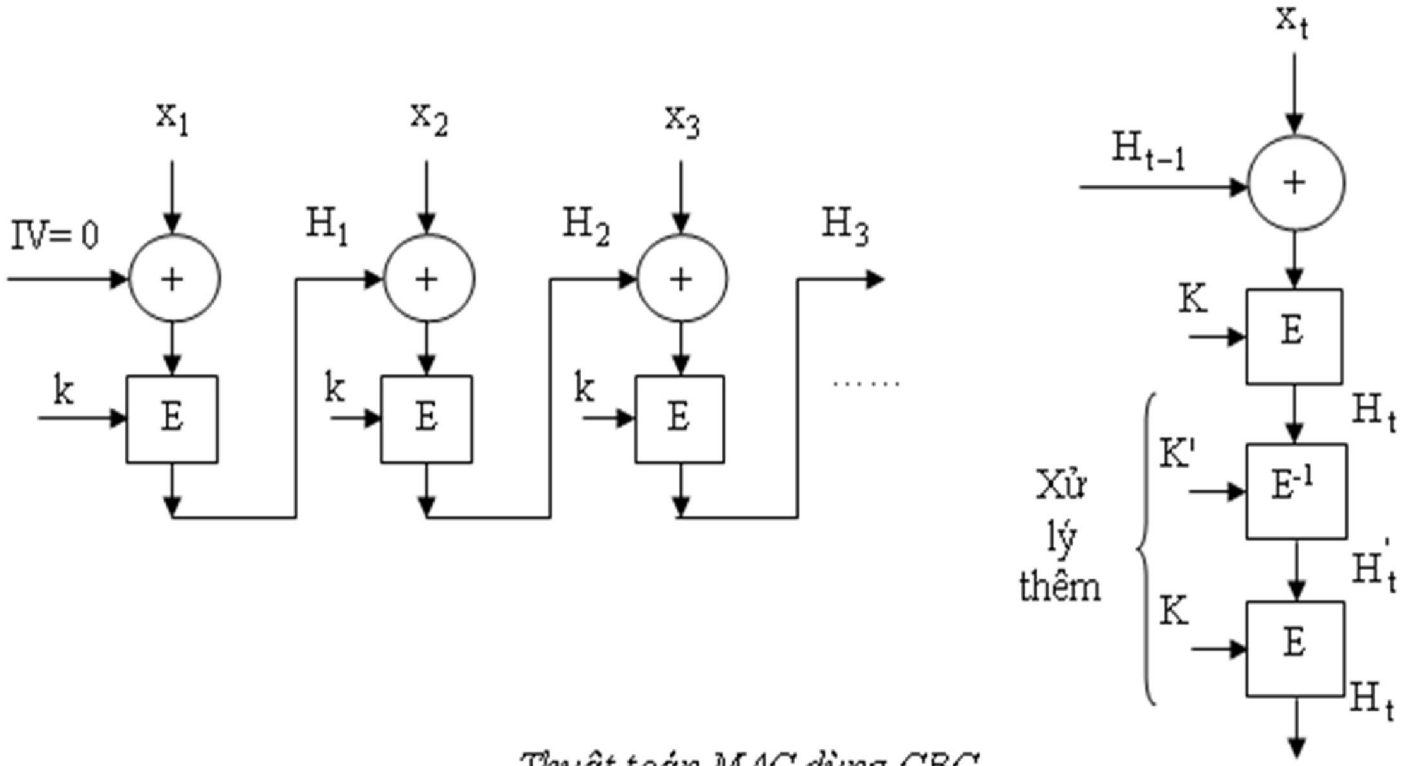
- (3) Xử lý thêm để tăng sức mạnh của MAC. Dùng một khoá bí mật thứ hai $k \neq k'$. Tính:

$$H_t' \leftarrow E_k^{-1}(H_t)$$

$$H_t \leftarrow E_k(H_t')$$

- (4) Xử lý thêm để tăng sức mạnh của MAC
- (5) Kết thúc: MAC là khối n bit H_t

4.1 Các hàm băm và tính toàn vẹn của dữ liệu

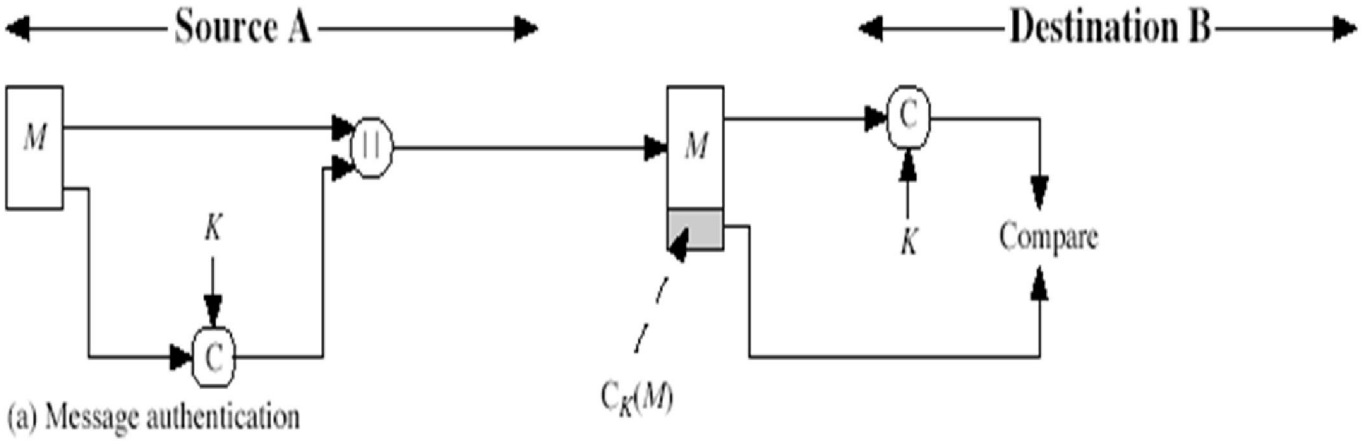


Thuật toán MAC dùng CBC

4.1 Các hàm băm và tính toàn vẹn của dữ liệu

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

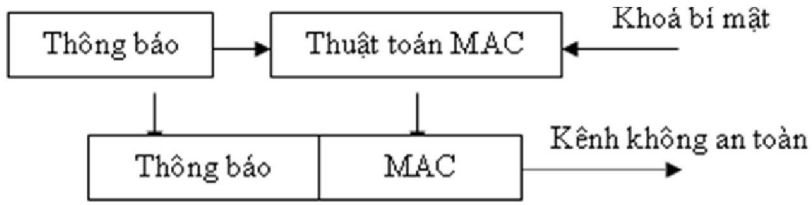
4.1 Các hàm băm và tính toàn vẹn của dữ liệu



(a) Message authentication

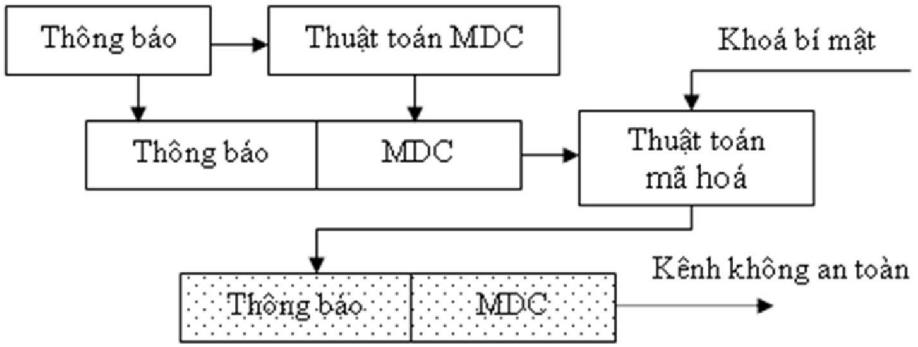
4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- **Tính toàn vẹn của dữ liệu và xác thực thông báo**
  - Có ba phương pháp cung cấp tính toàn vẹn của dữ liệu bằng cách dùng các hàm băm.
    - **Chỉ dùng MAC**

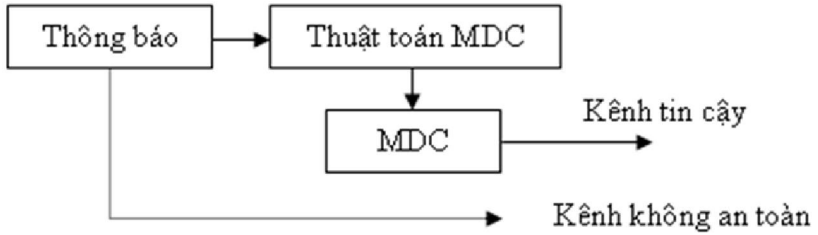


4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- **Dùng MDC và mã hóa:**



- **Sử dụng MDC và kênh tin cậy:**



4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- Các phương pháp đảm bảo xác thực tính nguyên vẹn của dữ liệu.
  - Dùng MAC.
  - Dùng các sơ đồ chữ ký số.
  - Gắn (trước khi mã hoá) một giá trị thẻ xác thực bí mật vào văn bản được mã.

4.1 Các hàm băm và tính toàn vẹn của dữ liệu

- Các mục tiêu của đối phương đối với các thuật toán MAC:
  - Tấn công với văn bản đã biết: một hoặc nhiều cặp (x_i, $h_k(x_i)$) là có giá trị.
  - Tấn công văn bản chọn lọc: một hoặc nhiều cặp (x_i, $h_k(x_i)$) là có giá trị với x_i được chọn bởi đối phương.
  - Tấn công với văn bản chọn lọc thích ứng: x_i có thể được chọn bởi đối phương như ở trên, bây giờ cho phép lựa chọn thành công dựa trên các kết quả truy vấn có ưu tiên.

4.2 Trao đổi và thoả thuận khoá

- Giả sử A và B muốn liên lạc sử dụng hệ mật khoá bí mật. Để thoả thuận mật khoá K chung cho cả hai bên qua một kênh không an toàn mà không ai khác có thể biết được, A và B có thể dùng thủ tục thoả thuận khoá Diffie -Hellman sau:
  - **(1)** Chọn trước một số nguyên tố p thích hợp và một phần tử sinh α của Z_p^* ($2 \leq \alpha \leq p - 2$) . Các giá trị p và α được công khai.
  - **(2)** A gửi cho B giá trị (2.1)

    B gửi cho A giá trị (2.2)

4.2 Trao đổi và thoả thuận khoá

- **(3)** Thực hiện các bước sau mỗi khi cần có khóa chung:
  - (a) A chọn một số nguyên bí mật x: $1 \leq x \leq p - 2$ và gửi cho B thông báo $\alpha^x \bmod p$ (2.1)
  - (b) B chọn một số nguyên bí mật y: $1 \leq y \leq p - 2$ và gửi cho A thông báo $\alpha^y \bmod p$ (2.2).
  - (c) B thu được α^x và tính khoá chung k: $k = (\alpha^x)^y \bmod p$
  - (d) A thu được α^y và tính khoá chung k: $k = (\alpha^y)^x \bmod p$

4.2 Trao đổi và thoả thuận khoá

- **Ví dụ:**
  - Giả sử A và B chọn p = 11 và $\alpha = 2$. Nhóm nhân xyclic sinh bởi α:

    $\{\alpha^i, i = 0, \ldots, 9\} = \{1, 2, 4, 8, 5, 10, 9, 7, 3, 6\}$. Các phần tử sinh của nhóm này bao gồm các phần tử sau: $\alpha = 2$, $\alpha^3 = 8$, $\alpha^7 = 7$, $\alpha^9 = 6$.
  - Giả sử A chọn giá trị ngẫu nhiên x = 4 và gửi cho B giá trị $2^4 \bmod 11 = 5$.
  - Giả sử B chọn giá trị ngẫu nhiên y = 7 và gửi cho A giá trị $2^7 \bmod 11 = 7$.
  - B nhận được 5 và tính khoá chung $k = 5^7 \bmod 11 = 3$
  - A nhận được 7 và tính khoá chung $k = 7^4 \bmod 11 = 3$

4.3 Hệ mật dựa trên định danh

- **Ý tưởng cơ bản**:
  - Hệ mật dựa trên định danh do Shamir đề xuất là một hệ mật bất đối xứng trong đó:
    - Thông tin định danh của thực thể (tên riêng) đóng vai trò khoá công khai của nó.
    - Trung tâm xác thực T được sử dụng để tính khoá riêng tương ứng của thực thể này

4.3 Hệ mật dựa trên định danh

- **Sơ đồ trao đổi khoá Okamoto-Tanaka: gồm 3 pha**
  - **(1) Pha chuẩn bị:** Trung tâm xác thực tin cậy chọn 2 số nguyên tố p và q và đưa công khai các giá trị n, g và e, trong đó:
    - $n = p.q$
    - g là phần tử sinh của cả Z_p^* và Z_q^*
    - $e \in Z^*_{\lambda(n)}$. Ở đây, hàm Carmichael của n được xác định như sau:
      - $\lambda(n) = BCNN(p - 1, q - 1)$
    - Tính khoá bí mật của trung tâm $d = e^{-1} \bmod \lambda(n)$ với $d \in Z^*_{\lambda(n)}$.

4.3 Hệ mật dựa trên định danh

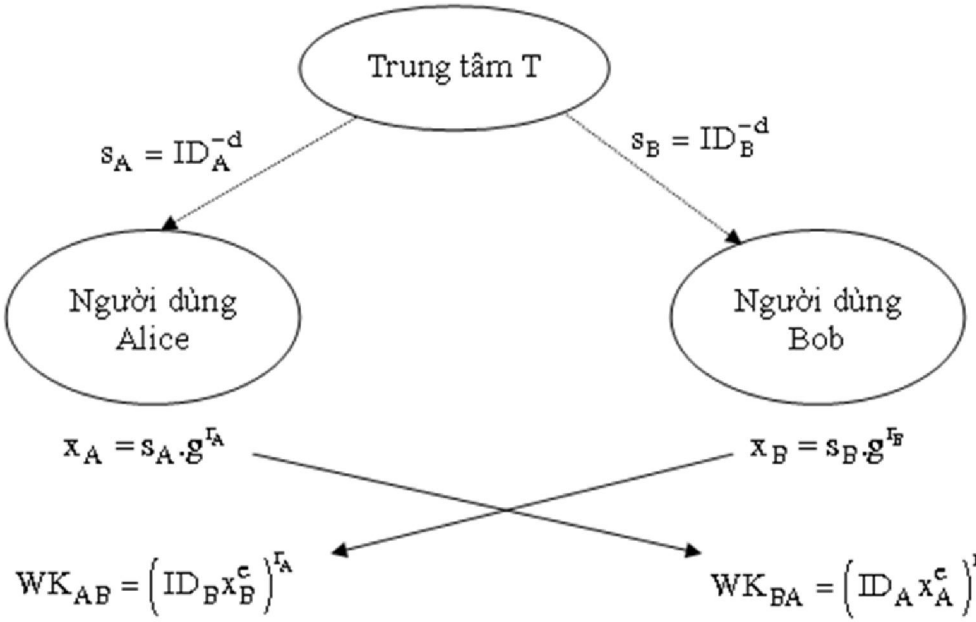
- **(2) Pha tham gia của người dùng**
  - Cho ID_i là thông tin định danh của người dùng thứ i (I = A, B, C, …). Cho a_i là khoá bí mật của người dùng i thoả mãn: $s_i \equiv Id_i^{-d} \bmod n$. Sau đó trung tâm sẽ công bố (e, n, g, Id_i) và phân phát s_i tới mỗi người dùng i qua một kênh an toàn (hoặc bằng cách dùng thẻ)
- **(3) Pha tạo khóa chung**
  - Ta giả sử ở đây rằng hai người dùng Alice và Bob muốn chia sẻ một khoá chung (chẳn hạn để dùng cho một hệ mật khoá bí mật).
  - Trước tiên Alice tạo một số ngẫu nhiên r_A và tính: $x_A \equiv s_A g^{r_A} \pmod n$ và gửi nó cho Bob.
  - Tương tự, Bob tạo một số ngẫu nhiên r_B và tính: $x_B \equiv s_B g^{r_B} \pmod n$ và gửi nó cho Alice.

4.3 Hệ mật dựa trên định danh

- Tiếp theo, Alice tính: $WK_{AB} = \left(ID_B x_B^e\right)^{r_A} \left(\mathrm{mod} n\right)$
- Tương tự, Bob tính: $WK_{BA} = \left(ID_A x_A^e\right)^{r_B} \left(\mathrm{mod} n\right)$
- WKAB và WKBA sẽ dùng làm khoá chung vì:

$$
\begin{aligned}
WK_{AB} &= \left(ID_B . x_B^e\right)^{r_A} \\
&= \left(ID_B \left(s_B . g^{r_B}\right)^e\right)^{r_A} \\
&= \left(ID_B \left(ID_B^{-d}\right)^e . g^{r_B e}\right)^{r_A} \\
&= g^{e . r_B . r_A} \\
&= WK_{BA} \left(\mathrm{mod} n\right)
\end{aligned}
$$

4.3 Hệ mật dựa trên định danh



Sơ đồ trao đổi khoá Okamoto-Tanaka

4.3 Hệ mật dựa trên định danh

- **Ví dụ:**
  - $p = 11, q = 13, n = p.q = 143, \lambda(143) = 60.$

    $Z^*(\lambda) = \{1, 7, 11, 13, 17, 19, 23, 29, 37, 41, 43, 47, 49, 53, 59\}$. Giả sử $e = 43$.
  - Mô phỏng quá trình trao đổi khóa?

4.3 Hệ mật dựa trên định danh

- Tính $d = e^{-1} \bmod \lambda(n) = 7$
- Với $ID_i = 2$ và $ID_j = 3$ ta có:
  - $s_i = 2^{-7} \bmod 143 = 19$; $s_j = 3^{-7} \bmod 143 = 126$.
- Ở pha tạo khóa chung:
  - Giả sử A chọn $r_i = 3$, khi đó $X_i = 19.2^3 \bmod 143 = 9$. A gửi X_i cho B
  - Giả sử B chọn $r_j = 2$, khi đó $X_j = 126.2^2 \bmod 143 = 75$. B gửi X_j cho A.
  - A tính $WK_{ij} = \left(ID_i X_i^e\right)^j \bmod n = \left(2.9^{43}\right)^2 \bmod 143 = 25$
  - B tính $WK_{ji} = 2^{43.2.3} \bmod 143 = 25$

4.4 Các sơ đồ chữ kí số không nén

- **Chữ kí số:**
  - Ta sẽ nghiên cứu một ứng dụng điển hình trong máy tính thể hiện một nhu cầu thông thường của con người: lệnh chuyển tiền từ một người này tới một người khác.
  - Về văn bản đây là một dạng séc đã được "máy tính hóa".
  - Giao dịch ở dạng giấy tờ được thực hiện như sau:
    - Séc là một đối tượng xác định có tư cách giao dịch thương mại
    - Chữ kí trên séc xác nhận tính xác thực bởi vì chắc chắn rằng chỉ có người kí hợp pháp mới có thể tạo được chữ kí này
    - Trong trường hợp bất hợp pháp thì sẽ có một bên thứ 3 có thể được gọi vào để phán xét tính xác thực.
    - Séc bị hủy để nó không được sử dụng lại
    - Séc giấy không thể thay đổi được, hay hầu hết các kiểu thay đổi đều có thể dễ dàng phát hiện được

4.4 Các sơ đồ chữ kí số không nén

- Giao dịch trên máy tính đòi hỏi một mô hình khác. Xét mô hình sau đây:
  - Sandy gửi cho ngân hàng của mình một thông báo ủy quyền ngân hàng chuyển 100$ cho Tim.
  - Ngân hàng của Sandy phải làm những việc sau:
    - Kiểm tra và chứng tỏ được rằng thông báo này thực sự tới từ Sandy, nếu sau đó cô ta không nhận là mình đã gửi nó
    - Phải biết chắc rằng toàn bộ thông báo này là của Sandy và nó đã không bị sửa đổi
  - Sandy cũng muốn biết chắc rằng ngân hàng của mình không thể giả mạo những thông báo tương tự.
  - Cả hai bên đều muốn đảm bảo rằng thông báo đó là thông báo mới, không phải là một thông báo trước đó được sử dụng lại và nó không bị sửa đổi trong khi truyền

4.4 Các sơ đồ chữ kí số không nén

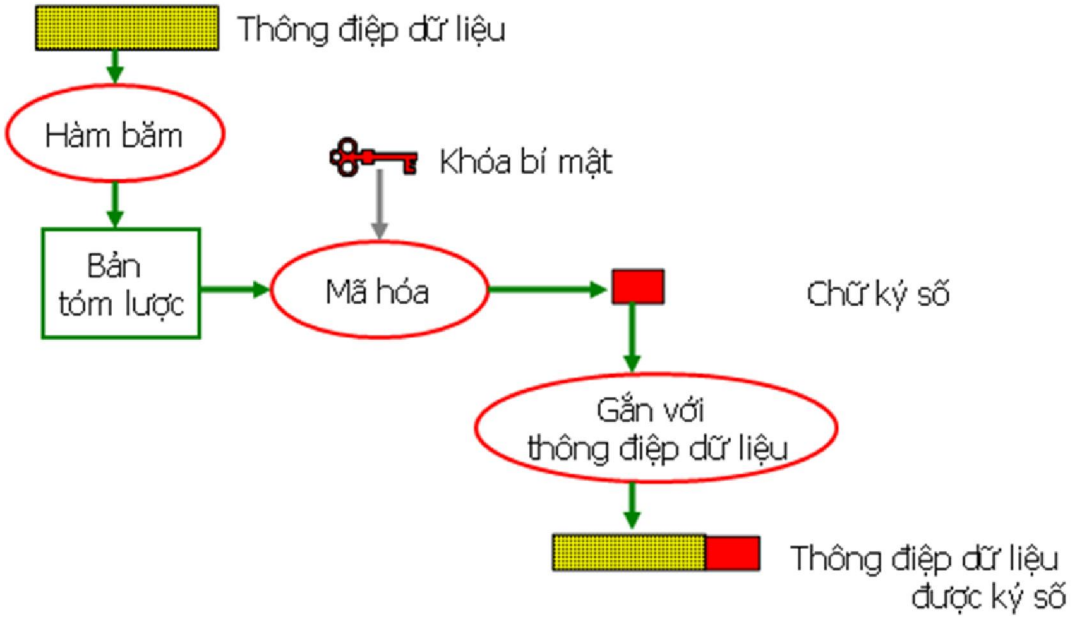
- Chữ kí số là một giao thức tạo ra một hiệu quả tương tự như chữ kí thực:
  - Nó là một dấu hiệu mà chỉ có người gửi mới có thể tạo ra nhưng những người khác có thể nhận thấy được rằng nó là của người gửi.
  - Giống như chữ kí thực, chữ kí số dùng để xác nhận nội dung thông báo

4.4 Các sơ đồ chữ kí số không nén

- **Chữ kí số phải thỏa mãn điều kiện sau đây:**
  - **Không thể giả mạo:** Nếu P kí thông báo M bằng chữ kí S(P, M) thì không một ai có thể tạo được cặp [M, S(M,P)]
  - **Xác thực:** Nếu R nhận được cặp [M, S(M,P)] được coi là của P thì R có thể kiểm tra được rằng chữ kí có thực sự là của P hay không. Chỉ P mới có thể tạo được chữ kí này và chữ kí được "gắn chặt" với M.
  - **Không thể thay đổi:** sau khi được phát M không thể bị thay đổi bởi S, R hoặc bởi một kẻ thu trộm nào
  - **Không thể sử dụng lại:** Một thông báo trước đó đã được đưa ra sẽ ngay lập tức bị R phát hiện

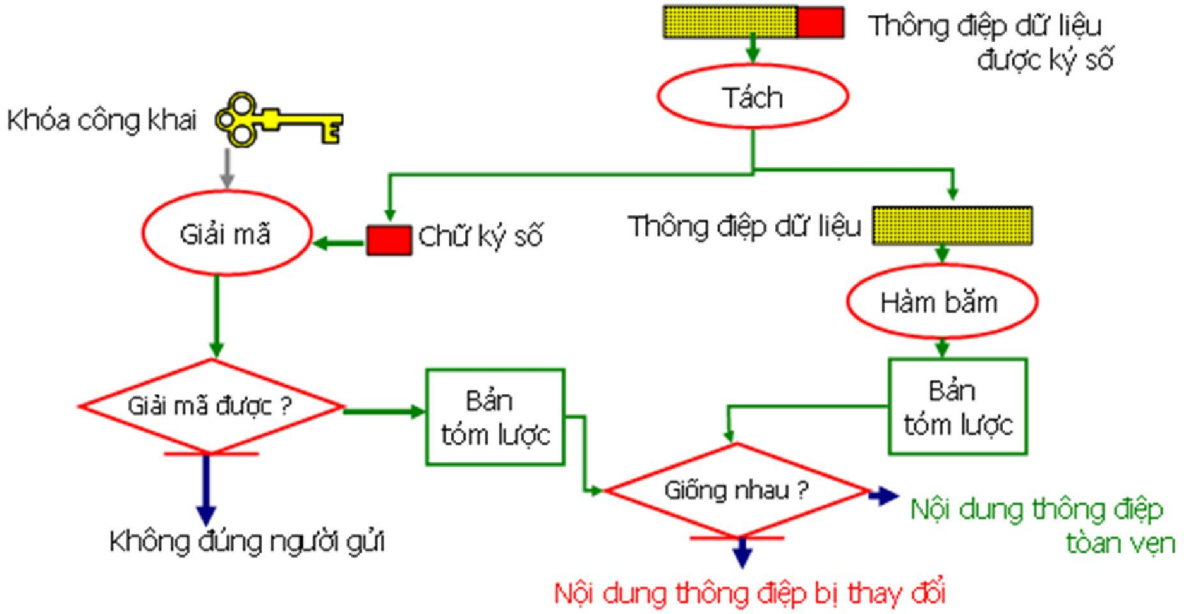
4.4 Các sơ đồ chữ kí số không nén

- **Tạo chữ ký số**



4.4 Các sơ đồ chữ kí số không nén

- **Thẩm định chữ ký số**



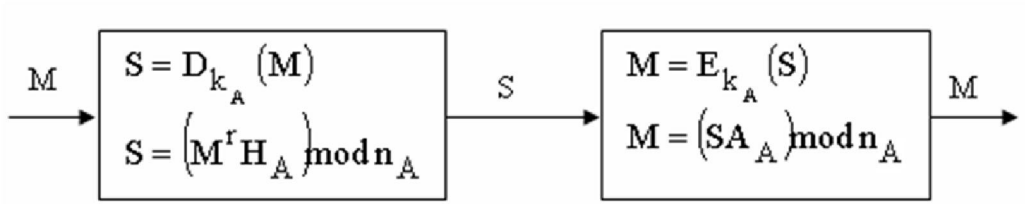
4.4 Các sơ đồ chữ kí số không nén

Chữ ký số Shamir:

- Chuỗi bít thông báo trước hết được tách thành các véctơ k bít M.
- Giả sử $M \in [0, n-1]$, $M = (m_1, \ldots, m_k)$
- Một ma trận nhị phân bí mật k x 2k (ma trận H) được chọn ngẫu nhiên cùng với một giá trị modulo n, trong đó n là một số nguyên tố ngẫu nhiên k bit. Một vecto A 2K bit (được dùng làm khóa công khai) được chọn trên cơ sở giải hệ phương trình tuyến tính sau:

$$\begin{pmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,2k-1} & h_{1,2k} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,2k-1} & h_{2,2k} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{k,1} & h_{k,2} & \cdots & h_{k,2k-1} & h_{k,2k} \end{pmatrix} \times \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_{2k} \end{pmatrix} \bmod n = \begin{pmatrix} 2^0 \\ 2^1 \\ \vdots \\ 2^{k-1} \end{pmatrix}$$

4.4 Các sơ đồ chữ kí số không nén



Xác thực thông báo dùng sơ đồ chữ kí

4.4 Các sơ đồ chữ kí số không nén

- **Xác thực thông báo dùng sơ đồ Shamir**
  - Người gửi A có thể chứng tỏ cho B tính xác thực của thông báo M bằng cách dùng khóa riêng của mình (H_A, n_A) đối với thông báo M: $S = D_{k_A}(M)$

    $$S = M^r \times H_A \bmod n_A$$
  - Trong đó: $M^r = (m_k, m_{k-1}, \ldots, m_2, m_1)$
  - Các bít của thông báo đã ký là: $s_i = \sum_{i=1}^{k} m_i h_{ij}$, với $1 \le j \le 2k$, $s_i \in [0, k]$
  - Chỉ có A có thể tạo ra 2Kbít $\{s_i\}$ từ k bít của thông báo $\{m_i\}$ vì chỉ có A mới tạo được $2.k^2$ phần tử của ma trận $\{h_{ij}\}$

4.4 Các sơ đồ chữ kí số không nén

Kiểm tra thông báo

- Mỗi người dùng trên mạng có thể kiểm tra tính xác thực của thông báo do A gửi bằng cách dùng thông tin công khai (A_A, n_A):

$$E_{k_A}(S) = S \times A_A \bmod n_A$$

$$E_{k_A}(S) = \left(M^r \times H_A\right) \times A_A \bmod n_A$$

$$E_{k_A}(S) = M$$

- Tức là:

$$\sum_{j=1}^{2k} s_j a_j \bmod n_A = \sum_{j=1}^{2k} \left[\sum_{i=1}^{k} m_i h_{ij} \right] a_j \bmod n_A$$

$$\sum_{j=1}^{2k} s_j a_j \bmod n_A = \sum_{i=1}^{k} m_i \left[\sum_{j=1}^{2k} h_{ij} a_j \right] \bmod n_A$$

$$= \sum_{i=1}^{k} m_i 2^{i-1} \bmod n_A$$

4.4 Các sơ đồ chữ kí số không nén

- **Ví dụ áp dụng:**
  - Cho k = 3, n = 5 và ma trận $H=\begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$
  - Tìm khóa công khai A_A khi cho trước các giá trị $a_1 = 1$, $a_2 = 3$, $a_3 = 4$?
  - Hãy xác thực thông báo M = 3 và kiểm tra tính xác thực của thông báo M đó?

4.4 Các sơ đồ chữ kí số không nén

- Để tránh nguy cơ thám mã có thể xác định được ma trận H với một cặp bản rõ – mã thích hợp. Ta sẽ tìm cách ngẫu nhiên hóa thông báo M trước khi kí. Ta làm như sau:
  - Vecto A sẽ được nhân với một vecto ngẫu nhiên R có K 2K bit: $R = (r_1, r_2, \ldots, r_{2k})$
  - Thực hiện phép biến đổi: M' = (M – R x A) mod n hay M = (M' + R x A) mod n
  - Để kí thông báo đã biến đổi M' ta tính theo công thức sau: $S' = M'^{r} \times H + R$
  - Khi đó để xác thực, bên nhận tính: S' x A mod n = M
  - Ví dụ:
    - Trở lại ví dụ trước, ta chọn ngẫu nhiên R = (1, 1, 0, 0, 0, 1)
    - Hãy xác thực thông báo M = 3 và kiểm tra tính xác thực của thông báo M đó?

4.4 Các sơ đồ chữ kí số không nén

- **Sơ đồ xác thực Ong-Schnorr-Shamir**
  - (1) Người gửi A chọn một số nguyên lớn n_A.
  - (2) Sau đó A chọn một số ngẫu nhiên k_A nguyên tố cùng nhau với n_A
  - (3) Khóa công khai k_A được tính như sau:
    - $K_A = -(k_A)^{-2} \bmod n_A$
    - Cặp (K_A, n_A) được công khai cho mọi người dùng trong mạng

4.4 Các sơ đồ chữ kí số không nén

- (4) Để xác thực một thông báo M ($\gcd(M, n_A) = 1$), người gửi sẽ chọn một số ngẫu nhiên R_A ($\gcd(R_A, n_A) = 1$) rồi tính thông báo được xác thực là cặp (S_1, S_2) sau:

$$S_1 = 2^{-1}\left[\left(MR_A^{-1}\right) + R_A\right] \bmod n_A$$

$$S_2 = 2^{-1} k_A \left[\left(MR_A^{-1}\right) - R_A\right] \bmod n_A$$

- (5) Sau đó A gửi S cho bên thu qua mạng
- (6) Việc kiểm tra tính xác thực ở bên thu được thực hiện như sau:

$$S_1^2 + \left(K_A S_2^2\right) \bmod n_A = M$$

4.4 Các sơ đồ chữ kí số không nén

- **Ví dụ:**
  - Cho $n_A = 23$, $k_A = 7$
  - Tính khóa công khai K_A?
  - Chọn $R_A = 13$, với M = 25, xác thực M và kiểm tra tính xác thực của M?

4.4 Các sơ đồ chữ kí số có nén

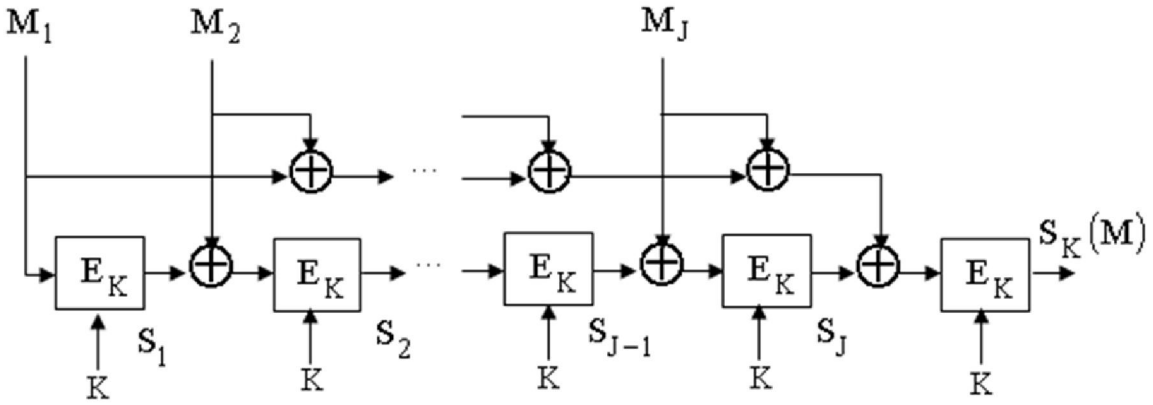
- **Nén chữ kí**

$$S_1 = E_K(M_1);\ S_2 = E_K(M_2 \oplus S_1)$$

$$\vdots$$

$$S_J = E_K(M_J \oplus S_{J-1})$$ Theo cách này ta tạo được một chữ ký $S_k(M)$

$$S_K(M) = E_K(M_1 \oplus M_2 \oplus \ldots \oplus M_J \oplus S_J)$$



4.4 Các sơ đồ chữ kí số có nén

Sơ đồ chữ ký Diffie – Lamport

- (1) Chọn n cặp khóa ngẫu nhiên (chẳng hạn như khóa 56 bít của DES) được gửi bí mật:

$$\begin{aligned} i = 1 &\Rightarrow (K_{1,0}, K_{1,1}) \\ i = 2 &\Rightarrow (K_{2,0}, K_{2,1}) \\ &\vdots \\ i = n &\Rightarrow (K_{n,0}, K_{n,1}) \end{aligned}$$

- (2) Chọn một dãy S gồm n cặp véctơ ngẫu nhiên (chẳng hạn như các khối đầu vào 64 bít của DES), dãy này được đưa ra công khai:

$$S = \{(S_{1,0}, S_{1,1}), (S_{2,0}, S_{2,1}), \ldots, (S_{n,0}, S_{n,1})\}$$

- (3) Tính R là dãy các khóa mã (chẳng hạn là các dãy ra của DES). Dãy R cũng được đưa công khai, trong đó $R_{ij} = E_{K_{i,j}}(S_{i,j})$, $1 \le i \le n$, $j = 0, 1$

$$R = \{(R_{1,0}, R_{1,1}), (R_{2,0}, R_{2,1}), \ldots, (R_{n,0}, R_{n,1})\}$$

4.4 Các sơ đồ chữ kí số có nén

- Chữ ký SG(M) của một bản tin n bít $M = (m_1, m_2, ..., m_n)$ chính là dãy khóa sau:

$$= \left(K_{1,i_1}, K_{2,i_2}, \ldots, K_{n,i_n} \right)$$

- Trong đó $i_j = m_j$
- Ví dụ: Thông báo $M = m_1 \quad m_2 \quad m_3 \quad m_4 \quad \ldots \quad m_{n-1} \quad m_n$

$$M = 1 \quad 0 \quad 0 \quad 1 \quad \ldots \quad 1 \quad 1$$

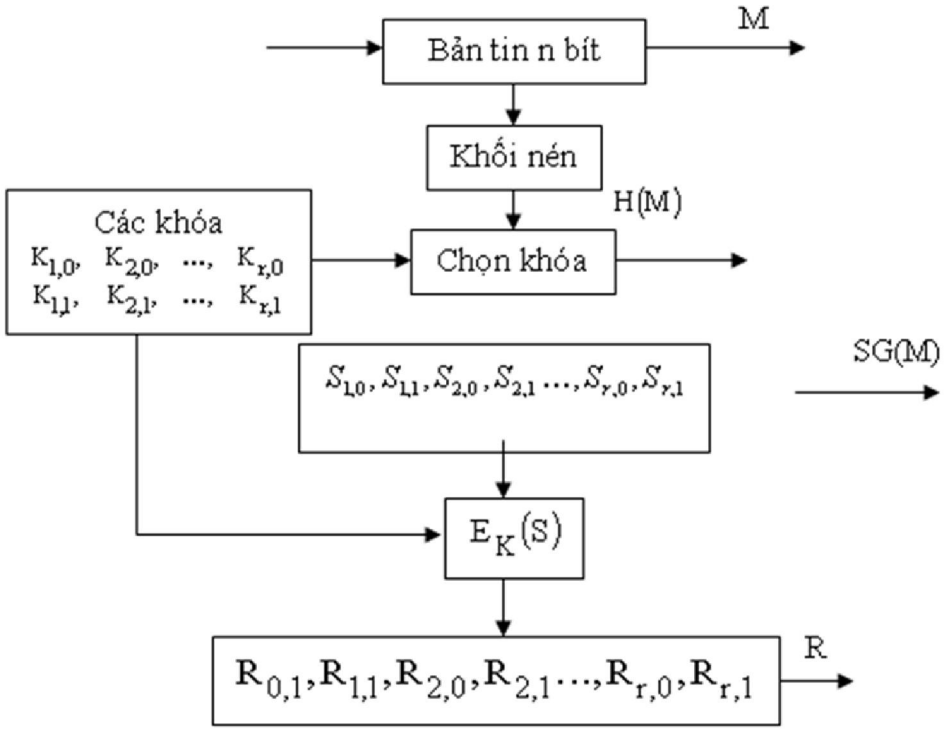
thì SG(M) là:

$$SG(M) = K_{1,i_1} \quad K_{2,i_2} \quad K_{3,i_3} \quad K_{4,i_4} \quad \ldots \quad K_{n-1,i_{n-1}} \quad K_{n,i_n}$$

$$SG(M) = K_{1,1} \quad K_{2,0} \quad K_{3,0} \quad K_{4,1} \quad \ldots \quad K_{n-1,1} \quad K_{n,1}$$

- Bản tin M và chữ ký SG(M) đều được gửi tới nơi thu

4.5 Các sơ đồ chữ kí số có nén



Sơ đồ chữ kí Diffie - Lamport (đầu phát)

4.5 Các sơ đồ chữ kí số có nén

- Bản tin có thể kiểm tra tính xác thực của thông báo bằng cách:
  - Mã hóa các véctơ tương ứng của dãy S đã biết với chữ ký SG(M) đã nhận
  - So sánh bản mã tạo ra với dãy R đã biết

$$E_{K_{1,i_1}}\left(S_{1,i_1}\right)\overset{?}{=}R_{1,i_1}$$

$$E_{K_{2,i_2}}\left(S_{2,i_2}\right)\overset{?}{=}R_{2,i_2}$$

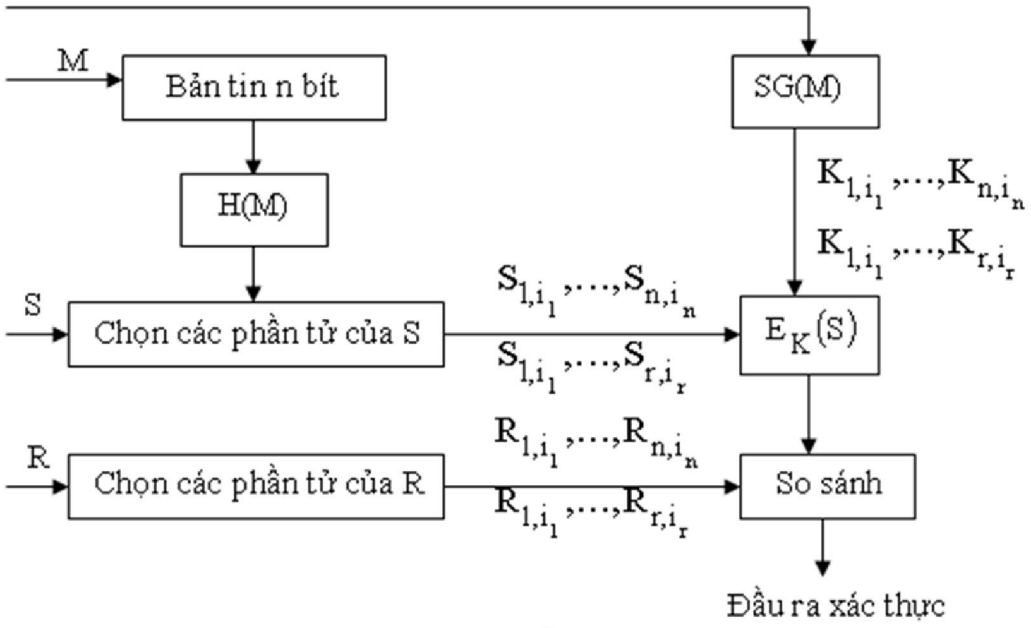
$$\vdots$$

$$E_{K_{n,i_n}}\left(S_{n,i_n}\right)\overset{?}{=}R_{n,i_n}$$

  - Nếu dãy n véctơ này bằng nhau thì chữ ký được xem là đã xác thực

$$\left(R_{1,i_1},R_{2,i_2},\ldots,R_{n,i_n}\right)=\left[E_{K_{1,i_1}}\left(S_{1,i_1}\right),\ldots,E_{K_{n,i_n}}\left(S_{n,i_n}\right)\right]$$

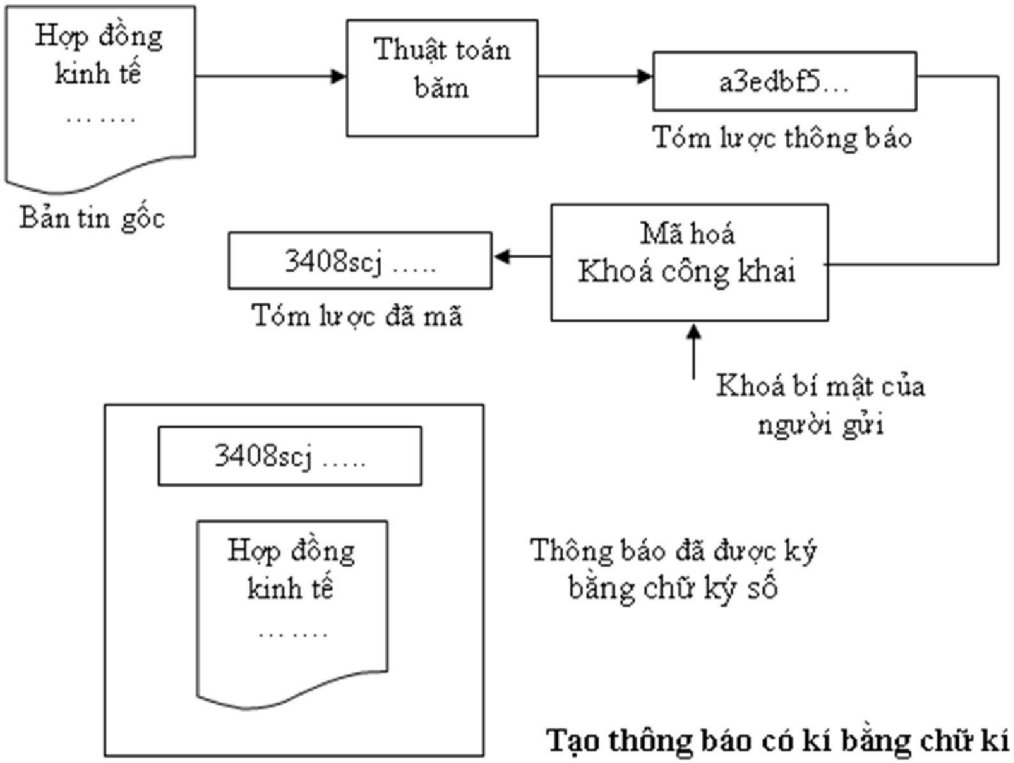
4.5 Các sơ đồ chữ kí số có nén



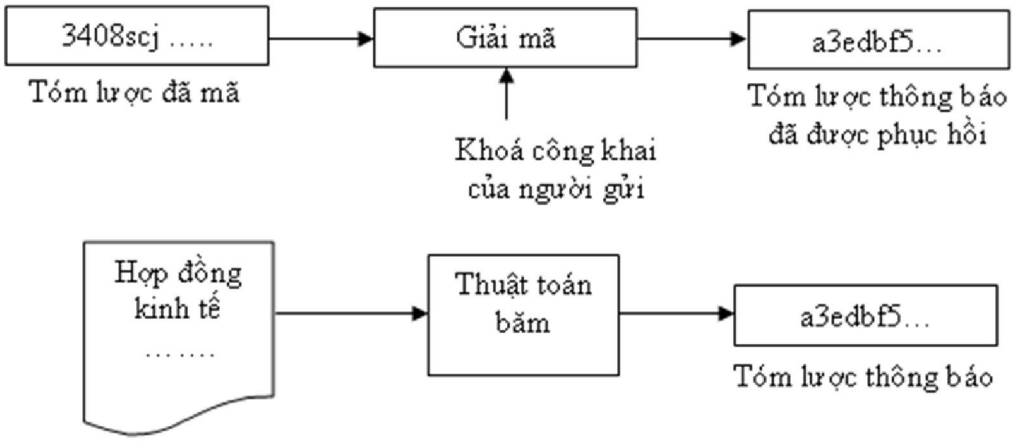
Quá trình kiểm tra chữ kí

4.5 Các sơ đồ chữ kí số có nén

- **Sơ đồ chữ ký RSA**



4.5 Các sơ đồ chữ kí số có nén



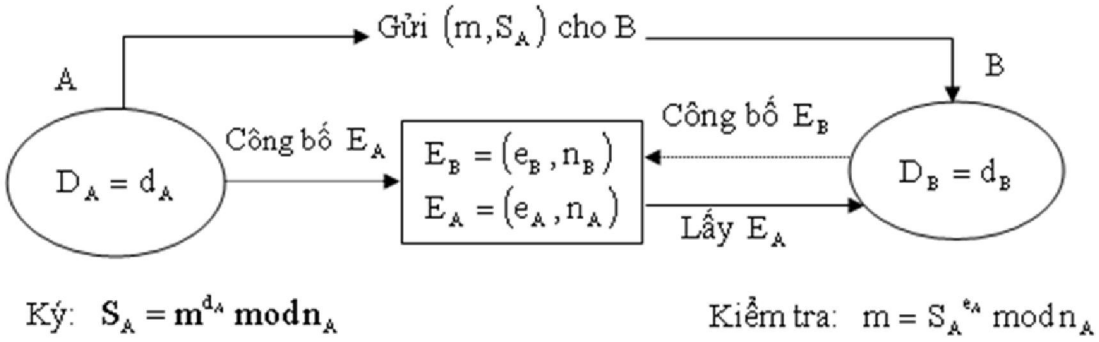
Các bước kiểm tra một thông báo đã kí

4.5 Các sơ đồ chữ kí số có nén

- **Ví dụ:** sơ đồ kí số RSA
  - n = p.q với p, q là các số nguyên tố lớn có kích thước tương đương
  - Với $K = \{(n, e, d): d \in Z_p^*, ed \equiv 1 \bmod (n)\}$
  - Ta có D = d là khóa bí mật, E = (n, e) là khóa công khai, m là bản tin cần kí
    - Tạo chữ kí: $S = sig_D(m) = m^d \bmod n$
    - Kiểm tra chữ kí: $ver_E(m, S) = TRUE \Leftrightarrow m \equiv S^e \bmod n$

4.5 Các sơ đồ chữ kí số có nén

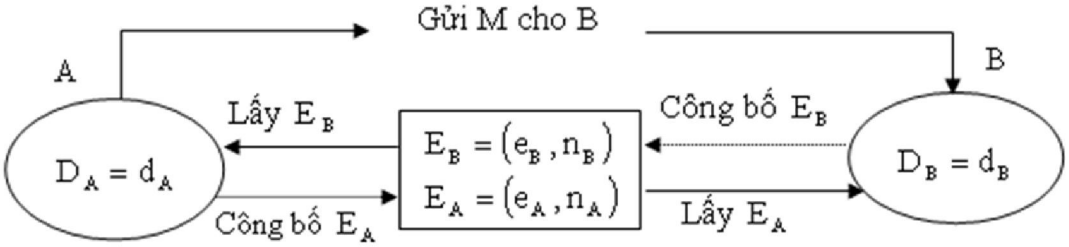
- **Trường hợp bản tin rõ m không cần bí mật:**
  - A ký bản tin m và gửi cho B.
  - B kiểm tra chữ ký của A



Sơ đồ chữ kí số RSA (không bí mật bản tin)

4.5 Các sơ đồ chữ kí số có nén

- **Trường hợp bản tin rõ m cần giữ bí mật:**
  - A ký bản tin rõ m để được chữ ký S_A.
  - Sau đó A dùng khoá mã công khai E_B của B để lập bản mã $M = E_B(m, S_A)$ rồi gửi đến B



A tính:

i. $S_A = m^{d_A} \bmod n_A$

ii. $X = m \| S_a$

iii. $M = X^{e_B} \bmod n_B$

B tính:

i. $X = M^{d_B} \bmod n_B$

ii. $X = m \| S_a$

iii. Kiểm tra: $m = S_A^{e_A} \bmod n_A$

Sơ đồ chữ kí số RSA (có bí mật bản tin)