**TCVN** TIÊU CHUẨN QUỐC GIA

# TCVN 12855-2 : 2020

# ISO/IEC 9796-2 : 2010

Xuất bản lần 1

# CÔNG NGHỆ THÔNG TIN – CÁC KỸ THUẬT AN TOÀN – LƯỢC ĐỒ CHỮ KÝ SỐ CHO KHÔI PHỤC THÔNG ĐIỆP - PHẦN 2: CÁC CƠ CHẾ DỰA TRÊN PHÂN TÍCH SỐ NGUYÊN

*Information technology – Security techniques – Digital signature achemes giving message recovery -*
*Part 2: Integer factorization based mechanisms*

HÀ NỘI – 2020

## Mục Lục

## Lời nói đầu

TCVN 12855-2:2020 hoàn toàn tương đương với ISO/IEC 9796-2:2010.

TCVN 12855-2:2020 (ISO/IEC 9796-2:2010) do Cục Quản lý mật mã dân sự và Kiểm định sản phẩm mật mã biên soạn, Ban Cơ yếu Chính phủ đề nghị, Tổng cục Tiêu chuẩn Đo lường Chất lượng thẩm định, Bộ Khoa học và Công nghệ công bố.

Bộ tiêu chuẩn TCVN 12855:2020 *Công nghệ thông tin – Các kỹ thuật an toàn – Lược đồ chữ ký số cho khôi phục thông điệp gồm* các tiêu chuẩn sau:

*TCVN 12855-2:2020 (ISO/IEC 9796-2:2010), Phần 2: Các cơ chế dựa trên phân tích số nguyên.*

*TCVN 12855-3:2020 (ISO/IEC 9796-3:2013), Phần 3: Các cơ chế dựa vào logarit rời rạc.*

Bộ tiêu chuẩn này có thể có các phần tiếp theo.

**TIÊU CHUẨN QUỐC GIA** **TCVN 12855 - 2 : 2020**

# Công nghệ thông tin – Các kỹ thuật an toàn – Các lược đồ chữ ký số cho khôi phục thông điệp – Phần 2: Các cơ chế dựa trên phân tích số nguyên

*Information technology – Security techniques – Digital signature schemes giving message recovery - Part 2: Integer factorization based mechanisms*

## 1 Phạm vi áp dụng

Tiêu chuẩn này quy định ba lược đồ chữ ký số cho khôi phục thông điệp, hai trong số đó là tất định (không ngẫu nhiên) và một là ngẫu nhiên. Sự an toàn của cả ba lược đồ đều dựa trên độ phức tạp của việc phân tích các số nguyên lớn. Tất cả ba lược đồ đều có thể cung cấp sự khôi phục toàn bộ hoặc một phần thông điệp.

Tiêu chuẩn này quy định phương pháp sản xuất khóa cho ba lược đồ chữ ký. Tuy nhiên, các kỹ thuật quản lý khóa và tạo số ngẫu nhiên (theo yêu cầu của lược đồ chữ ký ngẫu nhiên) nằm ngoài phạm vi của tiêu chuẩn này.

Cơ chế đầu tiên được quy định trong tiêu chuẩn này chỉ áp dụng cho các triển khai hiện có và được giữ lại vì lý do tương thích ngược.

## 2 Tài liệu viện dẫn

Các tài liệu viện dẫn sau rất cần thiết cho việc áp dụng tiêu chuẩn này. Đối với những tài liệu viện dẫn có năm công bố, thì áp dụng phiên bản được nêu. Đối với tài liệu viện dẫn không ghi năm công bố, thì áp dụng phiên bản mới nhất (bao gồm cả các sửa đổi, bổ sung).

*TCVN 11816 (ISO/IEC 10118) (tất cả các phần), Công nghệ thông tin - Các kỹ thuật an toàn - Hàm băm*

## 3 Thuật ngữ và định nghĩa

Tiêu chuẩn này áp dụng các thuật ngữ và định nghĩa sau:

**3.1**

**Khả năng** (capacity)

Số nguyên dương chỉ ra số bit có trong chữ ký thuộc phần có thể khôi phục được của thông điệp

**3.2**

**Miền chứng thư** (certificate domain)

Tập hợp các thực thể sử dụng các chứng thư khóa công khai được tạo bởi cùng một Cơ quan Chứng thực (CA) hoặc một tập hợp các CA hoạt động theo cùng một chính sách bảo mật

**3.3**

**Các tham số miền chứng thư** (certificate domain parameters)

Các tham số mật mã cụ thể cho một miền chứng thư , được biết đến và đồng ý bởi tất cả các thành viên trong miền chứng thư

**3.4**

**Hàm băm kháng va chạm** (collision-resistant hash-function)

Hàm băm thỏa mãn tính chất sau đây:

Không thể tính toán để tìm được hai giá trị đầu vào khác nhau mà ánh xạ đến cùng một đầu ra

[TCVN 11816 (ISO/IEC 10118-1)]

**3.5**

**Mã băm** (hash-code)

Xâu bit giá trị đầu ra của hàm băm

[TCVN 11816 (ISO/IEC 10118-1)]

**3.6**

**Hàm băm** (hash-function)

Hàm ánh xạ các xâu bit thành các xâu có độ dài cố định, thỏa mãn hai tính chất sau đây:

– Với một giá trị đầu ra cho trước, không thể tính toán để tìm được một giá trị đầu vào ánh xạ đến giá trị đầu ra đó;

– Với một giá trị đầu vào cho trước, không thể tính toán để tìm được một giá trị đầu vào khác sao cho ánh xạ đến cùng một giá trị đầu ra

[ISO/IEC 9797-2]

**3.7**

**Hàm tạo mặt nạ** (mask generation function)

Hàm ánh xạ các xâu bit thành các xâu bit có độ dài tùy ý, thỏa mãn tính chất sau đây:

– Với một phần cho trước của giá trị đầu ra chứ không phải giá trị đầu vào, không thể tính toán để dự đoán được phần còn lại của giá trị đầu ra

**3.8**

**Thông điệp** (message)

Xâu bit có độ dài bất kỳ

[TCVN 12214-1]

**3.9**

**Giá trị đại diện của thông điệp** (message representative)

Xâu bit được lấy từ một hàm của thông điệp và được kết hợp với khóa chữ ký bí mật để thu được chữ ký

**3.10**

**Xâu bộ bốn** (nibble)

Khối bốn bit liên tiếp nhau (một nửa xâu bộ tám)

**3.11**

**Phần không thể khôi phục** (non-recoverable part)

Phần của thông điệp được lưu trữ hoặc được truyền cùng với chữ ký; là rỗng khi thông điệp được khôi phục toàn bộ

**3.12**

**Xâu bộ tám** (octet)

Xâu tám bit

**3.13**

**Khóa riêng** (private key)

Khóa trong cặp khóa phi đối xứng của một thực thể và chỉ được sử dụng bởi thực thể đó

[TCVN 11817-1]

**3.14**

**Khóa chữ ký riêng** (private signature key)

Khóa bí mật định nghĩa phép biến đổi chữ ký bí mật

[TCVN 11817-1]

**3.15**

**Khóa công khai** (public key)

Khóa trong cặp khóa phi đối xứng của một thực thể và có thể được công khai

[TCVN 11817-1]

**3.16**

**Hệ thống khóa công khai** (public key system)

Lược đồ mật mã <chữ ký số> bao gồm ba hàm số sau đây:

- *Sản xuất khóa*, phương thức tạo một cặp khóa gồm một khóa chữ ký bí mật và một khóa xác thực công khai;

- *Tạo chữ ký*, phương thức tạo một chữ ký Σ từ một giá trị đại diện của thông điệp $F$ và một khóa chữ ký bí mật;

- *Mở chữ ký*, phương thức thu được giá trị đại diện của thông điệp $F^*$ từ một chữ ký Σ và một khóa xác thực công khai

CHÚ THÍCH Giá trị đầu ra của hàm này cũng chứa một số chỉ biểu thị thủ tục mở chữ ký đã thành công hay bị lỗi.

**3.17**

**Khóa xác thực công khai** (public verification key)

Khóa công khai định nghĩa phép biến đổi xác thực công khai

[TCVN 11817-1]

**3.18**

**Phần có thể khôi phục** (recoverable part)

Phần thông điệp được truyền tải bên trong chữ ký

**3.19**

**Salt** (Salt)

Mục dữ liệu ngẫu nhiên được tạo ra bởi thực thể ký trong quá trình tạo giá trị đại diện của thông điệp trong Lược đồ chữ ký 2

**3.20**

**Chữ ký (signature)**

Xâu bit kết quả của quá trình ký

[TCVN 12214-1]

**3.21**

**Trailer** (trailer)

Xâu bit có độ dài một hoặc hai xâu bộ tám, được nối vào cuối phần có thể khôi phục được của thông điệp trong quá trình tạo giá trị đại diện của thông điệp

## 4 Ký hiệu và chữ viết tắt

Tiêu chuẩn này áp dụng các ký hiệu và chữ viết tắt sau:

CHÚ THÍCH Trong hầu hết các trường hợp, các chữ cái viết hoa được sử dụng để biểu thị các xâu bit và các xâu bộ tám, còn các chữ cái viết thường được sử dụng để biểu thị các hàm số.

| | |
|---|---|
| $C$ | Độ dài bit xâu mã bộ tám của phần có thể khôi phục được của thông điệp (được sử dụng trong tạo giá trị đại diện của thông điệp trong Lược đồ chữ ký 2 và 3). |

| | |
|---|---|
| $c$ | Năng lực của lược đồ chữ ký, có nghĩa là số lượng bit tối đa sẵn sàng cho phần có thể khôi phục được của thông điệp. |
| $c^*$ | Độ dài thông điệp có thể khôi phục được, có nghĩa là độ dài được tính bằng bit của phần có thể khôi phục được của thông điệp ($c \geq c^*$). |
| $D, D'$ | Các xâu bit được tạo ra trong quá trình tạo giá trị đại diện của thông điệp trong Lược đồ chữ ký 2 và 3. |
| $D^*, D'^*$ | Các xâu bit được tạo ra trong lúc khôi phục thông điệp trong Lược đồ chữ ký 2 và 3. |
| $F$ | Giá trị đại diện của thông điệp (một xâu bit). |
| $F^*$ | Giá trị đại diện của thông điệp đã được khôi phục (giá trị đầu ra của bước mở chữ ký). |
| $g$ | Hàm tạo mặt nạ. |
| $H$ | Mã băm được tính toán bằng một hàm của thông điệp $M$ (một xâu bit). |
| $H^*$ | Mã băm đã được khôi phục bằng cách lấy từ việc khôi phục thông điệp |
| $h$ | Hàm băm kháng va chạm |
| $k$ | Độ dài bit của các mô-đun của khóa chữ ký bí mật và khóa xác thực công khai (xem Phụ Lục A). |
| $L_h$ | Độ dài bit của mã băm được tạo ra bởi hàm băm $h$. |
| $L_s$ | Độ dài bit của salt $S$. |
| $M$ | Thông điệp để ký (một xâu bit). |
| $M^*$ | Thông điệp đã được khôi phục từ chữ ký là kết quả của quá trình xác thực. |
| $M_1$ | Phần có thể khôi phục của thông điệp M, có nghĩa là $M = M_1 \| \| M_2$. |
| $M_1^*$ | Phần có thể khôi phục đã được khôi phục của thông điệp (được tạo ra trong quá trình khôi phục thông điệp). |
| $M_2$ | Phần không thể khôi phục của thông điệp M, có nghĩa là $M = M_1 \| \| M_2$. |
| $M_2^*$ | Phần không thể khôi phục của thông điệp, là đầu vào của quá trình xác thực. |
| $N$ | Xâu bit được xây dựng trong tạo giá trị đại diện của thông điệp trong Lược đồ |

| | |
|---|---|
| | chữ ký 2 và 3. |
| $N^*$ | Xâu bit được tạo ra trong quá trình khôi phục thông điệp trong Lược đồ chữ ký 2 và 3. |
| $P$ | Một xâu các bit 0 được xây dựng trong tạo giá trị đại diện của thông điệp trong Lược đồ chữ ký 2 và 3. |
| $S$ | Salt (một xâu bit). |
| $S^*$ | Salt đã được khôi phục (một xâu bit). |
| $t$ | Số xâu bộ tám trong trường Trailer ( $t$=1 hoặc 2). |
| $T$ | Trường Trailer (một xâu có độ dài 8$t$ bit được sử dụng trong tạo giá trị đại diện của thông điệp). |
| $\Delta$ | Số nguyên trong khoảng từ 0 đến 7 được sử dụng trong đặc tả phân bổ thông điệp. |
| $\delta$ | Số nguyên trong khoảng từ 0 đến 7 được sử dụng trong đặc tả của Lược đồ chữ ký 2 và 3. |
| $\Sigma$ | Chữ ký (một xâu bit chứa k-1 hoặc k bit). |
| $\|A\|$ | Độ dài bit của xâu bit $A$, nghĩa là số bit trong $A$. |
| $A \| B$ | Phép ghép nối xâu bit A và B (theo thứ tự đó). |
| $\lceil a \rceil$ | Đối với một số thực $a$, số nguyên nhỏ nhất không nhỏ hơn $a$. |
| $a$ mod $n$ | Đối với các số nguyên $a$ và $n$, ($a$ mod $n$) biểu thị số dư (không âm) thu được khi chia a cho n. Một cách tương đương nếu b = a mod n, thì b là số nguyên duy nhất thỏa mãn:<br>(i) 0 ≤ b < n, và<br>(ii) (b-a) là bội số nguyên của n. |
| $\oplus$ | Toán tử XOR thực hiện trên bit, được sử dụng để kết hợp hai xâu nhị phân có cùng độ dài. |

## 5 Sự chuyển đổi giữa các xâu bit và số nguyên

Để biểu diễn một số nguyên không âm $x$ dưới dạng một xâu bit có độ dài bằng $l$ ($l$ phải thỏa mãn $2^l > x$), số nguyên sẽ được viết lại dưới dạng biểu thức nhị phân duy nhất:

$$x = 2^{l-1}x_{l-1} + 2^{l-2}x_{l-2} + \cdots + 2x_1 + x_0$$

Trong đó $0 \le x_i \le 2$ (lưu ý rằng một hoặc vài chữ số đầu có thể bằng 0 nếu $x < 2^{l-1}$. Xâu bit sẽ là

$x_{l-1}x_{l-2}\dots x_0$ .

Để biểu diễn một xâu bit $x_{l-1}x_{l-2}\dots x_0$ (có độ dài l) thành một số nguyên x, quá trình đảo ngược sẽ như sau, x sẽ là số nguyên được định nghĩa bởi công thức

$x = 2^{l-1}x_{l-1} + 2^{l-2}x_{l-2} + \cdots + 2x_1 + x_0.$

## 6 Yêu cầu

Người sử dụng tiêu chuẩn này, nếu có thể, được khuyến nghị áp dụng cơ chế thứ hai (Lược đồ chữ ký số 2). Tuy nhiên, trong các môi trường mà việc tạo ra các biến ngẫu nhiên bởi người ký được coi là không khả thi, thì Lược đồ chữ ký số 3 được khuyến nghị sử dụng.

Người sử dụng muốn dùng cơ chế chữ ký số theo tiêu chuẩn này phải đảm bảo các thuộc tính sau đây được thỏa mãn:

a) Thông điệp *M* để ký là một chuỗi nhị phân có độ dài bất kỳ, có thể rỗng.

b) Hàm ký sử dụng khóa chữ ký bí mật, trong khi hàm xác thực sử dụng khóa xác thực công khai tương ứng.

– Mỗi thực thể ký sẽ sử dụng và giữ bí mật khóa chữ ký bí mật của mình tương ứng với khóa xác thực công khai.
– Mỗi thực thể xác thực phải biết khóa xác thực công khai của thực thể ký.

c) Việc sử dụng các lược đồ chữ ký được quy định trong tiêu chuẩn này đòi hỏi phải lựa chọn một hàm băm kháng va chạm *h*. Hàm băm được tiêu chuẩn hóa theo TCVN 11816. Sẽ có một ràng buộc giữa cơ chế ký và hàm băm trong khi sử dụng. Nếu không có ràng buộc này, kẻ thù có thể yêu cầu sử dụng một hàm băm yếu (chứ không phải là một hàm băm thực) và từ đó giả mạo chữ ký.

CHÚ THÍCH 1 Có nhiều cách để thực hiện điều kiện này. Các tùy chọn sau được liệt kê theo thứ tự tăng dần của rủi ro.

1. Yêu cầu một hàm băm cụ thể khi sử dụng một cơ chế ký cụ thể. Quá trình xác minh sẽ chỉ sử dụng hàm băm cụ thể đó. TCVN 12214-3 cung cấp một ví dụ về tùy chọn này khi cơ chế DSA yêu cầu sử dụng chức hàm băm chuyên dụng 3 theo TCVN 11816-3 (TCVN 11816-3) (còn gọi là SHA-1).

2. Cho phép tập hợp các hàm băm và chỉ định hàm băm được sử dụng trong các tham số miền chứng thư. Trong miền chứng thư này, quá trình xác minh sẽ sử dụng hàm băm được chỉ định trong chứng thư. Bên ngoài miền chứng thư này, rủi ro có thể phát sinh từ các cơ quan chứng thực (CA) không tuân thủ chính sách của người dùng. Nếu, ví dụ, một CA bên ngoài tạo ra một chứng thư số cho phép các hàm băm khác, thì các vấn đề giả mạo chữ ký có thể phát sinh. Trong trường hợp như vậy, một bên xác thực nhầm lẫn có thể đang tranh chấp với CA tạo ra chứng thư số khác.

3. Cho phép một tập hợp các hàm băm và chỉ định hàm băm được sử dụng bằng một số phương pháp khác, ví dụ như một số chỉ biểu thị trong thông điệp hoặc thỏa thuận song phương. Quá trình xác thực sẽ chỉ sử dụng hàm băm được chỉ ra bằng phương pháp khác này. Tuy nhiên, vẫn có nguy cơ kẻ thù có thể giả mạo chữ ký bằng cách sử dụng một hàm băm khác.

CHÚ THÍCH 2 Phương pháp khác được đề cập đến trong đoạn 3 ngay phía trên có thể dưới dạng một số mã nhận dạng hàm băm có trong giá trị đại diện của thông điệp *F* (xem mục 8.2.2 và 9.2.3). Nếu mã nhận dạng hàm băm được bao gồm trong *F* theo cách này thì kẻ tấn công không thể sử dụng lại một chữ ký hiện có với cùng $M_1$ và một $M_2$ khác, ngay cả khi bên xác thực có thể bị thuyết phục chấp nhận chữ ký được tạo ra bằng cách sử dụng một hàm băm đủ yếu mà có thể tìm được tiền

ảnh. Tuy nhiên, như đã thảo luận chi tiết trong [16] (xem phụ lục D), trong trường hợp này và sử dụng hàm băm yếu, kẻ tấn công vẫn có thể tìm thấy một chữ ký mới với một $M_1$ "ngẫu nhiên".

CHÚ THÍCH 3 Cuộc tấn công được đề cập trong CHÚ THÍCH 2 thu được một chữ ký mới với $M_1$ ' ngẫu nhiên' có thể được ngăn chặn bằng cách yêu cầu sự tồn tại của một cấu trúc cụ thể trong $M_1$. Chẳng hạn như, có thể áp đặt một giới hạn độ dài cho $M_1$ đủ nhỏ hơn năng lực của lược đồ chữ ký (xem thêm ở Phụ lục D). Đối với các lược đồ chữ ký số 2 và 3, giới hạn độ dài cho $M_1$ cũng có thể ngăn chặn kẻ tấn công sử dụng lại các chữ ký hiện có ngay cả khi không có mã nhận dạng hàm băm được đưa vào trong giá trị đại diện của thông điệp, với điều kiện hàm tạo mặt nạ $g$ dựa trên hàm băm. Điều này được dựa trên giả định hợp lý rằng hàm băm yếu là một hàm băm "mục đích chung" chứ không phải là một hàm băm được thiết kế chỉ nhằm mục đích giả mạo chữ ký.

Người sử dụng một cơ chế chữ ký số cần tiến hành một đánh giá rủi ro xem xét chi phí và lợi ích của các phương tiện thay thế khác nhau để đạt được điều kiện bắt buộc. Đánh giá này phải bao gồm một đánh giá về chi phí kết hợp với khả năng có một chữ ký giả mạo đang được sản xuất.

d) Bên xác thực chữ ký sẽ luôn luôn có một phương thức độc lập an toàn để xác định lược đồ nào trong ba lược đồ chữ ký được quy định trong tiêu chuẩn này đã được sử dụng để tạo ra chữ ký. Ngoài ra, nếu sử dụng Lược đồ chữ ký số 2 hoặc 3, bên xác thực chữ ký cũng phải có phương thức để xác định hàm tạo chữ ký nào trong hai hàm tạo chữ ký trong Phụ lục B đã được sử dụng. Điều này có thể thu được bằng cách quy định cơ chế và hàm tạo chữ ký trong 'các tham số miền' đã được đồng ý hoặc bằng cách đưa ra một mã nhận dạng rõ ràng cho lược đồ chữ ký và hàm tạo chữ ký trong chứng thư khóa công khai của người ký. Hàm tạo chữ ký cũng có thể được quy định trong một mã nhận dạng thuật toán liên kết với dữ liệu đã được ký.

e) Mỗi lược đồ chữ ký số được quy định trong tiêu chuẩn này đều có các tùy chọn cụ thể, phạm vi tùy chọn có thể của người ký phải được biết đến bởi bên xác thực bằng một phương thức độc lập an toàn. Các tùy chọn này bao gồm.

– Đối với tất cả ba lược đồ chữ ký số, bên xác thực phải biết xem trường trailer tùy chọn 1 hoặc 2 có đang làm việc hay không.
– Đối với lược đồ chữ ký số 2 và 3, bên xác thực phải biết $L_s$, độ dài của salt S.

Ví dụ, điều này có thể thu được bằng cách quy định lựa chọn tùy chọn trong "các tham số miền" hoặc bao gồm thông tin tùy chọn trong chứng thư khóa công khai của người ký.

## 7 Mô hình quá trình ký và xác thực

### 7.1 Tổng quan

Mô hình cho một lược đồ chữ ký cho khôi phục thông điệp được trình bày ở đây áp dụng cho cả ba lược đồ trong tiêu chuẩn. Khi được áp dụng cho một thông điệp $M$, một lược đồ chữ ký kiểu này có thể cung cấp khôi phục hoặc toàn bộ hoặc một phần thông điệp.

– Nếu $M$ là đủ ngắn, thì có thể khôi phục toàn bộ thông điệp vì có thể $M$ được bao gồm toàn bộ trong chữ ký.
– Nếu $M$ quá dài, thì có thể phục hồi thông điệp sẽ một phần. Trong trường hợp này, $M$ sẽ được chia thành phần thể khôi phục được, một xâu bit có độ dài giới hạn được bao gồm trong chữ ký, và phần không thể khôi phục được, một xâu các xâu bộ tám có độ dài bất kỳ được lưu trữ và / hoặc được truyền cùng với chữ ký.

Mô hình được chia thành ba phần: đặc tả thủ tục ký thông điệp, đặc tả thủ tục xác thực chữ ký và chi tiết của các khía cạnh bổ sung của ký và xác thực cần được định nghĩa để hoàn chỉnh đặc tả của một lược đồ chữ ký. Các điều 8, 9 và 10 quy định các khía cạnh bổ sung này cho ba lược đồ được định nghĩa trong tiêu chuẩn này.

### 7.2 Ký thông điệp

#### 7.2.1 Giới thiệu

Cho một thông điệp *M* được ký, cần phải thực hiện ba bước để tạo ra một chữ ký trên *M*, cụ thể là phân bổ thông điệp, tạo xâu có thể khôi phục được và tạo chữ ký.

– *Phân bổ thông điệp* bao gồm quá trình trong đó thông điệp được chia thành hai phần: phần có thể khôi phục được $M_1$ và phần không thể khôi phục $M_2$ (có thể rỗng). Độ dài của phần có thể khôi phục được được giới hạn trên bằng năng lực *c* của lược đồ chữ ký, một giá trị được xác định bởi việc lựa chọn lược đồ chữ ký và khóa của lược đồ. Phần có thể khôi phục sẽ được khôi phục từ chữ ký trong quá trình xác thực, trong khi phần không thể khôi phục phải được cung cấp cho bên xác thực bằng các phương thức khác (ví dụ nó có thể được gửi hoặc lưu trữ với chữ ký). Do đó, nếu thông điệp đủ ngắn, toàn bộ thông điệp có thể được phân bổ vào phần có thể khôi phục được, và phần không thể khôi phục được sẽ là rỗng.

– *Tạo giá trị đại diện của thông điệp* lấy đầu vào là hai phần này của thông điệp, và đầu ra là một xâu có định dạng, được gọi là *giá trị đại diện của thông điệp*, là đầu vào cho bước tạo chữ ký.

– *Tạo chữ ký* lấy đầu vào là giá trị đại diện của thông điệp và khóa chữ ký bí mật và đầu ra là chữ ký Σ. Quá trình này được thực hiện bằng cách sử dụng một hệ thống khoá công khai.

#### 7.2.2 Phân bổ thông điệp

Sự lựa chọn lược đồ chữ ký và khóa cho lược đồ xác định năng lực *c* của chữ ký, trong đó *c* phải thỏa mãn $c \geq 7$. Thông điệp *M* cần được ký sẽ được chia thành hai phần, $M_1$ và $M_2$ như sau.

Một độ dài thông điệp có thể khôi phục được $c^*$ sẽ được lựa chọn, trong đó $c^* \leq c$, $c^* \leq |M|$ và $c^* \equiv |M| \ (mod\ 8)$ . Đối với Lược đồ chữ ký 1, $c^*$ sẽ được thiết lập bằng với giá trị nhỏ nhất của $c - \Delta$ và $|M|$, trong đó $\Delta = (c - |M|) mod\ 8$.

– Nếu $|M| = c^*$ thì toàn bộ thông điệp sẽ có thể khôi phục được, có nghĩa là $M_1 = M$ và $M_2$ sẽ là rỗng.

– Nếu $|M| > c^*$ thì $M_1$ sẽ được thiết lập bằng với $c^*$ bit bên trái nhất của *M*, và $M_2$ sẽ được thiết lập bằng với phần còn lại của *M*, có nghĩa là $M_2$ chứa $|M|-c^*$ bit.

Trong cả hai trường hợp, $M = M_1||M_2$.

CHÚ THÍCH 1 Để phục vụ các mục đích thực tiễn, một ứng dụng có thể muốn cấu trúc thông điệp *M* để đảm bảo rằng dữ liệu mà nó cần được lưu trữ hoặc truyền dưới dạng rõ (ví dụ: thông tin địa chỉ) được phân bổ vào phần thông điệp không thể phục hồi $M_2$. Tuy nhiên, việc cấu trúc và biểu diễn của thông điệp *M* là nằm ngoài phạm vi của tiêu chuẩn này.

CHÚ THÍCH 2 Phương thức phân bổ thông điệp đảm bảo rằng $M_2$ luôn luôn có độ dài là số nguyên lần các xâu bộ tám. Hơn nữa, chọn $c^*$ là giá trị nhỏ nhất của $c - \Delta$ và $|M|$, trong đó $\Delta = (c - |M|) mod 8$, đảm bảo rằng $M_1$ càng dài càng tốt do điều kiện

này. Ngoài ra, nếu $M$ có độ dài là số nguyên lần các xâu bộ tám, tức là nếu |M| là bội số nguyên của 8, thì cả $M_1$ và $M_2$ sẽ bao gồm một số nguyên lần các xâu bộ tám.

### 7.2.3 Tạo giá trị đại diện của thông điệp

Bước này sẽ lấy đầu vào là phần có thể khôi phục và phần không thể khôi phục của thông điệp, $M_1$ và $M_2$, và đầu ra là giá trị đại diện của thông điệp $F$. Điều này sẽ đạt được bằng cách sử dụng một trong các phương pháp được quy định tại các điều 8, 9 và 10 trong tiêu chuẩn này. Các phương pháp này yêu cầu phải sử dụng hàm băm $h$, và trong trường hợp các cơ chế thứ hai và thứ ba, một hàm tạo mặt nạ $g$ cũng sử dụng $h$. Hàm băm $h$ được sử dụng sẽ được lựa chọn trong các hàm băm đã được tiêu chuẩn hóa theo TCVN 11816; hàm tạo mặt nạ $g$ sẽ được thiết lập bằng hàm được quy định tại Phụ lục C trong tiêu chuẩn này.

### 7.2.4 Tạo chữ ký

Bước này lấy đầu vào là giá trị đại diện của thông điệp và khóa chữ ký bí mật và đầu ra là *chữ ký* Σ. Điều này sẽ đạt được bằng cách sử dụng hệ thống khoá công khai được quy định trong Phục lục B trong tiêu chuẩn này.

## 7.3 Xác thực chữ ký

### 7.3.1 Giới thiệu

Một thông điệp đã được ký bao gồm hoặc chữ ký Σ trong trường hợp khôi phục toàn bộ hoặc phần không thể khôi phục được của thông điệp $M_2^*$ cùng với chữ ký Σ trong trường hợp khôi phục một phần. Một chữ ký sẽ được chấp nhận khi và chỉ khi quá trình xác thực thành công.

Cho chữ ký Σ và phần thông không thể khôi phục được $M_2^*$, cần phải thực hiện ba bước sau đây để xác thực Σ và khôi phục $M^*$, cụ thể là mở chữ ký, khôi phục thông điệp và lắp ghép thông điệp.

– Mở chữ ký lấy đầu vào là chữ ký Σ và khóa xác thực công khai và đầu ra là một giá trị đại diện của thông điệp đã được khôi phục $F^*$ hoặc trả về báo hiệu việc xác thực đã bị lỗi. Quá trình này được thực hiện bằng cách sử dụng một hệ thống khóa công khai.
– Khôi phục thông điệp lấy đầu vào là giá trị đại diện của thông điệp đã được khôi phục $F^*$ và phần không thể khôi phục của thông điệp $M_2^*$, và đầu ra là phần có thể khôi phục (đã được khôi phục) của thông điệp $M_1^*$ hoặc trả về báo hiệu việc xác thực đã bị lỗi.
– Lắp ghép thông điệp là quá trình mà thông điệp đã được phục hồi $M^*$ được khôi phục từ phần có thể khôi phục (đã được khôi phục) $M_1^*$ và phần không thể khôi phục $M_2^*$ (có thể rỗng).

### 7.3.2 Mở chữ ký

Bước này lấy đầu vào là chữ ký Σ và khóa xác thực công khai và đầu ra hoặc là một giá trị đại diện của thông điệp đã được khôi phục $F^*$ hoặc là trả về báo hiệu việc xác thực đã bị lỗi. Điều này sẽ đạt được bằng cách sử dụng hệ thống khoá công khai được quy định trong Phục lục B của tiêu chuẩn này.

### 7.3.3 Khôi phục thông điệp

Bước này lấy đầu vào là giá trị đại diện của thông điệp đã được khôi phục $F^*$ và phần không thể khôi phục của thông điệp $M_2^*$, và đầu ra là phần có thể khôi phục (đã được khôi phục) của thông điệp $M_1^*$ hoặc trả về báo hiệu việc xác thực đã bị lỗi. Điều này sẽ đạt được bằng cách sử dụng một trong các phương pháp được quy định tại các điều 8, 9 và 10 trong tiêu chuẩn này. Các phương pháp này yêu

cầu phải sử dụng hàm băm, và trong trường hợp các cơ chế thứ hai và thứ ba, là một hàm tạo mặt nạ. Hàm băm được sử dụng sẽ được lựa chọn trong các hàm băm đã được tiêu chuẩn hóa theo TCVN 11816; hàm tạo mặt nạ sẽ được thiết lập bằng hàm được quy định tại Phụ lục C trong tiêu chuẩn này.

### 7.3.4 Lắp ghép thông điệp

Lắp ghép thông điệp là quá trình mà thông điệp đã được phục hồi $M^*$ được khôi phục từ phần có thể khôi phục (đã được khôi phục) $M_1^*$ và phần không thể khôi phục $M_2^*$ (có thể rỗng). Có nghĩa là, thông điệp $M^*$ được ghép lại với nhau bằng $M^* = M_1^* || M_2^*$.

## 7.4 Quy định lược đồ chữ ký

Mục đích của mục 7.4 là định nghĩa các lựa chọn cần được thực hiện để quy định thống nhất các quá trình ký và xác thực được quy định trong tiêu chuẩn này.

a) Các bước phân bổ thông điệp và lắp ghép thông điệp được định nghĩa duy nhất trong tiêu chuẩn này.

b) Phải chọn một trong ba tùy chọn trong các bước tạo giá trị đại diện của thông điệp và khôi phục thông điệp, như được định nghĩa trong các điều 8, 9 và 10 trong tiêu chuẩn này. Bất kỳ lựa chọn nào trong ba lựa chọn này được chọn, một hàm băm cũng phải được chọn trong các hàm băm đã được tiêu chuẩn hóa theo TCVN 11816 tùy thuộc vào điều kiện rằng mã băm đầu ra sẽ chứa ít nhất 160 bit. Trong hai trong số ba trường hợp, một hàm tạo mặt nạ được yêu cầu thêm, và hàm được sử dụng được định nghĩa trong Phụ lục C trong tiêu chuẩn này.

c) Các bước tạo chữ ký và mở chữ ký được định nghĩa duy nhất trong Phụ lục B trong tiêu chuẩn này, sự lựa chọn của khoá chữ ký bí mật được sử dụng trong quá trình tạo chữ ký, và trong trường hợp Lược đồ chữ ký 2 và 3 với số mũ lẻ, sự lựa chọn giữa chữ ký cơ sở và thay thế và các hàm xác thực. Phương pháp được sử dụng để tạo ra các cặp khóa chữ ký bí mật và khóa xác thực công khai được định nghĩa trong Phụ lục B trong tiêu chuẩn này.

# 8 Lược đồ chữ ký số 1

## 8.1 Tổng quan

Điều 8 xác định quá trình tạo giá trị đại diện của thông điệp và khôi phục thông điệp cho lược đồ chữ ký số tất định cho phép khôi phục thông điệp.

Do các tấn công có thể xảy ra (xem [5] và [6]), lược đồ này sẽ chỉ được sử dụng trong các môi trường nơi các điều kiện về mặt tính toán đảm bảo rằng kẻ tấn công không thể có được chữ ký trên một số lượng lớn các thông điệp đã được lựa chọn.

CHÚ THÍCH Lược đồ chữ ký số 1 chỉ nên sử dụng trong các môi trường đòi hỏi tính tương thích với các hệ thống được cài đặt theo phiên bản đầu tiên của tiêu chuẩn này (xem [5] và [6]). Tuy nhiên, Lược đồ chữ ký số 1 chỉ tương thích với các hệ thống được cài đặt theo phiên bản đầu tiên của tiêu chuẩn này sử dụng mã băm có độ dài ít nhất 160 bit.

### 8.2 Tham số

#### 8.2.1 Độ dài theo mô-đun

Khóa chữ ký bí mật đang sử dụng được giả định để có một mô-đun với độ dài là *k* bit (xem Phụ lục B). Nó xác định cả *c*, năng lực của chữ ký, và độ dài của *F*, giá trị đại diện của thông điệp.

#### 8.2.2 Các tùy chọn trường trailer

Trong lược đồ này, trường trailer (được sử dụng như là một phần cấu tạo nên giá trị đại diện của thông điệp) có thể có độ dài là một hoặc hai xâu bộ tám. Trailer sẽ bao gồm *t* xâu bộ tám ($t = 1$ hoặc 2), trong đó xâu bộ bốn bên phải nhất luôn luôn bằng giá trị 'C' của hệ thập lục phân. Có hai lựa chọn sau đây được cho phép.

– Tùy chọn 1 ($t = 1$): trailer sẽ bao gồm một xâu bộ tám; xâu bộ tám này sẽ bằng giá trị "BC" của hệ thập lục phân.

– Tùy chọn 2 ($t = 2$): trailer sẽ bao gồm hai xâu bộ tám liên tiếp; xâu bộ tám bên phải sẽ bằng giá trị "CC" của hệ thập lục phân và xâu bộ tám bên trái sẽ là định danh hàm băm. Định danh hàm băm xác định hàm băm đang được sử dụng.

Khoảng từ "00" đến "7F" được dành cho tiêu chuẩn ISO / IEC JTC1; TCVN 11816 quy định một định danh duy nhất trong dải đó cho mỗi hàm băm đã được tiêu chuẩn hóa. Khoảng từ "80" đến "FF" được dành riêng cho việc sử dụng độc quyền.

CHÚ THÍCH Việc sử dụng tùy chọn thứ hai cần bên xác thực phải có một phương tiện độc lập an toàn để biết được hàm băm nào được sử dụng để xác minh chữ ký. Mặc dù điều này trước đây được cho là tùy trường hợp, tuy nhiên nó đã được chứng minh là sai, [16] (xem thêm Phụ lục D).

#### 8.2.3 Năng lực

Năng lực *c* của chữ ký trong lược đồ này được xác định bởi

$c = k - L_h - 8t - 4.$

Như đã được định nghĩa trong mục 7.2.2, độ dài $c^*$ của thông điệp có thể khôi phục được phải thỏa mãn:

a) $c^* = |M_1|$ trong trường hợp khôi phục thông điệp hoàn toàn;

b) $c - 7 \leq c^* \leq c$ trong trường hợp khôi phục một phần.

### 8.3 Tạo giá trị đại diện của thông điệp

Trong lược đồ này, việc tạo giá trị đại diện của thông điệp bao gồm hai bước chính:

– Băm thông điệp;

– Định dạng.

#### 8.3.1 Băm thông điệp

Thông điệp *M* (trong đó $M = M_1||M_2$) sẽ là đầu vào của hàm băm *h* để thu được mã băm *H*, có nghĩa là, $H = h(M)$. CHÚ THÍCH rằng *H* chứa $L_h$ bit.

### 8.3.2 Định dạng

Một xâu có độ dài k bit sẽ được xây dựng như sau (thực hiện từ trái sang phải):

- Hai bit được thiết lập bằng "01"
- Một bit được thiết lập bằng '0' trong trường hợp là khôi phục toàn bộ (nghĩa là $M = M_1$) và '1' trong trường hợp là khôi phục một phần (nghĩa là khi $|M_2| > 0$),
- $k - L_h - |M_1| - 8t - 4$ bit đệm tất cả được thiết lập bằng '0',
- Một bit được thiết lập bằng '1' (bit đệm cuối cùng),
- $|M_1|$ bit của $M_1$,
- $L_h$ bit của $H$, mã băm,
- $8t$ bit của trường trailer $T$.

CHÚ THÍCH 1 : Khi khôi phục một phần được cung cấp, $M_2$ được giữ càng ngắn càng tốt để phù hợp với điều kiện rằng nó sẽ là một số nguyên lần của xâu bộ tám, trong trường hợp này số bit đệm bằng '0' sẽ ít hơn 8.

Giá trị đại diện của thông điệp $F$ sẽ là kết quả của việc xử lý xâu bên trên từ trái sang phải trong các khối bốn bit liên tiếp, có nghĩa là, các xâu bộ bốn, theo các bước dưới đây.

1. Xâu bộ bốn bên trái nhất sẽ giữ nguyên không thay đổi.

2. Nếu xâu bộ bốn bên trái nhất có bit bên phải nhất là '0' thì

a) Tất cả xâu bộ bốn tiếp theo bằng "0000", nếu có, sẽ được thay bằng một xâu bộ bốn bằng 'B' trong hệ thập lục phân; nó là một phần của trường đệm.

b) Xâu bộ bốn tiếp theo đầu tiên không bằng "0000" sẽ được cộng XOR với 'B' trong hệ thập lục phân (nghĩa là "1011") ; đây là xâu bộ bốn chứa bit đệm cuối cùng.

3. Tất cả các bit tiếp theo sẽ giữ nguyên không thay đổi.

CHÚ THÍCH 2 Điều này có nghĩa là nếu xâu bộ bốn bên trái nhất có bit bên phải nhất được thiết lập là '1' (và do đó không có '0' bit đệm), thì không có thay đổi được thực hiện đối với xâu bit.

4. Bit đầu tiên của xâu kết quả (sẽ luôn bằng '0') sẽ bị xóa, kết quả là $F$ là một xâu có $k$-1 bit.

### 8.4 Khôi phục thông điệp

Như đã được quy định trong Điều 6, bên xác thực phải biết hàm băm $h$ nào đã được sử dụng trong quá trình ký trước đó. Do đó bên xác thực cũng sẽ biết $L_h$.

Nếu xâu bộ tám bên phải nhất của giá trị đại diện của thông điệp đã được khôi phục $F^*$, một xâu có $k$-1 bit, bằng với

- "BC" của hệ thập lục phân, thì trailer chỉ bao gồm một xâu bộ tám;
- "CC" của hệ thập lục phân, thì trailer bao gồm hai xâu bộ tám bên phải nhất của $F^*$, trong đó xâu bộ tám bên trái là định danh của hàm băm được sử dụng. Nó sẽ được kiểm tra để xác định xem nó có giống với hàm băm được sử dụng bởi bên xác thực hay không; nếu không thống nhất thì việc xác thực chữ ký sẽ bị lỗi.

Chữ ký Σ sẽ bị từ chối nếu hoặc trailer hoặc định danh hàm băm (nếu có) không thể được hiểu. Ngược lại, quá trình xác thực sẽ tiếp tục.

Chữ ký Σ sẽ bị từ chối nếu bit bên trái nhất của giá trị đại diện của thông điệp đã được khôi phục $F^*$ là '0'.

Một bit '0' sẽ được nối liền ở đầu bên trái của xâu (kết quả là một xâu $k$ bit). Xâu này tiếp theo sẽ được xử lý các khối bốn bit liên tiếp từ trái sang phải, nghĩa là các xâu bộ bốn, theo các bước dưới đây.

1. Xâu bộ bốn bên trái nhất sẽ giữ nguyên không thay đổi.

2. Nếu xâu bộ bốn bên trái nhất có bit bên phải nhất là '0' thì

a) Tất cả xâu bộ bốn tiếp theo bằng 'B' của hệ thập lục phân, nếu có, là một phần của trường đệm,

b) Xâu bộ bốn tiếp theo đầu tiên không bằng 'B' của hệ thập lục phân sẽ là sẽ là số đảo ngược thứ tự với 'B' của hệ thập lục phân để lấy lại giá trị ban đầu của xâu bộ bốn này.

3. Tất cả các bit tiếp theo sẽ giữ nguyên không thay đổi.

Vị trí của bit đệm (bên phải nhất) cuối cùng đã được xác định, và do đó đã tính toán được tổng số bit đệm. Bit thứ ba của xâu bộ bốn đầu tiên cũng có thể được xử lý để xác định xem chữ ký cung cấp khôi phục một phần hay toàn bộ. Trong trường hợp khôi phục một phần, chữ ký Σ sẽ bị từ chối nếu có lớn hơn hoặc bằng chín bit đệm (có nghĩa là có lớn hơn hoặc bằng tám bit 0). Ngược lại, quá trình xác thực sẽ tiếp tục.

Tất cả các bit cho đến cuối trường đệm sẽ được loại bỏ từ bên trái của phiên bản đã được chỉnh sửa của $F^*$, và trailer một hoặc hai xâu bộ tám sẽ được loại bỏ từ bên phải. Xâu nhị phân còn lại sẽ được chia làm hai phần.

- Mã băm đã được khôi phục $H^*$ sẽ bao gồm $L_h$ bit bên phải nhất.
- Phần đã được khôi phục của thông điệp $M_1^*$ sẽ bao gồm các bit còn lại bên trái.

Phần thông điệp đã được khôi phục $M_1^*$ sẽ được ghép với $M_2^*$, phần không thể khôi phục được của thông điệp, để đưa vào quá trình xác thực và cho hàm băm. Nếu kết quả là giống như $H^*$, có nghĩa là nếu $H^* = h(M_1^* || M_2^*)$, thì chữ ký được chấp nhận và $M_1^*$ sẽ được trả về; ngược lại chữ ký sẽ bị từ chối.

## 9. Lược đồ chữ ký số 2

### 9.1 Tổng quan

Điều 9 định nghĩa các quá trình tạo giá trị đại diện của thông điệp và khôi phục thông điệp cho lược đồ chữ ký số ngẫu nhiên cho phép khôi phục thông điệp.

CHÚ THÍCH Lược đồ chữ ký số này tương thích với lược đồ đã được biết đến dưới tên gọi là IFSSR được quy định trong IEEE P1363a, [10]. Nó được dựa trên một cách chặt chẽ vào một lược đồ đã được biết đến dưới tên gọi là PSS-R, [3]. Phương thức tạo giá trị đại diện của thông điệp là tương tự như phương thức đã được biết đến dưới tên gọi là EMSR3 trong IEEE P1363a, [10].

### 9.2 Tham số

#### 9.2.1 Độ dài mô-đun

Khóa chữ ký bí mật khi sử dụng được giả sử là có một độ dài mô-đun là *k* bit (xem Phụ lục B). Nó xác định cả *c*, năng lực của chữ ký, và độ dài của *F*, giá trị đại diện của thông điệp.

#### 9.2.2 Độ dài salt

Độ dài salt $L_s$ sẽ được lựa chọn. $L_s$ sẽ là một số nguyên dương ($L_s > 0$) ; là một giá trị điển hình của $L_h$.

#### 9.2.3 Các tùy chọn trường trailer

Trong lược đồ này, trường trailer (được sử dụng như là một phần cấu tạo nên giá trị đại diện của thông điệp) có thể có độ dài là một hoặc hai xâu bộ tám. Trailer sẽ bao gồm *t* xâu bộ tám ($t = 1$ hoặc 2), trong đó xâu bộ bốn bên phải nhất luôn luôn bằng giá trị 'C' của hệ thập lục phân. Có hai lựa chọn sau đây được cho phép.

– Tùy chọn 1 ($t = 1$): trailer sẽ bao gồm một xâu bộ tám; xâu bộ tám này sẽ bằng giá trị "BC" của hệ thập lục phân.

– Tùy chọn 2 ($t = 2$): trailer sẽ bao gồm hai xâu bộ tám liên tiếp; xâu bộ tám bên phải sẽ bằng giá trị "CC" của hệ thập lục phân và xâu bộ tám bên trái sẽ là định danh hàm băm. Định danh hàm băm xác định hàm băm đang được sử dụng.

Khoảng từ "00" đến "7F" được dành cho tiêu chuẩn ISO / IEC JTC1; TCVN 11816 quy định một định danh duy nhất trong dải đó cho mỗi hàm băm đã được tiêu chuẩn hóa. Khoảng từ "80" đến "FF" được dành riêng cho việc sử dụng độc quyền.

#### 9.2.4 Năng lực

Năng lực *c* của chữ ký trong lược đồ này được xác định bởi:

$c = k - L_h - L_s - 8t - 2.$

### 9.3 Tạo giá trị đại diện của thông điệp

Trong lược đồ này, việc tạo giá trị đại diện của thông điệp bao gồm hai bước chính:

– Băm thông điệp;

– Định dạng.

#### 9.3.1 Băm thông điệp

Mã băm *H* sẽ được tính như dưới đây hoặc theo một quy trình gồm các bước tương tự.

Chuyển đổi độ dài bit của $M_1$, hay $|M_1|$ thành một xâu *C* có độ dài 64 bit sử dụng phép chuyển đổi đã được mô tả trong điều 5.

Tạo ra một xâu bit ngẫu nhiên mới *S* có độ dài là $L_s$ bit.

Tính mã băm *H* bằng công thức $H = h(C||M_1||h(M_2)||S)$. CHÚ THÍCH rằng *H* chứa $L_h$ bit.

### 9.3.2 Định dạng

Giá trị đại diện của thông điệp *F* sẽ được tính toán như dưới đây hoặc theo một quy trình gồm các bước tương tự.

1. Cho *P* là một xâu bit chứa $k + \delta - L_h - L_s - |M_1| - 8t - 2$ bit '0' trong đó $\delta = (1 - k) mod\ 8$.

2. Cho xâu bit *D* được xác định bởi $D = P||'1'||M_1||S$, trong đó '1' là một bit. Độ dài của *D* là $k + \delta - L_h - 8t - 1$ bit.

CHÚ THÍCH : Nếu mã băm có độ dài là bội số của xâu bộ tám thì xâu bit *D* cũng sẽ có độ dài là bội số của xâu bộ tám.

3. Áp dụng hàm tạo mặt nạ *g* cho mã băm *H* để tạo ra một xâu bit *N* có độ dài $k + \delta - L_h - 8t - 1$ bit.

4. Độ dài của $D \oplus N$ là $k + \delta - L_h - 8t - 1$ bit. Cho D' là xâu $k - L_h - 8t - 1$ bit thu được bằng cách xóa đi $\delta$ bit ở bên trái nhất của $D \oplus N$ .

5. Cho $F = D'||H||T$, trong đó *T* là trường trailer 8 bit, *F* là xâu có độ dài $k - 1$ bit.

### 9.4 Khôi phục thông điệp

Nếu xâu bộ tám bên phải nhất của giá trị đại diện của thông điệp đã được khôi phục *F**, một xâu có *k*-1 bit, bằng với

- "BC" của hệ thập lục phân, thì trailer chỉ bao gồm một xâu bộ tám;
- "CC" của hệ thập lục phân, thì trailer bao gồm hai xâu bộ tám bên phải nhất của *F**, trong đó xâu bộ tám bên trái nhất là định danh của hàm băm được sử dụng. Nó sẽ được kiểm tra để xác định xem nó có giống với hàm băm được sử dụng bởi bên xác thực hay không; nếu không thống nhất thì việc xác thực chữ ký sẽ bị lỗi.

Chữ ký Σ sẽ bị từ chối nếu hoặc trailer hoặc định danh hàm băm (nếu có) không thể được hiểu. Ngược lại, quá trình xác thực sẽ tiếp tục.

Phần thông điệp có thể khôi phục được $M_1$ sau đó sẽ được khôi phục từ đại diện thông điệp *F** đã được khôi phục và phần không thể khôi phục được $M_2$ như dưới đây hoặc bằng một trình tự các bước tương tự.

1. Thêm $\delta$ bit '0' vào bên trái nhất của *F**.

2. Cho D'* là $k + \delta - L_h - 8t - 1$ bit bên trái nhất của xâu kết quả, và H* là $L_h$ bit tiếp theo.

3. Áp dụng hàm tạo mặt nạ *g* cho xâu *H** để tạo ra một xâu bit *N** có độ dài $k + \delta - L_h - 8t - 1$ bit.

4. Cho $D^* = D'^* \oplus N^*$.

5. Thiết lập $\delta$ bit bên trái nhất của *D** bằng '0'.

6. Thực hiện từ bên trái nhất của *D**, tìm kiếm bit '1' đầu tiên. Loại bỏ bit này và tất cả các bit '0' ở bên trái của nó, sau đó cho *S** là $L_s$ bit bên phải nhất của *D**, và $M_1^*$ là các bit còn lại của *D**. Nếu không có bit '1' đầu tiên thì trả về báo hiệu quá trình xác thực đã bị lỗi và dừng lại.

7. Chuyển đổi độ dài bit của $M_1^*$ thành một xâu *C* có độ dài 64 bit sử dụng phép chuyển đổi đã được mô tả trong điều 5.

8. Nếu $H^* = h(C||M_1{}^*||h(M_2{}^*)||S^*)$ thì trả về phần thông điệp có thể khôi phục được $M_1{}^*$. Ngược lại, trả về báo hiệu quá trình xác thực bị lỗi.

## 10 Lược đồ chữ ký số 3

Điều 10 định nghĩa các quá trình tạo giá trị đại diện của thông điệp và khôi phục thông điệp cho một lược đồ chữ ký số tất định cho phép khôi phục thông điệp.

Lược đồ này giống hệt như lược đồ đã được định nghĩa tại điều 9 ngoại trừ việc $S$ là một giá trị cố định được cho phép có độ dài bằng 0, có nghĩa là $L_s \geq 0$ (không giống như điều kiện $L_s > 0$ được áp dụng trong điều 9). Do đó lược đồ này là tất định và không ngẫu nhiên.

Salt cố định S có thể được người ký lựa chọn. Ngoài ra, nó có thể được quy định như một phần của các tham số miền.

CHÚ THÍCH 1 Độ an toàn của lược đồ này tương đương với mức độ có thể đạt được từ việc sử dụng "băm toàn miền", [1], [4].

CHÚ THÍCH 2 Lược đồ chữ ký số 3 được xem là nên được sử dụng hơn lược đồ chữ ký số 1 - xem điều 1. Đó là do các lý do sau đây.

- Các lược đồ giống lược đồ chữ ký số 3 có các bằng chứng toán học về độ an toàn (xem [4]). Tuy nhiên, các kỹ thuật chứng minh này không áp dụng cho Lược đồ chữ ký số 1.

- Hai lược đồ có hiệu quả tương đương.

# Phụ lục A
(quy định)
# Mô-đun ASN.1

## A.1 Tổng quan

```
MessageRecoverySignatureMechanisms {
   iso(1) standard(0) signature-schemes(9796) part2(2) asn1-module(1)
      message-recovery-signature-mechanisms(0) }
DEFINITIONS EXPLICIT TAGS ::= BEGIN

IMPORTS

   HashFunctions
      FROM DedicatedHashFunctions {
         iso(1) standard(0) hash-functions(10118) part(3)
            asn1-module(1) dedicated-hash-functions(0) } ;


SignatureWithMessageRecovery ::= SEQUENCE {
   algorithm  ALGORITHM.&id({MessageRecovery}),
   parameters ALGORITHM.&Type({MessageRecovery}{@algorithm}) OPTIONAL
}

MessageRecovery ALGORITHM ::= {
   dswmr-mechanism1A      |
   dswmr-mechanism2A      |
   dswmr-mechanism3A      |
   dswmr-mechanism1N      |
   dswmr-mechanism2N      |
   dswmr-mechanism3N      |
   dswmr-mechanism1A-sha1 |
   dswmr-mechanism2A-sha1 |
   dswmr-mechanism3A-sha1 |
   dswmr-mechanism1N-sha1 |
   dswmr-mechanism2N-sha1 |
   dswmr-mechanism3N-sha1,

   ...   -- Expect additional signature scheme objects --
}

dswmr-mechanism1A ALGORITHM ::= {
   OID mechanism1A PARMS HashFunctions
}

dswmr-mechanism2A ALGORITHM ::= {
   OID mechanism2A PARMS HashFunctions
}

dswmr-mechanism3A ALGORITHM ::= {
   OID mechanism3A PARMS HashFunctions
}

dswmr-mechanism1N ALGORITHM ::= {
```

```
}

dswmr-mechanism2N ALGORITHM ::= {
   OID mechanism2N PARMS HashFunctions
}

dswmr-mechanism3N ALGORITHM ::= {
   OID mechanism3N PARMS HashFunctions
}

dswmr-mechanism1A-sha1 ALGORITHM ::= { OID  mechanism1A-sha1 }

dswmr-mechanism2A-sha1 ALGORITHM ::= { OID  mechanism2A-sha1 }

dswmr-mechanism3A-sha1 ALGORITHM ::= { OID  mechanism3A-sha1 }

dswmr-mechanism1N-sha1 ALGORITHM ::= { OID  mechanism1N-sha1 }

dswmr-mechanism2N-sha1 ALGORITHM ::= { OID  mechanism2N-sha1 }

dswmr-mechanism3N-sha1 ALGORITHM ::= { OID  mechanism3N-sha1 }

-- Cryptographic algorithm identification --

ALGORITHM ::= CLASS {
   &id    OBJECT IDENTIFIER  UNIQUE,
   &Type  OPTIONAL
}
  WITH SYNTAX { OID &id [PARMS &Type] }

-- Message recovery signature mechanisms --

OID ::= OBJECT IDENTIFIER  -- Alias

signatureMechanismA OID ::= {
   iso(1)     standard(0)     signature-schemes(9796)     part2(2)     mechanism(0)
alternate(0) }

mechanism1A OID ::= { signatureMechanismA mechanism1(0) }

mechanism2A OID ::= { signatureMechanismA mechanism2(1) }

mechanism3A OID ::= { signatureMechanismA mechanism3(2) }

signatureMechanismN OID ::= {
   iso(1) standard(0) signature-schemes(9796) part2(2) mechanism(0) normal(1) }

mechanism1N OID ::= { signatureMechanismN mechanism1(0) }

mechanism2N OID ::= { signatureMechanismN mechanism2(1) }

mechanism3N OID ::= { signatureMechanismN mechanism3(2) }

-- Combined signature scheme and hash-function mechanisms --

mechanismA-Hash OID ::= {
   iso(1) standard(0) signature-schemes(9796) part2(2)
      mechanismHash(2) alternate(0) }

mechanism1A-sha1 OID ::= { mechanismA-Hash mechanism1-SHA1(0) }
```

```
mechanism2A-sha1 OID ::= { mechanismA-Hash mechanism2-SHA1(1) }

mechanism3A-sha1 OID ::= { mechanismA-Hash mechanism3-SHA1(2) }

mechanismN-Hash OID ::= {
   iso(1) standard(0) signature-schemes(9796) part2(2)
      mechanismHash(2) normal(1) }

mechanism1N-sha1 OID ::= { mechanismN-Hash mechanism1-SHA1(0) }

mechanism2N-sha1 OID ::= { mechanismN-Hash mechanism2-SHA1(1) }

mechanism3N-sha1 OID ::= { mechanismN-Hash mechanism3-SHA1(2) }

END  -- MessageRecoverySignatureMechanisms -
```

### A.2 Sử dụng các định danh đối tượng đi kèm

Mỗi lược đồ chữ ký đều sử dụng một hàm băm, một dãy bao gồm một định danh thuật toán và các tham số liên quan. Do đó, định danh đối tượng của lược đồ chữ ký có thể được đi kèm với một trong các định danh thuật toán hàm băm chuyên dụng được quy định trong ISO/IEC 10118-3 và các tham số liên quan.

Sử dụng ký hiệu giá trị ASN.1 XML, một giá trị có kiểu SignatureWithMessageRecovery sử dụng cơ chế xử lý chữ ký thông thường 1 được định nghĩa trong tiêu chuẩn này và hàm băm SHA-1 được định nghĩa trong ISO/IEC 10118-3 được biểu diễn như sau:

```
<SignatureWithMessageRecovery>
   <algorithm> 1.0.9796.2.0.1.0 </algorithm>
   <parameters>
      <HashFunctions>
         <algorithm> 1.3.14.3.2.26 </algorithm>
         <parameters/>
      </HashFunctions>
   </parameters>
</SignatureWithMessageRecovery>
```

Một giá trị có kiểu SignatureWithMessageRecovery sử dụng định danh đối tượng kết hợp cho cơ chế xử lý chữ ký thông thường 1 và hàm băm SHA-1 có dạng đơn giản như sau:

```
<SignatureWithMessageRecovery>
   <algorithm> 1.0.9796.2.2.1.0 </algorithm>
</SignatureWithMessageRecovery>
```

## Phụ lục B
(quy định)
## Hệ thống khóa công khai cho chữ ký số

Phụ lục này định nghĩa một hệ thống khóa công khai. Hệ thống khóa công khai này bao gồm ba phần chính:

– Tạo khóa, phương pháp tạo một cặp khóa gồm một khóa chữ ký bí mật và một khóa xác thực công khai,

– Tạo chữ ký, phương pháp tạo một chữ ký Σ từ giá trị đại diện của thông điệp *F* và một khóa chữ ký bí mật, và

– Mở chữ ký, phương pháp thu được giá trị đại diện của thông điệp đã được khôi phục *F** từ chữ ký Σ và khóa xác thực công khai. Đầu ra của hàm này cũng chứa một báo hiệu xác định xem thủ tục mở chữ ký đã thành công hay bị lỗi.

### B.1 Thuật ngữ và định nghĩa

Trong phụ lục này áp dụng các thuật ngữ và định nghĩa trong tiêu chuẩn này và dưới đây:

**B.1.1**

**Mô-đun**

Số nguyên kết quả của việc tạo ra hai số nguyên tố, là thành phần cấu tạo nên các khóa công khai và khóa bí mật

**B.1.2**

**Khóa chữ ký bí mật**

Mô-đun và số mũ chữ ký bí mật

**B.1.3**

**Khóa xác thực công khai**

Mô-đun và số mũ xác thực công khai

### B.2 Ký hiệu và chữ viết tắt

Trong phụ lục này áp dụng các ký hiệu và chữ viết tắt được quy định trong tiêu chuẩn này và dưới đây:

| | |
|---|---|
| *f* | Số nguyên với F là biểu diễn nhị phân của nó. |
| *f** | Số nguyên được tính toán trong quá tình mở chữ ký. |
| *J* | Số nguyên được tính toán trong quá trình tạo chữ ký. |
| *J** | Số nguyên được tính toán trong quá trình mở chữ ký. |
| *n* | Mô-đun (thành phần của khóa chữ ký bí mật và khóa xác thực công khai). |

| | |
|---|---|
| *p, q* | Các thừa số nguyên tố của mô-đun. |
| *s* | Số mũ chữ ký. |
| *v* | Số mũ xác thực |
| *lcm(a,b)* | Bội số chung nhỏ nhất của số nguyên *a* và *b*. |
| *min{a,b}* | Giá trị nhỏ hơn trong hai giá trị *a* và *b*. |
| *(a\|n)* | Ký hiệu Jacobi của *a* theo *n*. |

CHÚ THÍCH 1 Cho *p* là một số nguyên tố lẻ và *a* là một số nguyên dương. Biểu thức sau định nghĩa ký hiệu Legendre theo *p*.

$(a|p) = a^{(p-1)/2} mod\ p$.

Ký hiệu Legendre của bội số của *p* với *p* bằng 0. Khi *a* không phải là bội số của *p*, ký hiệu Legendre của *a* theo *p* là +1 hoặc -1 tùy thuộc vào *a* là hay không là bình phương mô-đun của *p*.

CHÚ THÍCH 2 Cho *n* là một số nguyên dương lẻ và *a* là một số nguyên lẻ. Ký hiệu Jacobi của *a* theo *n* là kết quả của ký hiệu Legendre của *a* theo các phần tử nguyên tố của *n* (lặp lại ký hiệu Jacobi cho các phần tử nguyên tố đã lặp lại). Do đó nếu $n = pq$, thì $(a|n) = (a|p)(a|q)$. Ký hiệu Jacobi của *a* theo *n* vẫn có thể tính được mà không cần biết các phần tử nguyên tố của *n*.

### B.3 Tạo khóa

CHÚ THÍCH Không có một phương pháp về xác thực khóa công khai nào được quy định trong tài liệu này, nó đảm bảo cho một tổ chức (là tổ chức tạo cặp khóa, tổ chức sử dụng khóa công khai hoặc một tổ chức thứ ba trung lập) rằng một khóa công khai nào đó có tuân theo định nghĩa số học của khóa công khai hay không. Một khóa công khai không hợp lệ có thể tồn tại do lỗi tính toán do sơ ý khi tạo khóa hoặc hành động có chủ ý của đối phương. Việc sử dụng một khóa công khai không hợp lệ sẽ chống lại tất cả mọi đảm bảo an toàn, bao gồm khả năng đối phương không thể giả mạo một chữ ký hoặc khám phá ra khóa bí mật liên quan. Người sử dụng mong muốn được đảm bảo tính hợp lệ về mặt số học của khóa công khai trước khi sử dụng nó nên sử dụng các phương pháp khác, ví dụ như các phương pháp trong ISO/IEC 9796-3. Là nguyên lý chung của tất cả các hệ thống mật mã, việc sử dụng một khóa công khai hợp lệ nhưng được tạo ra một cách không chính xác (ví dụ như được tạo ra từ một nguồn không đủ ngẫu nhiên), hoặc một khóa bí mật được bảo vệ một cách không chính xác cũng có thể chống lại tất cả mọi đảm bảo an toàn. Kiểm tra tính hợp lệ của các cài đặt có thể giảm thiểu các rủi ro này cũng như khả năng các khóa không hợp lệ được sử dụng. Tuy nhiên, nó không cung cấp sự đảm bảo cụ thể rằng khóa công khai cho trước đó có hợp lệ trên thực tế hay không.

#### B.3.1 Số mũ xác thực công khai

Mỗi thực thể ký sẽ lựa chọn một số nguyên dương v làm số mũ xác thực công khai. Số mũ xác thực công khai này có thể được tiêu chuẩn hóa trong các ứng dụng cụ thể.

CHÚ THÍCH Các giá trị 2, 3, 17 và 65537 có thể có một số ưu điểm về mặt thực tế.

#### B.3.2 Các thừa số nguyên tố bí mật và số mô đun công khai

Mỗi thực thể ký sẽ lựa chọn một cách bí mật và ngẫu nhiên hai số nguyên tố lớn khác nhau *p* và *q* theo các điều kiện sau đây.

– Nếu *v* là số lẻ, thì $p - 1$ và $q - 1$ sẽ là nguyên tố cùng nhau với *v*.

– Nếu *v* là số chẵn, thì $(p-1)/2$ và $(q-1)/2$ sẽ là nguyên tố cùng nhau với *v*. Hơn nữa, *p* và *q* phải không đồng dư với nhau theo mod 8.

Số đồng dư công khai *n* được thiết lập bằng tích của *p* và *q*, có nghĩa là $n = pq$. Kích thước của *n* theo bit xác định giá trị của *k* như sau

$2^{k-1} < n \leq 2^k - 1.$

CHÚ THÍCH 1 Một số điều kiện bổ sung cho việc lựa chọn số nguyên tố có thể để ngăn chặn việc bị phân tích của mô-đun.

CHÚ THÍCH 2 Một số định dạng của số đồng dư đơn giản hóa việc giảm thiểu mô-đun và yêu cầu ít lưu trữ bảng hơn. Ví dụ của các định dạng này là

$n = 2^{64x} - r$ của độ dài $k = 64x$ bit,

$n = 2^{64x} + r$ của độ dài $k = 64x + 1$ bit,

trong đó: $1 \leq y \leq 2x$ và $r < 2^{64x-8y} < 2r$.

Đối với mô-đun có định dạng $2^{64x} - r$, *8y* bit quan trọng nhất bằng 1, trong đó 8y nhiều nhất bằng một phần tư độ dài số đồng dư. Đối với mô-đun có định dạng $2^{64x} + r$, bit quan trọng nhất bằng 1 và theo sau nó là *8y* bit bằng 0, trong đó *8y* nhiều nhất bằng một phần tư độ dài số đồng dư.

### B.3.3 Số mũ chữ ký bí mật

Số mũ chữ ký bí mật sẽ là bất kỳ một số nguyên dương *s* nào sao cho $sv - 1$ là bội số của

– $lcm(p-1, q-1)$ nếu *v* là số lẻ;

– $lcm(p-1, q-1)/2$ nếu *v* là số chẵn.

CHÚ THÍCH Nhìn chung, s là giá trị nhỏ nhất có thể.

### B.4 Hàm tạo chữ ký

Giá trị đại diện của thông điệp *F* là một xâu $k-1$ bit, trong đó bốn bit bên phải nhất bằng "1100" ('C' trong hệ thập lục phân). Nó là biểu diễn nhị phân của một số nguyên dương được ký hiệu là *f*.

Số nguyên *J* được định nghĩa như sau:

– nếu v là số lẻ thì $J = f$,

– nếu v là số chẵn và $(f|n) = +1$ thì $J = f$, và

– nếu v là số chẵn và $(f|n) = -1$ thì $J = f/2$.

CHÚ THÍCH Nếu v là số lẻ thì biểu thức trên đảm bảo rằng ký hiệu Jacobi của *J* theo *n* luôn bằng +1.

Chữ ký Σ là một xâu bi có độ dài $k-1$ bit tương ứng với số nguyên $min\{J^s mod\ n, n - (J^s mod\ n)\}$ sử dụng quy ước đã được mô tả trong Điều 5.

### B.5 Hàm mở chữ ký

Chữ ký Σ là một xâu $k-1$ bit, đó là biểu diễn nhị phân của số nguyên dương nhỏ hơn *n*. Số nguyên này sẽ tăng lên theo số mũ *v* mod *n* để thu được *J**, có nghĩa là

$J^* = \Sigma^v mod\ n.$

Số nguyên *f** sẽ được tính như sau.

– Nếu *v* là số lẻ và

– nếu $J^* mod\ 16 = 12$ thì $f^* = J^*$,

– nếu $J^* mod\ 16 = n - 12\ mod\ 16$ thì $f^* = n - J^*$.

- Nếu *v* là số chẵn và
- nếu $J^* \, mod \, 8 \, = \, 1$ thì $f^* = n - J^*$,
- nếu $J^* \, mod \, 8 \, = \, 4$ thì $f^* = J^*$,
- nếu $J^* \, mod \, 8 \, = \, 6$ thì $f^* = 2J^*$,
- nếu $J^* \, mod \, 8 \, = \, 7$ thì $f^* = 2(n - J^*)$.

Chữ ký Σ sẽ bị từ chối trong tất cả các trường hợp khác; nó cũng sẽ bị từ chối nếu $f^* \, mod \, 16 \neq 12$, và nếu *f** không thỏa mãn $f^* \leq 2^{k-1} - 1$ .

Giá trị đại diện của thông điệp đã được khôi phục *F** là xâu bit có độ dài $k - 1$ tương ứng với số nguyên *f** sử dụng quy ước đã được mô tả trong Điều 5.

### B.6 Hàm tạo chữ ký thay thế

Nếu *v* là số lẻ thì hàm này có thể được sử dụng như là một sự thay thế cho hàm trong Điều B.4. Nó sẽ được sử dụng cùng với hàm mở chữ ký trong B.7.

Giá trị đại diện của thông điệp F là một xâu $k - 1$ bit, trong đó bốn bit bên phải nhất bằng "1100" ('C' trong hệ thập lục phân). Nó là biểu diễn nhị phân của số nguyên dương được ký hiệu là *f*.

Số nguyên *J* được định nghĩa là $J = f$.

Chữ ký Σ là một xâu bit có độ dài *k* bit tương ứng với số nguyên $J^s mod \, n$ sử dụng quy ước đã được mô tả trong Điều 5.

CHÚ THÍCH Sự khác nhau giữa hàm này và hàm ở Điều B.4 là chữ ký Σ luôn luôn là $J^s mod \, n$; không có bước "giá trị tuyệt đối" được thực hiện để lựa chọn giá trị nhỏ hơn giữa $J^s mod \, n$ và $n - (J^s mod \, n)$.

### B.7 Hàm mở chữ ký thay thế

Nếu *v* là số lẻ thì hàm này có thể được sử dụng như là một sự thay thế cho hàm trong Điều B.5. Nó sẽ được sử dụng cùng với hàm tạo chữ ký trong B.6.

Chữ ký Σ là một xâu *k* bit, đó là biểu diễn nhị phân của số nguyên dương nhỏ hơn *n*. Số nguyên này sẽ tăng lên theo số mũ *v* mod *n* để thu được *J**, có nghĩa là

$J^* = \Sigma^v mod \, n.$

Số nguyên *f** sẽ được tính bằng $f^* = J^*$.

Chữ ký Σ sẽ bị từ chối nếu $f^* \, mod \, 16 \neq 12$, và nếu *f** không thỏa mãn $f^* \leq 2^{k-1} - 1$ .

Giá trị đại diện của thông điệp đã được khôi phục *F** là xâu bit có độ dài $k - 1$ tương ứng với số nguyên *f** sử dụng quy ước đã được mô tả trong Điều 5.

CHÚ THÍCH Sự khác nhau giữa hàm này và hàm ở Điều B.5 là số nguyên *f** luôn bằng *J**; không cần thiết phải "disambiguation" giữa $J^*$ và $n - J^*$.

# Phụ lục C
(quy định)
## Hàm tạo mặt nạ

Phụ lục này định nghĩa một hàm tạo mặt nạ dựa trên hàm băm.

CHÚ THÍCH Hàm này là mở rộng của hàm đã được định nghĩa dưới tên gọi là MFG1 trong IEEE Std 1363-2000, cho phép đầu vào đầu ra là các xâu bit. Nó tương tự với đề xuất của Bellare và Rogaway.

Một hàm tạo mặt nạ lấy đầu vào là một xâu bit $Z$ và độ dài mong đợi của đầu ra $L_N$, và đầu ra là một xâu bit $N$ có độ dài bằng giá trị đó.

### C.1 Ký hiệu và chữ viết tắt

Trong phụ lục này cùng với các ký hiệu đã được định nghĩa trong Điều 4, các ký hiệu và chữ viết tắt dưới đây được áp dụng:

| | |
|---|---|
| $L_N$ | Độ dài (tính bằng bit) của đầu ra của hàm tạo mặt nạ $g$. |
| $L_Z$ | Độ dài (tính bằng bit) của xâu bộ tám $Z$. |
| $N$ | Đầu ra của hàm tạo mặt nạ g (một xâu bit). |
| $Z$ | Một xâu bit đầu vào của hàm tạo mặt nạ. |

### C.2 Yêu cầu

Việc sử dụng hàm này đòi hỏi phải lựa chọn một hàm băm. Hàm băm này, được ký hiệu là $h$, sẽ được thiết lập bằng hàm băm h như trong đoạn (c) của Điều 6. Chúng ta ký hiệu độ dài đầu ra của h theo bit là $L_h$.

### C.3 Chi tiết kỹ thuật

#### C.3.1 Tham số

Một đầu vào của hàm $g$ là độ dài mong muốn theo bit của đầu ra, là số nguyên dương $L_N$.

#### C.3.2 Tạo mặt nạ

Xâu bit $N$ sẽ được tính toán như sau hoặc theo một trình tự các bước tương tự.

1. Nếu $L_z$ vượt quá giới hạn về độ dài ($2^{64} - 33$ đối với Hàm băm chuyên dụng 1 và 3 trong ISO/IEC 10118-3), hoặc nếu $L_N > L_h \times 2^{32}$, xuất ra báo "lỗi" và dừng lại.
2. Cho $N$ là một xâu rỗng.
3. Cho $i = 0$.

3.1 Chuyển đổi $i$ thành một xâu bit $C$ có độ dài 32 bit sử dụng phép chuyển đổi đã được mô tả trong Điều 5.

3.2 Cho $N := N||h(Z||C)$.

3.3 Cho $i := i + 1$.

3.4 Nếu $i < \lceil L_N/L_h \rceil$, quay lại bước 3.1.

4. Xuất ra $L_N$ bit bên trái nhất của *N*.

## Phụ lục D
(tham khảo)
## Về định danh hàm băm và sự lựa chọn độ dài có thể khôi phục được của thông điệp

Như được quy định trong Điều 6 (Yêu cầu), người dùng các lược đồ chữ ký được quy định trong tiêu chuẩn này phải lựa chọn một hàm băm kháng va chạm *h*. Điều quan trọng là bên xác thực có nhiều cách để xác định hàm băm nào đã được sử dụng khi tạo ra chữ ký, để quá trình xác thực có thể diễn ra một cách an toàn. Nếu có một tổ chức thứ ba nguy hiểm có thể thuyết phục bên xác thực rằng một hàm băm "yếu" đã được sử dụng để tạo ra chữ ký (ví dụ như một hàm băm thiếu tính chất một chiều) thì tổ chức thứ ba này có thể thuyết phục bên xác thực rằng một chữ ký có hiệu lực thực sự đã được áp dụng cho một thông điệp "sai".

Ba lược đồ chữ ký số được quy định trong tiêu chuẩn này cho phép một định danh hàm băm chứa trong giá trị đại diện của thông điệp *F* (xem 8.2.2). Nếu định danh hàm băm đó chứa trong *F* theo cách này thì kẻ tấn công không thể sử dụng lại một cách gian lận một chữ ký đã tồn tại với cùng $M_1$ và khác $M_2$, thậm chí ngay cả khi bên xác thực có thể bị thuyết phục chấp nhận các chữ ký đã được tạo ra bằng một hàm băm đủ yếu mà tiền ảnh có thể được tìm thấy. Điều này được cho là có thể giải quyết được vấn đề đã được đề cập đến trong đoạn trước.

Tuy nhiên, như đã thảo luận chi tiết trong [16], ngay cả khi một định danh hàm băm được bao gồm trong giá trị đại diện của thông điệp, kẻ tấn công khác đều có thể xảy ra nếu bên xác thực có thể bị thuyết phục bằng một hàm băm "yếu" đã được sử dụng. Từ yếu ở đây có nghĩa là hàm băm thiếu tính chất một chiều, có nghĩa là với một hàm băm cho trước có thể tính toán để tìm ra xâu đầu vào ánh xạ đến mã băm này bởi hàm băm. (CHÚ THÍCH rằng chính xác thì kiểu yếu này trước tiên phải được thúc đẩy bởi sự bao gồm của một định danh hàm băm trong giá trị đại diện của thông điệp.

Các tấn công được mô tả trong [16] hoạt động theo cách thức chung sau đây. Kẻ tấn công tạo ra "chữ ký" ngẫu nhiên và đối với mỗi "chữ ký" này áp dụng một hàm xác thực công khai của thực thể mà chữ ký của nó muốn giả mạo, và thu được "giá trị đại diện của thông điệp đã được khôi phục" (đây là bước "mở chữ ký"). Phần tiếp theo của tấn công sẽ rất khác nhau tùy thuộc vào định dạng của giá trị đại diện của thông điệp, nhưng bắt buộc kẻ tấn công phải xem xét xem giá trị đại diện của thông điệp đã được khôi phục có định dạng đúng tương ứng với chữ ký thật và rằng định danh hàm băm trong xâu này là định danh tương ứng với một hàm băm yếu hay không. Khả năng xảy ra việc này là rất khác nhau, nhưng có thể lớn đến $2^{-16}$ (và do đó kẻ tấn công không cần phải thử quá nhiều "chữ ký ngẫu nhiên" trước khi tìm được một chữ ký có các thuộc tính mong muốn).

Với "chữ ký" này, kẻ tấn công có thể nhúng mã băm vào trong giá trị đại diện của thông điệp đã được khôi phục, và lợi dụng thực tế là hàm băm yếu để phát hiện ra một phần thông điệp không thể khôi phục được, mà khi kết hợp với phần có thể khôi phục được có trong giá trị đại diện của thông điệp, băm để có được mã băm mong muốn. Như thế, kẻ tấn công có thể giả mạo một chữ ký mới với một $M_1$ "ngẫu nhiên". Do đó, kể cả khi có định danh hàm băm trong giá trị đại diện của thông điệp cũng không tránh được việc đòi hỏi người xác thức phải có một phương thức độc lập an toàn để biết được hàm băm nào được sử dụng để xác thực chữ ký.

Thảo luận này cũng liên quan đến việc lựa chọn độ dài có thể khôi phục được $c^*$ cho lược đồ chữ ký 2 và 3. Như đã được mô tả trong 7.2.2, $c^*$ sẽ được lựa chọn thỏa mãn điều kiện $c^* \leq c$, năng lực của lược đồ chữ ký. $c^*$ thường được mong đợi là gần bằng $c$ để tối đa độ dài phần có thể khôi phục được của thông điệp, và do đó tối thiểu độ dài phần không thể khôi phục được của thông điệp. $c^*$ được khuyến nghị lựa chọn là một số bất kỳ nhỏ hơn $c$ (ví dụ như $c$-16, $c$-24 hoặc $c$-80, tùy theo độ khó mong muốn), để làm cho các cuộc tấn công theo hình thức đã được mô tả ở trên trở nên khó khăn hơn.

# Phụ lục E
(tham khảo)
## Ví dụ

Phụ lục này bao gồm tổng cộng 12 ví dụ về tạo chữ ký và xác thực chữ ký làm việc theo ba lược đồ đã được quy định trong tiêu chuẩn này, cùng với hai ví dụ về tạo khóa.

Phụ lục E.1 bao gồm các ví dụ với số mũ công khai bằng 3.

– E.1.1 bao gồm một ví dụ về tạo khóa.
– E.1.2 bao gồm ba ví dụ về tạo và xác thực chữ ký, tất cả đều là khôi phục toàn bộ thông điệp. Đối với mỗi lược đồ được quy định trong tiêu chuẩn này có một ví dụ tương ứng.
– E.1.3 bao gồm ba ví dụ về tạo và xác thực chữ ký, tất cả đều là khôi phục một phần thông điệp. Đối với mỗi lược đồ được quy định trong tiêu chuẩn này có một ví dụ tương ứng.

Phụ lục E.2 bao gồm các ví dụ với số mũ công khai bằng 2.

– E.2.1 bao gồm một ví dụ về tạo khóa.
– E.2.2 bao gồm ba ví dụ về tạo và xác thực chữ ký, tất cả đều là khôi phục toàn bộ thông điệp. Đối với mỗi lược đồ được quy định trong tiêu chuẩn này có một ví dụ tương ứng.
– E.2.3 bao gồm ba ví dụ về tạo và xác thực chữ ký, tất cả đều là khôi phục một phần thông điệp. Đối với mỗi lược đồ được quy định trong tiêu chuẩn này có một ví dụ tương ứng

### E.1 Các ví dụ với số mũ công khai bằng 3

Phụ lục E.1 bao gồm các ví dụ với khóa công khai có số mũ bằng 3.

#### E.1.1 Ví dụ vê quá trình tạo khóa

Khóa trong ví dụ có mô-đun $k = 1024$ bit với số mũ công khai $v = 3$.

```
p =   FB961451 995C82F9 527CAAAF B3FB4254 6D00A01D 8B2BDE3D 2E7B8F7D 0C9E781E
      B7FABFC8 E86E9F6D ACE3435A 9D043A99 93F3E473 D93FA888 D3577906 77A94931
q =   FF0EAFCA 70585166 A8CD8E90 36E75290 2F32B863 068016B6 A89F2EA3 418882EF
      6F570122 F92D2E9B EFFF7329 1818F251 BF095D6F 208F93CD CEF4767A 568AB241
```

Số đồng dư công khai *n* là kết quả của các thừa số nguyên tố bí mật *p* và *q*. Độ dài của nó là 1024 bit.

```
n =   FAA8ED34 EEF1CE38 D29814B6 EEAA154D C060BB37 EB1A51E8 AB0398DD ADDFD334
      CB9BE20C 087B1DDF 1F78A397 62B5F20A 7A730086 30913CD2 EE60183D E249DD16
      9CA4EB3A E0420E51 13D73050 4A73A926 BEFBFF32 C89858DE 5F5B3899 FEC52521
      04933163 625F2963 5AB8FAA7 AA14C4F3 C0DD2470 DEFCEB39 2429110A 0149A771
```

Số mũ của chữ ký bí mật s bằng nghịch đảo phép nhân của *v* mod $lcm(p-1, q-1)$.

```
s =   0A71B48C DF4A1342 5E1BAB87 9F471638 92AEB277 A9CBC369 B1CAD109 3C93FE22
      33267EC0 805A7693 F6A506D0 F9723F6B 1A6F755A ECB0B7DE 1F440102 94186936
      316AAC4B F39B37BF 6105DFA0 AEA60BB2 C17306F2 179F2ED4 704D5A6F BCB141C0
      C9380F5A 500823CE 67E8ED81 7FBA5100 59E9541B 498C91F4 1ABE8C10 6220E72B
```

#### E.1.2 Các ví dụ về khôi phục toàn bộ

Ở đây trình bày ba ví dụ về tạo và xác thực chữ ký, mỗi ví dụ tương ứng với một trong ba lược đồ.

### E.1.2.1 Ví dụ về lược đồ chữ ký 1

Ví dụ này sử dụng hàm băm chuyên dụng 3 trong ISO/IEC 10118-3 (còn được biết đến với tên gọi là SHA-1).

#### E.1.2.1.1 Quá trình ký

Thông điệp được ký là một xâu bao gồm 64 ký tự mã ASCII như sau.

abcdbcdecdefdefgefghfghighijhijkijkljklmklmnlmnomnopnopqopqrpqrs

Trong hệ thập lục phân, thông điệp $M$ là một xâu có độ dài là 64 xâu bộ tám, nghĩa là 512 bit như sau.

$M$ = 61626364 62636465 63646566 64656667 65666768 66676869 6768696A 68696A6B
696A6B6C 6A6B6C6D 6B6C6D6E 6C6D6E6F 6D6E6F70 6E6F7071 6F707172 70717273

160 bit mã băm được tính toán bằng cách áp dụng SHA-1 cho 512 bit của $M$.

$H$ = 79EA0C76 F0056373 FFD6A5AA D389DD90 8B0C0E94

Định danh trong trường trailer xác định hàm băm được sử dụng; ISO/IEC 10118-3 thiết lập định danh cho hàm băm chuyên dụng 3 giá trị "33". Do đó, trường trailer $T$ bao gồm 16 bit sau đây.

$T$ = 33CC

Thông điệp là đủ ngắn để khôi phục toàn bộ. 1024 bit của xâu trung gian $S_i$ kết quả của việc nối hai bit của tiêu đề bằng "01", bit dữ liệu thêm được thiết lập bằng '0', 332 (= 1024 – 512 – 160 – 16 – 4) bit đệm bằng '0', bit bao quanh bằng 1, 512 bit của $M_1$ (=$M$), 160 bit của $H$ và 16 bit của trường trailer $T$. Xâu $S_r$ có thể khôi phục kết quả của việc thay thế 82 xâu bộ bốn đệm bằng '0' bằng 82 xâu bộ bốn đệm bằng 'B' và tương tự đối với xâu bộ bốn bao quanh bằng '1' thay thế bằng 'A'.

$S_r$ = 4BBBBBBB BBBBBBBB BBBBBBBB BBBBBBBB BBBBBBBB BBBBBBBB BBBBBBBB
BBBBBBBB
BBBBBBBB BBBBBBBB BBBA6162 63646263 64656364 65666465 66676566 67686667
68696768 696A6869 6A6B696A 6B6C6A6B 6C6D6B6C 6D6E6C6D 6E6F6D6E 6F706E6F
70716F70 71727071 727379EA 0C76F005 6373FFD6 A5AAD389 DD908B0C 0E9433CC

Số nguyên có thể khôi phục $f_r$ là số nguyên dương không dấu biểu diễn $S_r$. $f_r$ tăng theo lũy thừa bậc $s$ theo mô-đun $n$. Kết quả được biểu diễn bởi một số nguyên dương không dấu tạm thời $t$.

$t$ = D6369220 6E1FE0A5 7DF603C1 E5EE6025 B4EF2E69 3E8C3C9E BA00057B 40860A35
FCA66D88 33795AC1 91191515 FE852CAD C80F315C 86142051 ED322775 9F307934
421D615F 39792C40 1319F233 CFFD18D0 12D17A02 442E5BBF B17DCFC5 654BEF59
5F500A15 365CD5D0 BD27948E C938F7C3 BA775982 472E8921 7424A74B 868B63A8

Vì kết quả trên lớn hơn $n/2$, chữ ký $\Sigma = n - t$.

$\Sigma$ = 24725B14 80D1ED93 54A210F5 08BBB528 0B718CCE AC8E1549 F1039362 6D59C8FE
CEF57483 D501C31D 8E5F8E81 6430C55C B263CF29 AA7D1C81 012DF0C8 431963E2
5A8789DB A6C8E211 00BD3E1C 7A769056 AC2A8530 8469FD1E ACDD68D4 997935C7
A543274E 2C025392 9D916618 E0DBCD30 0665CAEE 97CE6217 B00469BE 7ABE43C9

Thông điệp đã ký có 128 xâu bộ tám chỉ bao gồm chữ ký vì $M_2$ là rỗng.

#### E.1.2.1.2 Quá trình xác thực

Chữ ký $\Sigma$ là một xâu nhị phân đại diện một số nguyên dương không dấu, nhỏ hơn $n/2$. Số nguyên này tăng theo lũy thừa bậc 3 mô-đun $n$, do đó thu được số nguyên $f_s$.

$f_s$ = AEED3179 3336127D 16DC58FB 32EE5992 04A4FF7C 2F5E962C EF47DD21 F2241779
0FE02650 4CBF6223 63BE4234 FF518FA7 160D9D21 CB2AD86D 87F8B2D7 7AE176AF
343B83D2 76D7A5E7 A96BC6E5 D9073EB3 528E93C6 5B29EC70 EFEBCB2B 8F54B6B1
9421C1F2 F0ECB8F1 E84580BD 9D9DD4EE 5D69249A 395217AF 469885FD F2B573A5

Vì $f_s$ là đồng dư với $(n - 12)$ mô-đun 16, nó được thay thế bởi phần dư của nó với $n$, có nghĩa là số nguyên được khôi phục là $f'_r = n - f_s$.

$f'_r$ = 4BBBBBBB BBBBBBBB BBBBBBBB BBBBBBBB BBBBBBBB BBBBBBBB BBBBBBBB BBBBBBBB
BBBBBBBB BBBBBBBB BBBA6162 63646263 64656364 65666465 66676566 67686667
68696768 696A6869 6A6B696A 6B6C6A6B 6C6D6B6C 6D6E6C6D 6E6F6D6E 6F706E6F
70716F70 71727071 727379EA 0C76F005 6373FFD6 A5AAD389 DD908B0C 0E9433CC

$f'_r$ được biểu diễn dưới dạng một số nguyên dương không dấu bởi xâu đã được khôi phục $S'_r$.

– Xâu bộ tám bên trái nhất của $S'_r$ bằng "4B"; nó bao gồm tiêu đề bằng "01", bit dữ liệu thêm bằng '0' (khôi phục toàn bộ), một bit đệm bằng '0' và một xâu bộ bốn đệm bằng 'B'; theo sau là 81 xâu bộ bốn bằng 'B' và xâu bộ bốn bao quanh bằng 'A'; 42 xâu bộ tám này được chuyển sang bên phải của $S'_r$.

– Xâu bộ tám bên phải nhất của $S'_r$ bằng "CC"; do đó, trailer bao gồm hai xâu bộ tám và bằng "33CC"; hai xâu bộ tám đó cũng được chuyển sang bên phải của $S'_r$.

Định danh hàm băm bằng "33"; do đó hàm băm được sử dụng là hàm băm chuyên dụng 3.

Xâu còn lại 672 bit được chia làm hai phần.

– $M_1^*$ bao gồm 512 bit bên trái nhất.
– $H'$ bao gồm 160 bit bên phải nhất.

$M_1^*$ = 61626364 62636465 63646566 64656667 65666768 66676869 6768696A 68696A6B
696A6B6C 6A6B6C6D 6B6C6D6E 6C6D6E6F 6D6E6F70 6E6F7071 6F707172 70717273

$H'$ = 79EA0C76 F0056373 FFD6A5AA D389DD90 8B0C0E94

Thông điệp đã được khôi phục $M^*$ chỉ bao gồm $M_1^*$ vì khôi phục thông điệp là toàn bộ. Mã băm còn lại $H''$ được tính bằng các áp dụng SHA-1 cho $M^*$.

$H''$ = 79EA0C76 F0056373 FFD6A5AA D389DD90 8B0C0E94

Vì hai mã băm $H'$ và $H''$ là giống nhau, chữ ký $\Sigma$ được chấp nhận.

### E.1.2 Ví dụ về lược đồ chữ ký 2

Ví dụ này sử dụng hàm băm chuyên dụng 1 trong ISO/IEC 10118-3 (còn được biết đến với tên gọi là RIPEMD-160).

#### E.1.2.1 Quá trình ký

Trong hệ thập lục phân, thông điệp $M$ là một xâu có độ dài là 48 xâu bộ tám, nghĩa là 384 bit như sau.

*M* = FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210

160 bit của salt *S* được tạo ra.

*S* = 436BCA99 54EC376C 96B79C95 D4B82686 F3494AD3

160 bit mã băm được tính toán bằng cách áp dụng hàm băm chuyên dụng 1 để thu được một xâu nhị phân có độ dài là 768 (= 64 + 384 + 160 + 160) và kết quả được ghép thêm 64 bit độ dài thông điệp có thể khôi phục được *C*, 384 bit của phần có thể khôi phục được $M_1$ (=*M*), 160 bit của mã băm của phần không thể khôi phục được (phần này bằng rỗng) $h(M_2)$ và 160 bit của salt *S*. $H = h(C||M_1||h(M_2)||S)$.

*H* = 503E9461 4DA4AF5F 8E78C269 E0DFA03E 027CE74F

Định danh trong trường trailer xác định hàm băm được sử dụng; ISO/IEC 10118-3 thiết lập định danh cho hàm băm chuyên dụng 1 giá trị "31". Do đó, trường trailer *T* bao gồm 16 bit sau đây.

*T* = 31CC

Thông điệp là đủ ngắn để khôi phục toàn bộ. 1024 bit của xâu trung gian $S_i$ kết quả của việc nối 303 (= $1024 - 384 - 160 - 160 - 16 - 1$) bit đệm bằng '0', bit bao quanh bằng 1, 384 bit của $M_1$ (=*M*), 160 bit của *L*, 160 bit của *H* và 16 bit của trường trailer *T*.

$S_i$ = 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 0001FEDC BA987654 3210FEDC BA987654 3210FEDC BA987654 3210FEDC
BA987654 3210FEDC BA987654 3210FEDC BA987654 3210436B CA9954EC 376C96B7
9C95D4B8 2686F349 4AD350BE 94614DA4 AF5F8E78 C269E0DF A03E027C E74F31CC

Xâu có thể khôi phục $S_r$ thu được từ việc áp dụng hàm tạo mặt nạ *MGF1* đối với 848 (= 1024 – 160 – 16) bit bên trái nhất của $S_i$, và 1 bit bên trái nhất của $S_r$ được thiết lập bằng '0' vì δ =1 (δ= $1 - 1024) mod 8$).

$S_r$ = 7BB5D930 4572EE04 BECAE622 6939DC6D A6F19867 8B339668 B09581DA 7DC69063
CFE49956 108754DD BC3AF3FF A6F562C3 6C91DAB4 BFF8CE66 29AC5B1B 6C1B524A
49B7669B 549E679C ABDAD642 A565394D 7373C4C9 4ECADF09 08A5C00D 0511B9F6
D79039FC 7F4BD793 420A50BE 94614DA4 AF5F8E78 C269E0DF A03E027C E74F31CC

Số nguyên có thể khôi phục $f_r$ là số nguyên dương không dấu biểu diễn $S_r$. $f_r$ tăng theo lũy thừa bậc *s* theo mô-đun *n*. Kết quả được biểu diễn bởi một số nguyên dương không dấu tạm thời *t*.

*t* = A4958BAD DA6AB0F5 E7F544BB 1313DB93 BB733605 3678459A 31386D3A 9F0A477F
37B853DF 63BBA87B ECAC7CD2 B19FFACD 98B40E82 0B638D5F 7DDAAE56 FF198EF6
AB1002C3 76C1FFDE 03041201 FF8E6AF9 4AFDF056 06E10E32 F3F69091 34864AEB
D983AAA2 BD725FCC A288DECE 27810D34 807956DC 78F3CFC4 EA45A8DF ADA4226C

Xâu nhị phân biểu diễn số nguyên t dưới dạng một số nguyên dương không dấu là chữ ký được tạo ra bởi hàm tạo chữ ký thay thế (xem Phụ lục A.6) $\Sigma' = t$.

Vì kết quả trên lớn hơn $n/2$, nó được thay thế bởi phần dư của nó với n. Xâu nhị phân biểu diễn số nguyên này dưới dạng một số nguyên dương không dấu là chữ ký $\Sigma = n - t$.

$\Sigma$ = 56136187 14871D42 EAA2CFFB DB9639BA 04ED8532 B4A20C4E 79CB2BA3 0ED58BB5

```
93E38E2C 9CBF7563 32CC26C4 B115F73C E1BEF204 252DAF73 708569E6 E3304E1F
F194E877 69800E73 10D31E4E 4AE53E2D 73FE0EDC C1B74AAB 6A64A808 CA3EDA35
2B0F86C0 A4ECC996 B8301BD9 8293B7BF 4063CD94 66091B74 39E3682A 53A58505
```

Thông điệp đã ký có 128 xâu bộ tám chỉ bao gồm chữ ký vì $M_2$ là rỗng.

**E.1.2.2.2 Quá trình xác thực**

Chữ ký $\Sigma$ là một xâu nhị phân đại diện một số nguyên dương không dấu, nhỏ hơn $n/2$. Số nguyên này tăng theo lũy thừa bậc 3 mô-đun $n$, do đó thu được số nguyên $f_s$.

```
f_s =   7EF31404 A97EE034 13CD2E94 857038E0 196F22D0 5FE6BB7F FA6E1703 301942D0
        FBB748B5 F7F3C901 633DAF97 BBC08F47 0DE125D1 70986E6C C4B3BD22 762E8ACC
        52ED849F 8BA3A6C4 67FC5A0D A50E6FD9 4B883A69 79CD79D5 55B5788C F9836B2A
        2D12F766 E31351D0 18AEA9E9 15B3774F 117D95F8 1C930A59 83EB0E8D 19FA75A5
```

Vì $f_s$ là đồng dư với $(n - 12)$ mô-đun 16, nó được thay thế bởi phần dư của nó với $n$, có nghĩa là số nguyên được khôi phục là $f'_r = n - f_s$.

```
f_r' =  7BB5D930 4572EE04 BECAE622 6939DC6D A6F19867 8B339668 B09581DA 7DC69063
        CFE49956 108754DD BC3AF3FF A6F562C3 6C91DAB4 BFF8CE66 29AC5B1B 6C1B524A
        49B7669B 549E678C ABDAD642 A565394D 7373C4C9 4ECADF09 08A5C00D 0511B9F6
        D78039FC 7F4BD793 420A50BE 94614DA4 AF5F8E78 C269E0DF A03E027C E74F31CC
```

$f'_r$ được biểu diễn dưới dạng một số nguyên dương không dấu bởi xâu đã được khôi phục $S'_r$. Hàm tạo mặt nạ *MGF1* áp dụng cho 848 (= 1024 – 160 – 16) bit bên trái nhất của $S'_r$, từ đó thu được xâu được khôi phục tạm thời $S'_i$.

```
S_i' =  80000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
        00000000 0001FEDC BA987654 3210FEDC BA987654 3210FEDC BA987654 3210FEDC
        BA987654 3210FEDC BA987654 3210FEDC BA987654 3210436B CA9954EC 376C96B7
        9C95D4B8 2686F349 4AD350BE 94614DA4 AF5F8E78 C269E0DF A03E027C E74F31CC
```

$S'_i$ biểu diễn xâu được khôi phục trung gian như sau.

– Bit bên trái nhất của $S'_i$ được thiết lập bằng '0' vì $\delta = 1$ ($\delta = (1 - 1024) mod 8$). 37 xâu bộ tám bên trái nhất của xâu nhị phân còn lại bằng '0'; nó được theo sau bởi xâu bộ tám bao quanh "01"; 38 xâu bộ tám đó được chuyển sang bên trái của $S'_i$.

– Xâu bộ tám bên phải nhất của $S'_i$ bằng "CC"; do đó, trailer bao gồm hai xâu bộ tám và bằng "33CC"; hai xâu bộ tám đó cũng được chuyển sang bên phải của $S'_i$.

Định danh hàm băm bằng "31"; do đó hàm băm được sử dụng là hàm băm chuyên dụng 1.

Xâu còn lại 704 bit được chia làm ba phần.

– $M_1^*$ bao gồm 384 bit bên trái nhất.
– $S^*$ bao gồm 160 bit bên phải nhất.
– $H'$ bao gồm 160 bit bên phải nhất.

$M_1^*$ = FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210

$S^*$ = 436BCA99 54EC376C 96B79C95 D4B82686 F3494AD3

$H'$ = 50BE9461 4DA4AF5F 8E78C269 E0DFA03E 027CE74F

Thông điệp đã được khôi phục $M^*$ chỉ bao gồm $M_1^*$ vì khôi phục thông điệp là toàn bộ. Mã băm còn lại $H''$ được tính bằng các áp dụng hàm băm chuyên dụng 1 cho xâu nhị phân có độ dài 768 (=64+384+160+160), kết quả của việc ghép 64 bit của độ dài thông điệp đã được khôi phục $C'$, 384 bit của thông điệp đã được khôi phục $M^*$, 160 bit của mã băm của phần thông điệp không thể khôi phục (phần này bằng rỗng) $h(M_2^*)$ và 160 bit của salt đã được khôi phục $S^*$. $H'' = h(C'||M_1^*||h(M_2^*)||S^*)$.

$H''$ = 50BE9461 4DA4AF5F 8E78C269 E0DFA03E 027CE74F

Vì hai mã băm $H'$ và $H''$ là giống nhau, chữ ký $\Sigma$ được chấp nhận.

### E.1.2.3 Ví dụ về lược đồ chữ ký 3

Ví dụ này sử dụng hàm băm chuyên dụng 3 trong ISO/IEC 10118-3 (còn được biết đến với tên gọi là SHA-1).

#### E.1.2.3.1 Quá trình ký

Thông điệp để ký là rỗng, có nghĩa là một xâu nhị phân có độ dài bằng 0.

Vì lược đồ chữ ký này là thuộc kiểu tất định, giá trị salt S có độ dài bằng 0 được lựa chọn.

160 bit mã băm được tính toán bằng cách áp dụng hàm băm chuyên dụng 3 cho xâu nhị phân có độ dài 224 (= 64 + 160), kết quả của việc ghép thêm 64 bit độ dài thông điệp có thể khôi phục được $C$ và 160 bit của mã băm của phần không thể khôi phục được (phần này bằng rỗng) $h(M_2)$. $H = h(C||h(M_2))$.

$H$ = A35D1688 A60AC69F D53E4442 8BFD380E 94DB9176

Hàm băm được sử dụng đã được biết đến hoàn toàn. Do đó, trường trailer $T$ bao gồm một xâu bộ tám duy nhất.

$T$ = BC

Thông điệp là đủ ngắn để khôi phục toàn bộ. 1024 bit của xâu trung gian $S_i$ kết quả của việc nối 855 (= 1024 − 160 − 8 − 1) bit đệm bằng '0', bit bao quanh bằng 1, 160 bit của $H$ và 8 bit của trường trailer $T$.

$S_i$ = 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 000001A3 5D1688A6
0AC69FD5 3E44428B FD380E94 DB9176BC

Xâu có thể khôi phục $S_r$ thu được từ việc áp dụng hàm tạo mặt nạ *MGF1* đối với 856 (= 1024 − 160 − 8) bit bên trái nhất của $S_i$, và 1 bit bên trái nhất của $S_r$ được thiết lập bằng '0' vì δ =1 (δ= 1 − 1024)$mod$ 8).

$S_r$ = 7CCB5422 2079C84C 343B0AB1 6307273B 36359229 BD3DFDEC A9FE8054 AD1EF319
44758A67 3B7C70C2 FACB6FE9 12690EE2 6DF58975 585A78C2 723F0C71 50535C80
8F0868F6 CA94F36C FB079FBB 9126286D 5EECA3CA ACA12593 033A0D64 136A7A72
D6050B0A 6CF68B6D DA0AE6A3 5D1688A6 0AC69FD5 3E44428B FD380E94 DB9176BC

Số nguyên có thể khôi phục $f_r$ là số nguyên dương không dấu biểu diễn $S_r$. $f_r$ tăng theo lũy thừa bậc $s$ theo mô-đun $n$. Kết quả được biểu diễn bởi một số nguyên dương không dấu tạm thời $t$.

```
t =   F9DD9F72 FAB4AFFC ED3B0538 C5848B27 756AC50C B2890F4C BC268D96 C5E91EE8
      8E3B058F 2EF6585F EF5323CA 4E2C30BC C6140CF5 F5357960 5B3BF0CC 621082EB
      77F4A42D 3567355E AA151FB4 652BAFFE 58A4B310 7A064669 FD4177C8 D79F5DE5
      E8C562FF A2D0F5D9 C409AEA0 D5B9F8DF 493AF2F1 8F91D828 CE32C4CC 35C13113
```

Xâu nhị phân biểu diễn số nguyên t dưới dạng một số nguyên dương không dấu là chữ ký được tạo ra bởi hàm tạo chữ ký thay thế (xem Phụ lục A.6) $\Sigma' = t$.

Vì kết quả trên lớn hơn $n/2$, chữ ký $\Sigma = n - t$.

```
Σ =   00CB4DC1 F43D1E3B E55D0F7E 29258A26 4AF5F62B 3891429B EEDD0B46 E7F6B44C
      3D60DC7C D984C57F 30257FCD 1489C17D B45EF390 3B5BC372 93242771 B0395A2B
      24B0470D AADAD8F2 69C2109B E547F928 66574C22 4E921274 6119C0D1 2725C73B
      15CDCE63 BF8E3389 96AF4C06 D45ACC14 77A2317F 4F6B1310 55F64C3D CB88765E
```

Thông điệp đã ký có 128 xâu bộ tám chỉ bao gồm chữ ký vì $M_2$ là rỗng.

**E.1.2.3.2 Quá trình xác thực**

Chữ ký $\Sigma$ là một xâu nhị phân đại diện một số nguyên dương không dấu, nhỏ hơn $n/2$. Số nguyên này tăng theo lũy thừa bậc 3 mô-đun $n$, do đó thu được số nguyên $f_s$.

```
f_s =  7DDD9912 CE7805EC 9E5D0A05 8BA2EE12 8A2B290E 2DDC53FC 01051889 00C0E01B
       872657A4 CCFEAD1C 24AD33AE 504CE328 0C7D7710 D836C410 7C210BCC 91F68096
       0D9C8244 15AD1AE4 18CF9094 B94D80B9 600F5B68 1BF7334B 5B212B35 EB5AAAAE
       2E8E2958 F56B9DF5 80AE1404 4CFE3C4D B616849B A0B8A8AD 26F10275 25B830B5
```

Vì $f_s$ là đồng dư với $(n - 12)$ mô-đun 16, số nguyên được khôi phục là $f'_r = n - f_s$.

```
f'_r =  7CCB5422 2079C84C 343B0AB1 6307273B 36359229 BD3DFDEC A9FE8054 AD1EF319
        44758A67 3B7C70C2 FACB6FE9 12690EE2 6DF58975 585A78C2 723F0C71 50535C80
        8F0868F6 CA94F36C FB079FBB 9126286D 5EECA3CA ACA12593 033A0D64 136A7A72
        D6050B0A 6CF68B6D DA0AE6A3 5D1688A6 0AC69FD5 3E44428B FD3B0E94 DB9176BC
```

$f'_r$ được biểu diễn dưới dạng một số nguyên dương không dấu bởi xâu đã được khôi phục $S'_r$. Hàm tạo mặt nạ *MGF1* áp dụng cho 856 (= 1024 – 160 – 8) bit bên trái nhất của $S'_r$, từ đó thu được xâu được khôi phục trung gian $S'_i$.

```
S'_i =  00000000 00000000 00000000 00000000 00000000 00000000 00000000
        00000000 00000000 00000000 00000000 00000000 00000000 00000000
        00000000 00000000 00000000 00000000 00000000 00000000 00000000
        00000000 00000000 00000000 00000000 00000000 000001A3 5D1688A6
        0AC69FD5 3E44428B FD3B0E94 DB9176BC
```

$S'_i$ biểu diễn xâu được khôi phục trung gian như sau.

– Bit bên trái nhất của $S'_i$ được thiết lập bằng '0' vì $\delta = 1$ ($\delta = (1 - 1024) mod 8$). 106 xâu bộ tám bên trái nhất của xâu nhị phân còn lại bằng '0'; nó được theo sau bởi xâu bộ tám bao quanh "01"; 107 xâu bộ tám đó được chuyển sang bên trái của $S'_i$.

– Xâu bộ tám bên phải nhất của $S'_i$ bằng "BC"; xâu bộ tám đó cũng được chuyển sang bên phải của $S'_i$.

Vì trailer bằng "BC", hàm băm được sử dụng đã được biết đến hoàn toàn; hàm băm chuyên dụng 3 trong ví dụ này.

Xâu còn lại 160 bit được giả định là mã băm $H'$ vì không còn thừa dữ liệu.

$H'$ = A35D1688 A60AC69F D53E4442 8BFD380E 94DB9176

Thông điệp đã được khôi phục $M'$ được giả định là rỗng và do đó, khôi phục là toàn bộ. Mã băm khác $H''$ được tính bằng cách áp dụng SHA-1 cho xâu nhị phân có độ dài 224 (=64+160), kết quả của việc ghép thêm 64 bit của độ dài thông điệp đã được khôi phục $C'$ và 160 bit của mã băm của phần thông điệp không thể khôi phục được (phần này bằng rỗng) $h(M_2')$. $H'' = h(C'||h(M_2))$.

$H''$ = A35D1688 A60AC69F D53E4442 8BFD380E 94DB9176

Vì hai mã băm $H'$ và $H''$ là giống nhau, chữ ký $\Sigma$ được chấp nhận.

### E.1.3 Các ví dụ về khôi phục một phần

Ở đây trình bày ba ví dụ về tạo và xác thực chữ ký, mỗi ví dụ tương ứng với một trong ba lược đồ.

#### E.1.3.1 Ví dụ về lược đồ chữ ký 1

Ví dụ này sử dụng hàm băm chuyên dụng 1 trong ISO/IEC 10118-3 (còn được biết đến với tên gọi là RIPEMD-160).

##### E.1.3.1.1 Quá trình ký

Ví dụ này mô tả chữ ký của một thông điệp 132 xâu bộ tám, có nghĩa là 1056 bit.

$M$ = FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98

160 bit mã băm được tính toán bằng cách áp dụng hàm băm chuyên dụng 1 cho 1056 bit của $M$.

$H$ = F0EA911A F528FA38 777D4B9A 58B6FDA4 2D7E1999

Hàm băm được sử dụng đã được biết đến hoàn toàn. Do đó, trường trailer $T$ bao gồm 8 bit sau đây.

$T$ = BC

Thông điệp là quá dài để có thể được khôi phục hoàn toàn bởi quá trình xác thực. Do đó, nó được chia làm hai phần.

– $M_1$ bao gồm 848 bit bên trái nhất.
– $M_2$ bao gồm 208 bit còn lại, có nghĩa là 26 xâu bộ tám.

$M_1$ = FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDC

$M_2$ = BA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98

1024 bit của xâu trung gian $S_i$ kết quả của việc ghép thêm hai bit tiêu đề bằng "01", bit dữ liệu thêm bằng '1', bốn (= 1024 – 848 – 160 – 8 – 4) bit đệm bằng '0', bit bao quanh bằng 1, 848 bit của $M_1$, 160 bit của $H$ và 8 bit của trường trailer $T$. Xâu có thể khôi phục $S_r$ kết quả của việc xâu bộ bốn bao quanh bằng 1 được thay thế bằng 'A'.

```
Sr =   6AFEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432
       10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432
       10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432
       10FEDCBA 98765432 10FEDCF0 EA911AF5 28FA3877 7D4B9A58 B6FDA42D 7E1999BC
```

Số nguyên có thể khôi phục $f_r$ là số nguyên dương không dấu biểu diễn $S_r$. $f_r$ tăng theo lũy thừa bậc $s$ theo mô-đun $n$. Kết quả được biểu diễn bởi một số nguyên dương không dấu tạm thời $t$.

```
t =    C9DE5B79 67CFD8BE 506749A2 F2E5035C 9C2C5E94 3DD46838 AEF7144E A01283F0
       95C35FE5 53A87553 AEADBBCE 2B9876EC 14EA5C31 EA11BCC1 F33E5161 7B4B73C2
       38EB6D4C AA3DF32D 1434E846 E2E74146 E24C7171 D2A0FBED 77E37371 1444360B
       962A9C27 D9CC2E15 4FE30BEC A3E20B4C 0CCF472F 70E64A9C 9FFAA56A 98BC1079
```

Vì kết quả trên lớn hơn $n/2$, chữ ký $\Sigma = n - t$.

```
Σ =    30CA91BB 8721F57A 8230CB13 FBC511F1 24345CA3 AD45E9AF FC0C848F 0DCD4F44
       35D88226 B4D2A88B 70CAE7C9 371D7B1E 6588A454 467F8010 FB21C6DC 66FE6954
       63B97DEE 36041B23 FFA24809 678C67DF DCAF8DC0 F5F75CF0 E677C528 EA80EF15
       6E68953B 8892FB4E 0AD5EEBB 0632B9A7 B40DDD41 6E16A09C 842E6B9F 688D96F8
```

Thông điệp đã ký có 128 xâu bộ tám chữ ký $\Sigma$ cùng với 26 xâu bộ tám của thông điệp không thể khôi phục được $M_2$, có nghĩa là chỉ nhiều hơn 22 xâu bộ tám so với thông điệp $M$.

**E.1.3.1.2 Quá trình xác thực**

Chữ ký $\Sigma$ là một xâu nhị phân đại diện một số nguyên dương không dấu, nhỏ hơn $n/2$. Số nguyên này tăng theo lũy thừa bậc 3 mô-đun $n$, do đó thu được số nguyên $f_s$.

```
fs =   8FAA107A 567B7A06 C19937FC 5633C11B AF61DE7D 52A3FDB6 9A04BC23 15697F02
       BA9D0551 7004C9AD 0E79C6DC CA3F9DD8 697423CB 981AE8A0 DD613B83 49D388E4
       EBA60E80 47CBBA1F 02D85395 B1FD54F4 ADFD2278 302204AC 4D5C5BDF 664ED0EE
       F39454A8 C9E8D531 49BA1DB6 BF83A9FE 97E2EBF9 61B150E0 6D2B6CDC 83300DB5
```

Vì $f_s$ là đồng dư với $(n - 12)$ mô-đun 16, nó được thay thế bởi phần dư của nó với $n$, có nghĩa là số nguyên được khôi phục là $f'_r = n - f_s$.

```
fr' =  6AFEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432
       10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432
       10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432
       10FEDCBA 98765432 10FEDCF0 EA911AF5 28FA3877 7D4B9A58 B6FDA42D 7E1999BC
```

$f'_r$ được biểu diễn dưới dạng một số nguyên dương không dấu bởi xâu đã được khôi phục $S'_r$.

– Xâu bộ tám bên trái nhất của $S'_r$ bằng "6A"; nó bao gồm tiêu đề bằng "01", bit dữ liệu thêm bằng '1' (khôi phục một phần), một bit đệm bằng '0' và một xâu bộ bốn đệm bằng 'A'; xâu bộ tám này được chuyển sang bên trái của $S'_r$.

– Xâu bộ tám bên phải nhất của $S'_T$ bằng "BC"; xâu bộ tám đó cũng được chuyển sang bên phải của $S'_T$.

Vì trailer bằng "BC"; hàm băm được sử dụng đã được biết đến hoàn toàn, hàm băm chuyên dụng 1 trong ví dụ này.

Xâu còn lại 1008 bit được chia làm hai phần.

– $M_1^*$ bao gồm 848 bit bên trái nhất.
– $H'$ bao gồm 160 bit bên phải nhất.

$M_1^*$ = FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDC

$H'$ = F0EA911A F528FA38 777D4B9A 58B6FDA4 2D7E1999

Vì khôi phục là một phần, thông điệp đã được khôi phục $M^*$ bao gồm $M_1^*$ và $M_2^*$, phần có thể và không thể khôi phục được.

$M^*$ = FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98

Mã băm còn lại $H''$ được tính bằng các áp dụng hàm băm chuyên dụng 1 cho $M^*$.

$H''$ = F0EA911A F528FA38 777D4B9A 58B6FDA4 2D7E1999

Vì hai mã băm $H'$ và $H''$ là giống nhau, chữ ký $\Sigma$ được chấp nhận.

### E.1.3.2 Ví dụ về lược đồ chữ ký 2

Ví dụ này sử dụng hàm băm chuyên dụng 3 trong ISO/IEC 10118-3 (còn được biết đến với tên gọi là SHA-1).

#### E.1.3.2.1 Quá trình ký

Thông điệp để ký là xâu 112 ký tự mã ASCII sau đây.

abcdbcdecdefdefgefghfghighijhijkijkljklmklmnlmnomnopnopq
opqrpqrsqrstrstustuvtuvwuvwxvwxywxyzxyzayzabzabcabcdbcde

Trong hệ thập lục phân, thông điệp $M$ là xâu bộ tám có độ dài 112 xâu bộ tám, có nghĩa là 896 bit sau đây.

$M$ = 61626364 62636465 63646566 64656667 65666768 66676869 6768696A 68696A6B
696A6B6C 6A6B6C6D 6B6C6D6E 6C6D6E6F 6D6E6F70 6E6F7071 6F707172 70717273
71727374 72737475 73747576 74757677 75767778 76777879 7778797A 78797A61
797A6162 7A616263 61626364 62636465

160 bit của salt $S$ được tạo ra.

$S$ = 4C95C1B8 7A1DE8AC C193C14C F3147FE9 C6636078

Thông điệp là quá dài để có thể được khôi phục hoàn toàn bởi quá trình xác thực. Do đó, nó được chia làm hai phần.

- $M_1$ bao gồm 688 bit bên trái nhất.
- $M_2$ bao gồm 208 bit còn lại, có nghĩa là 26 xâu bộ tám.

$M_1$ = 61626364 62636465 63646566 64656667 65666768 66676869 6768696A 68696A6B
696A6B6C 6A6B6C6D 6B6C6D6E 6C6D6E6F 6D6E6F70 6E6F7071 6F707172 70717273
71727374 72737475 73747576 74757677 75767778 7677

$M_2$ = 7879 7778797A 78797A61 797A6162 7A616263 61626364 62636465

160 bit của mã băm được tính bằng cách áp dụng hàm băm chuyên dụng 3 với xâu nhị phân có độ dài 1072 (= 64 + 688 + 160 + 160), kết quả của việc ghép 64 bit của độ dài phần không thể khôi phục được C, 688 bit của phần thông điệp có thể khôi phục được $M_1$, 160 bit của mã băm của phần không thể khôi phục được $h(M_2)$ và 160 bit của salt $S$. $H = h(C||M_1||h(M_2)||S)$.

$H$ = 16671F61 4F2954A8 6E51CB81 102A3D47 E2C11EBD

Hàm băm được sử dụng đã được biết đến hoàn toàn. Do đó, trường trailer $T$ bao gồm 8 bit sau đây.

$T$ = BC

1024 bit của xâu trung gian $S_i$ kết quả của việc ghép thêm bảy (= 1024 – 688 – 160 – 160 – 8 – 1) bit đệm bằng '0', bit bao quanh bằng 1, 688 bit của $M_1$, 160 bit của $L$, 160 bit của $H$ và 8 bit của trường trailer $T$.

$S_i$ = 01616263 64626364 65636465 66646566 67656667 68666768 69676869 6A68696A
6B696A6B 6C6A6B6C 6D6B6C6D 6E6C6D6E 6F6D6E6F 706E6F70 716F7071 72707172
73717273 74727374 75737475 76747576 77757677 7876774C 95C1B87A 1DE8ACC1
93C14CF3 147FE9C6 63607816 671F614F 2954A86E 51CB8110 2A3D47E2 C11EBDBC

Xâu có thể khôi phục $S_r$ thu được từ việc áp dụng hàm tạo mặt nạ *MGF1* đối với 856 (= 1024 – 160 – 8) bit bên trái nhất của $S_i$, và 1 bit bên trái nhất của $S_r$ được thiết lập bằng '0' vì δ =1 (δ= 1 – 1024)$mod$ 8).

$S_r$ = 390871A1 2BB3F417 63782F59 BB700DD4 63C071B6 98C7D152 9B8616A9 B72DF9DE
B899BFA8 C3839DD0 903CEF72 A9C18496 64129992 6FE8D3F8 FDA1D07D 2251EAAD
34017F7D C66DD60F D3F0014D 23CA2D2E 43383F30 9724B846 2529C5FE 73205FB5
D1FCC8CF D18C687F B9E35616 671F614F 2954A86E 51CB8110 2A3D47E2 C11EBDBC

Số nguyên có thể khôi phục $f_r$ là số nguyên dương không dấu biểu diễn $S_r$. $f_r$ tăng theo lũy thừa bậc $s$ theo mô-đun $n$. Kết quả được biểu diễn bởi một số nguyên dương không dấu tạm thời $t$.

$t$ = 92ACA17F 28426177 1E4A1313 C0510483 8C3CC91C 1CB6F576 CF95090A 5FDEA51E
3C189F65 E6BA3F28 4268B4FF 2363B3B9 12D023A9 1C96541A C1F9E60E 58F6B3DA
8DFB1B69 41792AA6 341DB184 88366A5E 1E18DBBA E4A2E390 77A2B4FE 1DFB34A2
CCAD1812 C4AFFAF5 5570855A AEB685DA 2E1F124F F70F529F ED02F515 BFD572AE

Xâu nhị phân biểu diễn số nguyên $t$ dưới dạng một số nguyên dương không dấu là chữ ký được tạo ra bởi hàm tạo chữ ký thay thế (xem phụ lục B.6) $\Sigma' = t$.

Vì kết quả trên lớn hơn $n/2$, chữ ký $\Sigma = n - t$.

```
Σ=   67FC4BB5 C6AF6CC1 B44E01A3 2E5910CA 3423F21B CE635C71 DB6E8FD3 4E012E16
     8F5342A6 21C0DEB6 DD0FEE98 3F523E51 67A2DCDD 13FAE8B8 2C66322F 8953293C
     0EB9CFD1 9EC8E3AA DFB97ECB C23D3EC8 A0E32377 E3F5754D E6B8839B E0C9F07E
     37E61950 9DAF2E6E 0548754C FB5E3F19 92BE1220 E7ED9899 37261BF4 417434C3
```

Thông điệp đã ký có 128 xâu bộ tám chữ ký $\Sigma$ cùng với 26 xâu bộ tám của thông điệp không thể khôi phục được $M_2$, có nghĩa là chỉ nhiều hơn 42 xâu bộ tám so với thông điệp *M*.

**E.1.3.2.2 Quá trình xác thực**

Chữ ký $\Sigma$ là một xâu nhị phân đại diện một số nguyên dương không dấu, nhỏ hơn $n/2$. Số nguyên này tăng theo lũy thừa bậc 3 mô-đun *n*, do đó thu được số nguyên $f_s$.

```
f_s =  C1A07B93 C36DDA21 6F1FE55D 333A0779 5CA04981 52528096 0F7D8233 F6B1D956
       13022263 44F7800E 8F3BB424 B8F46D74 166066F3 C0A868D9 F0BE47C0 BFF7F269
       68A36BBD 19D43841 3FE72F03 26A97BF8 7BC3C002 3173A098 3931729B 8BA4C56B
       32966893 90D2C0E3 A0D5A491 42F563A4 97887C02 8D316A28 F9EBC927 402AE9B5
```

Vì $f_s$ là đồng dư với $(n - 12)$ mô-đun 16, nó được thay thế bởi phần dư của nó với *n*, có nghĩa là số nguyên được khôi phục là $f'_r = n - f_s$.

```
f'_r =  390871A1 2B83F417 63782F59 BB700DD4 63C071B6 98C7D152 9B8616A9 B72DF9DE
        B899BFA8 C3839DD0 903CEF72 A9C18496 64129992 6FE8D3F8 FDA1D07D 2251EAAD
        34017F7D C66DD60F D3F0014D 23CA2D2E 43383F30 9724B846 2529C5FE 73205FB5
        01FCC8CF D19C687F B9E35616 671F614F 2954A86E 51CB8110 2A3D47E2 C11EBDBC
```

$f'_r$ được biểu diễn dưới dạng một số nguyên dương không dấu bởi xâu đã được khôi phục $S'_r$. Hàm tạo mặt nạ *MGF1* đối với 856 (= 1024 – 160 – 8) bit bên trái nhất của $S'_r$ , và thu được xâu đã được khôi phục trung gian $S'_i$.

```
S'_i =  01616263 64626364 65636465 66646566 67656667 68666768 69676869 6A68696A
        6B696A6B 6C6A6B6C 6D6B6C6D 6E6C6D6E 6F6D6E6F 706E6F70 716F7071 72707172
        73717273 74727374 75737475 76747576 77757677 7876774C 95C1D87A 1DE9ACC1
        93C14CF3 147FE9C6 63607816 671F614F 2954A86E 51CB8110 2A3D47E2 C11EBDBC
```

$S'_i$ biểu diễn xâu trung gian đã được khôi phục như sau.

– Bit bên trái nhất của $S'_i$ được thiết lập bằng '0' vì δ =1 (δ= 1 – 1024)$mod$ 8). 7 bit bên trái nhất của xâu còn lại bằng '0'; nó được theo sau bởi bit bao quanh '1'; xâu bộ tám này được chuyển sang bên trái của $S'_i$.

– Xâu bộ tám bên phải nhất của $S'_i$ bằng "BC"; xâu bộ tám đó cũng được chuyển sang bên phải của $S'_i$.

Vì trường trailer *T* bằng "BC"; hàm băm được sử dụng đã được biết đến hoàn toàn, hàm băm chuyên dụng 3 trong ví dụ này.

Xâu còn lại 1008 bit được chia làm ba phần.

– $M_1{}^*$ bao gồm 688 bit bên trái nhất.

– $S^*$ bao gồm 160 bit bên phải nhất.

– $H'$ bao gồm 160 bit bên phải nhất.

$M_1^*$ = 61626364 62636465 63646566 64656667 65666768 66676869 6768696A 68696A6B
696A6B6C 6A6B6C6D 6B6C6D6E 6C6D6E6F 6D6E6F70 6E6F7071 6F707172 70717273
71727374 72737475 73747576 74757677 75767778 7677

$S^*$ = 4C95C1B8 7A1DE8AC C193C14C F3147FE9 C0636078

$H'$ = 16671F61 4F2954A8 6E51CB81 102A3D47 E2C11EBD

Vì khôi phục là một phần, thông điệp đã được khôi phục $M^*$ bao gồm $M_1^*$ và $M_2^*$, phần có thể và không thể khôi phục được.

$M^*$ = 61626364 62636465 63646566 64656667 65666768 66676869 6768696A 68696A6B
696A6B6C 6A6B6C6D 6B6C6D6E 6C6D6E6F 6D6E6F70 6E6F7071 6F707172 70717273
71727374 72737475 73747576 74757677 75767778 76777879 7778797A 78797A61
797A6162 7A616263 61626364 62636465

Mã băm còn lại $H''$ được tính bằng các áp dụng hàm băm SHA-1 cho xâu nhị phân có độ dài 1072 (= 64 + 688 + 160 + 160), kết quả của việc ghép thêm 64 bit độ dài phần có thể khôi phục được $C'$, 688 bit của phần thông điệp đã được khôi phục $M_1^*$, 160 bit của mã băm của phần thông điệp không thể khôi phục h($M_2^*$) và 160 bit của salt đã được khôi phục S*. $H'' = h(C'||M_1^*||h(M_2^*)||S^*)$

$H''$ = 16671F61 4F2954A8 6E51CB81 102A3D47 E2C11EBD

Vì hai mã băm $H'$ và $H''$ là giống nhau, chữ ký $\Sigma$ được chấp nhận.

**E.1.3.3 Ví dụ về lược đồ chữ ký 3**

Ví dụ này sử dụng hàm băm chuyên dụng 3 trong ISO/IEC 10118-3 (còn được biết đến với tên gọi là SHA-1).

**E.1.3.3.1 Quá trình ký**

Ví dụ này mô tả chữ ký của một thông điệp 132 xâu bộ tám, có nghĩa là 1056 bit sau đây.

$M$ = FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98

Vì lược đồ chữ ký này là thuộc kiểu tất định, một giá trị salt $S$ có độ dài bằng 0 được lựa chọn.

Thông điệp là quá dài để có thể được khôi phục hoàn toàn bởi quá trình xác thực. Do đó, nó được chia làm hai phần.

- $M_1$ bao gồm 840 bit bên trái nhất.
- $M_2$ bao gồm 216 bit còn lại, có nghĩa là 27 xâu bộ tám.

$M_1$ = FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FE

$M_2$ = DCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98

160 bit của mã băm *H* được tính bằng cách áp dụng hàm băm chuyên dụng 3 với xâu nhị phân có độ dài 1064 (= 64 + 840 + 160), kết quả của việc ghép 64 bit của độ dài phần không thể khôi phục được C, 840 bit của phần thông điệp có thể khôi phục được $M_1$, 160 bit của mã băm của phần không thể khôi phục được $h(M_2)$. $H = h(C||M_1||h(M_2))$.

*H* = E30A9CB8 F10DC3C8 1897D9E8 D3945S5A AC6DEC79

Định danh trong trường trailer *T* xác định hàm băm được sử dụng, là hàm băm chuyên dụng 3; ISO/IEC 10118-3 thiết lập định danh cho hàm băm giá trị "33". Do đó, trường trailer *T* bao gồm 16 bit sau đây.

*T* = 33CC

1024 bit của xâu trung gian $S_i$ kết quả của việc ghép thêm bảy (= 1024 − 840 − 160 − 16 − 1) bit đệm bằng '0', bit bao quanh bằng 1, 840 bit của phần có thể khôi phục được $M_1$, 160 bit của mã băm của phần thông điệp không thể khôi phục được $h(M_2)$ và 16 bit của trường trailer *T*.

$S_i$ = 01FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432
10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432
10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432
10FEDCBA 98765432 10FEE30A 9CB8F10D C3C81897 D9E8D394 555AAC6D EC7933CC

Xâu có thể khôi phục $S_r$ thu được từ việc áp dụng hàm tạo mặt nạ *MGF1* đối với 848 (= 1024 − 160 − 16) bit bên trái nhất của $S_i$, và 1 bit bên trái nhất của $S_r$ được thiết lập bằng '0' vì δ =1 (δ= 1 − 1024)$mod$ 8).

$S_r$ = 1E1F9F67 4356F609 0062DEA3 FC994589 8259AE44 7F4ACAD2 0655D646 0435C851
ED9D1754 837A1269 1886C244 DED123EB 470510BC 459AD416 99310030 C824907D
0DC63B7F 422008FA 4D6BE912 0DFDD534 8FAD5056 DD7A43F1 BB38E64A EEDF14CD
2549A7F6 B1869D77 C4E5E30A 9CB8F10D C3C81897 D9E8D394 555AAC6D EC7933CC

Số nguyên có thể khôi phục $f_r$ là số nguyên dương không dấu biểu diễn $S_r$. $f_r$ tăng theo lũy thừa bậc *s* theo mô-đun *n*. Kết quả nhỏ hơn *n*/2 vẫn được giữ nguyên. Xâu nhị phân biểu diễn số nguyên đó dưới dạng một số nguyên dương không dấu là chữ ký $\Sigma$.

$\Sigma$ = 30147ECB 074705DD F33EF765 D0EE1017 D5535AB3 9A7727C4 D8D4DC42 42C693BD
1FB544EC AE2323D1 185BED05 C8AA5F69 9D3AAED4 1FC3ECF9 DF297A61 56D6BC86
5196A619 806E3FDF F8A8416D 2984EF9E 33940013 4A6D1712 2FCF0946 783AEBD4
6F11397E 66863E74 28F4542D E2AE8A30 7355633F 380F937B 308C149F 14194487

Trong ví dụ này, chữ ký được tạo ra bởi hàm tạo chữ ký thay thế (xem phụ lục B.6) cũng là một xâu nhị phân $\Sigma$, $\Sigma' = \Sigma$.

Thông điệp đã ký có 128 xâu bộ tám chữ ký $\Sigma$ cùng với 27 xâu bộ tám của thông điệp không thể khôi phục được $M_2$, có nghĩa là chỉ nhiều hơn 23 xâu bộ tám so với thông điệp *M*.

**E.1.3.3.2 Quá trình xác thực**

Chữ ký $\Sigma$ là một xâu nhị phân đại diện một số nguyên dương không dấu, nhỏ hơn $n/2$. Số nguyên này tăng theo lũy thừa bậc 3 mô-đun *n*, do đó thu được số nguyên $f_s$.

$f_s$ = 1E1F9F67 4356F609 0062DEA3 FC994589 8259AE44 7F4ACAD2 0655D646 0435C851
ED9D1754 837A1269 1886C244 DED123EB 470510BC 459AD416 99310030 C824907D
0DC63B7F 422008FA 4D68E912 0DFDD534 8FAD5056 DD7A43F1 BB38E64A EEDF14CD
2549A7F6 B1869D77 C4E5E30A 9CB8F10D C3C81897 D9E8D394 555AAC6D EC7933CC

Vì $f_s$ là đồng dư với $(n - 12)$ mô-đun 16, có nghĩa là số nguyên được khôi phục là $f'_r = f_s$.

$f'_r$ được biểu diễn dưới dạng một số nguyên dương không dấu bởi xâu đã được khôi phục $S'_r$. Hàm tạo mặt nạ *MGF1* đối với 848 (= 1024 – 160 – 16) bit bên trái nhất của $S'_r$ , và thu được xâu đã được khôi phục trung gian $S'_i$.

$S_i'$ = 81FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432
10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432
10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432
10FEDCBA 98765432 10FEE30A 9CB8F10D C3C81897 D9E8D394 555AAC6D EC7933CC

$S'_i$ biểu diễn xâu trung gian đã được khôi phục như sau.

– Bit bên trái nhất của $S'_i$ được thiết lập bằng '0' vì δ =1 (δ= 1 – 1024)$mod$ 8). 7 bit bên trái nhất của xâu còn lại bằng '0'; nó được theo sau bởi bit bao quanh '1'; xâu bộ tám này được chuyển sang bên trái của $S'_i$.

– Xâu bộ tám bên phải nhất của $S'_i$ bằng "CC"; do đó, trường trailer bao gồm hai xâu bộ tám bằng "33CC"; hai xâu bộ tám đó cũng được chuyển sang bên phải của $S'_i$.

Định danh hàm băm bằng "33"; do đó, hàm băm được sử dụng là hàm băm chuyên dụng 3.

Xâu còn lại 1000 bit được chia làm hai phần.

– $M_1^*$ bao gồm 840 bit bên trái nhất.
– $H'$ bao gồm 160 bit bên phải nhất.

$M_1^*$ = FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FE
$H^*$ = E30A9CB8 F10DC3C8 1897D9E8 D394555A AC6DEC79

Vì khôi phục là một phần, thông điệp đã được khôi phục $M^*$ bao gồm $M_1^*$ và $M_2^*$, phần có thể và không thể khôi phục được.

$M^*$ = FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98

Mã băm còn lại $H''$ được tính bằng các áp dụng hàm băm SHA-1 cho xâu nhị phân có độ dài 1064 (= 64 + 840 + 160), kết quả của việc ghép thêm 64 bit độ dài phần có thể khôi phục được $C'$, 840 bit của phần thông điệp đã được khôi phục $M_1^*$ và 160 bit của mã băm của phần thông điệp không thể khôi phục h($M_2^*$). $H'' = h(C'||M_1^*||h(M_2^*))$

$H''$ = E30A9CB8 F10DC3C8 1897D9E8 D394555A AC6DEC79

Vì hai mã băm *H'* và *H''* là giống nhau, chữ ký *Σ* được chấp nhận.

### E.2 Các ví dụ với số mũ công khai bằng 2

Phụ lục E.2 bao gồm các ví dụ với khóa công khai có số mũ bằng 2.

#### E.2.1 Ví dụ về quá trình tạo khóa

Khóa trong ví dụ có mô-đun $k = 1024$ bit với số mũ công khai $v = 2$. Vì số mũ xác thực công khai *v* là số chẵn, một thừa số nguyên tố bí mật là đồng dư với 3 theo mô-đun 8 và thừa số còn lại là đồng dư với 7 theo mod 8.

*p* = F69AD66B F97E4CCC B4A76FD3 1F43871D C71100CA F9256C3D BE98CC23 BEC06324
A2282D3C CFCAF00B 0E7492C0 1FB19CE5 0F73EEFD 1A08B0AE 6756E7DF 5670D69B

*q* = C41DB9CC D8777062 2BFA8836 1E49ADA2 B5B6CBD0 28479585 472150A1 96C65E89
C2114580 FDE60F6B E12CA9DD A370A3EA 74D33B52 8EB791A9 0FD52818 3D8F612F

Số đồng dư công khai *n* là kết quả của các thừa số nguyên tố bí mật *p* và *q*. Độ dài của nó là 1024 bit.

*n* = BCEB2EB0 2E1C8E99 99BC9603 F8F91DA6 084EA6E7 C75BD18D D0CDBEDB 21DA29F1
9E731125 9DB0D190 B1920186 A8126B58 2D13ABA6 9958763A DA8F79F1 62C7379D
6109D2C9 4AA2E041 B383A74B BF17FFCC 145760AA 8B58BE3C 00C52BA3 BD05A9D0
BE5BA503 E6721FC1 066D37A8 9BF072C9 7BABB26C F6B29633 043DB474 6F9D2175

Số mũ của chữ ký bí mật s bằng nghịch đảo phép nhân của *v* mod $lcm(p-1, q-1)/2$.

*s* = 029FB5FB 55F94917 7777F3DC 7FE703F7 A3ABC251 70FDB83E 6A02DB8A 2794CECE
05C19920 85BEE677 57CCB1CC 8972089A 1D120D0C FB04C8C0 D141FE23 5A42C453
F0883D5E 73742EB5 98435B52 B393B491 F053C59C A8950D48 CA990ADF 888C6DE4
085CEB5D 6B02AEAB BCC2D543 B4C9F995 3FE16572 2F4E0846 9AD92248 D8622DEA

#### E.2.2 Các ví dụ về khôi phục toàn bộ

Ở đây trình bày ba ví dụ về tạo và xác thực chữ ký, mỗi ví dụ tương ứng với một trong ba lược đồ.

#### E.2.2.1 Ví dụ về lược đồ chữ ký 1

Ví dụ này sử dụng hàm băm chuyên dụng 3 trong ISO/IEC 10118-3 (còn được biết đến với tên gọi là SHA-1).

##### E.2.2.1.1 Quá trình ký

Trong hệ thập lục phân, thông điệp *M* là một xâu có độ dài là 48 xâu bộ tám, nghĩa là 384 bit như sau.

*M* = FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210

160 bit mã băm được tính toán bằng cách áp dụng hàm băm chuyên dụng 3 cho 384 bit của *M*.

*H* = 85DCC7FC 51371637 5A059D02 5439FCD9 25C828AC

Hàm băm được sử dụng đã được biết hoàn toàn. Do đó, trường trailer bao gồm 8 bit sau đây.

*T* = BC

Thông điệp là đủ ngắn để khôi phục toàn bộ. 1024 bit của xâu trung gian $S_i$ kết quả của việc nối hai bit của tiêu đề bằng "01", bit dữ liệu thêm được thiết lập bằng '0', 468 (= 1024 – 384 – 160 – 8 – 4) bit đệm bằng '0', bit bao quanh bằng 1, 384 bit của $M_1$ (=*M*), 160 bit của *H* và 8 bit của trường trailer *T*.

Xâu $S_r$ có thể khôi phục kết quả của việc thay thế 116 xâu bộ bốn đệm bằng '0' bằng 116 xâu bộ bốn đệm bằng 'B' và tương tự đối với xâu bộ bốn bao quanh bằng '1' thay thế bằng 'A'.

```
S_r =  4BBBBBBB BBBBBBBB BBBBBBBB BBBBBBBB BBBBBBBB BBBBBBBB BBBBBBBB BBBBBBBB
       BBBBBBBB BBBBBBBB BBBBBBBB BBBBBBBB BBBBBBBB BBBBBBBB BBBBBAFE DCBA9876
       543210FE DCBA9876 543210FE DCBA9876 543210FE DCBA9876 543210FE DCBA9876
       543210FE DCBA9876 54321085 DCC7FC51 3716375A 059D0254 39FCD925 C828ACBC
```

Số nguyên có thể khôi phục $f_r$ là số nguyên dương không dấu biểu diễn $S_r$. Vì ký hiệu Jacobi của $f_r$ theo n bằng 1, kết quả được giữ lại. $I_r$ tăng theo lũy thừa bậc $s$ theo mô-đun $n$. Vì nhỏ hơn n/2 nên kết quả được giữ lại. Xâu nhị phân biểu diễn số nguyên đó dưới dạng số nguyên dương không dấu là chữ ký $\Sigma$.

```
Σ =    0C0C62D3 523F2DA3 972679D0 348D9A50 38E93AE3 D19E97DF 875DCC04 6B2637DB
       CE7D4CCC 5967529A B96D27B6 D9B41F54 56E65EEA 328FDB7D AE6F4E7D A0CFC1CF
       F8AB5A80 CC7C9B9F 487EC2B5 90CBC2F3 1AFDC5CF 9C3478B9 3C46D575 A0E08D21
       D965A9C4 FCAFE356 2D64B1C3 0706AF0D 43288156 DA3FF990 CB040D5C 0863F262
```

Thông điệp đã ký có 128 xâu bộ tám chỉ bao gồm chữ ký vì $M_2$ là rỗng.

**E.2.2.1.2 Quá trình xác thực**

Chữ ký $\Sigma$ là một xâu nhị phân đại diện một số nguyên dương không dấu, nhỏ hơn $n/2$. Số nguyên này tăng theo lũy thừa bậc 2 mô-đun $n$, do đó thu được số nguyên $f_s$.

```
f_s =  712F72F4 7260D2DD DE00DA48 3D3D61EA 4C92EB2C 0BA015D2 1512031F 661E6E35
       E2B75569 E1F515D4 F5D645CA EC56AF9C 7157EFEA DD9CBA7F 1ED3BEF2 860C9F27
       0CD7C1CA 6DE847CB 5F51964C E25D6755 C0254FAB AE9E25C5 AC931AA4 E04B115A
       6A299405 09B7874D B23B2722 BF287678 44957B12 F11593DE CA40DB4E A77474B9
```

Quá trình xác thực không liên quan đến ký hiệu Jacobi. Vì ba bit trọng số nhỏ nhất của số nguyên kết quả $f_s$ bằng "001", $f'_r = n - f_s$.

```
f'_r =  4BBBBBBB BBBBBBBB BBBBBBBB BBBBBBBB BBBBBBBB BBBBBBBB BBBBBBBB BBBBBBBB
        BBBBBBBB BBBBBBBB BBBBBBBB BBBBBBBB BBBBBBBB BBBBBBBB BBBBBAFE DCBA9876
        543210FE DCBA9876 543210FE DCBA9876 543210FE DCBA9876 543210FE DCBA9876
        543210FE DCBA9876 54321085 DCC7FC51 3716375A 059D0254 39FCD925 C828ACBC
```

$f'_r$ được biểu diễn dưới dạng một số nguyên dương không dấu bởi xâu đã được khôi phục $S'_r$.

– Xâu bộ tám bên trái nhất của $S'_r$ bằng "4B"; nó bao gồm tiêu đề bằng "01", bit dữ liệu thêm bằng '0' (khôi phục toàn bộ), một bit đệm bằng '0' và một xâu bộ bốn đệm bằng 'B'; theo sau là 115 xâu bộ bốn bằng 'B' và xâu bộ bốn bao quanh bằng 'A'; 59 xâu bộ tám này được chuyển sang bên trái của $S'_r$.

– Xâu bộ tám bên phải nhất của $S'_r$ bằng "BC"; xâu bộ tám đó cũng được chuyển sang bên phải của $S'_r$.

Vì trường trailer $T$ bằng "BC", hàm băm được sử dụng đã được biết đến hoàn toàn: hàm băm chuyên dụng 3 trong ví dụ này.

Xâu còn lại 544 bit được chia làm hai phần.

- $M_1^*$ bao gồm 384 bit bên trái nhất.
- $H'$ bao gồm 160 bit bên phải nhất.

$M_1^*$ = FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210

$H'$ = 85DCC7FC 51371637 5A059D02 5439FCD9 25C828AC

Thông điệp đã được khôi phục $M^*$ chỉ bao gồm $M_1^*$ vì khôi phục thông điệp là toàn bộ. Mã băm còn lại $H''$ được tính bằng các áp dụng SHA-1 cho $M^*$.

$H''$ = 85DCC7FC 51371637 5A059D02 5439FCD9 25C828AC

Vì hai mã băm $H'$ và $H''$ là giống nhau, chữ ký $\Sigma$ được chấp nhận.

**E.2.2.2 Ví dụ về lược đồ chữ ký 2**

Ví dụ này sử dụng hàm băm chuyên dụng 1 trong ISO/IEC 10118-3 (còn được biết đến với tên gọi là RIPEMD-160).

**E.2.2.2.1 Quá trình ký**

Thông điệp để ký là rỗng, có nghĩa là một xâu nhị phân có độ dài bằng 0.

160 bit của salt $S$ được tạo ra.

$S$ = 61DF870C 4890FE85 D6E3DD87 C3DCE372 3F91DB49

160 bit mã băm được tính toán bằng cách áp dụng hàm băm chuyên dụng 1 để thu được một xâu nhị phân có độ dài là 384 (= 64 + 160 + 160) và kết quả được ghép thêm 64 bit độ dài thông điệp có thể khôi phục được $C$, 160 bit của mã băm của phần không thể khôi phục được $h(M_2)$ và 160 bit của salt $S$. $H = h(C||h(M_2)||S)$.

$H$ = 632E21FD 52D2B95C 5F7023DA 63DE9509 C01F6C7B

Hàm băm được sử dụng đã được biết đến hoàn toàn. Do đó, trường trailer $T$ bao gồm 8 bit sau đây.

$T$ = BC

Thông điệp là rỗng, do đó là khôi phục toàn bộ. 1024 bit của xâu trung gian $S_i$ kết quả của việc nối 695 (= 1024 − 160 − 160 − 8 − 1) bit đệm bằng '0', bit bao quanh bằng 1, 160 bit của $S$, 160 bit của $H$ và 8 bit của trường trailer $T$.

$S_i$ = 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000161 DF870C48 90FE85D6 E3DD87C3 DCE3723F 91DB4963 2E21FD52
D2B95C5F 7023DA63 DE9509C0 1F6C7BBC

Xâu có thể khôi phục $S_r$ thu được từ việc áp dụng hàm tạo mặt nạ *MGF1* đối với 856 (= 1024 − 160 − 8) bit bên trái nhất của $S_i$, và 1 bit bên trái nhất của $S_r$ được thiết lập bằng '0' vì δ =1 (δ= 1 − 1024)$mod$ 8).

$S_r$ = 73FEAF13 EB12914A 43FE6350 22BB4AB8 108A9F3A BD8D8A9E 4AD6C355 EE920359
C7F237AE 36B1212F E947F676 C68FE362 247D27D1 F298CA93 02EB21F4 A64C26CE
44471EF8 C0DFE1A5 4606F0BA 8E63E87C DACA993B FA62973B 567473B4 D38FAF73
AB226600 934A9CC1 D3263E63 2E21FD52 D2B95C5F 7023DA63 DE9509C0 1F6C7BBC

Số nguyên có thể khôi phục $f_r$ là số nguyên dương không dấu biểu diễn $S_r$. Vì ký hiệu Jacobi của $f_r$ theo n bằng -1, số nguyên đại diện là $J = f_r / 2$.

```
J =   39FF5789 F58948A5 21FF31A8 115DA55C 0C45479D 5EC6C54F 256B61AA F74901AC
      E3F91BD7 1B589097 F4A3FB3B 6347F1B1 123E93E8 F94C6549 817590FA 53261367
      22238F7C 606FF0D2 A303785D 4731F43E 6D654C9D FD314B9D AB3A39DA 69C7D739
      D5914300 49A54E60 E9931F31 9710FEA9 695CAE2F B811ED31 EF4A84E0 0FB63DDE
```

$J$ tăng theo lũy thừa bậc $s$ mô-đun $n$. Kết quả như sau.

```
J^s =  B6935ACC DCABB323 D7A7125A CA86B2E6 AF7937DE 4F523629 93B07BF2 895A4677
       50553ECE 92570E7F 975CDB89 D3EC9487 CA626E9B 4E7FD5A4 16ED9C7A 9E619DCF
       DC05A5A9 4089E593 50C9E86B 4DD10E5B DD709150 843D755B 057C99F6 71330258
       E56474B9 6A7A4848 DC1F4100 1603BBAB DBA44AE7 1A6F8211 40137572 67C97D0C
```

Vì kết quả trên lớn hơn $n/2$, chữ ký $\Sigma = n - t$.

```
Σ =    0657D3E3 5170DB75 C21583A9 2E726ABF 58D56F09 78099B64 3D1D42E8 987FE37A
       4E1DD257 0B59C311 1A3525FC D425D6D0 62B13D0B 4AD8A096 C3A1DD76 C46599CD
       85042D20 0A18FAAE 62B9BEE0 7146F170 36E6CF5A 071B48E0 FB4891AD 4BD2A777
       D8F7304A 7BF7D77B 2A4DF6A8 85ECB71D A0076785 DC431421 C42A3F02 07D3A469
```

Thông điệp đã ký có 128 xâu bộ tám chỉ bao gồm chữ ký vì $M_2$ là rỗng.

#### E.2.2.2.2 Quá trình xác thực

Chữ ký $\Sigma$ là một xâu nhị phân đại diện một số nguyên dương không dấu, nhỏ hơn $n/2$. Số nguyên này tăng theo lũy thừa bậc 2 mô-đun $n$, do đó thu được số nguyên $f_s$.

```
f_s =  39FF5789 F58948A5 21FF31A8 115DA55C 0C45479D 5EC6C54F 256B61AA F74901AC
       E3F91BD7 1B589097 F4A3FB3B 6347F1B1 123E93E8 F94C6549 817590FA 53261367
       22238F7C 606FF0D2 A303785D 4731F43E 6D654C9D FD314B9D AB3A39DA 69C7D739
       D5914300 49A54E60 E9931F31 9710FEA9 695CAE2F B811ED31 EF4A84E0 0FB63DDE
```

Quá trình xác thực không liên quan đến ký hiệu Jacobi. Vì ba bit trọng số nhỏ nhất của số nguyên kết quả $f_s$ bằng "100", $f'_r = 2f_s$.

```
f'_r =  73FEAF13 EB12914A 43FE6350 22BB4AB8 188A8F3A BD8D8A9E 4AD6C355 EE920359
        C7F237AE 36B1212F E947F676 C68FE362 247D27D1 F298CA93 02EB21F4 A64C26CE
        44471EF8 C0DFE1A5 4606F0BA BE63E87C DACA993B FA62973B 567473B4 D38FAE73
        AB228600 934A9CC1 D3263E63 2E21FD52 D2B95C5F 7023DA63 DE9509C0 1F6C7BBC
```

$f'_r$ được biểu diễn dưới dạng một số nguyên dương không dấu bởi xâu đã được khôi phục $S'_r$. Hàm tạo mặt nạ *MGF1* áp dụng cho 856 (= 1024 – 160 – 8) bit bên trái nhất của $S'_r$, từ đó thu được xâu được khôi phục tạm thời $S'_i$.

```
S'_i =  00000000 00000000 00000000 00000000 00000000 00000000 00000000
        00000000 00000000 00000000 00000000 00000000 00000000 00000000
        00000000 00000000 00000000 00000000 00000000 00000000 00000000
        00000161 DF870C48 90FE85D6 F3DD87C3 DCE3723F 91DB4963 2E21FD52
        D2B95C5F 7023DA63 DE9509C0 1F6C7BBC
```

$S'_i$ biểu diễn xâu được khôi phục trung gian như sau.

– Bit bên trái nhất của $S'_i$ được thiết lập bằng '0' vì $\delta = 1$ ($\delta = (1 - 1024) mod 8$). 695 bit bên trái nhất của xâu nhị phân còn lại bằng '0'; nó được theo sau bởi bit bao quanh "1"; 87 xâu bộ tám đó được chuyển sang bên trái của $S'_i$.

– Xâu bộ tám bên phải nhất của $S'_i$ bằng "BC"; xâu bộ tám đó cũng được chuyển sang bên phải của $S'_i$.

Vì trailer bằng "BC", hàm băm được sử dụng đã được biết đến hoàn toàn: hàm băm chuyên dụng 1 trong ví dụ này.

Xâu còn lại 320 bit được chia làm hai phần.

– $S^*$ bao gồm 160 bit bên phải nhất.
– $H'$ bao gồm 160 bit bên phải nhất.

$S^*$ = 61DF870C 4890FE85 D6E3DD87 C3DCE372 3F91DB49

$H'$ = 632E21FD 52D2B95C 5F7023DA 63DE9509 C01F6C7B

Thông điệp đã được khôi phục $M^*$ được giả định là rỗng, và do đó khôi phục thông điệp là toàn bộ. Mã băm còn lại $H''$ được tính bằng các áp dụng hàm băm chuyên dụng 1 cho xâu nhị phân có độ dài 384 (=64+160+160), kết quả của việc ghép 64 bit của độ dài thông điệp đã được khôi phục $C'$, 160 bit của mã băm của phần thông điệp không thể khôi phục $h(M_2)$ và 160 bit của salt đã được khôi phục $S^*$. $H'' = h(C'||h(M_2)||S^*)$.

$H''$ = 632E21FD 52D2B95C 5F7023DA 63DE9509 C01F6C7B

Vì hai mã băm $H'$ và $H''$ là giống nhau, chữ ký $\Sigma$ được chấp nhận.

### E.2.2.3 Ví dụ về lược đồ chữ ký 3

Ví dụ này sử dụng hàm băm chuyên dụng 3 trong ISO/IEC 10118-3 (còn được biết đến với tên gọi là SHA-1).

#### E.1.2.3.1 Quá trình ký

Thông điệp để ký là xâu 64 ký tự mã ASCII sau đây.

abcdbcdecdefdefgefghfghighijhijkijkljklmklmnlmnomnopnopqopqrpqrs

Trong hệ thập lục phân, thông điệp $M$ là xâu có độ dài là 64 xâu bộ tám, có nghĩa là 512 bit sau đây.

$M$ = 61626364 62636465 63646566 64656667 65666768 66676869 6768696A 68696A6B
696A6B6C 6A6B6C6D 6B6C6D6E 6C6D6E6F 6D6E6F70 6E6F7071 6F707172 70717273

Vì lược đồ chữ ký này là thuộc kiểu tất định, giá trị salt S có độ dài bằng 0 được lựa chọn.

160 bit mã băm được tính toán bằng cách áp dụng hàm băm chuyên dụng 3 cho xâu nhị phân có độ dài 736 ($= 64 + 512 + 160$), kết quả của việc ghép thêm 64 bit độ dài thông điệp có thể khôi phục được $C$, 512 bit của phần thông điệp có thể khôi phục được $M_1$ và 160 bit của mã băm của phần không thể khôi phục được $h(M_2)$. $H = h(C||M_1||h(M_2))$.

$H$ = D74009C4 638462E6 9D5923E7 433AEC02 8B9A90E6

Định danh trong trường trailer xác định hàm băm được sử dụng, ISO/IEC thiết lập định danh cho hàm băm chuyên dụng 3 giá trị "33". Do đó, trường trailer *T* bao gồm 16 bit sau đây.

*T* = 33CC

Thông điệp là đủ ngắn để khôi phục toàn bộ. 1024 bit của xâu trung gian $S_i$ kết quả của việc nối 335 (= 1024 – 512 – 160 – 16 – 1) bit đệm bằng '0', bit bao quanh bằng 1, 512 bit của $M_1(=M)$, 160 bit của S, 160 bit của *H* và 16 bit của trường trailer *T*.

$S_i$ = 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00016162 63646263 64656364 65666465
66676566 67686667 68696768 696A6869 6A6B696A 6B6C6A6B 6C6D6B6C
6D6E6C6D 6E6F6D6E 6F706E6F 70716F70 71727071 7273D740 09C46384
62E69D59 23E7433A EC028B9A 90E633CC

Xâu có thể khôi phục $S_r$ thu được từ việc áp dụng hàm tạo mặt nạ *MGF1* đối với 848 (= 1024 – 160 – 16) bit bên trái nhất của $S_i$, và 1 bit bên trái nhất của $S_r$ được thiết lập bằng '0' vì δ =1 (δ= 1 – 1024)$mod$ 8).

$S_r$ = 296B0622 4010E1EC 230D4560 A5F88F03 550AAFCE 31C805CE 81E811E5 E53E5F71
AE64FC2A 2A486B19 3E87972D 90C54B80 7A862F21 A21919A4 3ECF0672 40A8C8C6
41DE8DCD F1942CF7 90D13672 8FFC0D98 FB906E79 39C1EC0E 64C0E067 F0A7443D
6170E411 DF91F797 D1FFD740 09C46384 62E69D59 23E7433A EC028B9A 90E633CC

Số nguyên có thể khôi phục $f_r$ là số nguyên dương không dấu biểu diễn $S_r$. Vì ký hiệu Jacobi của $f_r$ ttheo n bằng -1, số nguyên đại diện là $J = f_r/2$.

*J* = 14B58311 200870F6 1186A2B0 52FC4781 AA8557E7 18E402E7 40F408F2 F29F2FB8
D7327E15 1524358C 9F43CB96 C862A5C0 3D431790 D10C8CD2 1F678339 20546463
20EF46E6 F8CA167B C8689B39 47FE06CC 7DC8373C 9CE0F607 32607033 F853A21E
B0B87208 EFC8FBCB E8FFEBA0 04E231C2 31734EAC 91F3A19D 760145CD 487319E6

*J* tăng theo lũy thừa bậc *s* mô-đun *n*. Vì kết quả nhỏ hơn $n/2$, chữ ký $\Sigma = J^s$.

Σ = 4F9FE3FA 21E8EAE7 786363CC D14D0AE6 401174BC B94AFBE8 3E24D014 4CB8CDF1
075E4D92 F4E08091 7DFC3C66 3A65457A 3178F280 DFF7E16C A9D29BCD B18AE2AE
C483A97F 2EF1FB4C 7BBFA1D1 269BFAF5 245C27DA E6DF3531 CADEE605 74A97378
21454089 91530D1B F8AED104 CB95149B 28E552DB 1A611286 2C099D7A 442A462B

Thông điệp đã ký có 128 xâu bộ tám chỉ bao gồm chữ ký vì $M_2$ là rỗng.

### E.2.2.3.2 Quá trình xác thực

Chữ ký Σ là một xâu nhị phân đại diện một số nguyên dương không dấu, nhỏ hơn $n/2$. Số nguyên này tăng theo lũy thừa bậc 2 mô-đun *n*, do đó thu được số nguyên $f_s$.

$f_s$ = 14B58311 200870F6 1186A2B0 52FC4781 AA8557E7 18E402E7 40F408F2 F29F2FB8
D7327E15 1524358C 9F43CB96 C862A5C0 3D431790 D10C8CD2 1F678339 20546463
20EF46E6 F8CA167B C8689B39 47FE06CC 7DC8373C 9CE0F607 32607033 F853A21E
B0B87208 EFC8FBCB E8FFEBA0 04E231C2 31734EAC 91F3A19D 760145CD 487319E6

Quá trình xác thực không liên quan đến ký hiệu Jacobi. Vì ba bit trọng số nhỏ nhất của số nguyên kết quả $f_s$ bằng "110", $f'_r = 2f_s$.

$f'$ = 296B0622 4010E1EC 230D4560 A5F88F03 550AAFCE 31C805CE 81E811E5 E53E5F71
AB64FC2A 2A486B19 3E87972D 90C54B80 7A862F21 A21919A4 3ECF0672 40ABC8C6
41DE8DCD F1942CF7 90D13672 8FFC0D98 FB906E79 39C1EC0E 64C0E067 F0A7443D
6170E411 DF91F797 D1FFD740 09C46384 62E69D59 23E7433A EC028B9A 90E633CC

$f'_r$ được biểu diễn dưới dạng một số nguyên dương không dấu bởi xâu đã được khôi phục $S'_r$. Hàm tạo mặt nạ *MGF1* áp dụng cho 848 (= 1024 – 160 – 16) bit bên trái nhất của $S'_r$, từ đó thu được xâu được khôi phục trung gian $S'_i$.

$S'_i$ = 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00016162 63646263 64656364 65666465
66676566 67686667 68696768 696A6869 6A6B696A 6B6C6A6B 6C6D6B6C
6D6E6C6D 6E6F6D6E 6F706E6F 70716F70 71727071 7273D740 09C46384
62E69D59 23E7433A EC028B9A 90E633CC

$S'_i$ biểu diễn xâu được khôi phục trung gian như sau.

– Bit bên trái nhất của $S'_i$ được thiết lập bằng '0' vì $\delta = 1$ ($\delta = (1 - 1024) mod 8$). 335 xâu bộ tám bên trái nhất của xâu nhị phân còn lại bằng '0'; nó được theo sau bởi bit bao quanh '1'; 42 xâu bộ tám đó được chuyển sang bên trái của $S'_i$.

– Xâu bộ tám bên phải nhất của $S'_i$ bằng "CC"; do đó, trailer bao gồm hai xâu bộ tám bằng "33CC"; hai xâu bộ tám đó cũng được chuyển sang bên phải của $S'_i$.

Định danh hàm băm bằng "33"; do đó hàm băm được sử dụng là hàm băm chuyên dụng 3.

Xâu còn lại 672 bit được chia làm hai phần.

– $M_1$* bao gồm 512 bit bên trái nhất.

– *H'* bao gồm 160 bit bên phải nhất.

$M_1$* = 61626364 62636465 63646566 64656667 65666768 66676869 6768696A 68696A6B
696A6B6C 6A6B6C6D 6B6C6D6E 6C6D6E6F 6D6E6F70 6E6F7071 6F707172 70717273

*H'* = D74009C4 638462E6 9D5923E7 433AEC02 8B9A90E6

Thông điệp đã được khôi phục *M** chỉ bao gồm $M_1$* và do đó, khôi phục là toàn bộ. Mã băm khác *H''* được tính bằng cách áp dụng SHA-1 cho xâu nhị phân có độ dài 736 (= 64 + 512 + 160), kết quả của việc ghép thêm 64 bit của độ dài thông điệp đã được khôi phục *C'*, 512 bit của phần thông điệp đã được khôi phục $M_1$* và 160 bit của mã băm của phần thông điệp không thể khôi phục được (rỗng) $h(M_2^*)$. $H'' = h(C'||M_1^*||h(M_2^*))$.

*H''* = D74009C4 638462E6 9D5923E7 433AEC02 8B9A90E6

Vì hai mã băm *H'* và *H''* là giống nhau, chữ ký $\Sigma$ được chấp nhận.

### E.2.3 Các ví dụ về khôi phục một phần

Ở đây trình bày ba ví dụ về tạo và xác thực chữ ký, mỗi ví dụ tương ứng với một trong ba lược đồ.

#### E.2.3.1 Ví dụ về lược đồ chữ ký 1

Ví dụ này sử dụng hàm băm chuyên dụng 3 trong ISO/IEC 10118-3 (còn được biết đến với tên gọi là SHA-1).

##### E.2.3.1.1 Quá trình ký

Thông điệp để ký là xâu 112 ký tự mã ASCII sau đây.

```
abcdbcdecdefdefgefghfghighijhijkijkljklmklmnlmnomnopnopq
opqrpqrsqrstrstustuvtuvwuvwxvwxywxyzxyzayzabzabcabcdbcde
```

Trong hệ thập lục phân, thông điệp *M* là xâu có độ dài 112 xâu bộ tám, có nghĩa là 896 bit sau đây.

```
M =   61626364 62636465 63646566 64656667 65666768 66676869 6768696A 68696A6B
      696A6B6C 6A6B6C6D 6B6C6D6E 6C6D6E6F 6D6E6F70 6E6F7071 6F707172 70717273
      71727374 72737475 73747576 74757677 75767778 76777879 7778797A 78797A61
      797A6162 7A616263 61626364 62636465
```

160 bit mã băm được tính toán bằng cách áp dụng SHA- 1 cho 896 bit của *M*.

```
H =   1CF7A997 4518B555 C1802CB8 10A23C27 4FCFAA73
```

Định danh trong trường trailer xác định hàm băm được sử dụng; ISO/IEC thiết lập định danh cho hàm băm chuyên dụng 3 giá trị "33". Do đó, trường trailer *T* bao gồm 16 bit sau đây.

```
T =   33CC
```

Thông điệp là quá dài để có thể được khôi phục hoàn toàn bởi quá trình xác thực. Do đó, nó được chia làm hai phần.

- $M_1$ bao gồm 840 bit bên trái nhất.
- $M_2$ bao gồm 56 bit còn lại, có nghĩa là 7 xâu bộ tám.

```
M1 =  61626364 62636465 63646566 64656667 65666768 66676869 6768696A 68696A6B
      696A6B6C 6A6B6C6D 6B6C6D6E 6C6D6E6F 6D6E6F70 6E6F7071 6F707172 70717273
      71727374 72737475 73747576 74757677 75767778 76777879 7778797A 78797A61
      797A6162 7A616263 61
M2 =  626364 62636465
```

1024 bit của xâu trung gian $S_i$ kết quả của việc ghép thêm hai bit tiêu đề bằng "01", bit dữ liệu thêm bằng '1', bốn (= 1024 – 840 – 160 – 16 – 4) bit đệm bằng '0', bit bao quanh bằng 1, 840 bit của $M_1$, 160 bit của *H* và 16 bit của trường trailer *T*. Xâu có thể khôi phục $S_r$ kết quả của việc xâu bộ bốn bao quanh bằng 1 được thay thế bằng 'A'.

```
Sr =  6A616263 64626364 65636465 66646566 67656667 68666768 69676869 6A68696A
      6B696A6B 6C6A6B6C 6D6B6C6D 6E6C6D6E 6F6D6E6F 706E6F70 716F7071 72707172
      73717273 74727374 75737475 76747576 77757677 78767778 79777879 7A787A7A
      61797A61 627A6162 63611CF7 A9974518 E555C180 2CB810A2 3C274FCF AA7333CC
```

Số nguyên có thể khôi phục $f_r$ là số nguyên dương không dấu biểu diễn $S_r$. Vì ký hiệu Jacobi của $f_r$ theo n bằng -1, số nguyên đại diện là $J = f_r/2$.

```
J =   3530B131 B23131B2 32B1B232 B332B2B3 33B2B333 B43333B4 34B3B434 B5343435
      35B4B535 B6353536 36B5B636 B7363637 37B6B737 B8373738 38B7B838 B9383839
      39B8B939 BA39393A 3AB9BA3A BB3A3ABB 3BBABB3B BC3B3BBC 3CBBBC3C BD3C3CBD
      3DBCBD3D B13D3DB1 31B08E7B D4C9A28C 72AAE0C0 165C0851 1E13A7E7 D53999E6
```

*J* tăng theo lũy thừa bậc *s* mô-đun *n*. Kết quả như sau.

$S^s$ = AD83029D FB27EC14 E5F8FDC0 8E10481F 35CB879F 62BC2180 A17D84DE 9C65FF91
729DEAC0 6848885A AC99863F 0B937046 2FF8C5ED 33DA98CB E75D54BA 59CC12BF
E9AF94B7 80E31542 A96FD25A 4B1AD996 0032373A 208D4965 0870EEB0 A771F302
74B08389 95F0B131 F0CE526F 679B1618 A1BCAAAB 45AE4669 421339D3 6398C111

Vì kết quả trên lớn hơn $n/2$, chữ ký $\Sigma = n - J^s$.

$\Sigma$ = 0F682C12 32F4A2B4 B3C39843 6AE8D596 D2B31F48 649FB00D 2F5039FC 85742A60
2BB52665 356B4936 04F87B47 9C7EFB11 FD1AB5B9 657DDD6E F3322537 08FB24DD
7F5A3E11 C9BFCAFF 0A13D4F1 73FD2636 14252970 6ACB74D6 F8543CF3 1593B6CE
49AB217A 50816E92 159EE539 34555CB0 D9EF07C1 B1044FC9 C22A7AA1 0C046064

Thông điệp đã ký có 128 xâu bộ tám chữ ký $\Sigma$ cùng với 7 xâu bộ tám của thông điệp không thể khôi phục được $M_2$, có nghĩa là chỉ nhiều hơn 23 xâu bộ tám so với thông điệp $M$.

**E.2.3.1.2 Quá trình xác thực**

Chữ ký $\Sigma$ là một xâu nhị phân đại diện một số nguyên dương không dấu, nhỏ hơn $n/2$. Số nguyên này tăng theo lũy thừa bậc 2 mô-đun $n$, do đó thu được số nguyên $f_s$.

$f_s$ = 3530B131 B23131B2 32B1B232 B33232B3 33B2B333 B43333B4 34B3B434 B53434B5
35B4B535 B63535B6 36B5B636 B73636B7 37B6B737 B83737B8 38B7B838 B93838B9
39B8B939 BA3939BA 3AB9BA3A BB3A3ABB 3BBABB3B BC3B3BBC 3CBBBC3C BD3C3CBD
3CBCBD3C B13D3CB1 31B0BE7B D4CBA28C 72AAE0C0 165C0851 1E13A7E7 D53999E6

Quá trình xác thực không liên quan đến ký hiệu Jacobi. Vì ba bit trọng số nhỏ nhất của số nguyên kết quả $f_s$ bằng "110", $f'_r = 2f_s$.

$f'_r$ = 6A616263 64626364 65636465 66646566 67656667 68666768 69676869 6A686968
6B696A6B 6C6A6B6C 6D6B6C6D 6E6C6D6E 6F6D6E6F 706E6F70 716F7071 72707172
73717273 74727374 75737475 76747576 77757677 78767778 79777879 7A787A7A
61797A61 627A6162 63611CF7 A9974518 E555C180 2CB810A2 3C274FCF AA7333CC

$f'_r$ được biểu diễn dưới dạng một số nguyên dương không dấu bởi xâu đã được khôi phục $S'_r$.

– Xâu bộ tám bên trái nhất của $S'_r$ bằng "6A"; nó bao gồm tiêu đề bằng "01", bit dữ liệu thêm bằng '1' (khôi phục một phần), một bit đệm bằng '0' và một xâu bộ bốn đệm bằng 'A'; xâu bộ tám này được chuyển sang bên trái của $S'_r$.

– Xâu bộ tám bên phải nhất của $S'_r$ bằng "CC"; do đó trailer bao gồm hai xâu bộ tám bằng "33CC"; hai xâu bộ tám đó cũng được chuyển sang bên phải của $S'_r$.

Định danh hàm băm là "33"; do đó, hàm băm được sử dụng là hàm băm chuyên dụng 3.

Xâu còn lại 1000 bit được chia làm hai phần.

– $M_1^*$ bao gồm 840 bit bên trái nhất.

– $H'$ bao gồm 160 bit bên phải nhất.

$M_1^*$ = 61626364 62636465 63646566 64656667 65666768 66676869 6768696A 68696A6B
696A6B6C 6A6B6C6D 6B6C6D6E 6C6D6E6F 6D6E6F70 6E6F7071 6F707172 70717273
71727374 72737475 73747576 74757677 75767778 76777879 7778797A 78797A61
797A6162 7A616263 61

$H'$ = 1CF7A997 4518E555 C1802CB8 10A23C27 4FCFAA73

Vì khôi phục là một phần, thông điệp đã được khôi phục $M^*$ bao gồm $M_1^*$ và $M_2^*$, phần có thể và không thể khôi phục được.

$M^*$ = 61626364 62636465 63646566 64656667 65666768 66676869 6768696A 68696A6B
696A6B6C 6A6B6C6D 6B6C6D6E 6C6D6E6F 6D6E6F70 6E6F7071 6F707172 70717273
71727374 72737475 73747576 74757677 75767778 76777879 7778797A 78797A61
797A6162 7A616263 61626364 62636465

Mã băm còn lại $H''$ được tính bằng các áp dụng hàm băm chuyên dụng 3 cho $M^*$.

$H''$ = 1CF7A997 4518E555 C1B02CB8 10A23C27 4FCFAA73

Vì hai mã băm $H'$ và $H''$ là giống nhau, chữ ký $\Sigma$ được chấp nhận.

### E.2.3.2 Ví dụ về lược đồ chữ ký 2

Ví dụ này sử dụng hàm băm chuyên dụng 1 trong ISO/IEC 10118-3 (còn được biết đến với tên gọi là RIPEMD-160).

#### E.2.3.2.1 Quá trình ký

Ví dụ này mô tả chữ ký của một thông điệp có độ dài 132 xâu bộ tám, có nghĩa là 1056 bit sau đây.

$M$ = FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98

160 bit của salt $S$ được tạo ra.

$S$ = 78E29320 3CBA1B7F 92F05F4D 171FF8CA 3E7385F8

Thông điệp là quá dài để có thể được khôi phục hoàn toàn bởi quá trình xác thực. Do đó, nó được chia làm hai phần.

- $M_1$ bao gồm 680 bit bên trái nhất.
- $M_2$ bao gồm 376 bit còn lại, có nghĩa là 47 xâu bộ tám.

FEDCBA98

160 bit của mã băm được tính bằng cách áp dụng hàm băm chuyên dụng 1 với xâu nhị phân có độ dài 1064 (= 64 + 680 + 160 + 160), kết quả của việc ghép 64 bit của độ dài phần không thể khôi phục được C, 680 bit của phần thông điệp có thể khôi phục được $M_1$, 160 bit của mã băm của phần không thể khôi phục được $h(M_2)$ và 160 bit của salt $S$. $H = h(C||M_1||h(M_2)||S)$.

$H$ = A4B517F2 E820B81F 26BCE7C6 6F48A2DB 12A8F3D7

Định danh trong trailer xác định hàm băm được sử dụng; ISO/IEC thiết lập định danh cho hàm băm chuyên dụng 1 giá trị "31". Do đó, trường trailer $T$ bao gồm 16 bit sau đây.

$T$ = 31CC

1024 bit của xâu trung gian $S_i$ kết quả của việc ghép thêm bảy (= 1024 – 680 – 160 – 160 – 16 – 1) bit đệm bằng '0', bit bao quanh bằng 1, 680 bit của $M_1$, 160 bit của $L$, 160 bit của $H$ và 16 bit của trường trailer $T$.

$S_i$ = 01FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432
10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432
10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 987678E2 93203CBA 1B7F92F0
5F4D171F F8CA3E73 8FF8A4B5 17F2E820 B81F26BC E7C66F48 A2DB12A8 F3D731CC

Xâu có thể khôi phục $S_r$ thu được từ việc áp dụng hàm tạo mặt nạ *MGF1* đối với 848 (= 1024 – 160 – 16) bit bên trái nhất của $S_i$, và 1 bit bên trái nhất của $S_r$ được thiết lập bằng '0' vì δ =1 (δ= 1 – 1024)$mod$ 8).

$S_r$ = 01402B29 ABA10407 9677CE7F C3D5A84D B24494D6 F9508B45 964B4F5B 3CC7E8AF
CC4DDE70 81F21CAE 9D4F94D6 D2CCCB43 FCEDA098 8FFD4EF2 EAE72CFD EB4A2630
F0A34A0C 49664CD9 DB723315 759D7588 36C8BA26 AC4348B6 6958AC94 AECB5A75
195B57AB FB9971E2 1337A4B5 17F2E820 B81F26BC E7C66F48 A2DB12A8 F3D731CC

Số nguyên có thể khôi phục $f_r$ là số nguyên dương không dấu biểu diễn $S_r$. Vì ký hiệu Jacobi của $f_r$ theo n bằng -1, số nguyên đại diện là $J = f_r/2$.

$J$ = 00A01594 D5D08203 CB3BE73F E1EAD426 D9224A6B 7CA845A2 CB2427AD 9E63F457
E626EF38 40F90E57 4EA7CA6B 696665A1 FE76D04C 47FEA779 7573967E F5A5131C
7851A506 24B3266C EDB9198A BACEBAC4 1B645D13 5621A45B 34AC564A 5705AD3A
BCADABD5 FDCCB8F1 099BD25A 8BF97410 5C0F935E 73E337A4 516D8954 79EB98E6

$J$ tăng theo lũy thừa bậc $s$ mô-đun $n$. Kết quả như sau.

$J^s$ = 66313F1F BCE72AD5 7D32353B DAF0D50C 3915C837 D12F5C46 DFC76A59 557D27F9
B41CF256 94EB6105 6E029BD5 C4F91C20 D687FAC0 26BA47EF 05AD83DC FE5CF985
EF6B1B91 282460B8 77A8C111 2628F25A 519EF21E E2C1EB0F 019CE73C 747F435C
6DB21E45 28A96AB9 76289AB7 99D32256 4167D935 5C3F3D7F 84AE87C2 2AA23A5A

Vì kết quả trên lớn hơn $n/2$, chữ ký $\Sigma = n - f^s$.

$\Sigma$ = 56F9EF90 713563C4 1C8A60C8 1E084899 CF38DEAF F62C7546 F1065481 CC5D01F7
EA561ECF 08C5708B 438F65B0 E3194F37 568BB0E6 729E2E4B D4E1F614 646A3E17
719EB738 227E7F89 3BDAE63A 98EF0D71 C2B86E8B A896D32C FF284467 48866674
50A986BE BDC8B50A 90449CF1 021D5073 3A43D937 9A7358B3 7F8F2CB2 44FAE71B

Thông điệp đã ký có 128 xâu bộ tám chữ ký $\Sigma$ cùng với 47 xâu bộ tám của thông điệp không thể khôi phục được $M_2$, có nghĩa là chỉ nhiều hơn 43 xâu bộ tám so với thông điệp $M$.

**E.2.3.2.2 Quá trình xác thực**

Chữ ký $\Sigma$ là một xâu nhị phân đại diện một số nguyên dương không dấu, nhỏ hơn $n/2$. Số nguyên này tăng theo lũy thừa bậc 2 mô-đun $n$, do đó thu được số nguyên $f_s$.

$f_s$ = BC4B191B 584C0C95 CE80AEC4 170E497F 2F2C5C7C 4AB38BEB 05A9972D 83763599
B84C21ED 5CB7C339 62EA371B 3EAC05B6 2E9CDB5A 5159CEC1 651BE372 6D222480
E8B82DC3 25EFB9D4 C5CA8DC1 04494507 F8F30397 353719E0 CC18D559 65FFFC96
31ADF92D E8A566D2 FCD1654E 0FF6FEB9 1F9C1F0E 82CF5E8E B2D02B1F F5B1888F

Quá trình xác thực không liên quan đến ký hiệu Jacobi. Vì ba bit trọng số nhỏ nhất của số nguyên kết quả $f_s$ bằng "111", $f'_r = 2(n - f_s)$.

$f_r'$ = 01402B29 ABA10407 9677CE7F C3D5A84D B24494D6 F9508B45 96484F5B 3CC7E8AF
CC4DDE70 81F21CAE 9D4F94D6 D2CCCB43 FCEDA098 8FFD4EF2 EAE72CFD EB4A2638
F0A34A0C 49664CD9 DB723315 759D7588 36C8BA26 AC4348B6 6958AC94 AE0B5A75
195B57AB FB9971E2 1337A4B5 17F2E820 B81F26BC E7C66F48 A2DB12A8 F3D731CC

$f'_r$ được biểu diễn dưới dạng một số nguyên dương không dấu bởi xâu đã được khôi phục $S'_r$. Hàm tạo mặt nạ *MGF1* đối với 848 (= 1024 – 160 – 16) bit bên trái nhất của $S'_r$ , và thu được xâu đã được khôi phục trung gian $S'_i$.

$S_i'$ = 01FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432
10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432
10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 987678E2 93203CBA 1B7F92F0
5F4D171F F8CA3E73 BFF8A4B5 17F2E820 B81F26BC E7C66F48 A2DB12A8 F3D731CC

$S'_i$ biểu diễn xâu trung gian đã được khôi phục như sau.

– Bit bên trái nhất của $S'_i$ được thiết lập bằng '0' vì ō =1 (ō= $1 - 1024) mod 8$). 7 bit bên trái nhất của xâu còn lại bằng '0'; nó được theo sau bởi bit bao quanh '1'; xâu bộ tám này được chuyển sang bên trái của $S'_i$.

– Xâu bộ tám bên phải nhất của $S'_i$ bằng "CC"; do đó trailer bao gồm hai xâu bộ tám bằng "31CC"; hai xâu bộ tám đó cũng được chuyển sang bên phải của $S'_i$.

Định danh hàm băm bằng "31"; do đó hàm băm được sử dụng là hàm băm chuyên dụng 1.

Xâu còn lại 1000 bit được chia làm ba phần.

– $M_1^*$ bao gồm 680 bit bên trái nhất.
– $S^*$ bao gồm 160 bit bên phải nhất.
– $H'$ bao gồm 160 bit bên phải nhất.

$M_1^*$ = FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76

$S^*$ = 78E29320 3CBA1B7F 92F05F4D 171FF8CA 3E738FF8

$H'$ = A4B517F2 E820B81F 26BCE7C6 6F48A2DB 12A8F3D7

Vì khôi phục là một phần, thông điệp đã được khôi phục $M^*$ bao gồm $M_1^*$ và $M_2^*$, phần có thể và không thể khôi phục được.

$M^*$ = FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98

Mã băm còn lại $H''$ được tính bằng các áp dụng hàm băm chuyên dụng 1 cho xâu nhị phân có độ dài 1064 (= 64 + 680 + 160 + 160), kết quả của việc ghép thêm 64 bit độ dài phần có thể khôi phục được $C'$, 680 bit của phần thông điệp đã được khôi phục $M_1^*$, 160 bit của mã băm của phần thông điệp không thể khôi phục h($M_2^*$) và 160 bit của salt đã được khôi phục S*. $H'' = h(C'||M_1^*||h(M_2^*)||S^*)$

*H''*= A4B517F2 E820B81F 26BCE7C6 6F48A2DB 12A8F3D7

Vì hai mã băm *H'* và *H''* là giống nhau, chữ ký *Σ* được chấp nhận.

### E.2.3.3 Ví dụ về lược đồ chữ ký 3

Ví dụ này sử dụng hàm băm chuyên dụng 1 trong ISO/IEC 10118-3 (còn được biết đến với tên gọi là RIPEMD-160).

#### E.2.3.3.1 Quá trình ký

Thông điệp để ký là xâu 112 ký tự mã ASCII sau đây.

abcdbcdecdefdefgefghfghighijhijkijkljklmklmnlmnomnopnopq
opqrpqrsqrstrstustuvtuvwuvwxvwxywxyzxyzayzabzabcabcdbcde

Trong hệ thập lục phân, thông điệp *M* là xâu có độ dài 112 xâu bộ tám, có nghĩa là 896 bit sau đây.

*M* = 61626364 62636465 63646566 64656667 65666768 66676869 6768696A 68696A6B
696A6B6C 6A6B6C6D 6B6C6D6E 6C6D6E6F 6D6E6F70 6E6F7071 6F707172 70717273
71727374 72737475 73747576 74757677 75767778 76777879 7778797A 78797A61
797A6162 7A616263 61626364 62636465

Vì lược đồ chữ ký này là thuộc kiểu tất định, một giá trị salt *S* có độ dài bằng 0 được lựa chọn.

Thông điệp là quá dài để có thể được khôi phục hoàn toàn bởi quá trình xác thực. Do đó, nó được chia làm hai phần.

- $M_1$ bao gồm 848 bit bên trái nhất.
- $M_2$ bao gồm 48 bit còn lại, có nghĩa là 6 xâu bộ tám.

$M_1$ = 61626364 62636465 63646566 64656667 65666768 66676869 6768696A 68696A6B
696A6B6C 6A6B6C6D 6B6C6D6E 6C6D6E6F 6D6E6F70 6E6F7071 6F707172 70717273
71727374 72737475 73747576 74757677 75767778 76777879 7778797A 78797A61
797A6162 7A616263 6162

$M_2$ = 6364 62636465

160 bit của mã băm *H* được tính bằng cách áp dụng hàm băm chuyên dụng 1 với xâu nhị phân có độ dài 1072 (= 64 + 848 + 160), kết quả của việc ghép 64 bit của độ dài phần không thể khôi phục được C, 848 bit của phần thông điệp có thể khôi phục được $M_1$, 160 bit của mã băm của phần không thể khôi phục được $h(M_2)$. $H = h(C||M_1||h(M_2))$.

*H* = 15F000AC 58EE3FFF 144845E7 71907C0C 83324A6F

Hàm băm được sử dụng đã được biết đến hoàn toàn. Do đó, trường trailer *T* bao gồm 8 bit sau đây.

*T* = BC

1024 bit của xâu trung gian $S_i$ kết quả của việc ghép thêm bảy (= 1024 − 848 − 160 − 8 − 1) bit đệm bằng '0', bit bao quanh bằng 1, 848 bit của phần có thể khôi phục được $M_1$, 160 bit của mã băm của phần thông điệp không thể khôi phục được $h(M_2)$ và 8 bit của trường trailer *T*.

$S_i$ = 01616263 64626364 65636465 66646566 67656667 68666768 69676869 6A68696A
6B696A6B 6C6A6B6C 6D6B6C6D 6E6C6D6E 6F6D6E6F 706E6F70 716F7071 72707172
73717273 74727374 75737475 76747576 77757677 78767778 79777879 7A78797A
61797A61 627A6162 63616215 F000AC58 EE3FFF14 4845E771 907C0C83 324A6FBC

Xâu có thể khôi phục $S_r$ thu được từ việc áp dụng hàm tạo mặt nạ *MGF1* đối với 856 (= 1024 – 160 – 8) bit bên trái nhất của $S_i$, và 1 bit bên trái nhất của $S_r$ được thiết lập bằng '0' vì δ =1 (δ= 1 – 1024)$mod$ 8).

$S_r$ = 6F2BB975 71FE2EF2 05B66000 E9DD0665 6655C197 7F374E86 66D63655 6A5FEEEE
AF645555 B25F4556 7C4EE534 1F96FED8 6508C90A 9E3F11B2 6E8D4961 39ED3E55
ECE42860 A6FB3A08 17DAFBF1 3019D93E 1D382DA0 7264FE99 D9797D2F 0B777935
7CA7E74E E440D885 5B7DDF15 F000AC58 EE3FFF14 4845E771 907C0C83 324A6FBC

Số nguyên có thể khôi phục $f_r$ là số nguyên dương không dấu biểu diễn $S_r$. Vì ký hiệu Jacobi của $f_r$ theo n bằng 1, kết quả được giữ nguyên. $f_r$ tăng theo lũy thừa bậc *s* theo mô-đun *n*. Kết quả được biểu diễn bởi một số nguyên dương không dấu tạm thời *t*.

*t* = A1B2C623 971FD3F2 83DE75E1 6A69EA07 262F2A0E DB41B5D5 E048DEB2 15DDCA0D
37A04E1E 87372B77 740CE252 3EF1EE7F F68D4781 819F797E DA8AD7AE F4D7EB7B
C7CED9AE 937FC765 452BABF2 56C9DE4D CEE7C866 C285AA58 09674464 BDD4708F
F37BC8F1 753C7E84 9D76EB1D 522F7FB9 6ABCC26F 591DF88A 33B736C2 82690912

Vì kết quả trên lớn hơn n/2, nó được thay thế bởi phần dư của nó với n. Xâu nhị phân biểu diễn số nguyên này dưới dạng một số nguyên dương không dấu là chữ ký $\Sigma = n - t$.

$\Sigma$ = 1B38688C 96FCBAA7 15DE2022 8E8F339E E21F7CD8 EC1A1BB7 F084E029 0BFC5FE4
66D2C307 1679A619 3D851F34 69207CD8 36866425 17B8FCBC 0004A242 6DEF4C21
993BC91A B7231BDC 6E57FB59 684E217E 456F9843 C8D313E3 F75DE73E FF313940
CADFDC12 7135A13F 68F64C8B 49C0F310 10EEEFFD 9D949DA8 D0867DB1 ED341863

Thông điệp đã ký có 128 xâu bộ tám chữ ký $\Sigma$ cùng với 6 xâu bộ tám của thông điệp không thể khôi phục được $M_2$, có nghĩa là chỉ nhiều hơn 22 xâu bộ tám so với thông điệp *M*.

**E.2.3.3.2 Quá trình xác thực**

Chữ ký $\Sigma$ là một xâu nhị phân đại diện một số nguyên dương không dấu, nhỏ hơn $n/2$. Số nguyên này tăng theo lũy thừa bậc 2 mô-đun *n*, do đó thu được số nguyên $f_s$.

$f_s$ = 6F2BB975 71FE2EF2 05B66000 E9DD0665 6655C197 7F374E86 66D63655 6A5FEEEE
AF645555 B25F4556 7C4EE534 1F96FED8 6508C90A 9E3F11B2 6E8D4961 39ED3E55
ECE42860 A6FB3A08 17DAFBF1 3019D93E 1D382DA0 7264FE99 D9797D2F 0B777935
7CA7E74E E440D885 5B7DDF15 F000AC58 EE3FFF14 4845E771 907C0C83 324A6FBC

Quá trình xác thực không liên quan đến ký hiệu Jacobi. Vì ba bit trọng số nhỏ nhất của số nguyên kết quả $f_s$ bằng "100", $f'_r = f_s$.

$f'_r$ được biểu diễn dưới dạng một số nguyên dương không dấu bởi xâu đã được khôi phục $S'_r$. Hàm tạo mặt nạ *MGF1* đối với 856 (= 1024 – 160 – 8) bit bên trái nhất của $S'_r$ , và thu được xâu đã được khôi phục trung gian $S'_i$.

$S'_i$ = 01616263 64626364 65636465 66646566 67656667 68666768 69676869 6A68696A
6B696A6B 6C6A6B6C 6D6B6C6D 6E6C6D6E 6F6D6E6F 706E6F70 716F7071 72707172
73717273 74727374 75737475 76747576 77757677 78767778 79777879 7A78797A
61797A61 627A6162 63616215 F000AC58 EE3FFF14 4845E771 907C0C83 324A6FBC

$S'_i$ biểu diễn xâu trung gian đã được khôi phục như sau.

– Bit bên trái nhất của $S'_i$ được thiết lập bằng '0' vì δ =1 (δ= $1 - 1024) mod 8$). 7 bit bên trái nhất của xâu còn lại bằng '0'; nó được theo sau bởi bit bao quanh '1'; xâu bộ tám này được chuyển sang bên trái của $S'_i$.

– Xâu bộ tám bên phải nhất của $S'_i$ bằng "BC"; xâu bộ tám đó cũng được chuyển sang bên phải của $S'_i$.

Vì trailer bằng "BC", hàm băm được sử dụng đã được biết đến hoàn toàn: RIPEMD-160 trong ví dụ này.

Xâu còn lại 1008 bit được chia làm hai phần.

– $M_1^*$ bao gồm 848 bit bên trái nhất.

– $H'$ bao gồm 160 bit bên phải nhất.

$M_1^*$ = 61626364 62636465 63646566 64656667 65666768 66676869 6768696A 68696A6B
696A6B6C 6A6B6C6D 6B6C6D6E 6C6D6E6F 6D6E6F70 6E6F7071 6F707172 70717273
71727374 72737475 73747576 74757677 75767778 76777879 7778797A 78797A61
797A6162 7A616263 6162

$H'$ = 15F000AC 58EE3FFF 144845E7 71907C0C 83324A6F

Vì khôi phục là một phần, thông điệp đã được khôi phục $M^*$ bao gồm $M_1^*$ và $M_2^*$, phần có thể và không thể khôi phục được.

$M^*$ = 61626364 62636465 63646566 64656667 65666768 66676869 6768696A 68696A6B
696A6B6C 6A6B6C6D 6B6C6D6E 6C6D6E6F 6D6E6F70 6E6F7071 6F707172 70717273
71727374 72737475 73747576 74757677 75767778 76777879 7778797A 78797A61
797A6162 7A616263 61626364 62636465

Mã băm còn lại $H''$ được tính bằng các áp dụng hàm băm chuyên dụng 1 cho xâu nhị phân có độ dài 1072 (= 64 + 848 + 160), kết quả của việc ghép thêm 64 bit độ dài phần có thể khôi phục được $C'$, 848 bit của phần thông điệp đã được khôi phục $M_1^*$ và 160 bit của mã băm của phần thông điệp không thể khôi phục h($M_2^*$). $H'' = h(C'||M_1^*||h(M_2^*))$

$H''$ = 15F000AC 58EE3FFF 144845E7 71907C0C 83324A6F

Vì hai mã băm $H'$ và $H''$ là giống nhau, chữ ký $\Sigma$ được chấp nhận.

## Thư mục tài liệu tham khảo


[1] M. Bellare and P. Rogaway, *Random oracles are practical: a paradigm for designing efficient protocol.* In: Proceedings of the first annual conference on Computer and Communications Security, ACM, 1993, pp.62-73.

[2] M. Bellare and P. Rogaway, *Optimal asymmetric encryption — how to encrypt with RSA'.* In: A. De Santis (editor), Advances in Cryptology — Eurocrypt '94, Lecture Notes in Computer Science 950 (1995), Springer-Verlag, pp.92-111.

[3] M. Bellare and P. Rogaway, *The exact security of digital signatures: How to sign with RSA and Rabin.* In: U.M. Maurer (editor), Advances in Cryptology — Eurocrypt '96, Lecture Notes in Computer Science 1070 (1996), Springer-Verlag, pp.399-416.

[4] J.-S. Coron, *On the exact security of full domain hashing.* In: M. Bellare (editor), Advances in Cryptology — Crypto 2000, Lecture Notes in Computer Science 1880 (2000), Springer-Verlag, pp.229-235.

[5] J.-S. Coron, D. Naccache, and J.P. Stern, *On the security of RSA padding.* In: M.J. Wiener (editor), Advances in Cryptology — Crypto '99, Lecture Notes in Computer Science 1666 (1999), Springer-Verlag, pp.1-18.

[6] J.-S. Coron, D. Naccache, M. Tibouchi, and R.-P. Weinmann. *Practical Cryptanalysis of ISO 9796-2 and Europay-Mastercard-Visa Signatures.* In: S. Halevi (editor), Advances in Cryptology — Crypto 2009, Lecture Notes in Computer Science 5677 (2009), Springer-Verlag, pp.428-444.

[7] M. Girault and J.-F. Misarsky, *Cryptanalysis of countermeasures proposed for repairing ISO 9796-1.* In: B. Preneel (editor), Advances in Cryptology — Eurocrypt 2000, Lecture Notes in Computer Science 1807 (2000), Springer-Verlag, pp.81-90.

[8] F. Grieu, *A chosen messages attack on the ISO/IEC 9796-1 signature scheme.* In: B. Preneel (editor), Advances in Cryptology — Eurocrypt 2000, Lecture Notes in Computer Science 1807 (2000), Springer-Verlag, pp.70-80.

[9] IEEE Std 1363-2000, *Standard specifications for public key cryptography.*

[10] IEEE Std 1363a-2004, *Standard specifications for public key cryptography — Amendment 1: Additional techniques.*

[11] ISO/IEC 9796-3:2006, *Information technology — Security techniques — Digital signature schemes giving message recovery — Part 3: Discrete logarithm based mechanisms.*

[12] ISO/IEC 9797-2:2002, *Information technology — Security techniques — Message Authentication*

*Codes (MACs) — Part 2: Mechanisms using a dedicated hash-function.*

[13] TCVN 11817-1:19971), *Information technology — Security techniques — Entity authentication — Part 1: General.*

[14] TCVN 12214 (all parts), *Information technology — Security techniques — Digital signatures with appendix.*

[15] J. Jonsson, *Security proofs for the RSA-PSS signature scheme and its variants.* Proceedings of the 2nd NESS/E Workshop, Royal Holloway, University of London, September 2001. Full version available in IACR cryptology archive 2001/053.

[16] B. Kaliski, *On hash function firewalls in signature schemes.* In: B. Preneel (editor), Cryptographers' Track RSA Conference 2002, Lecture Notes in Computer Science 2271 (2002), Springer-Verlag, pp.1-16.