TCVN TIÊU CHUẨN QUỐC GIA

TCVN 12214-2:2018
ISO/IEC 14888-2:2008 VÀ ĐÍNH CHÍNH KỸ THUẬT 1:2015

Xuất bản lần 1

CÔNG NGHỆ THÔNG TIN - CÁC KỸ THUẬT AN TOÀN - CHỮ KÝ SỐ KÈM PHỤ LỤC - PHẦN 2: CÁC CƠ CHẾ DỰA TRÊN PHÂN TÍCH SỐ NGUYÊN

Information technology - Security techniques - Digital signatures with appendix - Part 2: Integer factorization based mechanisms

HÀ NỘI - 2018

Mục Lục

Lời nói đầu

TCVN 12214-2 : 2018 hoàn toàn tương đương với ISO/IEC 14888-2:2008 và đính chính kỹ thuật 1:2015.

TCVN 12214-2 : 2018 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 12214 (ISO/IEC 14888) *Công nghệ thông tin – Các kỹ thuật an toàn – Chữ ký số kèm phụ lục* gồm các tiêu chuẩn sau:

- TCVN 12214-1 : 2018 (ISO/IEC 14888-1:2008) Phần 1: Tổng quan

- TCVN 12214-2 : 2018 (ISO/IEC 14888-2:2008) Phần 2: Các cơ chế dựa trên phân tích số nguyên

- TCVN 12214-3 : 2018 (ISO/IEC 14888-3:2016) Phần 3: Các cơ chế dựa trên logarit rời rạc

TIÊU CHUẨN QUỐC GIA TCVN 12214-2 : 2018

Công nghệ thông tin - Các kỹ thuật an toàn - Chữ ký số kèm phụ lục Phần 2: Các cơ chế dựa trên phân tích số nguyên

Information technology - Security techniques - Digital signature with appendix - Part 2: Integer factorization based mechanisms

1 Phạm vi áp dụng

Tiêu chuẩn này quy định chữ ký số kèm phụ lục với độ an toàn dựa trên độ khó của phân tích số theo mô-đun. Với mỗi lược đồ chữ ký, quy định:

a) Mối quan hệ và ràng buộc giữa tất cả các thành phần dữ liệu được yêu cầu để ký và kiểm tra;

b) Cơ chế ký, tức là cách để tạo ra chữ ký của một thông điệp với các thành phần dữ liệu được yêu cầu để ký;

c) Cơ chế kiểm tra, tức là cách để kiểm tra chữ ký của một thông điệp với các thành phần dữ liệu được yêu cầu để kiểm tra.

Việc tạo các cặp khóa yêu cầu bit ngẫu nhiên và số nguyên tố. Quá trình tạo chữ ký thường yêu cầu bit ngẫu nhiên. Các kỹ thuật sinh các bit ngẫu nhiên và các số nguyên tố nằm ngoài phạm vi tiêu chuẩn này. Để biết thêm thông tin, xem ISO/IEC 18031 [33] và ISO/IEC 18032 [34].

Có nhiều cách khác nhau để có được một bản sao tin cậy của khóa kiểm tra công khai, ví dụ: một chứng thư khóa công khai. Kỹ thuật quản lý khóa và chứng thư nằm ngoài phạm vi tiêu chuẩn này. Để biết thêm thông tin, xem ISO/IEC 9594-8 [27], ISO/IEC 11770 [31] và ISO/IEC 15945 [32].

2 Tài liệu viện dẫn

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

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

TCVN 12214-1 (ISO/IEC 14888-1), *Công nghệ thông tin – Kỹ thuật an toàn – Chữ ký số kèm phụ lục – Phần 1: Tổng quan.*

3 Thuật ngữ và định nghĩa

Với mục đích của tiêu chuẩn này, các thuật ngữ và định nghĩa trong phần 1 tiêu chuẩn này và dưới đây được áp dụng:

3.1

Mô-đun (modulus)

Số nguyên với phân tích số được giữ bí mật và không thể tính toán được các thừa số nguyên tố của số đó.

3.2

Giá trị đặc trưng (representative)

Xâu bit được tạo ra bằng một cơ chế định dạng.

3.3

Salt (salt)

Xâu bit tùy chọn để tạo ra một giá trị đặc trưng.

3.4

Số mũ ký (signature exponent)

Số mũ bí mật để tạo chữ ký.

3.5

Trailer (trailer)

Xâu bit tùy chọn nằm bên phải của một giá trị đặc trưng.

3.6

Số mũ kiểm tra (verification exponent)

Số mũ công khai để kiểm tra thông điệp được ký và cũng có khi được dùng để tạo ra chữ ký.

4 Ký hiệu và chữ viết tắt

Trong tiêu chuẩn này áp dụng các ký hiệu và chữ viết tắt dưới đây:

$A\|\|B$	Xâu bit kết quả của phép ghép hai xâu bit A và B theo thứ tự.
$A \oplus B$	Xâu bit kết quả của phép XOR hai xâu bit A và B có cùng độ dài.
h	Tham số thay thế (adaptation) (GQ2).
Cr	Hệ số CRT.

CRT	Định lý phần dư Trung Hoa.
$\|D\|$	Độ dài bit của D nếu D là một xâu bit, hoặc độ lớn bit của D nếu D là một số (tức là, bằng 0 nếu $D = 0$, hoặc bằng số nguyên duy nhất i sao cho $2^{i-1} \leq D < 2^i$ nếu $D > 0$. Ví dụ $\|65537 = 2^{16} + 1\| = 17$).
$\lfloor D \rfloor$	Số nguyên lớn nhất nhỏ hơn hoặc bằng D.
$\lceil D \rceil$	Số nguyên nhỏ nhất lớn hơn hoặc bằng D.
E	Salt (RSA, RW, ESIGN).
F	Giá trị đặc trưng của (RSA, RW, GQ1, ESIGN).
f	Số lượng các thừa số nguyên tố.
G, G_i	Số công khai.
g, g_i	Số cơ sở.
$(g\|n)$	Ký hiệu Jacobi của một số nguyên dương g đối với một hợp số lẻ n. CHÚ THÍCH 1 Theo định nghĩa, ký hiệu Jacobi của g đối với n là tích của ký hiệu Legendre của g đối với mọi thừa số nguyên tố của n (tính lặp lại các ký hiệu Legendre đối với các thừa số nguyên tố lặp lại nhiều lần). Có thể tính ký hiệu Jacobi [13, 15] mà không cần biết về các thừa số nguyên tố của n.
$(g\|p)$	Ký hiệu Legendre của một số nguyên dương g đối với một số nguyên tố lẻ p. CHÚ THÍCH 2 Theo định nghĩa, nếu p là số nguyên tố thì $(g\|p) = g^{(p-1)/2} mod\ p$. Nghĩa là $(g\|p)$ bằng 0 nếu g là bội của p, ngược lại bằng +1 hoặc -1 phụ thuộc g có là bình phương theo mô-đun p hay không.
$\gcd(a, b)$	Ước chung lớn nhất của hai số nguyên dương a và b.
H, HH	Mã băm.
h	Hàm băm.
$i\ mod\ n$	Số nguyên duy nhất j từ 0 đến $n - 1$ sao cho n chia hết cho $i - j$.
Id	Chuỗi dữ liệu định danh (GQ1).
$Indic$	Chỉ số của một cơ chế được dùng (hàm băm, cơ chế định dạng, biến thể băm.
k	Tham số an toàn (GQ2).
$lcm(a, b)$	Bội số chung nhỏ nhất của hai số nguyên dương a và b.
M	Thông điệp.

m	Số lượng các số cơ sở (GQ2).
n	Mô-đun.
p_i	Thừa số nguyên tố.
Q, Q_i	Số bí mật.
$Q_{i,j}$	Số mũ bí mật (GQ2).
R	Phần đầu tiên của chữ ký (GQ1, GQ2, GPS1, GPS2).
r, r_i, r_{ij}	Số ngẫu nhiên (GQ1, GQ2, GPS1, GPS2, ESIGN).
S	Chữ số (RSA, RW, ESIGN) hoặc phần thứ hai của chữ ký (GQ1, GQ2, GPS1, GPS2).
s, s_i	Số mũ chữ ký (RSA, RW, GQ1, GQ2).
T	coupon (GPS1, GPS2).
t	Tham số độ dài chữ ký (GQ1, GQ2).
u, u_i	Số mũ (GQ1, GQ2).
v	Số mũ kiểm tra (RSA, RW, GQ1, GPS2, ESIGN).
W	Xâu bit (GQ1, GQ2, GPS1, GPS2).
$'XY'$	Ký hiệu sử dụng các chữ số hexa '0' tới '9' và 'A' tới 'F', bằng XY hệ cơ số 16.
x, y, z	Các số nguyên.
α	Độ lớn bit của mô-đun.
γ	Độ dài bit của các giá trị đặc trưng (RSA, RW, GQ1, ESIGN).
ϵ	Độ dài bit của các giá trị salt (các cơ chế định dạng).
τ	Độ dài bit của các giá trị trailer (các cơ chế định dạng).

5 Tổng quan

5.1 Các yêu cầu an toàn

Cơ chế chữ ký sử dụng một tập các thành phần dữ liệu bắt buộc để ký. Tập hợp này bao gồm khóa ký riêng của người ký, được gọi đơn giản là "khóa ký" trong tiêu chuẩn này của bộ tiêu chuẩn. Một số thành phần dữ liệu của khóa ký phải được giữ bí mật (ít nhất phải giữ bí mật một thành phần dữ liệu).

CHÚ THÍCH Mọi thành phần dữ liệu bí mật phải được lưu trong một thiết bị phần cứng hoặc phần mềm dưới sự kiểm soát của người ký tránh kẻ tấn công có thể lấy cắp được. Các thẻ mạch điện tử tích hợp [24] có thể sử dụng để tạo chữ ký. Cấu hình bảo vệ cho các thiết bị tạo chữ ký nằm ngoài phạm vi tiêu chuẩn này của bộ tiêu chuẩn này.

Quá trình tạo chữ ký RSA và RW chỉ mang tính xác suất khi và chỉ khi mọi chữ ký yêu cầu một giá trị salt mới. Quá trình tạo chữ ký GQ1, GQ2, GPS1, GPS2 và ESIGN hoàn toàn mang tính xác suất. Khi quá trình tạo chữ ký mang tính xác suất, thì mọi người ký phải có phương pháp để chọn các bit ngẫu nhiên.

Cơ chế kiểm tra sử dụng một tập hợp các thành phần dữ liệu bắt buộc để kiểm tra, tất cả dữ liệu này phải được công khai trên miền.

- Mọi thành phần dữ liệu công khai dùng chung cho tất cả người ký được gọi là tham số miền.
- Mọi thành phần dữ liệu công khai đặc trưng cho một người ký duy nhất là một phần của khóa kiểm tra công khai của người ký, được gọi đơn giản là "khóa kiểm tra" trong tiêu chuẩn này của bộ tiêu chuẩn này.

Với một tên miền đã cho, mọi người kiểm tra đều biết được tập hợp các tham số miền và nhận được một bản sao tin cậy của khóa kiểm tra của người ký.

Người ký và người kiểm tra phải đảm bảo đầy đủ rằng tập hợp các tham số miền là hợp lệ, tức là nó thỏa mãn các ràng buộc cụ thể trong lược đồ. Ngược lại, sẽ không có đảm bảo nào về độ an toàn ngay cả khi thông điệp đã ký được chấp nhận. Sự đảm bảo này có được bằng nhiều cách, bao gồm một hoặc nhiều cách sau:

a) Lựa chọn một tập hợp các giá trị từ một nguồn công khai tin cậy, ví dụ: một tiêu chuẩn quốc tế;
b) Tạo ra một tập hợp các giá trị bởi bên thứ ba tin cậy, ví dụ: một tổ chức cung cấp chứng thư số [27];
c) Xác nhận một tập hợp các giá trị bởi bên thứ ba tin cậy, ví dụ: một tổ chức cung cấp chứng thư số [27];
d) Đối với người ký, tạo ra một tập hợp các giá trị bởi một hệ thống tin cậy;
e) Đối với người ký và người kiểm tra, xác nhận một tập hợp các giá trị.

Người ký và người kiểm tra phải đảm bảo đầy đủ rằng khóa kiểm tra là hợp lệ, tức là nó thỏa mãn các ràng buộc cụ thể trong lược đồ. Sự đảm bảo này có được bằng nhiều cách, bao gồm một hoặc nhiều cách sau:

a) Truy cập tới một thư mục hoặc kiểm tra một chứng thư;
b) Một giao thức kiểm tra khóa hoạt động với khóa kiểm tra và có thể là các thông tin khác liên quan đến sự tương tác với phần cứng hoặc phần mềm tạo chữ ký;
c) Tin tưởng vào lời khẳng định của bên thứ ba về việc đảm bảo rằng khóa kiểm tra là hợp lệ;
d) Tin tưởng rằng quá trình tạo khóa được thực hiện hoàn toàn chính xác.

Các giao thức và phương pháp kiểm tra khóa cụ thể để thu thập và chuyển tải sự đảm bảo tính hợp lệ của khóa nằm ngoài phạm vi tiêu chuẩn này của bộ tiêu chuẩn này.

Độ an toàn của mọi lược đồ chữ ký được quy định trong tiêu chuẩn này phụ thuộc vào một số mô-đun và một hàm băm.

- Số mô-đun là an toàn (tức là kháng phân tích số) khi giá trị phân tích số không bị lộ. Khi sử dụng lược đồ, không có chủ thể nào có khả năng phân tích số mô-đun đang được sử dụng.
- Hàm băm sử dụng là một trong những hàm băm được quy định trong TCVN 11816:2017 (ISO/IEC 10118); Nó có khả năng kháng va chạm.

5.2 Khóa kiểm tra

Bảng 1 tổng hợp các khóa kiểm tra (xem 6.1, 7.1, 8.1, 9.1, 10.1 và 11.1).

Bảng 1 – Các khóa kiểm tra

Lược đồ	Bắc buộc	Tùy chọn [a]		Tùy chọn [b]	
RSA, RW, ESIGN	n	v	*Indic*(h)	α	$Indic(\text{format}, \varepsilon, \tau)$
GQ1 [c]		n, v	*Indic*(h)	α	$Indic(\text{variant}), Indic(\text{format}, \varepsilon, \tau)$
GQ2	n		*Indic*(h)	$b, (g_1, g_2 \dots g_m), \alpha$	$Indic(\text{variant})$
GPS1	G	n	*Indic*(h)	g, α	$Indic(\text{variant})$
GPS2	n	v	*Indic*(h)	g, α	$Indic(\text{variant})$

[a] Nếu không là thành phần của khóa kiểm tra, thì một thành phần dữ liệu sẽ là một tham số miền.

[b] Nếu không phải là một tham số miền, cũng không là thành phần của khóa kiểm tra, thì một thành phần dữ liệu sẽ được coi là một giá trị mặc định.

[c] Khóa kiểm tra GQ1 có thể là rỗng.

Mọi lược đồ chữ ký được đặc tả trong tiêu chuẩn này sử dụng một mô-đun, ký hiệu là n.

- Trong lược đồ RSA, RW, GQ2, GPS2 và ESIGN, khóa kiểm tra bao gồm n.
- Trong lược đồ GQ1 và GPS1, các tham số miền hoặc khóa kiểm tra bao gồm n.

CHÚ THÍCH Thời hạn sử dụng một giá trị mô-đun cho trước thường bị giới hạn trong một khoảng thời gian nhất định trong một miền nhất định.

Để quy định độ lớn bit của mô-đun được sử dụng, các tham số miền hoặc khóa kiểm tra sẽ bao gồm một thành phần dữ liệu, ký hiệu là α. Nếu không quy định về α, thì giá trị mặc định của α được thiết lập bằng độ lớn bit của mô-đun được sử dụng (nghĩa là không quy định về kích thước mô-đun).

Trong lược đồ GPS1, khóa kiểm tra bao gồm một số công khai được sử dụng, ký hiệu là G.

Để tương thích với cơ sở hạ tầng khóa công khai đã được triển khai, ngay cả khi tất cả những người ký sử dụng cùng một giá trị trong miền, khóa kiểm tra có thể bao gồm:

- Số mũ kiểm tra được dùng, ký hiệu là v trong lược đồ RSA, RW, GQ1, GPS2 và ESIGN;

- Số mô-đun được dùng, ký hiệu là n trong lược đồ GQ1 và GPS1.

Mọi lược đồ chữ ký quy định trong tiêu chuẩn này sử dụng hàm băm, ký hiệu là h.

- Trong lược đồ RSA, RW và ESIGN, một cơ chế định dạng sử dụng hàm h để chuyển đổi thông điệp thành giá trị đặc trưng và kiểm tra giá trị đặc trưng sau khi được khôi phục lại.
- Trong lược đồ GQ1, một cơ chế định dạng sử dụng hàm h để chuyển đổi những chuỗi dữ liệu định danh thành các số công khai và một biến thể băm sử dụng hàm h để tạo ra các xâu bit.
- Trong lược đồ GQ2, GPS1 và GPS2, một biến thể băm sử dụng hàm h để tạo ra các xâu bit.

Để xác định hàm băm đang được sử dụng, các tham số miền hoặc khóa kiểm tra sẽ bao gồm một thành phần dữ liệu, ký hiệu là $Indic(h)$.

Tiêu chuẩn này đặc tả ba cơ chế định dạng (PSS trong mục 6.4, 7.4 và 11.4; D1 và D2 trong phụ lục D). Mỗi cơ chế định dạng sử dụng hai tham số, ký hiệu là ε và τ. Nhận giá trị 0,64 hoặc $|H|$, ε cho biết độ dài bit của giá trị salt. Nhận giá trị 0,8 hoặc 16, τ cho biết độ dài bit của giá trị trailer.

Tiêu chuẩn này đặc tả bốn biến thể băm, trong đó W ký hiệu một xâu bit và M là một thông điệp.

1) $h(W||M)$ 2) $h(W||h(M))$ 3) $h(h(W)||M)$ 4) $h(h(W)||h(M))$

Để xác định cơ chế định dạng đang được sử dụng, cùng với những giá trị tùy chọn ε và τ đang dùng, và/hoặc biến thể băm đang sử dụng, các tham số miền hoặc khóa kiểm tra có thể bao gồm một hoặc hai thành phần dữ liệu, ký hiệu là $Indic(format, \varepsilon, \tau)$ và $Indic(variant)$ nếu cần.

Khóa ưu tiên – Khi các tham số miền và khóa kiểm tra bao gồm thành phần dữ liệu tương tự nhau với những giá trị khác nhau, thì khóa kiểm tra có quyền ưu tiên hơn.

CHÚ THÍCH Trong một miền nhất định, do có khóa ưu tiên, những người ký khác nhau có thể sử dụng các hàm băm khác nhau và/hoặc các kích thước mô-đun khác nhau.

5.3 Kỹ thuật CRT

Xem xét hai số nguyên x_1 và x_2 là hai số nguyên tố cùng nhau, nhưng không cần là số nguyên tố. Theo định nghĩa, hệ số CRT của x_1 và x_2, ký hiệu là Cr là một số nguyên dương duy nhất, nhỏ hơn x_1, sao cho $Cr \times x_2 - 1$ là bội số của x_1.

Số nguyên bất kỳ X thuộc $\{0,1 \dots x_1 \times x_2 - 1\}$ có thể được phân tích thành cặp duy nhất các thành phần $X_1 = X \bmod x_1$ nhận giá trị từ $\{0,1 \dots x_1 - 1\}$ và $X_2 = X \bmod x_2$ nhận giá trị từ $\{0,1 \dots x_2 - 1\}$.

Hợp số CRT là giá trị nghịch đảo của phân tích ở trên. Sử dụng ba số nguyên x_1, x_2 và Cr để biến đổi hai thành phần X_1 nhận giá trị từ $\{0,1 \dots x_1 - 1\}$ và X_2 nhận giá trị từ $\{0,1 \dots x_2 - 1\}$ thành số nguyên duy nhất X nhận giá trị từ $\{0,1 \dots x_1 \times x_2 - 1\}$ sao cho $X_1 = X \bmod x_1$ và $X_2 = X \bmod x_2$.

$$Y = X_1 - X_2 \bmod x_1; Z = Y \times Cr \bmod x_1; X = Z \times x_2 + X_2$$

Lần lượt biến đổi ba thành phần X_1 từ $\{0,1 \dots x_1 - 1\}$, X_2 từ $\{0,1 \dots x_2 - 1\}$ và X_3 từ $\{0,1 \dots x_3 - 1\}$, trong đó x_1, x_2 và x_3 nguyên tố cùng nhau từng đôi một, thành số nguyên duy nhất X nhận giá trị từ $\{0,1 \dots x_1 \times x_2 \times x_3 - 1\}$ thỏa mãn $X_1 = X \bmod x_1$, $X_2 = X \bmod x_2$ và $X_3 = X \bmod x_3$, hợp số CRT được sử dụng hai lần:

1) để tính toán T nhận giá trị từ $\{0,1 \dots x_1 \times x_2 - 1\}$ sao cho $X_1 = T \bmod x_1$ và $X_2 = T \bmod x_2$;

2) để tính toán X nhận giá trị từ $\{0,1 \dots x_1 \times x_2 \times x_3 - 1\}$ sao cho $T = X \bmod x_1 \times x_2$ và $X_3 = X \bmod x_3$.

Khi biết các thừa số nguyên tố của n (xem mục 6.2, 7.1, 8.1, 8.2, 9.1, 9.2.2 và 10.2.2), kỹ thuật CRT làm giảm độ phức tạp của tính toán số học mod n (xem B.2.3). Thay vì tính toán trực tiếp kết quả cuối

cùng nhận giá trị từ $\{0,1 \ldots n-1\}$, sẽ tính toán một tập hợp các thành phần sau đó biến đổi thành kết quả cuối cùng.

CHÚ THÍCH Hiệu quả của kỹ thuật CRT sẽ tăng theo số lượng các thừa số nguyên tố khác nhau.

5.4 Biến đổi giữa xâu bit, số nguyên và chuỗi octet

Một xâu bit, ký hiệu là D, bao gồm $|D|$ bit trong đó giá trị của mỗi bit là 0 hoặc 1; các bit được đánh số theo thứ tự từ bit trái nhất, ký hiệu là d_1, đến bit phải nhất, ký hiệu là $d_{|D|}$.

$$D = d_1 d_2 d_3 \ldots d_{|D|-1} d_{|D|}$$

Để chuyển đổi D thành một số nguyên, ký hiệu là A, bit trái nhất, ký hiệu là d_1, là bit có trọng số cao nhất và bit phải nhất, ký hiệu là $d_{|D|}$ là bit có trọng số thấp nhất.

$$A = 2^{|D|-1} \times d_1 + 2^{|D|-2} \times d_2 + \cdots + 2^2 \times d_{|D|-2} + 2 \times d_{|D|-1} + d_{|D|}$$

Độ lớn bit của số nguyên A, ký hiệu là $|A|$ (nghĩa là $2^{|A|-1} \leq A \leq 2^{|A|}$ nếu $A > 0$, do đó $0 \leq A < 2^{|D|}$), bằng $|D|$ nếu $d_1 = 1$, hoặc nhỏ hơn $|D|$ nếu $d_1 = 0$. Biểu diễn nhị phân của số nguyên A bằng một xâu bit có độ dài lớn hơn $|A|$ là xâu bit duy nhất mà khi biến đổi thành số nguyên thì cho giá trị bằng A.

Khi độ dài bit của một xâu là bội số của 8, xâu bit có thể dễ dàng biểu diễn bằng một xâu octet trong đó mỗi octet có giá trị từ "00" đến "FF" trong ký hiệu hệ tập lục phân. Trong một xâu octet, octet được đánh số thứ tự từ octet trái nhất đến octet phải nhất. Để biến đổi một xâu octet thành một số nguyên, octet trái nhất là octet có trọng số cao nhất và octet phải nhất là octet có trọng số thấp nhất.

6 Lược đồ RSA và RW

6.1 Yêu cầu các thành phần dữ liệu để ký/kiểm tra

Các quan hệ và ràng buộc chuỗi được áp dụng cho các thành phần dữ liệu sau:

- Số mũ kiểm tra;
- Tập hợp các thừa số nguyên tố khác nhau;
- Số mô-đun;
- Số mũ ký;
- Tập hợp các số mũ ký CRT.

Số mũ kiểm tra ký hiệu là v. Các giá trị $v = 0$ và $v = 1$ không được sử dụng.

CHÚ THÍCH Các giá trị $v = 2,3$ và $65537 (= 2^{16} + 1)$ có những ưu điểm trong thực nghiệm.

Tập hợp các thừa số nguyên tố khác nhau ký hiệu là $p_1, p_2 \ldots p_f$ được sắp xếp theo thứ tự tăng dần $(f > 1)$.

Lược đồ RSA sử dụng số mũ kiểm tra lẻ. Có thể có nhiều hơn hai thừa số nguyên tố $(f \geq 2)$. Với i từ 1 đến f, v sẽ nguyên tố cùng nhau với $p_i - 1$, tức là $gcd(v, p_i - 1) = 1$.

Lược đồ RW sử dụng số mũ kiểm tra chẵn. Tiêu chuẩn này quy định giá trị $v = 2$, với chỉ hai thừa số nguyên tố $(f = 2)$, cả hai đều đồng dư với 3 mod 4, nhưng không đồng dư với nhau mod 8.

Số mô-đun, ký hiệu là n là tích của các thừa số nguyên tố $(n = p_1 \times \ldots \times p_f)$. Độ lớn của nó là α bit.

Số mũ ký được ký hiệu là s, là số nguyên dương bất kỳ (thường sử dụng số nhỏ nhất) sao cho $v \times s - 1$ là bội số của $lcm(p_1 - 1, \ldots, p_f - 1)$ nếu v là số lẻ, hoặc là bội số của $lcm(p_1 - 1, p_2 - 1)/2$ nếu $v = 2$.

Tập hợp các số mũ ký CRT ký hiệu là s_1 đến s_f. Với i từ 1 đến f, s_i là số nguyên dương bất kỳ (thường sử dụng số nhỏ nhất) sao cho $v \times s_i - 1$ là bội số của $p_i - 1$ nếu v là số lẻ, hoặc là bội số của $(p_i - 1)/2$ nếu $v = 2$.

CHÚ THÍCH Trong lược đồ RW, có một thừa số nguyên tố đồng dư với 3 mod 8 và một số khác đồng dư với 7 mod 8, $n \equiv 5 \bmod 8, (\pm 2|n) = -1, s = \frac{n - p_1 - p_2 + 5}{8}, s_1 = (p_1 + 1)/4$ và $s_2 = (p_2 + 1)/4$.

Quá trình ký yêu cầu một hàm băm (xem mục 5.1), một cơ chế định dạng và một khóa ký. Khuyến nghị sử dụng cơ chế định dạng được quy định trong 6.4; nó sử dụng hai tham số, ký hiệu là ε và τ. Khóa ký có hai dạng sau:

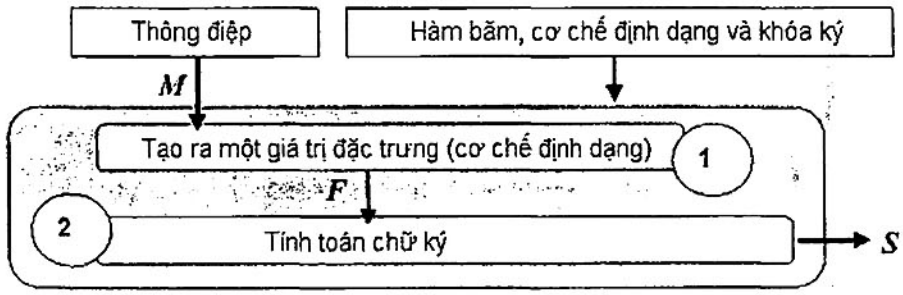
- Với CRT: p_1 đến $p_f, f - 1$ hệ số CRT (xem mục 5.3) và s_1 đến s_f.
- Không có CRT: n và s (n công khai).

CHÚ THÍCH Cơ chế định dạng quy định trong mục 6.4 được tin tưởng là an toàn. Hai cơ chế định dạng quy định trong phụ lục D có giới hạn an toàn nhỏ hơn.

Quá trình kiểm tra yêu cầu một tập hợp các tham số miền và một khóa kiểm tra. Các tham số miền hoặc khóa kiểm tra sẽ bao gồm v và $Indic(h)$, và có thể gồm cả α (mặc định $\alpha = |n|$) và $Indic(format, \varepsilon, \tau)$ (mặc định theo mục 6.4 với các giá trị tùy chọn $\varepsilon = |H|$ và $\tau = 8$). Khóa kiểm tra bao gồm n.

6.2 Cơ chế ký

Cơ chế ký được minh họa trong hình 1 sử dụng một hàm băm, một cơ chế định dạng và một khóa ký để ký một thông điệp (một xâu bit, ký hiệu là M), nghĩa là tạo ra một chữ ký của M (một xâu bit, ký hiệu là S).



Hình 1 – Ký với RSA hoặc RW

Bước 1 – Chuyển đổi thông điệp M thành một giá trị đặc trưng $\gamma = |n|$ bit, ký hiệu là F, theo cơ chế định dạng đang sử dụng. Xâu bit F biểu diễn một số, chia hết cho 4, cũng ký hiệu là F ($0 < F < n$).

Bước 2 – Tạo ra một số, ký hiệu là G ($0 < G < n$).

- Nếu v là số lẻ thì $G = F$.
- Nếu $v = 2$, tính toán ký hiệu Jacobi $(F|n)$ và biến đổi ký hiệu Jacobi $(G|n)$ về +1.

- Nếu $(F|n) = +1$, thì $G = F$.
- Nếu $(F|n) = -1$, thì $G = F/2$.
- Nếu $(F|n) = 0$ (trường hợp rất hiếm gặp), thì quá trình tạo số thất bại.

Tạo ra một số, ký hiệu là S bằng một trong hai cách sau:

- Với CRT, với i từ 1 đến f, tính $G_i = G \; mod \; p_i$ và $S_i = G_i^{Si} \; mod \; p_i$. Số S là hợp số CRT (xem mục 5.3) S_1 đến S_f.
- Không sử dụng CRT, tính toán $S = G^S mod \; n$.

Nếu $v = 2$, thì số S được thay thế bằng $n - S$.

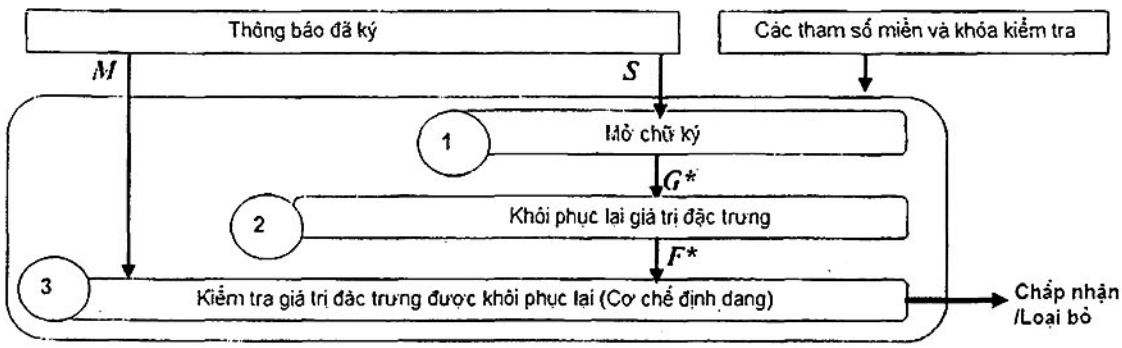
Chữ ký là một xâu bit bất kỳ đại diện cho S, thường là một xâu gồm $|n|$ bit và cũng ký hiệu là S.

6.3 Cơ chế kiểm tra

Cơ chế kiểm tra được minh họa trong hình 2 sử dụng một tập hợp của các tham số miền và một khóa kiểm tra (xem bảng 1), với khóa ưu tiên (xem mục 5.2) để kiểm tra một thông điệp và một chữ ký của thông điệp đó, tức là hai xâu bit, ký hiệu M và S.

Bước 0 – Loại bỏ nếu $|n| \neq \alpha$, hoặc nếu $v = 0$ hoặc 1, hoặc nếu n không đồng dư với 5 mod 8 khi $v = 2$.

Bước 1 – Xâu bit S biểu diễn một số, cũng ký hiệu là S. Loại bỏ nếu $S = 0$ hoặc 1, hoặc nếu $S \geq n - 1$. Tính toán $G^* = S^v mod \; n$.



Hình 2 – Kiểm tra với RSA hoặc RW

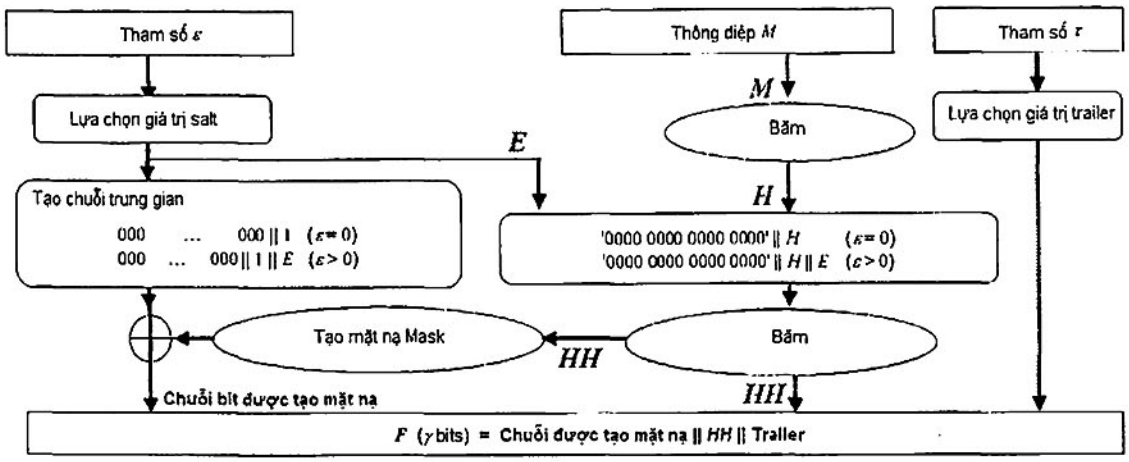
Bước 2 – Khôi phục lại giá trị đặc trưng, ký hiệu là F^*.

- Nếu v là một số lẻ, thì F^* là một xâu có độ dài $|n|$ bit biểu diễn của G^*.
- Nếu $v = 2$, F^* là một chuỗi có độ dài $|n|$ bit biểu diễn:
  - G^* nếu G^* đồng dư với 4 mod 8;
  - $n - G^*$ nếu G^* đồng dư với 1 mod 8;
  - $2G^*$ nếu G^* đồng dư với 6 mod 8;
  - $2(n - G^*)$ nếu G^* đồng dư với 7 mod 8.
  - Loại bỏ trong các trường hợp còn lại (Không biểu diễn được giá trị trailer).

Bước 3 – Kiểm tra giá trị đặc trưng F^* đã được khôi phục lại theo cơ chế định dạng sử dụng.

6.4 Cơ chế định dạng

Quá trình chuyển đổi thông điệp M sử dụng hai tham số (ε biểu thị độ dài của giá trị salt và τ biểu thị độ dài giá trị trailer) thành một giá trị đặc trưng gồm γ bit, ký hiệu là F. Hình 3 minh họa cho cơ chế này.



Hình 3 – Tạo giá trị đặc trưng

1) Các tùy chọn như sau.
   - Tùy chọn $\varepsilon = 0$. Giá trị salt là một xâu rỗng và quá trình tạo chữ ký là quá trình tất định.
   - Tùy chọn $\varepsilon = |H|$. Giá trị salt, ký hiệu là E là một xâu gồm $|H|$ bit ngẫu nhiên.
     - Nếu giá trị salt là giá trị cố định cho nhiều chữ ký, thì quá trình tạo chữ ký là quá trình tất định.
     - Nếu giá trị salt là một giá trị mới cho từng chữ ký, thì quá trình tạo chữ ký msng tính xác suất.
   - Tùy chọn $\tau = 8$. Giá trị trailer là một octet đơn, đặt bằng "BC".
   - Tùy chọn $\tau = 16$. Giá trị trailer là hai octet liên tiếp: octet phải nhất được đặt bằng "CC"; octect trái nhất xác định hàm băm sử dụng. Octet trái nhất được biểu diễn như sau.
     - Giá trị từ "00" đến "7F" dành cho tiêu chuẩn ISO/IEC JTC 1 SC 27; ISO/IEC 10118 quy định một định danh duy nhất trong dãy giá trị đó cho từng hàm băm tiêu chuẩn, ví dụ: "31" đại diện cho hàm đầu tiên trong Phần 3, có tên gọi là RIPEMD-160 và "33" đại diện cho hàm thứ ba trong Phần 3, có tên gọi là SHA-1.
     - Giá trị từ "80" đến "FF" được dành cho trường hợp đặc biệt.

Trailer = Định danh hàm băm || "CC".

CHÚ THÍCH Một số nghiên cứu [12] đặt câu hỏi về ưu điểm khi sử dụng định danh như trên trong giá trị trailer.

2) Băm M thành một xâu bit, ký hiệu là H. Từ trái sang phải, nối 8 octet có giá trị "00", H và E. Băm chuỗi vừa nối thành một xâu bit, ký hiệu là HH.

$$H = h(M) \qquad HH = h(("0000\ 0000\ 0000\ 0000")||H||E)$$

3) Tạo ra một chuỗi gồm ít nhất $\gamma - \tau - |H|$ bit từ HH theo các bước sau sử dụng hai biến: một xâu có độ dài tùy ý, ký hiệu là *String*, và một xâu 32 bit, ký hiệu là *Counter*.

a) Đặt *String* bằng một xâu rỗng.

b) Đặt *Counter* bằng 0.

c) Thay *String* bằng $String||h(HH||Counter)$.
d) Thay *Counter* bằng Counter + 1.
e) Nếu $|H| \times Counter < \gamma - \tau - |H|$, thì quay lại bước c.

Tạo ra một giá trị mặt nạ với $\gamma - \tau - |H|$ bit trái nhất của *String* trong đó bit trái nhất có giá trị bắt buộc bằng 0.

4) Tạo ra một xâu trung gian gồm $\gamma - \tau - |H|$ bit được nối từ trái sang phải theo thứ tự như sau:
- $\gamma - \tau - |H| - 1 - \varepsilon$ bit 0;
- Một bit giới hạn bằng 1;
- Giá trị salt *E*.

5) Áp dụng mặt nạ cho xâu trung gian, để tạo ra một xâu mặt nạ bằng cách thực hiện phép XOR.
6) Tạo ra *F* bằng cách nối xâu bit đã được tạo mặt nạ, *HH* và trailer theo thứ tự từ trái sang phải. Trả về *F*.

F = Xâu bit được tạo mặt nạ || *HH* || Trailer

Kiểm tra giá trị đặc trưng đã được khôi phục lại gồm γ bit, ký hiệu là F^* tương ứng với thông điệp *M* và sử dụng hai giá trị tùy chọn ε và τ (được cho bởi khóa kiểm tra hoặc các tham số miền, hoặc là giá trị mặc định).

1) Kiểm tra giá trị trailer như sau.
- Nếu octet phải nhất của F^* có giá trị bằng "BC", thì tùy chọn được khôi phục lại là $\tau^* = 8$.
- Nếu octet phải nhất của F^* có giá trị bằng "CC" và nếu octet bên trái của "CC" xác định hàm băm sử dụng, thì tùy chọn được khôi phục lại là $\tau^* = 16$.
- Loại bỏ trong các trường hợp còn lại (Không thể biểu diễn giá trị trailer) và khi τ^* và τ khác nhau.

2) Chia $\gamma - \tau$ bit trái nhất của F^* thành hai phần: một xâu đã được tạo mặt nạ gồm $\gamma - \tau - |H|$ bit nằm ở bên trái và xâu gồm $|H|$ bit, ký hiệu là HH^* nằm ở bên phải.
3) Tạo mặt nạ gồm $|n| - \tau - |H|$ bit từ HH^* giống bước 3 ở trên.
4) Áp dụng mặt nạ vào xâu đã được tạo mặt nạ, để khôi phục lại một xâu trung gian bằng cách Thực hiện phép XOR, trong đó bit giới hạn là bit đầu tiên có giá trị bằng 1 tính từ trái sang.
- Nếu còn lại ε bit ở bên phải của bit giới hạn trong xâu trung gian đã được khôi phục lại, thì tạo ra một xâu bit, ký hiệu là E^*.
- Ngược lại, thì loại bỏ.

5) Băm M thành một xâu bit, ký hiệu là *H*. Từ trái sang phải, nối 8 octet có giá trị "00", H và E^*. Băm chuỗi vừa nối thành một xâu bit, ký hiệu là *HH*.

$$H = h(M) \qquad HH = h(\text{"0000 0000 0000 0000"} \parallel H \parallel E^*)$$

6) Chấp nhận hoặc loại bỏ tùy thuộc vào *HH* và HH^* giống nhau hay khác nhau.

7 Lược đồ GQ1 (lược đồ dựa trên định danh)

7.1 Tập hợp các thành phần dữ liệu cần để ký/kiểm tra

CHÚ THÍCH Tập hợp các thừa số nguyên tố là giá trị bí mật của thực thể với số mô-đun công khai; Số mô-đun là tham số miền, hoặc một phần của khóa kiểm tra. Do đó, lược đồ có thể thực hiện bằng một trong hai cách sau.

1) Nếu số mô-đun là một tham số miền, thì thực thể tạo ra số mô-đun công khai là một tổ chức tin cậy cung cấp chứng thư số cung cấp cho mỗi người ký một số bí mật riêng, do đó đảm bảo chuỗi dữ liệu định danh của người ký. Ví dụ, nhà sản xuất thẻ điện tử tích hợp [24] có một số mô-đun.

- Đối với thẻ cá nhân, một chủ thể được ủy quyền sử dụng giá trị bí mật của nhà sản xuất để ký các chuỗi dữ liệu định danh; Trong mỗi thẻ, nó lưu trữ một chuỗi dữ liệu định danh và một số bí mật.
- Trong thời hạn sử dụng, thẻ điện tử sử dụng số bí mật theo kỹ thuật tri thức không.

2) Nếu số mô-đun là một phần của khóa kiểm tra, thì với mỗi phiên (session), người ký được cung cấp một số bí mật, do đó bảo đảm chuỗi dữ liệu định danh phiên. Trong một mạng cục bộ, một tổ chức cung cấp chứng thư số giám sát từng thao tác đăng nhập và quản lý một thư mục trong đó mỗi bên kiểm tra có thể nhận được một bản sao tin cậy của số mô-đun cho mọi chủ thẻ.

- Khi một máy tính kết nối tới mạng cục bộ, tức là, trong một thao tác đăng nhập, nó sử dụng giá trị bí mật của chủ thể tương ứng để tạo ra một số bí mật bằng cách đăng nhập một lần đối với một chuỗi dữ liệu định danh phiên.
- Trong phiên liên lạc, máy tính không thể sử dụng giá trị bí mật của chủ thể (một giá trị bí mật dài hạn) vì không biết gì về giá trị đó; nó sử dụng số bí mật theo kỹ thuật tri thức không. Số bí mật (một giá trị bí mật ngắn hạn) chỉ có giá trị trong một vài giờ: nó mất giá trị sau phiên liên lạc.

Các quan hệ và ràng buộc chuỗi được áp dụng cho các thành phần dữ liệu sau:

- Số mũ kiểm tra và một tham số độ dài chữ ký;
- Tập hợp các thừa số nguyên tố khác nhau;
- Số mô-đun;
- Chuỗi dữ liệu định danh;
- Số công khai;
- Số bí mật.

Số mũ kiểm tra, ký hiệu là v là một số nguyên tố. Tham số độ dài chữ ký được ký hiệu là t. Tích $(|v| - 1) \times t$ nhỏ hơn hoặc bằng $|H|$.

CHÚ THÍCH Với $(|v| - 1) \times t = 80$, các giá trị thường gặp của v và t là $(2^{80} + 13{,}1), (2^{40} + 15{,}2), (2^{26} + 7{,}4), (2^{16} + 1{,}5)$.

Tập hợp các thừa số nguyên tố khác nhau ký hiệu là $p_1, p_2 \ldots p_f$ được sắp xếp theo thứ tự tăng dần $(f > 1)$.

Với i từ 1 đến f, v không chia hết cho $p_i - 1$.

Số mô-đun, ký hiệu là n, là tích của các thừa số nguyên tố $(n = p_1 \times \ldots \times p_f)$. Độ lớn của nó là α bit.

Quá trình tạo của mỗi người ký tuân thủ theo ba bước sau.

Bước 1 – Lựa chọn một chuỗi dữ liệu định danh, ký hiệu là *Id*. Nó là một xâu bit, xác định người ký duy nhất và hợp lệ theo một thỏa thuận có sẵn ở cấp độ miền.

CHÚ THÍCH Chuỗi dữ liệu định danh bao gồm một số tài khoản, số seri, ngày giờ hết hạn, các quyền của một định danh. Bắt buộc phải tuân theo thời hạn theo ngày và giờ hết hạn trong chuỗi; số seri giúp đơn giản hóa quá trình thu hồi.

Bước 2 – Chuyển đổi *Id* thành một giá trị đặc trưng gồm $\gamma = |n|$ bit theo cơ chế định dạng sử dụng. Nó đại diện cho số công khai, ký hiệu là G $(1 < G < n)$.

Bước 3 – Tạo ra số bí mật, ký hiệu là Q bằng một trong hai cách sau.

- Với CRT, với i từ 1 đến f, tính một số, ký hiệu là s_i là số nguyên dương nhỏ nhất sao cho $v \times s_i - 1$ là bội số của $p_i - 1$, do đó $u_i = p_i - 1 - s_i$, $G_i = G \bmod p_i$ và $Q_i = G_i^{u_i} \bmod p_i$. Số Q là hợp số CRT (xem mục 5.3) của Q_1 đến Q_f.

- Không có CRT, tính một số, ký hiệu s là số nguyên dương nhỏ nhất sao cho $v \times s - 1$ là bội số của $lcm(p_1 - 1, ..., p_f - 1)$, thì $u = lcm(p_1 - 1, ..., p_f - 1) - s$ và $Q = G^u \bmod n$.

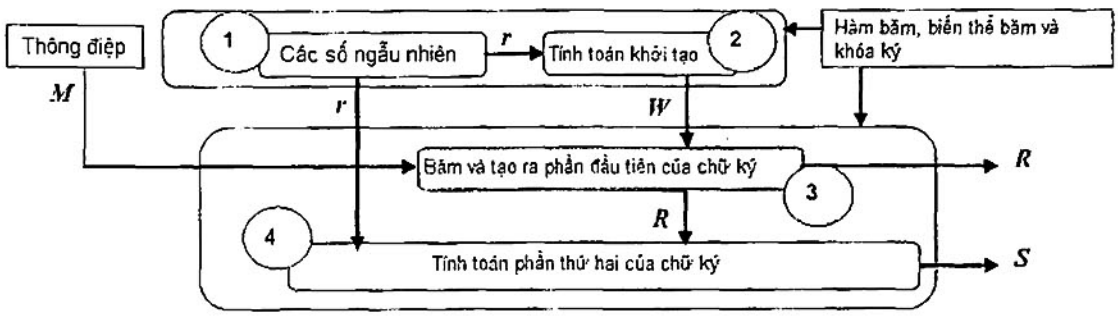
CHÚ THÍCH Số Q là số nghịch đảo mod n của chữ ký được định nghĩa trong 6.1. Cặp G và Q thỏa mãn $G \times Q^v \bmod n = 1$.

Quá trình ký yêu cầu một hàm băm (xem mục 5.1), một biến thể băm, một cơ chế định dạng và một khóa ký. Khuyến nghị sử dụng cơ chế định dạng quy định trong 7.4. Khóa ký bao gồm t, v, n và Q (t, v, n công khai).

Quá trình kiểm tra yêu cầu một tập hợp các tham số miền, một khóa kiểm tra và *Id*. Các tham số miền có thể bao gồm t (mặc định $t = 1$). Các tham số miền hoặc khóa kiểm tra bao gồm v, n và $Indic(h)$ và có thể gồm α (mặc định $\alpha = |n|$), $Indic(variant)$ (mặc định sử dụng biến thể đầu tiên) và $Indic(format, \varepsilon, \tau)$ (mặc định trong 7.4).

7.2 Cơ chế ký

Cơ chế ký được minh họa trong hình 4 sử dụng một hàm băm, một biến thể băm và một khóa ký để ký một thông điệp (một xâu bit, ký hiệu *M*), tức là tạo ra một chữ ký của *M* (hai xâu bit, ký hiệu là *R* và *S*).



Hình 4 – Ký với GQ1

Bước 1 – Lựa chọn t xâu gồm $|n|$ bit ngẫu nhiên.

Biểu diễn các số ngẫu nhiên (giữ bí mật), ký hiệu là r_1 đến r_t (ký hiệu là *r* trong hình 1).

CHÚ THÍCH Xác suất để một xâu gồm $|n|$ bit ngẫu nhiên có giá trị bằng 0 hoặc một bội số của một thừa số nguyên tố của n là không đáng kể.

Bước 2 – Với i từ 1 đến t, tính toán $r_i^v \bmod n$ và biểu diễn giá trị đó bằng một xâu gồm $|n|$ bit, ký hiệu là W_i.

Tạo một xâu gồm $|n| \times t$ bit, ký hiệu là W gồm $W_1 \parallel W_2 \parallel \cdots \parallel W_t$.

Bước 3 – Tạo ra một xâu bit, ký hiệu là H theo biến thể băm sử dụng.

$$H = h(W \parallel M) \text{trong biến thể đầu tiên}$$
$$h\big(W \parallel h(M)\big) \text{trong biến thể thứ hai}$$
$$h(h(W \parallel M) \text{trong biến thể thứ ba}$$
$$h\big(h(W) \parallel h(M)\big) \text{trong biến thể thứ tư}$$

Tạo ra phần đầu tiên của chữ ký, ký hiệu là R gồm $(|v| - 1) \times t$ bit trái nhất của H.

Bước 4 – Tách R thành t xâu gồm $|v| - 1$ bit, cụ thể gồm $R_1 \| R_2 \| \cdots \| R_t$. Mỗi xâu bit R_i biểu diễn một số, cũng ký hiệu là R_i (nhỏ hơn $2^{|v|-1}$, do đó nhỏ hơn v).

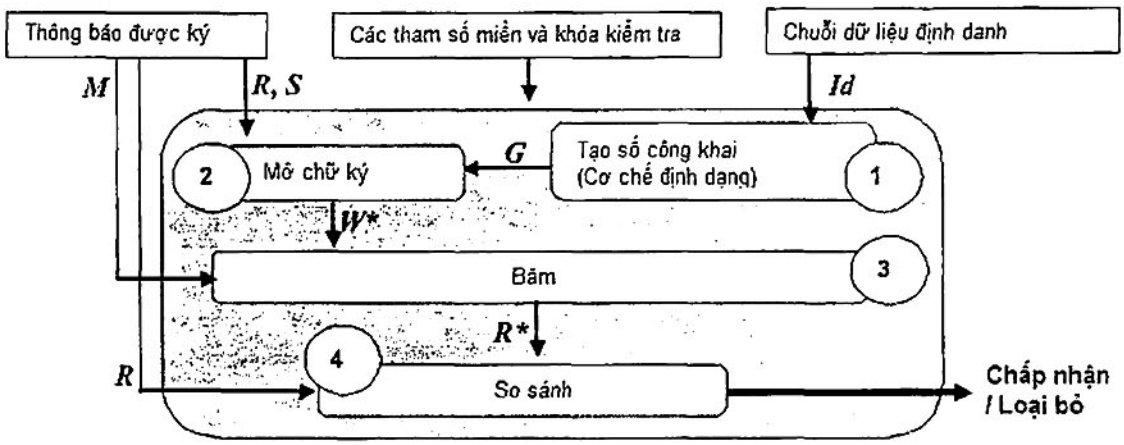
Với i từ 1 đến t, tính toán $r_i \times Q^{R_i} mod\ n$ và biểu diễn bằng một xâu gồm $|n|$ bit, ký hiệu là S_i.

Tạo ra phần thứ hai của chữ ký, ký hiệu là S, gồm $S_1 \| S_2 \| \cdots \| S_t$ ($|n| \times t$ bit).

7.3 Cơ chế kiểm tra

Cơ chế kiểm tra minh họa trong hình 5 sử dụng một tập hợp các tham số miền, một khóa kiểm tra (xem bảng 1) với khóa ưu tiên (xem 5.2) và một chuỗi dữ liệu định danh (một xâu bit, ký hiệu là *Id*), để kiểm tra một thông điệp và chữ ký của thông điệp đó, tức là ba xâu bit, ký hiệu là *M*, *R* và *S*.

Bước 0 – Loại bỏ nếu $|n| \neq \alpha$, hoặc nếu v không là số nguyên tố lẻ, hoặc nếu $|R| \neq (|v| - 1) \times t$, hoặc nếu $|S| \neq |n| \times t$, hoặc nếu *Id* hết hạn hoặc bị thu hồi.



Hình 5 – Kiểm tra với GQ1

Bước 1 – Biến đổi *Id* thành một giá trị đặc trưng gồm $\gamma = |n|$ bit theo cơ chế định dạng sử dụng.

Xâu bit này biểu diễn một số công khai $G (0 < G < n)$.

CHÚ THÍCH Số G trong mỗi lần tạo có thể được lưu trong bộ nhớ cache để tiếp tục sử dụng.

Bước 2 – Chia R thành t xâu gồm $|v| - 1$ bit là $R_1 \| R_2 \| \cdots \| R_t$ và S thành t xâu gồm $|n|$ bit là $S_1 \| S_2 \| \cdots \| S_t$. Mỗi xâu R_i hoặc S_i biểu diễn một số, cũng ký hiệu là R_i hoặc S_i. Loại bỏ nếu $S_i = 0$ hoặc $\geq n$.

Với i từ 1 đến t, tính toán $S_i^{\gamma} \times G^{R_i} mod\ n$ và biểu diễn bằng một xâu gồm $|n|$ bit, ký hiệu là W_i^*.

Tạo ra một xâu gồm $|n| \times t$ bit, ký hiệu là W^* với $W_1^* \| W_2^* \| \cdots \| W_t^*$.

Bước 3 – Tạo ra một xâu bit, ký hiệu là H^* theo biến thể băm sử dụng.

$$H^* = h(W^* \| M) \text{trong biến thể đầu tiên}$$
$$h(W^* \| h(M)) \text{trong biến thể thứ hai}$$

$$h(h(W^*) \| M) \text{trong biến thể thứ ba}$$
$$h(h(W^*) \| h(M)) \text{trong biến thể thứ tư}$$

Tạo ra một xâu bit, ký hiệu là R^*, gồm $(|v| - 1) \times t$ bit trái nhất của H^*.

Bước 4 – Chấp nhận hoặc loại bỏ phụ thuộc vào *R* và R^* giống nhau hay khác nhau.

7.4 Cơ chế định dạng

Biến đổi một chuỗi dữ liệu định dạng *Id* thành một giá trị đặc trưng gồm γ bit, ký hiệu là *F*.

1) Băm *Id* thành một xâu bit, ký hiệu là *H*. Nối 8 octet có giá trị bằng "00" vào bên trái xâu bit *H*. Băm chuỗi vừa nối thành một xâu bit, ký hiệu là *HH*.

$$H = h(Id) \qquad HH = h('00000000\ 00000000'||H)$$

2) Tạo ra một xâu gồm ít nhất $\gamma - |H|$ bit từ *HH* theo các bước sau sử dụng hai biến: một xâu có độ dài tùy ý, ký hiệu là *String* và một xâu 32 bit, ký hiệu là *Counter*.
   a) Đặt *String* bằng một xâu rỗng.
   b) Đặt *Counter* bằng 0.
   c) Thay thế *String* bằng $String \| h(HH \| Counter)$.
   d) Thay thế *Counter* bằng *Counter* + 1.
   e) Nếu $|H| \times Counter < \gamma - |H|$, thì quay lại bước c.

Tạo ra một xâu có mặt nạ với $\gamma - |H|$ bit trái nhất của *String* trong đó bit trái nhất được đặt bằng 0 và nghịch đảo bit phải nhất.

3) Tạo ra *F* bằng cách nối xâu vừa được tạo mặt nạ vào bên trái của *HH*.
   F = Xâu bit được tạo mặt nạ || *HH*
4) Nếu tất cả $\gamma - 1$ bit trái nhất của *F* bằng 0 (trường hợp rất hiếm gặp), thì quá trình thất bại (xâu bit Id không phù hợp). Ngược lại trả về giá trị *F*.

8 Lược đồ GQ2

8.1 Tập hợp các thành phần dữ liệu cần để ký/kiểm tra

Các quan hệ và ràng buộc chuỗi được áp dụng cho các thành phần dữ liệu sau:

- Tham số an toàn, số các số cơ sở và tham số độ dài chữ ký;
- Tập hợp các số cơ sở;
- Tập hợp các thừa số nguyên tố khác nhau;
- Số mô-đun;
- Tham số thay thế;
- Tập hợp các số mũ bí mật;
- Tập hợp các số bí mật.

Tham số an toàn ký hiệu là *k*. Số các số cơ sở ký hiệu là *m*. Tham số độ dài chữ ký ký hiệu là *t*. Tích của $k \times m \times t$ sẽ nhỏ hơn hoặc bằng $|H|$.

CHÚ THÍCH Với $k \times m \times t = 80$, các giá trị thường gặp của k, m và t là $(80,1,1), (20,4,1), (16,5,1)(10,8,1)$ và $(8,10,1)$.

Tập hợp các số cơ sở được ký hiệu là $g_1, g_2, \ldots, g_m$ được sắp xếp theo thứ tự tăng dần. Đây chính là *m* số nguyên tố khác nhau nhỏ hơn 256.

Tập hợp các thừa số nguyên tố khác nhau ký hiệu là $p_1, p_2, \ldots, p_f$ được sắp xếp theo thứ tự tăng dần ($f > 1$). Mỗi thừa số nguyên tố p_j có công thức như sau $p_j = 1 + q_j \times 2^{h_j}$ trong đó q_j là số lẻ (tức là $p_j - 1$ chia hết cho 2^{h_j}, nhưng không chia hết cho 2^{h_j+1}).

Có ít nhất một số cơ sở g_i và hai thừa số nguyên tố p_j và p_{jj} có ký hiệu Legendre như sau:

- Nếu $h_j = h_{jj}$, thì $(g_i|p_j) = -(g_i|p_{jj})$.
- Nếu $h_j > h_{jj}$, thì $(g_i|p_j) = -1$.

CHÚ THÍCH Quá trình sinh khóa theo một trong hai cách sau:

- Cho trước một tập hợp các số cơ sở, ví dụ: các số nguyên tố đầu tiên 2, 3, 5, 7, 11, 13, 17, 19..., lựa chọn một tập hợp các thừa số nguyên tố.
- Cho trước một tập hợp các thừa số nguyên tố, ví dụ các thừa số nguyên tố của mô-đun RSA hoặc RW, lựa chọn một tập hợp các số cơ sở.

Số mô-đun, ký hiệu là n, là tích của các thừa số nguyên tố ($n = p_1 \times \ldots \times p_f$). Độ lớn của nó bằng α bit.

Tham số thay thế, ký hiệu là b, là số lớn nhất từ f số h_1 đến h_f.

Tập hợp các phần tử bí mật ký hiệu là $Q_{1,1}$ đến $Q_{m,f}$. Với mỗi thừa số nguyên tố p_j, m phần tử bí mật (mỗi phần tử tương ứng với một số cơ sở g_i) được tính toán như sau:

$$s_j = \left(\frac{q_j + 1}{2}\right)^{b+k} mod\ q_j; \quad u_j = q_j - s_j; \quad Q_{i,j} = g_i^{2^b \times u_j} mod\ p_j$$

Tập hợp các số bí mật ký hiệu là Q_1 đến Q_m. Với i từ 1 đến m, số Q_i là hợp số CRT (xem mục 5.3) của $Q_{i,1}$ đến $Q_{i,f}$.

CHÚ THÍCH 1 $q = lcm(q_1, \ldots, q_f); \quad s = \left(\frac{q+1}{2}\right)^{b+k} mod\ q; \quad u = q - s; \quad Q_i = g_i^{2^b \times u} mod\ n.$

CHÚ THÍCH 2 Với $v = 2^{b+k}$ và $G_i = g_i^{2^b}$, mọi cặp G_i và Q_i kiểm tra $G_i \times Q_i^v mod\ n = 1$.

CHÚ THÍCH 3 Nếu $\chi_i = g_i \times Q_i^{2^k} mod\ n \neq 1$, thì với b là bình phương mod n của χ_i, số lượng phần tử đơn vị trước đó là một căn bậc hai mod n của phần tử đơn vị, ký hiệu là ω_i, là nghiệm tầm thường ($\omega_i = n - 1$) hoặc không tầm thường ($1 < \omega_i < n - 1$). Nếu g_i kiểm tra ràng buộc trên các ký hiệu Legendre, thì ω_i là không tầm thường, tức là n chia hết $\omega_i^2 - 1$, nhưng không chia hết $\omega_i - 1$ hoặc $\omega_i + 1$, do đó biết được phân tích không tầm thường $n = \gcd(n, \omega_i - 1) \times \gcd(n, \omega_i + 1)$.

CHÚ THÍCH 4 Nếu ký hiệu Jacobi bằng $(\pm g_i|n) = -1$ (có nghĩa là $n \equiv 1\ mod\ 4$, thì ω_i là không tầm thường.

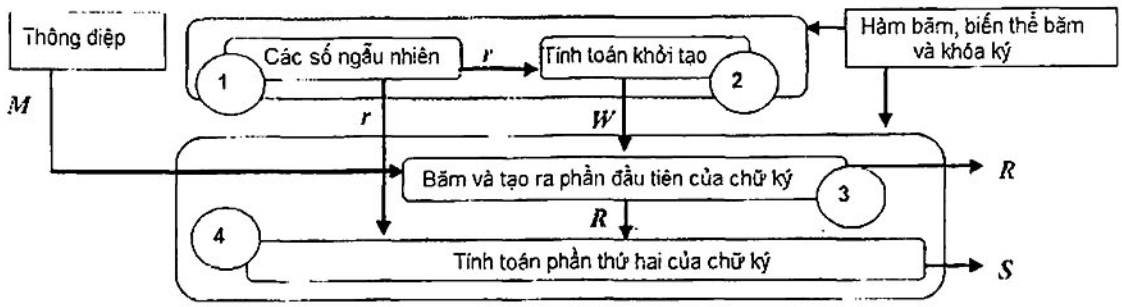
Quá trình ký yêu cầu một hàm băm (xem mục 5.1), một biến thể băm và một khóa ký. Khóa ký có một trong hai dạng sau:

- Với CRT: k, t, b, p_1 đến p_f, $f - 1$ hệ số CRT (xem mục 5.3) và $Q_{1,1}$ đến $Q_{m,f}$ (k, t, b công khai).
- Không có CRT: k, t, b, n và Q_1 đến Q_m (k, t, b, n công khai).

Quá trình kiểm tra yêu cầu một tập hợp các tham số miền và một khóa kiểm tra. Các tham số miền bao gồm k và có thể có m (mặc định, $m = 1$) và t (mặc định, $t = 1$). Các tham số miền hoặc khóa kiểm tra sẽ bao gồm $Indic(h)$, và có thể bao gồm $g_1, g_2 \ldots g_m$ (mặc định, m số nguyên tố đầu tiên, tức là 2,3,5,7,11 ...), α (mặc định, $\alpha = |n|$) và $Indic$(variant) (mặc định, là biến thể đầu tiên). Khóa kiểm tra bao gồm n và có thể có b (mặc định, $b = 1$).

8.2 Cơ chế ký

Cơ chế ký được minh họa trong hình 6, sử dụng một hàm băm, một biến thể băm và một khóa ký để ký một thông điệp (một xâu bit, ký hiệu M), tức là tạo ra chữ ký của M (hai xâu bit, ký hiệ là R và S).



Hình 6 – Ký với GQ2

Bước 1 – Tạo ra các số ngẫu nhiên (thường ký hiệu là r trong hình 1) bằng một trong hai cách sau:

- Với CRT, với j từ 1 đến f, chọn t xâu gồm $|p_j|$ bit ngẫu nhiên. Biểu diễn các số và giữ bí mật, ký hiệu là $r_{1,1}$ đến $r_{t,f}$.
  CHÚ THÍCH Xác suất để một xâu gồm $|p_j|$ bit ngẫu nhiên bằng 0 hoặc p_i là không đáng kể.
- Không có CRT, chọn t xâu gồm $|n|$ bit ngẫu nhiên. Biểu diễn các số và giữ bí mật, ký hiệu là r_1 đến r_t.
  CHÚ THÍCH Xác suất để một xâu gồm $|n|$ bit ngẫu nhiên bằng 0 hoặc bội của một thừa số nguyên tố bất kỳ của n là không đáng kể.

Bước 2 – Tạo ra các xâu bit, ký hiệu là W_1 đến W_t bằng một trong hai cách sau.

- Với CRT, với i từ 1 đến t và j từ 1 đến f, tính toán $W_{i,j} = r_{i,j}^{2^{b+k}} mod\ p_j$. Với i từ 1 đến t, biểu diễn hợp số CRT (xem mục 5.3) của $W_{i,1}$ đến $W_{i,f}$ bằng một xâu gồm $|n|$ bit, ký hiệu là W_i.
- Không có CRT, với i từ 1 đến t, tính toán $r_i^{2^{b+k}} mod\ n$ và biểu diễn nó bằng một xâu gồm $|n|$ bit, ký hiệu là W_i.

Tạo ra một xâu bit, ký hiệu là W, với $W_1 \parallel W_2 \parallel \cdots \parallel W_t$ $(|n| \times t\ bit)$.

Bước 3 – Tạo ra một xâu bit, ký hiệu là H theo biến thể băm sử dụng.

$$
\begin{aligned}
&H = h(W \parallel M \text{trong biến thể đầu tiên}\\
&h(W \parallel h(M)) \text{trong biến thể thứ hai}\\
&h(h(W) \parallel M) \text{trong biến thể thứ ba}\\
&h(h(W) \parallel h(M)) \text{trong biến thể thứ tư}
\end{aligned}
$$

Tạo ra phần đầu tiên của chữ ký, ký hiệu là R gồm $k \times m \times t$ bit trái nhất của H.

Bước 4 – Tách R thành t xâu gồm $k \times m$ bit, cụ thể gồm $R_1 \parallel R_2 \parallel \cdots \parallel R_t$. Tách mỗi R_i thành m xâu gồm k bit như sau $R_{i,1} \parallel R_{i,2} \parallel \cdots \parallel R_{i,m}$. Mỗi xâu $R_{i,j}$ gồm k bit, từ bit trái nhất ký hiệu là $R_{i,j,1}$ đến bit phải nhất ký hiệu là $R_{i,j,k}$. Mỗi xâu $R_{i,j}$ biểu diễn một số, cũng ký hiệu là $R_{i,j}$ $(< 2^k)$.

Tạo ra các số, ký hiệu S_1 đến S_t bằng một trong hai cách sau.

- Với CRT, với i từ 1 đến t và j từ 1 đến f, tính toán $S_{i,j} = r_{i,j} \times Q_{1,j}^{R_{i,1}} \times ... \times Q_{m,j}^{R_{i,m}} \bmod p_j$. Với i từ 1 đến t, số S_i là hợp số CRT (xem mục 5.3) của $S_{i,1}$ đến $S_{i,f}$.
- Không có CRT, với i từ 1 đến t, tính toán $S_i = r_i \times Q_1^{R_{i,1}} \times ... \times Q_m^{R_{i,m}} \bmod n$.

Số S_i bất kỳ có thể được thay thế bằng $n - S_i$.

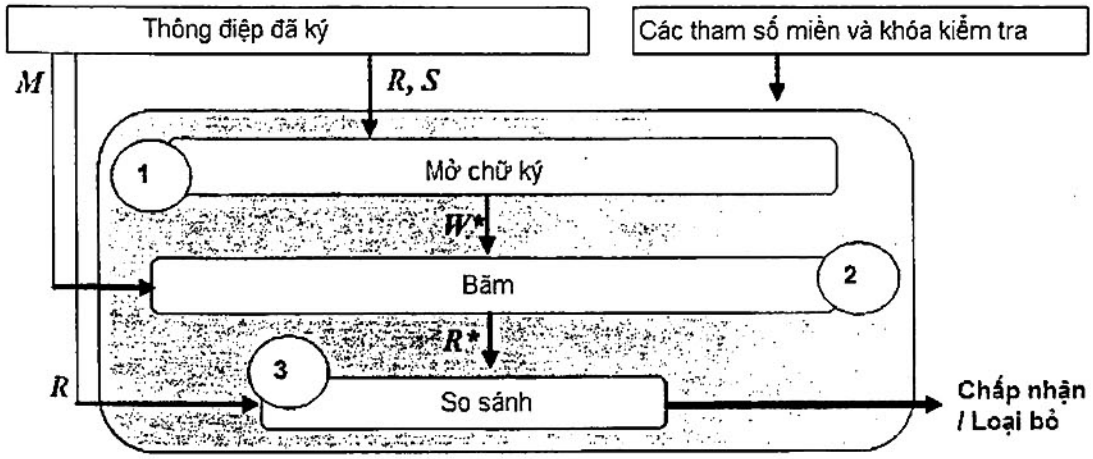
Biểu diễn từng số S_i bằng một xâu gồm $|n|$ bit, cũng ký hiệu là S_i.

Tạo ra phần thứ hai của chữ ký, ký hiệu là S, với $S_1 \| S_2 \| \cdots \| S_t$ (gồm $|n| \times t$ bit).

8.3 Cơ chế kiểm tra

Cơ chế kiểm tra được minh họa trong hình 7 sử dụng một tập hợp các tham số miền và một khóa kiểm tra (xem bảng 1), với khóa ưu tiên (xem mục 5.2) để kiểm tra một thông điệp và chữ ký của thông điệp đó, tức là có ba xâu bit, ký hiệu là M, R và S.

Bước 0 – Loại bỏ nếu $|n| \neq \alpha$, hoặc nếu $|R| \neq k \times m \times t$, hoặc nếu $|S| \neq |n| \times t$, hoặc nếu m các số cơ sở không phải là các số nguyên tố khác nhau nhỏ hơn 256.



Hình 7 – Kiểm tra với GQ2

Bước 1 – Tách S thành t xâu gồm $|n|$ bit như sau $S_1 \| S_2 \| \cdots \| S_t$. Mỗi xâu bit S_i biểu diễn một số, cũng ký hiệu là S_i. Loại bỏ nếu bất kỳ $S_i = 0$ hoặc $\geq n$.

Tách R thành t xâu gồm $k \times m$ bit như sau $R_1 \| R_2 \| \cdots \| R_t$. Tách mỗi R_i thành m xâu gồm k bit như sau $R_{i,1} \| R_{i,2} \| \cdots \| R_{i,m}$. Mỗi xâu $R_{i,j}$ gồm k bit, từ bit trái nhất ký hiệu là $R_{i,j,1}$ đến bit phải nhất ký hiệu là $R_{i,j,k}$. Mỗi xâu $R_{i,j}$ biểu diễn một số, cũng ký hiệu là $R_{i,j}$ $(< 2^k)$.

Với i từ 1 đến t, tính toán $S_i^{2^{b+k}} \times \left(g_1^{2^b}\right)^{R_{i,1}} \times ... \times \left(g_m^{2^b}\right)^{R_{i,m}} \bmod n$ và biểu diễn bằng một xâu gồm $|n|$ bit, ký hiệu là W_i^*.

CHÚ THÍCH Bắt đầu từ một tập giá trị bằng phép nhân của S, k được xen kẽ với $b + k$ phép bình phương. Sau giá trị bình phương thứ l là phép nhân thứ l: với j từ 1 đến m, bit tương ứng (ký hiệu là $R_{i,j,l}$ ở vị trí mà giá trị hiện thời là bội số của g_j (bit được đặt bằng 1) hoặc không là bội số của g_j (bit được đặt bằng 0). Giá trị cuối cùng sau bình phương thứ $b + k$ là W^*.

Tạo ra một xâu bit, ký hiệu là W^* với $W_1^* \| W_2^* \| \cdots \| W_t^*$ (gồm $|n| \times t$ bit).

Bước 2 – Tạo ra một xâu bit, ký hiệu là H^* theo biến thể băm sử dụng.

$$H^* = h(W^* \| M) \text{trong biến thể đầu tiên}$$
$$h(W^* \| h(M)) \text{trong biến thể thứ hai}$$
$$h(h(W^*)M) \text{ trong biến thể thứ ba}$$
$$h(h(W^*) \| h(M)) \text{trong biến thể thứ tư}$$

Tạo ra một xâu bit, ký hiệu là R^*, gồm $k \times m \times t$ bit trái nhất của H^*.

Bước 3 – Chấp nhận hoặc loại bỏ phụ thuộc vào R và R^* giống nhau hay khác nhau.

9 Lược đồ GPS1

9.1 Tập hợp các thành phần dữ liệu cần để ký/kiểm tra

Các quan hệ và ràng buộc chuỗi được áp dụng cho các thành phần dữ liệu sau:

- Số mô-đun;
- Tập hợp các thừa số nguyên tố;
- Số bí mật;
- Số cơ sở;
- Số công khai.

Số mô-đun ký hiệu là n. Độ lớn của nó bằng α bit. Không thể biết được phân tích của số mô-đun thành các thừa số nguyên tố.

Tập hợp các thừa số nguyên tố ký hiệu là $p_1, p_2 \ldots p_f$ được sắp xếp theo thứ tự tăng dần ($f > 1$).

Số bí mật, ký hiệu là Q, được biểu diễn bởi một xâu gồm $|H|$ bit ngẫu nhiên.

Số cơ sở ký hiệu là g. Cấm sử dụng giá trị $g = 0$ và $g = 1$.

CHÚ THÍCH Giá trị $g = 2$ có nhiều ưu điểm trong thực tế ứng dụng.

Số công khai, ký hiệu là G được tạo ra bằng một trong hai cách sau :

- Với CRT, với i từ 1 đến f, tính toán $Q_i = Q \; mod \; (p_i - 1)$ và $G_i = g^{Q_i} \; mod \; p_i$. Số G là hợp số CRT (xem mục 5.3) của G_1 đến G_f.
- Không có CRT, tính toán $G = g^Q \; mod \; n$.

Quá trình ký yêu cầu một hàm băm (xem mục 5.1), một biến thể băm và một khóa ký. Khóa ký được tạo ra bằng một trong hai cách sau :

- Với CRT: P_1 đến P_f, $f - 1$ hệ số CRT (xem mục 5.3), g và Q (g công khai);
- Không có CRT: n, g và Q (n, g công khai).

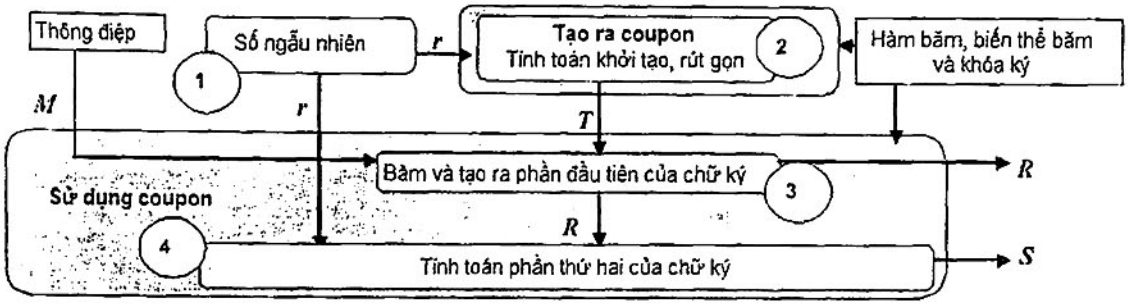
Quá trình kiểm tra yêu cầu một tập hợp các tham số miền và một khóa kiểm tra. Các tham số miền hoặc khóa kiểm tra bao gồm n và $Indic(h)$, và có thể gồm g (mặc định, $g = 2$), α (mặc định, $\alpha = |n|$) và $Indic(variant)$ (mặc định, biến thể thứ ba). Khóa kiểm tra bao gồm G.

9.2 Cơ chế ký

9.2.1 Giới thiệu chung

Cơ chế ký được minh họa bằng hình 8 sử dụng một hàm băm, một biến thể băm và một khóa ký để ký một thông điệp (một xâu bit, ký hiệu là M), tức là tạo ra một chữ ký của M (hai xâu bit, ký hiệu là R và S).

Mỗi người ký được cung cấp một hoặc nhiều coupon. Theo định nghĩa, một coupon là một xâu bit, độc lập với thông điệp, tiền tính toán từ một xâu các bit ngẫu nhiên, phải giữ bí mật và chỉ sử dụng một lần.



Hình 8 – Ký với GPS1

9.2.2 Số ngẫu nhiên

Bước 1 – Lựa chọn một xâu của $2|H| + 80$ bit ngẫu nhiên.

Nó biểu diễn một số ngẫu nhiên, cần phải giữ bí mật, ký hiệu là r.

Quá trình tạo số ngẫu nhiên liên quan đến quá trình tạo coupon hoặc sử dụng coupon.

- Bước 2 sử dụng r và n hoặc p_1 đến p_f.
- Bước 4 sử dụng r và Q.

CHÚ THÍCH Nếu thiết bị tạo chữ ký tạo ra các xâu bit giả ngẫu nhiên bằng một hàm tất định của các chỉ số coupon, thì nó lưu trữ chỉ số của coupon được tạo ra cuối cùng và chỉ số của coupon được sử dụng cuối cùng. Ngược lại, nó lưu trữ các xâu bit cho đến khi sử dụng các coupon.

9.2.3 Tạo coupon

Bước 2 – Tạo ra một xâu bit, ký hiệu là W bằng một trong hai cách sau.

- Với CRT, với i từ 1 đến f, tính toán $r_i = r \; mod(p_i - 1)$ và $W_i = g^{r_i} \; mod \; p_i$. Biểu diễn hợp số CRT (xem mục 5.3) của W_1 đến W_f bằng một xâu gồm $|n|$ bit, ký hiệu là W.
- Không có CRT, tính toán $g^r \; mod \; n$ và biểu diễn nó bằng một xâu gồm $|n|$ bit, ký hiệu là W.

Coupon, ký hiệu là T được đặt bằng mã băm $h(W)$.

9.2.4 Sử dụng coupon

Bước 3 – Tạo ra phần đầu tiên của chữ ký, ký hiệu là R theo biến thể băm sử dụng.

$$R = h(T||M), \text{ tức là } = h(h(W)||M) \text{ trong biến thể thứ ba}$$

$h(T||h(M))$, tức là $= h(h(W)||h(M))$ trong biến thể thứ tư.

Bước 4 – Xâu bit R biểu diễn một số, cũng ký hiệu là R.

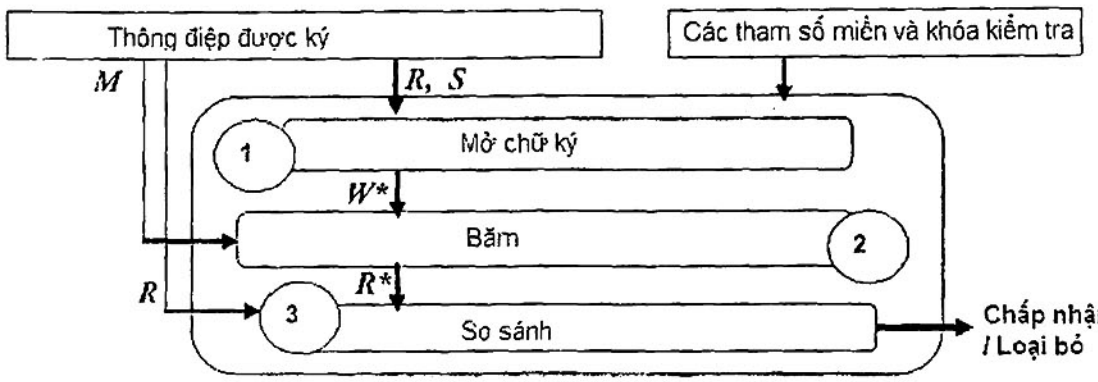
Tính toán $S = r - R \times Q$.

Phần thứ hai của chữ ký, cũng ký hiệu là S là một xâu gồm $2|H| + 80$ bit biểu diễn số S.

9.3 Cơ chế kiểm tra

Cơ chế kiểm tra được minh họa bằng hình 9 sử dụng một tập hợp các tham số miền và một khóa kiểm tra (xem bảng 1) với khóa ưu tiên (xem 5.2) để kiểm tra một thông điệp và một chữ ký của thông điệp khác, tức là có ba xâu bit, ký hiệu là M, R và S.

Bước 0 – Loại bỏ nếu $|n| \neq \alpha$, hoặc nếu $g = 0$ hoặc 1, hoặc nếu $|R| \neq |H|$, hoặc nếu $|S| \neq 2|H| + 80$.



Hình 9 – Kiểm tra với GPS1

Bước 1 – Các xâu bit R và S biểu diễn hai số, cũng ký hiệu là R và S.

Tính toán $G^R \times g^S mod\, n$ và biểu diễn nó bằng một xâu gồm $|n|$ bit, ký hiệu là W^*.

Bước 2 – Tạo ra một xâu bit, ký hiệu là R^* theo biến thể băm sử dụng.

$$R^* = h(h(W^*) \parallel M) trong\ biến\ thể\ thứ\ ba$$
$$h(h(W^*) \parallel h(M))\ trong\ biến\ thể\ thứ\ tư$$

CHÚ THÍCH Mã băm $h(W^*)$ là coupon đã được khôi phục lại.

Bước 3 – Chấp nhận hoặc loại bỏ phụ thuộc vào R và R^* giống nhau hay khác nhau.

10 Lược đồ GPS2

10.1 Tập hợp các thành phần dữ liệu cần để ký/kiểm tra

Các quan hệ và ràng buộc chuỗi được áp dụng cho các thành phần dữ liệu sau:

- Số mũ kiểm tra;
- Tập hợp các thừa số nguyên tố khác nhau;
- Số mô-đun;
- Số bí mật;

- Số cơ sở.

Số mũ kiểm tra, ký hiệu là v là một số nguyên tố do đó $|v| = |H| + 1$.

CHÚ THÍCH Nếu $|H| = 160$, thì giá trị $v = 2^{160} + 7$ có nhiều ưu điểm trong thực tế sử dụng.

Tập hợp các thừa số nguyên tố khác nhau được ký hiệu là $p_1, p_2 \ldots p_f$ được sắp xếp theo thứ tự tăng dần ($f > 1$).

Với *i* từ 1 đến *f*, *v* không chia hết $p_i - 1$.

Số mô-đun, ký hiệu là n là tích của các thừa số nguyên tố ($n = p_1 \times \ldots \times p_f$). Độ lớn của nó bằng α bit.

Số bí mật được ký hiệu là Q. Nó là một số nguyên dương bất kỳ (thường sử dụng số nhỏ nhất) do đó $v \times Q - 1$ là bội số của $lcm(p_1 - 1, \ldots, p_f - 1)$, được biểu diễn bằng một xâu gồm $|n|$ bit.

CHÚ THÍCH Số Q được định nghĩa giống như số mũ ký quy định trong 6.1.

Số cơ sở được ký hiệu là g. Cấm sử dụng giá trị $g = 0$ và $g = 1$.

CHÚ THÍCH Giá trị $g = 2$ có nhiều ưu điểm trong thực tế sử dụng.

Quá trình ký yêu cầu một hàm băm (xem mục 5.3), một biến thể băm và một khóa ký. Khóa ký được tạo ra bằng một trong hai cách sau:

- Với CRT: v, p_1 đến $p_f, f - 1$ hệ số CRT (xem mục 5.3), Q và g (v, g công khai);
- Không có CRT: v, n, Q và g (v, n, g công khai).

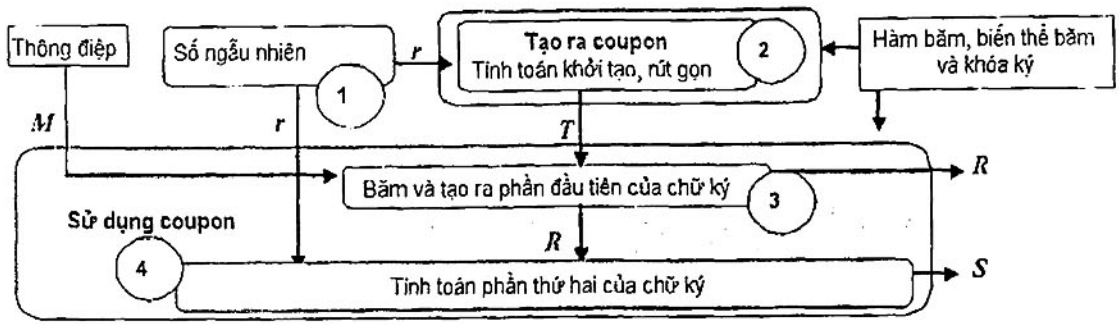
Quá trình kiểm tra yêu cầu một tập hợp các tham số miền và một khóa kiểm tra. Các tham số miền hoặc khóa kiểm tra bao gồm v và $Indic(h)$ và có thể bao gồm g (mặc định $g = 2$), α (mặc định, $\alpha = |n|$) và $Indic(variant)$ (mặc định là biến thể thứ ba). Khóa kiểm tra bao gồm n.

10.2 Cơ chế ký

10.2.1 Giới thiệu chung

Cơ chế ký được minh họa bằng hình 10, sử dụng một hàm băm, một biến thể và một khóa ký để ký một thông báo (một xâu bit, ký hiệu là M) tức là tạo ra một chữ ký của M (hai xâu bit, ký hiệu là R và S).

Mỗi người ký được cung cấp một hoặc nhiều coupon. Theo định nghĩa, một coupon là một xâu bit, độc lập với thông điệp, tiền tính toán từ một xâu các bit ngẫu nhiên, được giữ bí mật và chỉ sử dụng một lần.



Hình 10 – Ký với GPS2

10.2.2 Số ngẫu nhiên

Bước 1 – Lựa chọn một xâu gồm $|n| + |H| + 80$ bit ngẫu nhiên.

Biểu diễn một số ngẫu nhiên cần giữ bí mật, ký hiệu là r.

Quá trình tạo số ngẫu nhiên liên quan đến quá trình tạo coupon hoặc sử dụng coupon.

- Bước 2 sử dụng r, v và n hoặc p_1 đến p_f.
- Bước 4 sử dụng r và Q.

CHÚ THÍCH Nếu thiết bị tạo chữ ký tạo ra các xâu bit giả ngẫu nhiên bằng một hàm tất định của các chỉ số coupon, thì nó lưu trữ chỉ số của coupon được tạo ra cuối cùng và chỉ số của coupon được sử dụng cuối cùng. Ngược lại, nó lưu trữ các xâu bit cho đến khi sử dụng các coupon.

10.2.3 Tạo coupon

Bước 2 – Tạo ra một xâu bit, ký hiệu là W bằng một trong hai cách sau.

- Với CRT, với i từ 1 đến f, tính toán $r_i = v \times r \, mod(p_i - 1)$ và $W_i = g^{r_i} \, mod \, p_i$. Biểu diễn hợp số CRT (xem mục 5.3) của W_1 đến W_f bằng một xâu gồm $|n|$ bit, ký hiệu là W.
- Không có CRT, tính toán $g^{v \times r} mod \, n$ và biểu diễn bằng một xâu gồm $|n|$ bit, ký hiệu là W.

Coupon, ký hiệu là T được đặt bằng mã băm $h(W)$.

10.2.4 Sử dụng coupon

Bước 3 – Tạo ra phần thứ nhất của chữ ký, ký hiệu là R theo biến thể băm sử dụng.

$R = h(T \parallel M)$, tức là $= h(h(W) \parallel M)$ trong biến thể thứ ba

$h(T \parallel h(M))$, tức là $= h(h(W) \parallel h(M))$ trong biến thể thứ tư.

Bước 4 – Xâu bit R biểu diễn một số, cũng ký hiệu là R.

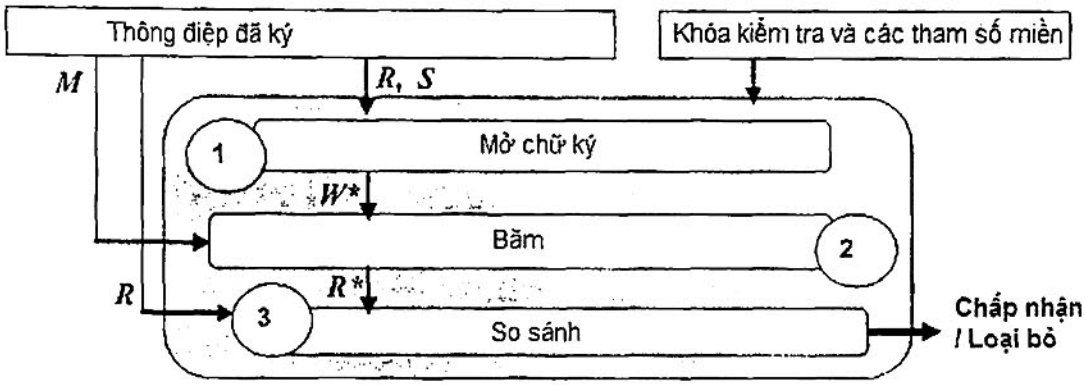
Tính toán $S = r - R \times Q$.

Phần thứ hai của chữ ký, cũng ký hiệu là S là một xâu gồm $|n| + |H| + 80$ bit biểu diễn số S.

10.3 Cơ chế kiểm tra

Cơ chế kiểm tra được minh họa trong hình 11, sử dụng một tập hợp các tham số miền và một khóa kiểm tra (xem bảng 1) với khóa ưu tiên (xem 5.2) để kiểm tra một thông điệp và một chữ ký của thông điệp khác, tức là có ba xâu bit, ký hiệu là M, R và S.

Bước 0 – Loại bỏ nếu $|n| \neq \alpha$, hoặc nếu v không là số nguyên tố lẻ, hoặc nếu $g = 0$ hoặc 1, hoặc nếu $|R| \neq |H|$, hoặc nếu $|S| \neq |n| + |H| + 80$.



Hình 11 – Kiểm tra với GPS2

Bước 1 - Các xâu bit R và S biểu diễn hai số, cũng ký hiệu là R và S.

Tính toán $g^{v \times S+R} \bmod n$ và biểu diễn nó bằng một xâu gồm $|n|$ bit, ký hiệu là W^*.

Bước 2 – Tạo ra một xâu bit, ký hiệu là R^* theo biến thể băm sử dụng.

$$R^* = h(h(W^*)M) \textit{ trong biến thể thứ ba}$$
$$h(h(W^*) \| h(M)) \textit{ trong biến thể thứ tư}$$

CHÚ THÍCH Mã băm $h(W^*)$ là coupon đã được khôi phục lại.

Bước 3 – Chấp nhận hoặc loại bỏ phụ thuộc vào R và R^* giống nhau hay khác nhau.

11 Lược đồ ESIGN

11.1 Tập hợp các thành phần dữ liệu cần để ký/kiểm tra

Các quan hệ và ràng buộc chuỗi được áp dụng cho các thành phần dữ liệu sau:

- Số mũ kiểm tra;
- Cặp các thừa số nguyên tố khác nhau;
- Số mô-đun.

Số mũ kiểm tra, ký hiệu là v lớn hơn hoặc bằng 8, nhưng nhỏ hơn $2^{\alpha-1}$.

CHÚ THÍCH Giá trị $v = 1024$ có nhiều ưu điểm trong thực tế sử dụng.

Cặp các thừa số nguyên tố khác nhau ký hiệu là p_1 và p_2 được sắp xếp theo thứ tự tăng dần. Độ lớn của mỗi thừa số nguyên tố là $\alpha/3$ bit (α là một bội số của 3).

CHÚ THÍCH Cho ví dụ, $\alpha = 1023$ (không bằng 1024), 1536, 2046 hoặc 2049 (không bằng 2048), 2304.

Số mô-đun, ký hiệu là n là tích của $p_1 \times p_2^2$ (sử dụng lại thừa số nguyên tố lớn nhất). Độ lớn của nó là α bit.

- Ước chung lớn nhất của v và n bằng 1, tức là $\gcd(v, n) = 1$.
- Ước chung lớn nhất của $v, p_1 - 1$ và $p_2 - 1$ tối đa bằng α, tức là $\gcd(v, p_1 - 1, p_2 - 1) \leq \alpha$.

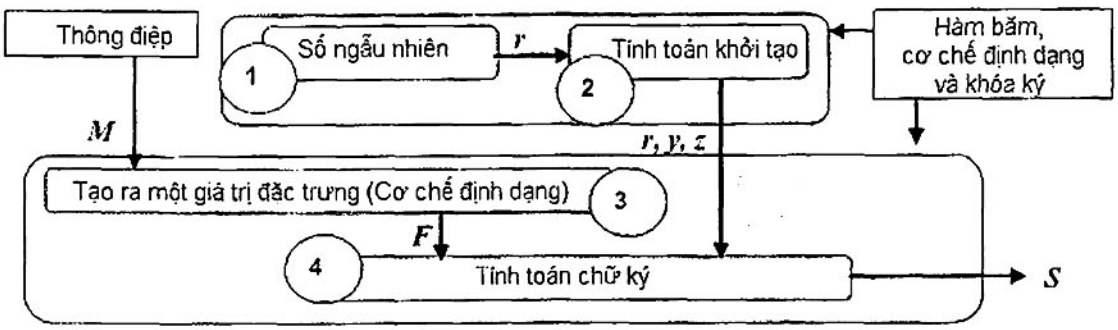
CHÚ THÍCH Giá trị v bất kỳ nhỏ hơn hoặc bằng α thỏa mãn cả hai ràng buộc trên.

Quá trình ký yêu cầu một hàm băm (xem mục 5.1), một cơ chế định dạng và một khóa ký. Khuyến nghị sử dụng cơ chế định dạng quy định trong 11.4. Khóa ký bao gồm v, p_1 và p_2 (v công khai).

Quá trình kiểm tra yêu cầu một tập hợp các tham số miền và một khóa kiểm tra. Các tham số miền hoặc khóa kiểm tra bao gồm v và $Indic(h)$ và có thể bao gồm α (mặc định $\alpha = |n|$) và $Indic(\text{format}, \varepsilon, \tau)$ (mặc định theo mục 11.4). Khóa kiểm tra bao gồm n.

11.2 Cơ chế ký

Cơ chế ký được minh họa trong hình 12, sử dụng một hàm băm, một cơ chế định dạng và một khóa ký để ký một thông điệp (một xâu bit, ký hiệu là M), tức là tạo ra một chữ ký của M (một xâu bit, ký hiệu là S).



Hình 12 – Ký với ESIGN

Bước 1 – Lựa chọn một xâu gồm $2|n|/3$ bit ngẫu nhiên.

Biểu diễn một số ngẫu nhiên, cần giữ bí mật, ký hiệu là r. Số r nhỏ hơn $p_1 \times p_2$.

CHÚ THÍCH Xác suất để một xâu gồm $2|n|/3$ bit ngẫu nhiên bằng 0 hoặc bội số của một thừa số nguyên tố bất kỳ của n là không đáng kể.

Bước 2 – Tính toán hai số, ký hiệu là $y(< n)$ và $z(< p_2)$. Số z cần được giữ bí mật.

$$y = r^v \bmod n$$
$$z = (v \times r^{v-1})^{-1} \bmod p_2$$

CHÚ THÍCH Công thức $z = r \times (v \times y)^{-1} \bmod p_2$ có nhiều ưu điểm tính toán.

Bước 3 – Biến đổi thông điệp M thành một giá trị đặc trưng gồm $\gamma = |n|/3$ bit, ký hiệu là F theo cơ chế định dạng sử dụng. Xâu bit F biểu diễn một số, cũng ký hiệu là F $(0 < F < p_1)$.

Bước 4 – Tính toán một số, ký hiệu là S $(0 < S < n)$.

$$a = \left(2^{\frac{2|n|}{3}} F - y\right) mod\, n$$
$$w = \lceil a/(p_1 \times p_2) \rceil$$

Nếu $w \times p_1 \times p_2 - a \geq 2^{\left(\frac{2|n|}{3}\right)-1}$ (trường hợp này xảy ra trên 50%), thì quay lại bước 1.

$$S = r + (w \times z\, mod\, p_2) \times p_1 \times p_2 mod\, n$$

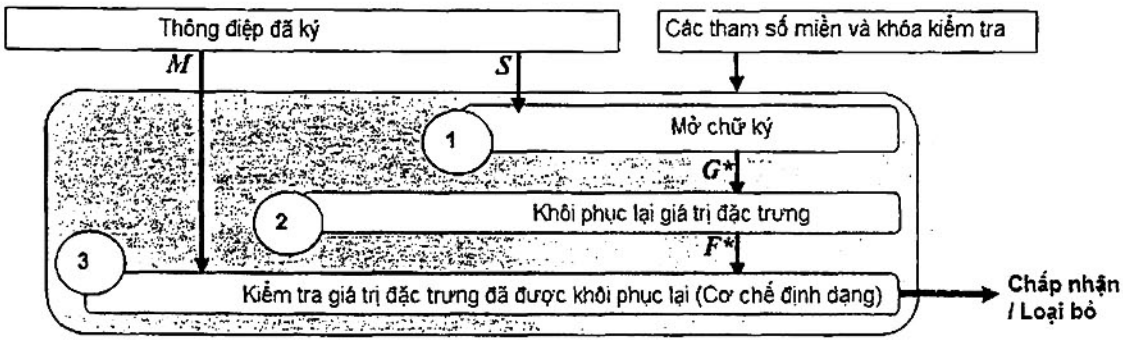
Nếu v là số chẵn, thì số S được thay thế bằng $n - S$.

Chữ ký, cũng ký hiệu là S là xâu bit bất kỳ biểu diễn số S, thường là một xâu gồm $|n|$ bit.

11.3 Cơ chế kiểm tra

Cơ chế kiểm tra được minh họa trong hình 13, sử dụng một tập hợp các tham số miền và một khóa kiểm tra (xem bảng 1) và khóa ưu tiên (xem mục 5.2) để kiểm tra một thông điệp và chữ ký của thông điệp đó, tức là có hai xâu bit, ký hiệu là M và S.

Bước 0 – Loại bỏ nếu α không phải là bội số của 3, hoặc nếu $|n| \neq \alpha$, hoặc nếu $v < 8$, hoặc nếu $v \geq 2^{\alpha-1}$.



Hình 13 – Kiểm tra với ESIGN

Bước 1 – Xâu bit S biểu diễn một số, cũng ký hiệu là S. Loại bỏ nếu $S = 0$ hoặc 1, hoặc nếu $S \geq n - 1$.

Tính toán $S^v mod\, n$ và biểu diễn bằng một xâu gồm $|n|$ bit, ký hiệu là G^*.

Bước 2 – Khôi phục lại giá trị đặc trưng, ký hiệu là F^* chính là $\gamma = |n|/3$ bit trái nhất của G^*.

Bước 3 – Kiểm tra giá trị đặc trưng đã được khôi phục lại F^* theo cơ chế định dạng sử dụng.

11.4 Cơ chế định dạng

Biến đổi thông điệp M thành một giá trị đặc trưng gồm γ bit, ký hiệu là F.

1) Lựa chọn một xâu mới gồm $\varepsilon = |H|$ bit ngẫu nhiên. Nó tạo ra một giá trị salt, ký hiệu là E.
2) Băm M thành một xâu bit, ký hiệu là H. Từ trái sang phải, nối 8 octet có giá trị bằng "00" vào H và giá trị salt E. Băm chuỗi vừa nối thành một xâu bit, ký hiệu là HH.

$$H = h(M) \qquad HH = h(("0000\ 0000\ 0000\ 0000")||H||E)$$

3) Tạo ra một chuỗi gồm ít nhất $\gamma - |H|$ bit từ *HH* theo các bước sau sử dụng hai biến: một xâu có độ dài tùy ý, ký hiệu là *String*, và một xâu 32 bit, ký hiệu là *Counter*.

a) Đặt *String* bằng một xâu rỗng.
b) Đặt *Counter* bằng 0.
c) Thay *String* bằng $String||h(HH||Counter)$.
d) Thay *Counter* bằng Counter + 1.
e) Nếu $|H| \times Counter < \gamma - |H|$, thì quay lại bước c.

Tạo ra một giá trị mặt nạ với $\gamma - |H|$ bit trái nhất của *String* trong đó bit trái nhất có giá trị bắt buộc bằng 0.

4) Tạo ra một xâu trung gian gồm $\gamma - |H|$ bit được nối từ trái sang phải theo thứ tự như sau:

- $\gamma - |H| - \varepsilon - 1$ bit 0;
- Một bit giới hạn bằng 1;
- Giá trị salt *E*.

5) Bằng cách thực hiện phép XOR, áp dụng mặt nạ vào xâu trung gian, từ đó tạo ra một xâu bit đã được tạo mặt nạ.
6) Tạo ra *F* bằng cách nối xâu bit đã được tạo mặt nạ vào bên trái của *HH*.

F = Xâu bit được tạo mặt nạ || *HH*

7) Nếu γ bit của *F* đều bằng 0 (trường hợp rất hiếm gặp), thì quay lại bước 1 (giá trị salt *E* không phù hợp). Ngược lại, trả về *F*.

Kiểm tra giá trị đặc trưng đã được khôi phục lại gồm γ bit, ký hiệu là F^* tương ứng với thông điệp *M*.

1) Nếu γ bit của F^* đều bằng 0, thì loại bỏ. Ngược lại, tiếp tục.
2) Từ $|H|$ bit phải nhất của F^*, ký hiệu là HH^*, tạo mặt nạ gồm $\gamma - |H|$ bit giống bước 3 ở trên.
3) Thực hiện phép XOR, áp dụng mặt nạ tới $\gamma - |H|$ bit trái nhất của F^*, để khôi phục lại một xâu trung gian trong đó bit giới hạn là bit đầu tiên được đặt bằng 1 tính từ trái sang.

- Nếu còn lại ε bit ở bên phải của bit giới hạn trong xâu trung gian đã được khôi phục lại, thì tạo ra một xâu bit, ký hiệu là E^*.
- Ngược lại, thì loại bỏ.

4) Băm *M* thành một xâu bit, ký hiệu là *H*. Từ trái sang phải, nối 8 octet có giá trị "00", *H* và E^*. Băm chuỗi vừa nối thành một xâu bit, ký hiệu là *HH*.

$$H = h(M) \qquad HH = h(("0000\ 0000\ 0000\ 0000")||H||E^*)$$

5) Chấp nhận hoặc loại bỏ tùy thuộc vào *HH* và HH^* giống nhau hay khác nhau.

Phụ lục A
(Quy định)
Định danh đối tượng

A.1 Định nghĩa

Bảng A.1 tổng hợp các tùy chọn được quy định trong tiêu chuẩn này.

Bảng A.1 – Các tùy chọn được quy định trong tiêu chuẩn này này

Lược đồ	Cơ chế định dạng [a]	Biến thể băm
RSA	formatPSS, formatD1, formatD2	novariant
RW	formatPSS, formatD1, formatD2	novariant
GQ1	formatPSS	variant1, variant2, variant3, variant4
GQ2	noformat	variant1, variant2, variant3, variant4
GPS1	noformat	variant3, variant4
GPS2	noformat	variant3, variant4
ESIGN	formatPSS	novariant

[a] Tiêu chuẩn này quy định ba dạng cài đặt của formatPSS: 6.4 ($\varepsilon = 0$ hoặc $|H|$ và $\tau = 8$ hoặc 16) đối với RSA và RW, 7.4 ($\varepsilon = \tau = 0$) đối với GQ1 và 11.4 ($\varepsilon = |H|$ và $\tau = 0$) đối với ESIGN. Phụ lục D quy định formatD1 ($\varepsilon = 0$ và $\tau = 8$ hoặc 16) và formatD2 ($\varepsilon = 64$ và $\tau = 8$).

Mô-đun sau phù hợp với ký hiệu được quy định trong ISO/IEC 8824-1 [25].

```
IntegerFactorizationBasedDigitalSignaturesWithAppendix {
  iso(1) standard(0) digital-signatures-with-appendix(14888) part2(2)
    asn1-module(1) integer-factorization-based-mechanisms(0) version1(1)}
DEFINITIONS EXPLICIT TAGS ::= BEGIN
-- XUẤT RA TOÀN BỘ--;
IMPORTS
  HashFunctions
    FROM DedicatedHashFunctions {
      iso(1) standard(0) hash-functions(10118) part(3)
        asn1-module(1) dedicated-hash-functions(0) } ;
SignatureWithAppendix ::= SEQUENCE {
  algorithm ALGORITHM.&id({SchemeOptions}),
  parameters ALGORITHM.&Type({SchemeOptions}{@algorithm}) OPTIONAL
```

```
}
SchemeOptions ALGORITHM ::= {
  RSA |
  RW  |
  GQ1 |
  GQ2 |
  GPS1|
  GPS2|
  ESIGN,
  ... -- Expect additional signature scheme objects
}
-- Integer factorization signature scheme object sets -

-- RSA scheme options --
  RSA ALGORITHM ::= {
  rsa-formatPSS-novariant |
  rsa-formatD1-novariant |
  rsa-formatD2-novariant,
  ... -- Expect additional RSA scheme objects --
}
rsa-formatPSS-novariant ALGORITHM ::= {
  OID id-rsa-formatPSS-novariant PARMS HashFunctions
}
rsa-formatD1-novariant ALGORITHM ::= {
  OID id-rsa-formatD1-novariant PARMS HashFunctions
}
rsa-formatD2-novariant ALGORITHM ::= {
  OID id-rsa-formatD2-novariant PARMS HashFunctions
}
-- RW scheme options --
RW ALGORITHM ::= {
  rw-formatPSS-novariant |
  rw-formatD1-novariant |
  rw-formatD2-novariant,
  ... -- Expect additional RW scheme objects --
}
rw-formatPSS-novariant ALGORITHM ::= {
  OID id-rw-formatPSS-novariant PARMS HashFunctions
```

```
}
rw-formatD1-novariant ALGORITHM ::= {
  OID id-rw-formatD1-novariant PARMS HashFunctions
}
rw-formatD2-novariant ALGORITHM ::= {
  OID id-rw-formatD2-novariant PARMS HashFunctions
}
-- GQ1 scheme options --
GQ1 ALGORITHM ::= {
  gq1-formatPSS-variant1 |
  gq1-formatPSS-variant2 |
  gq1-formatPSS-variant3 |
  gq1-formatPSS-variant4,
  ... -- Expect additional GQ1 scheme objects --
}
gq1-formatPSS-variant1 ALGORITHM ::= {
  OID id-gq1-formatPSS-variant1 PARMS HashFunctions
}
gq1-formatPSS-variant2 ALGORITHM ::= {
  OID id-gq1-formatPSS-variant2 PARMS HashFunctions
}
gq1-formatPSS-variant3 ALGORITHM ::= {
  OID id-gq1-formatPSS-variant3 PARMS HashFunctions
}
gq1-formatPSS-variant4 ALGORITHM ::= {
  OID id-gq1-formatPSS-variant4 PARMS HashFunctions
}

-- Lựa chọn lược đồ GQ2 --
GQ2 ALGORITHM ::= {
  gq2-noformat-variant1 |
  gq2-noformat-variant2 |
  gq2-noformat-variant3 |
  gq2-noformat-variant4,
  ... -- Expect additional GQ2 scheme objects --
}
gq2-noformat-variant1 ALGORITHM ::= {
  OID id-gq2-noformat-variant1 PARMS HashFunctions
```

```
}
gq2-noformat-variant2 ALGORITHM ::= {
  OID id-gq2-noformat-variant2 PARMS HashFunctions
}
gq2-noformat-variant3 ALGORITHM ::= {
  OID id-gq2-noformat-variant3 PARMS HashFunctions
}
gq2-noformat-variant4 ALGORITHM ::= {
  OID id-gq2-noformat-variant4 PARMS HashFunctions
}

-- Các lựa chọn lược đồ GPS1 --
GPS1 ALGORITHM ::= {
  id-gps1-noformat-variant3 |
  id-gps1-noformat-variant4,
  ... -- Expect additional GPS1 scheme objects --
}
gps1-noformat-variant3 ALGORITHM ::= {
  OID id-gps1-noformat-variant3 PARMS HashFunctions
}
gps1-noformat-variant4 ALGORITHM ::= {
  OID id-gps1-noformat-variant4 PARMS HashFunctions
}

-- Các lựa chọn lược đồ GPS2 --
GPS2 ALGORITHM ::= {
  id-gps2-noformat-variant3 |
  id-gps2-noformat-variant4,
  ... -- Expect additional GPS2 scheme objects --
}
gps2-noformat-variant3 ALGORITHM ::= {
  OID id-gps2-noformat-variant3 PARMS HashFunctions
}
gps2-noformat-variant4 ALGORITHM ::= {
  OID id-gps2-noformat-variant4 PARMS HashFunctions
}

-- Các lựa chọn của lược đồ ESIGN --
```

```
ESIGN ALGORITHM ::= {
  esign-formatPSS-novariant,
  ... -- Expect additional ESIGN scheme objects --
}
esign-formatPSS-novariant ALGORITHM ::= {
  OID id-esign-formatPSS-novariant PARMS HashFunctions
}

-- Cryptographic algorithm identification --
ALGORITHM ::= CLASS {
  &id OBJECT IDENTIFIER UNIQUE,
  &Type OPTIONAL
}
WITH SYNTAX { OID &id [PARMS &Type] }
OID ::= OBJECT IDENTIFIER -- alias
is14888-2 OID ::= {
  iso(1) standard(0) digital-signatures-with-appendix(14888) part2(2) }
signatureScheme OID ::= { is14888-2 scheme(0) }
-- Integer factorization signature scheme identifiers -
rsa OID ::= { signatureScheme rsa(1) }
rw OID ::= { signatureScheme rw(2) }
gq1 OID ::= { signatureScheme gq1(3) }
gq2 OID ::= { signatureScheme gq2(4) }
gps1 OID ::= { signatureScheme gps1(5) }
gps2 OID ::= { signatureScheme gps2(6) }
esign OID ::= { signatureScheme esign(7) }
-- Table A.1 format mechanism option types
noformat RELATIVE-OID ::= { noformat(0) }
formatPSS RELATIVE-OID ::= { formatPSS(1) }
formatD1 RELATIVE-OID ::= { formatD1(10) } -- see D.2
formatD2 RELATIVE-OID ::= { formatD2(11) } -- see D.3
-- Table A.1 hash-variant option types
novariant RELATIVE-OID ::= { novariant(0) }
variant1 RELATIVE-OID ::= { variant1(1) }
variant2 RELATIVE-OID ::= { variant2(2) }
variant3 RELATIVE-OID ::= { variant3(3) }
variant4 RELATIVE-OID ::= { variant4(4) }
-- Bảng A.1 Các lựa chọn cho lược đồ chữ ký phân tích số nguyên --
```

```
id-rsa-formatPSS-novariant OID ::= { rsa formatPSS novariant }
id-rsa-formatD1-novariant OID ::= { rsa formatD1 novariant }
id-rsa-formatD2-novariant OID ::= { rsa formatD2 novariant }
id-rw-formatPSS-novariant OID ::= { rw formatPSS novariant }
id-rw-formatD1-novariant OID ::= { rw formatD1 novariant }
id-rw-formatD2-novariant OID ::= { rw formatD2 novariant }
id-gq1-formatPSS-variant1 OID ::= { gq1 formatPSS variant1 }
id-gq1-formatPSS-variant2 OID ::= { gq1 formatPSS variant2 }
id-gq1-formatPSS-variant3 OID ::= { gq1 formatPSS variant3 }
id-gq1-formatPSS-variant4 OID ::= { gq1 formatPSS variant4 }
id-gq2-noformat-variant1 OID ::= { gq2 noformat variant1 }
id-gq2-noformat-variant2 OID ::= { gq2 noformat variant2 }
id-gq2-noformat-variant3 OID ::= { gq2 noformat variant3 }
id-gq2-noformat-variant4 OID ::= { gq2 noformat variant4 }
id-gps1-noformat-variant3 OID ::= { gps1 noformat variant3 }
id-gps1-noformat-variant4 OID ::= { gps1 noformat variant4 }
id-gps2-noformat-variant3 OID ::= { gps2 noformat variant3 }
id-gps2-noformat-variant4 OID ::= { gps2 noformat variant4 }
id-esign-formatPSS-novariant OID ::= { esign formatPSS novariant }
END -- IntegerFactorizationBasedDigitalSignaturesWithAppendix -
```

CHÚ THÍCH: Theo Quy tắc mã hóa cơ bản trong ASN.1 (xem ISO/IEC 8825-1 [26]), mỗi định danh là một hoặc nhiều chuỗi octet; bit thứ 8 (bit có trọng số cao nhất) bằng 0 trong octet cuối cùng của một chuỗi và bằng 1 trong các octet trước đó nếu chuỗi gồm nhiều octet. Nối 1 vào bit thứ 7 của các octet trong một chuỗi biểu diễn một số nguyên. Mỗi số nguyên sẽ được mã hóa trên số lượng ít nhất các octet có thể, nghĩa là octet "80" là không hợp lệ ở vị trí đầu tiên của một chuỗi.

- Octet đầu tiên được đặt bằng "28", tức là bằng 40 ở hệ thập phân để xác định một tiêu chuẩn ISO (xem ISO/IEC 8825-1 [26]).
- Hai octet tiếp theo được đặt bằng "F428". 14888 bằng "3A28" ở hệ thập lục phân, tức là 0011 1010 0010 1000, tức là hai khối gồm 7 bit: 1110100 0101000. Sau khi chèn giá trị thích hợp vào bit thứ 8 trong mỗi octet, do đó mã của chuỗi là 11110100 00101000, bằng "F428".
- Octet tiếp theo được đặt bằng "02" để xác định phần 2.
- Octet tiếp theo được đặt bằng "00" để xác định cơ chế biểu hiện cung (0).
- Octet tiếp theo xác định lược đồ chữ ký với giá trị từ "01" đến "07".
- Octet tiếp theo xác định cơ chế định dạng với giá trị là "00", "01", "0A" hoặc "0B" theo bảng A.1.
- Octet tiếp thep xác định biến thể băm với giá trị từ "00" đến "04" theo bảng A.1.

Ví DỤ 1: Thành phần dữ liệu "28 F4 28 02 00 01 01 00" có nghĩa là {tiêu chuẩn iso 14888 2 0 1 1 0}, tức là lược đồ chữ ký đầu tiên với PSS là cơ chế định dạng, với ISO/IEC 14888-2, tức là RSA-PSS. Nó được chuyển tải trong một đối tượng dữ liệu BER-TLV với thẻ lớp là "06".

Đối tượng dữ liệu = {"06 08 28 F4 28 02 00 01 01 00"}

VÍ DỤ 2: Thành phần dữ liệu "28 F4 28 02 00 03 01 02" có nghĩa là {tiêu chuẩn iso 14888 2 0 3 1 1}, tức là lược đồ chữ ký thứ ba (GQ1) với PSS là cơ chế định dạng và biến thể băm đầu tiên ($h(W||M)$), với ISO/IEC 14888-2. Nó được chuyển tải trong một đối tượng dữ liệu BER-TLV với thẻ lớp là "06".

Đối tượng dữ liệu = {"06 06 28 F4 28 02 00 03 01 01"}

VÍ DỤ 3: Thành phần dữ liệu "28 F4 28 02 00 04 00 02" có nghĩa là {tiêu chuẩn iso 14888 2 0 4 0 4}, tức là lược đồ chữ ký thứ tư (GQ2) với biến thể băm thứ hai ($h(W||h(M))$), với ISO/IEC 14888-2. Nó được chuyển tải trong một đối tượng dữ liệu BER-TLV với thẻ lớp là "06".

Đối tượng dữ liệu = {"06 06 28 F4 28 02 01 04 00 04"}

VÍ DỤ 4: Thành phần dữ liệu "28 F4 28 02 00 07 01 00" có nghĩa là {tiêu chuẩn iso 14888 2 0 7 1 0}, tức là lược đồ chữ ký thứ bảy với PSS là cơ chế định dạng, với ISO/IEC 14888-2 (ESIGN-PSS). Nó được chuyển tải trong một đối tượng dữ liệu BER-TLV với thẻ lớp là "06".

Đối tượng dữ liệu = {"06 06 28 F4 28 02 00 07 01 00"}

A.2 Sử dụng các định danh đối tượng

Mỗi lược đồ chữ ký được quy định trong tiêu chuẩn này sử dụng một hàm băm, một chuỗi gồm định danh thuật toán hàm băm và các tham số liên quan. Do đó, định danh đối tượng của lược đồ chữ ký liên quan đến 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 trong ASN.1 XML, lược đồ ESIGN-PSS (cơ chế định dạng 1 và không có biến thể băm được quy định trong tiêu chuẩn này), và hàm băm SHA-256 được quy định trong TCVN 11816-3 (ISO/IEC 10118-3), một giá trị kiểu SignatureWithAppendix sẽ được biểu diễn như sau:

```
<SignatureWithAppendix>
  <algorithm> 1.0.14888.2.0.7.1.0 </algorithm>
  <parameters>
    <HashFunctions>
      <algorithm> 2.16.840.1.101.3.4.2.1 </algorithm>
      <parameters/>
    </HashFunctions>
  </parameters>
</SignatureWithAppendix>
```

Phụ lục B
(Tham khảo)
Hướng dẫn lựa chọn tham số và so sánh các lược đồ chữ ký

B.1 Hướng dẫn lựa chọn tham số

B.1.1 Kích thước mô-đun

Trong tiêu chuẩn này, mọi lược đồ chữ ký sử dụng một số mô-đun là tích của các thừa số nguyên tố lớn, ít nhất hai trong số đó là khác nhau. Tất cả các thừa số nguyên tố phải có độ lớn tương đương nhau.

Năm 1995, Odlyzko [16] đã ước lượng độ khó của việc phân tích số nguyên. Trong kết luận ở cuối bài báo được trích dẫn [16], Kaliski đã nhấn mạnh tầm quan trọng của việc chọn các kích thước mô-đun khác nhau trong triển khai và cung cấp khuyến nghị về kích thước mô-đun: 768 bit cho độ an toàn ngắn hạn, 1024 bit cho độ an toàn trung hạn và 2048 bit cho độ an toàn dài hạn. Xem thêm Silverman [20], Lenstra và Verheul [14] để biết phân tích sâu hơn về kích thước mô-đun.

Bảng B.1 đặc tả ba khoảng an toàn cho số mô-đun ($|n|$ bit): trung, dài và rất dài hạn. Đồng thời thiết lập các điều kiện phụ thuộc vào độ lớn mô-đun tính theo bit; Những điều kiện này là về các thừa số nguyên tố ($|p|$ bit), mã băm ($|H|$ bit) và phần đầu tiên của chữ ký ($|R|$ bit trái nhất trong giá trị đầu ra của biến thể băm): không tuân thủ theo những điều kiện này sẽ ảnh hưởng đến độ an toàn.

- Nếu một thừa số nguyên tố p là quá nhỏ, nó có thể được khôi phục lại.
- Nếu mã băm H quá ngắn, có thể tìm ra hai xâu bit có cùng một mã băm.
- Nếu phần đầu tiên R quá ngắn, có thể ký mà không cần biết các số bí mật (xem mục B.1.3).

Bảng B.1 – Các điều kiện đối với $|p|$, $|H|$ và $|R|$ theo $|n|$

$\|n\|$	$\|p\|$	$\|H\|$		$\|R\|$
		RSA, RW, GQ1, ESIGN	GQ2, GPS1, GPS2	GQ1, GQ2, GPS1, GPS2
Từ 1024 đến 1599	> 340	≥ 160	≥ 128	≥ 80
Từ 1600 đến 2999	> 510	≥ 224	≥ 160	≥ 112
Từ 3000 đến 4999	> 680	≥ 256	≥ 192	≥ 144

B.1.2 Số mô-đun và các thừa số nguyên tố

Trong tiêu chuẩn, số lượng các thừa số nguyên tố ký hiệu là f và các thừa số nguyên tố lớn ký hiệu là $p_1, p_2 \dots p_f$ sắp xếp theo thứ tự tăng dần. Số mô-đun là tích của các thừa số nguyên tố.

$$n = p_1 \times p_2 \times \dots \times p_f.$$

Để đạt ưu thế trong thực tế ứng dụng, kích thước của số mô-đun phải là bội số của f, do đó theo bảng B.1, mọi thừa số nguyên tố phải có kích thước tương đương nhau.

CHÚ THÍCH Trong ESIGN, thừa số nguyên tố lớn nhất được sử dụng nhiều lần: $n = p_1 \times p_2^2$.

Phương pháp sau xác định các khoảng biến liên tiếp để lựa chọn các thừa số nguyên tố lớn, kích thước của số nguyên tố lớn được ký hiệu là π. Giá trị hiện thời của tích các thừa số nguyên tố ký hiệu là z.

- Thừa số nguyên tố đầu tiên được lựa chọn trong khoảng từ $2^{\pi-1}$ đến 2^{π}. Giá trị khởi tạo của z được thiết lập bằng thừa số nguyên tố đầu tiên.
- Bước này được lặp lại $f-1$ lần. Một thừa số nguyên tố mới được lựa chọn trong khoảng từ $(2^{|z|}/z)2^{\pi-1}$ đến 2^{π}. Giá trị hiện thời của z được nhân với thừa số nguyên tố mới.
- Các thừa số nguyên tố được ký hiệu từ p_1 đến p_f được sắp xếp theo thứ tự tăng dần, và n được đặt bằng giá trị cuối cùng của z.

Phương pháp sau xác định một khoảng giá trị có kích thước nhỏ hơn để lựa chọn các thừa số nguyên tố.

- Mọi thừa số nguyên tố được chọn trong khoảng từ $\gamma\, 2^{\pi}$ đến 2^{π}, trong đó γ là ký hiệu của căn bậc thứ f của $1/2$.

CHÚ THÍCH Giá trị của γ có thể xấp xỉ bằng một số thích hợp lớn hơn γ (ví dụ: 5/7 là căn bậc hai của 1/2, 4/5 là căn bậc ba của 1/2).

B.1.3 Các lược đồ sử dụng kỹ thuật tri thức không

B.1.3.1 Bộ ba tri thức không

Goldwasser, Micali và Rackoff [7] trình bày khái niệm về tri thức không. Lược đồ GQ1, GQ2, GPS1 và GPS2 sử dụng kỹ thuật tri thức không.

CHÚ THÍCH ISO/IEC 9798-5 [30] quy định các cơ chế xác thực sử dụng kỹ thuật tri thức không.

Ví dụ, trong GQ1, các bước sau chứng minh tri thức về một số bí mật Q.

- Lựa chọn một số nguyên dương ngẫu nhiên r $(0 < r < n)$.
- Tính toán $W = r^{v} mod n$ $(0 < W < n)$.
- Để hồi đáp mọi giá trị thách đố R $(0 \leq R < v)$, tính toán $S = r \times Q^{R}\ mod\ n$ $(0 < S < n)$.

Trong lược đồ GQ1, bên kiểm tra biết được số nguyên R là một giá trị thách đố sau khi nhận được bằng chứng W. Bên kiểm tra tính toán một giá trị bằng chứng khác $W^{*} = S^{v} \times G^{R} mod\ n$. Xác thực thành công khi và chỉ khi bằng chứng W và W^{*} bằng nhau và không bằng 0.

Do đó, tập hợp tất cả các bộ ba GQ1 $\{W, R, S\}$ được xem là một tập v phép hoán vị (được đánh số bởi R) trên vành đồng dư modulo n. Hoán vị với $R = 0$ là phép hoán vị RSA.

B.1.3.2 Độ an toàn chung

Để sử dụng một lược đồ chữ ký từ trao đổi xác thực tri thức không, sự tương tác với bên kiểm tra sẽ được loại bỏ. Số W được biểu diễn bằng một xâu gồm $|n|$ bit, cũng được ký hiệu là W. Trước tiên, các xâu bit W và M được băm (ví dụ: $h(W||M)$, xem các biến thể băm trong 5.2). Sau đó, biểu diễn số

bằng cách rút gọn xâu bit kết quả, ví dụ: rút gọn theo $mod\ v$ thành số nguyên R từ 0 đến $v-1$. Phép tính được ký hiệu là $R = R(W, M)$. Xâu bit M liên quan đến một bộ ba ZK $\{W, R, S\}$. Chữ ký chính là cặp (R, S).

Bộ ba GQ1 vừa hợp lệ vừa liên kết tới M.

- Bộ ba GQ1 hợp lệ khi và chỉ khi $0 < W < n$ và W giống với $W^* = S^v \times G^R mod\ n$.
- Bộ ba GQ1 liên kết tới M khi và chỉ khi $0 \leq R < v$ và R giống với xâu bit $R^* = R(W^*, M)$.

Tấn công sau nhằm mục đích đánh giá kích thước thích hợp của số mũ kiểm tra GQ1 là v khi $t = 1$. Với một thông điệp cho trước M và giá trị S bất kỳ từ 1 đến $n-1$, với $i = 1,2...$, tính toán $x = S^v \times G^i mod\ n$, do đó $y = R(x, M)$ nhỏ hơn v và tiếp tục tính toán cho đến khi $y = i$. Vì vậy, bộ ba GQ1 $\{S^v \times G^i mod\ n, i, S\}$ là hợp lệ và liên kết tới thông điệp M, thông điệp được ký (M, i, S) cũng hợp lệ. Kiểu tấn công này yêu cầu quá trình tính toán bậc của v. Do đó, để chống lại tấn công trên với năng lực tính toán hiện thời, bảng B.1 đã chỉ ra độ dài tối thiểu của R. Tất cả các giá trị có thể của R phải gần như nhau.

Có thể áp dụng tấn công tương tự lên lược đồ GQ2.

Lược đồ GPS1 và GPS2 cũng bị tấn công tương tự như trên. Ngay cả khi rút gọn độ dài của R cũng không làm giảm khối lượng công việc, độ dài của coupon nên rút gọn càng nhiều càng tốt. Để nhất quán thì coupon và R phải có độ dài giống nhau.

B.1.3.3 Kích thước tham số ngẫu nhiên

Trong các lược đồ GQ1, GQ2, GPS1 và GPS2, một tham số ngẫu nhiên r được chuyển đổi thành một giá trị khởi tạo W và phần thứ hai của chữ ký S được tính toán theo phần đầu tiên của chữ ký R. W, R và S tạo ra một bộ ba ZK, ký hiệu là $\{W, R, S\}$ thỏa mãn mọi ràng buộc đối với chữ ký công khai. Tập hợp tất cả các bộ ba ZK là một họ R phép hoán vị của tập hợp (hoặc một tập con) trên vành đồng dư modulo n.

Điều quan trọng là bên ký lựa chọn các tham số ngẫu nhiên sao cho xác suất để đoán giá trị đó và xác suất để giá trị đó được lựa chọn hai lần trong thời hạn ký là không đáng kể. Ví dụ, bên ký sử dụng cùng một giá trị hai lần, thì họ sẽ tạo ra một cặp bộ ba đồng bộ, tức là hỏi đáp hai giá trị thách đố bằng cùng một bằng chứng tri thức không, ký hiệu là $\{W, R_1, S_1\}$ và $\{W, R_2, S_2\}$. Cặp như vậy được gọi là claw (xem [8]) trong họ các phép hoán vị.

- Trong các lược đồ GQ1, GPS1 và GPS2, có thể dễ dàng suy luận số bí mật từ một cặp bộ ba đồng bộ bất kỳ. Có được số bí mật có thể giả mạo bên ký.
- Ràng buộc khóa GQ2 đảm bảo rằng với các giá trị m và k bất kỳ, nếu có hơn một nửa tất cả các cặp bộ ba đồng bộ sẽ giúp tìm ra một nghiệm căn bậc hai không tầm thường mod n của giá trị thỏa thuận. Số đó giúp tìm ra phân tích của n, chính là phân tích số nếu $f = 2$. Biết các thừa số nguyên tố có thể giả mạo bên ký.

Đối với chữ ký công khai, cần tính toán một bằng chứng W từ một cặp (R,S) bất kỳ được lựa chọn một cách ngẫu nhiên, tức là tạo ra các bộ ba một cách ngẫu nhiên. Điều quan trọng là tập hợp tất cả các bộ ba ZK phải đủ lớn để lợi thế thu được bằng cách tạo ra càng nhiều bộ ba càng tốt là không đáng kể.

Tóm lại, các xâu bit ngẫu nhiên là xâu bit gồm:

- $|n|$ bit trong lược đồ GQ1 và GQ2;
- $2\,|H| + 80$ bit trong lược đồ GPS1;
- $|n| + |H| + 80$ bit trong lược đồ GPS2.

B.2 So sánh các lược đồ chữ ký

B.2.1 Các ký hiệu và từ viết tắt

Tiến hành so sánh theo các tiêu chí sau: độ lớn của tập các thành phần dữ liệu để ký, độ phức tạp tính toán khi ký, độ phức tạp tính toán khi kiểm tra và độ lớn của tập các thành phần dữ liệu để kiểm tra.

CHÚ THÍCH Nếu bên ký là một thiết bị di động (ví dụ: một thẻ mạch tích hợp [24]), thì độ phức tạp khi tính toán và truyền tin cũng như không gian lưu trữ yêu cầu là các yếu tố rất quan trọng, vì năng lực xử lý và lưu trữ của thẻ sẽ bị giới hạn khi so sánh với bên kiểm tra.

Trong tiêu chuẩn này áp dụng các ký hiệu và chữ viết tắt dưới đây:

$HW(v)$ Số lượng các bit bằng 1 trong biểu diễn nhị phân của v, ví dụ: $HW(65537 = 2^{16} + 1) = 2$

M_α Độ phức tạp tính toán của phép nhân theo mô-đun

X_α Độ phức tạp tính toán của phép bình phương theo mô-đun

π Độ lớn bit của mỗi thừa số nguyên tố ($\pi = |p_1| = |p_2|$)

B.2.2 Độ phức tạp tính toán của các phép toán mô-đun

Trong mục này sẽ đánh giá độ phức tạp tính toán của các phép tính theo mô-đun, cụ thể là phép nhân theo mô-đun, phép bình phương theo mô-đun, phép mũ hóa theo mô-đun và phép mũ hóa kết hợp theo mô-đun.

Phép nhân theo mô-đun ký hiệu là $A \times B \bmod C$, được thực hiện bằng hai phép tính liên tiếp: phép nhân và phép rút gọn theo mô-đun. Trong thực tế, lượng công việc thực hiện phép nhân xấp xỉ bằng lượng công việc khi thực hiện phép rút gọn.

- Khi A và B có cùng độ lớn với C, kết quả của phép nhân dài gấp đôi so với C.
- Phép rút gọn là số dư của phép chia kết quả trên cho C.

Khi A và B có cùng độ lớn với C, độ phức tạp của phép nhân theo mô-đun ký hiệu là $M_{|C|}$.

Nếu số mô-đun lớn gấp f lần các thừa số nguyên tố, tức là $\alpha = f \times \pi$, thì tỷ lệ giữa số mô-đun n và số mô-đun là một thừa số nguyên tố xấp xỉ f^2 ($M_\alpha \approx f^2\, M_\pi$). Do đó, giá trị $M_{|C|}$ tỷ lệ thuận với $|C|^2$.

Ví dụ: nếu có hai thừa số nguyên tố, tức là $\alpha = 2\,\pi$, thì $M_\alpha \approx 4\, M_\pi$.

Phép bình phương theo mô-đun ký hiệu là $A^2 \bmod C$, được thực hiện bằng hai phép tính liên tiếp: phép bình phương và phép rút gọn theo mô-đun.

- Khi A có cùng độ lớn với C, phép bình phương cho kết quả có độ lớn gấp đôi C. Theo Menezes, van Oorschot và Vanstone [15], độ phức tạp của phép bình phương bằng một nửa so với phép nhân.

  CHÚ THÍCH Vì $A \times B = ((A+B)^2 - (A-B)^2)/4$, phép nhân nhận kết quả từ việc sử dụng hai lần phép bình phương.
- Phép rút gọn tính phần dư của phép chia kết quả cho C. Độ phức tạp của phép rút gọn như trên.

Khi A có cùng độ lớn với C, độ phức tạp tính bình phương theo mô-đun được ký hiệu là $X_{|C|}$.

$$X_{|C|} \approx 0{,}75\, M_{|C|}$$

Phép mũ hóa theo mô-đun ký hiệu là $A^B \bmod C$, được thực hiện bằng thuật toán nhân và bình phương [13, 15], tức là $|B| - 1$ phép bình phương theo mô-đun và $HW(B) - 1$ phép nhân với A theo mô-đun.

Phép mũ hóa kết hợp theo mô-đun ký hiệu là $A_1^{B_1} \times \ldots \times A_x^{B_x} \bmod C$, được thực hiện bằng $\max\{|B_1|, \ldots |B_x|\} - 1$ phép bình phương theo mô-đun và $HW(B_1) + \cdots + HW(B_x) - 1$ phép nhân với A_i. theo mô-đun.

- Nếu A_i nhỏ (tức là $|A_i| \leq 8$), thì phép nhân với B_i theo mô-đun là không đáng kể so với phép bình phương theo mô-đun.
- Phép mũ hóa theo mô-đun ngắn hoặc trung bình hoặc dài tùy thuộc vào kích thước số mũ tính theo bit ($\max\{|B_1|, \ldots |B_x|\}$) là nhỏ (lớn hơn 40), hoặc trung bình (80, 160, 240 đến 280), hoặc lớn ($|C|, |C| + 80$ đến $|C| + 120$).

B.2.3 Độ phức tạp tính toán của kỹ thuật CRT với hai thừa số nguyên tố có cùng kích thước

Hợp số CRT liên quan đến phép nhân theo mô-đun một thừa số nguyên tố và phép nhân của hai số nguyên có cùng kích thước với một thừa số nguyên tố, kết quả là một số nguyên có kích thước với số mô-đun. Khi hai thừa số nguyên tố có cùng kích thước, ví dụ: $\pi = |p_1| = |p_2| = \alpha/2$, độ phức tạp thành phần được ký hiệu là ChC.

$$ChC \approx 1{,}5\, M_\pi \approx (3/8)\, M_\alpha$$

Phân tích CRT liên quan đến hai phép rút gọn theo mô-đun của một thừa số nguyên tố. Khi hai thừa số nguyên tố có cùng kích thước, ví dụ: $\pi = |p_1| = |p_2| = \alpha/2$, độ phức tạp phân tích được ký hiệu là ChD.

$$ChD \approx M_\pi \approx 0{,}25\, M_\alpha$$

Ví dụ: kỹ thuật CRT giúp giảm độ phức tạp của quá trình tạo một chữ ký RSA hoặc RW từ một phép mũ hóa $mod\ n$ (tức là $\left(\frac{5}{1}\right) \alpha M_\alpha$) thành một ChD cộng hai phép mũ hóa $mod\ p_i$ (với số mũ được rút gọn theo $mod\ p_i - 1$) cộng với một ChC (tức là $(1 + 2{,}5\pi + 1{,}5)M_\pi = 2{,}5(\pi + 1)M_\pi$). Vì $\alpha = 2\,\pi$ và $M_\alpha \approx 4\, M_\pi$, độ phức tạp được giảm xuống là $\approx (5/16)\, \alpha M_\alpha$.

B.2.4 Phân tích độ phức tạp tính toán

B.2.4.1 RSA và RW

Để ký không có CRT: n và s	2 α bit
Tính toán chữ ký: lũy thừa bậc $s \bmod n$	$(\lvert s\rvert - 1)X_\alpha + (HW(s) - 1)M_\alpha$
Tổng cộng ($\lvert s\rvert = \alpha, HW(s) = \alpha/2$ và $X_\alpha = 0{,}75\ M_\alpha$)	$(5/4)\alpha M_\alpha$
Để ký với CRT : p_1, p_2, Cr, s_1 và s_2	2,5 α bit
Tính toán chữ ký: lũy thừa bậc $s_i \bmod p_i$	$(5/16)\alpha M_\alpha$
Để kiểm tra: n, v	α bit (v không đáng kể)
Kiểm tra chữ ký RSA: lũy thừa bậc $v \bmod n$	$(\lvert v\rvert - 1)X_\alpha + (HW(v) - 1)M_\alpha$
Tổng cộng	$(0{,}75\lvert v\rvert + HW(v) - 1{,}75)M_\alpha$
VÍ DỤ: 13 M_α nếu $v = 2^{16} + 1$ và 1,75 M_α nếu $v = 3$	
Kiểm tra chữ ký RW: bình phương $\bmod n$ ($X_\alpha \approx 0{,}75\ M_\alpha$)	0,75 M_α

B.2.4.2 GQ1

Để ký: n, v, t, Q	2 α bit (v, t không đáng kể)
Tính toán khởi tạo: $W_i = r_i^v \bmod n$	$(\lvert v\rvert - 1)X_\alpha + (HW(v) - 1)M_\alpha$
Phần thứ hai của chữ ký: $S_i = r_i \times Q^{R_i} \bmod n$	$M_\alpha + (\lvert R_i\rvert - 1)X_\alpha + (HW(R_i) - 1)M_\alpha$
$\lvert R_i\rvert = \lvert v\rvert - 1$ và $HW(R_i) = (\lvert v\rvert - 1)/2$,	$(2\lvert v\rvert + HW(v) - 3{,}75)M_\alpha$
Lặp lại t lần,	$(t \times (2\lvert v\rvert - 1) + HW(v) - 1{,}75))M_\alpha$
Tổng cộng	$(2(\lvert v\rvert - 1) \times t + t \times (HW(v) - 1{,}75))M_\alpha$
Để ký: n, v và t	α bit (v, t không đáng kể)
Tính toán chữ ký: $W_i^* = S_i^v \times G^{R_i} \bmod n$	$(\lvert v\rvert - 1)X_\alpha + (HW(R_i) + HW(v) - 1)M_\alpha$
$\lvert R_i\rvert = \lvert v\rvert - 1$ và $HW(R_i) = (\lvert v\rvert - 1)/2$,	$(1{,}25\lvert v\rvert + HW(v) - 2{,}25)M_\alpha$
Lặp lại t lần,	$(t \times (1{,}25\lvert v\rvert - 1) + HW(v) - 1))M_\alpha$
Tổng cộng	$(1{,}25(\lvert v\rvert - 1) \times t + t \times (HW(v) - 1))M_\alpha$

B.2.4.3 GQ2

Để ký không có CRT: n, k và b, Q_1 đến Q_m	$(m + 1)$ α bit (k, b không đáng kể)
Tính toán khởi tạo: $W = r^{2^{b+k}} \bmod n$	$(k + b)X_\alpha$
Phần thứ hai của chữ ký: $S = r \times \prod_{i=1}^{m} Q_i^{R_i} \bmod n$	$(\lvert R_i\rvert_{max} - 1)X_\alpha + (HW(R_1)+..+HW(R_m) - 1)M_\alpha + M_\alpha$
$\lvert R_i\rvert = k$ và $HW(R_m) = k/2$,	$(k - 1)X_\alpha + 0{,}5\ k\ m\ M_\alpha$
Tổng cộng	$(0{,}5\ k\ (m + 3) + 0{,}75(b - 1))M_\alpha$
Để ký với CRT: p_1, p_2, Cr, k và $b, Q_{1,1}$ đến $Q_{m,2}$	$(m + 1{,}5)\alpha$ bit (k, b không đáng kể)
Tính toán khởi tạo: $W_i = r_i^{2^{k+b}} \bmod p_i$	$2(k + b)X_\pi + ChC$
	$2(\lvert R_i\rvert_{max} - 1)X_\pi + 2\big(HW(R_1)+..+HW(R_m)\big)M_\pi + ChC$

Phần thứ hai của chữ ký: $S_j = r_i \times \prod_{i=1}^{m} Q_{i,j}^{R_i} \bmod p_j$	$2(k-1)X_\pi + k\,m\,M_\pi + ChC$
$\lvert R_i\rvert = k$ và $HW(R_m) = k/2$,	$(0{,}25\,k\,(m+3) + 0{,}375\,(b+1))M_\alpha$
Tổng cộng ($ChC \approx 1{,}5\,M_\pi$ và $M_\pi \approx M_\alpha/4$)	α bit (k, b, g_1 đến g_m không đáng kể)
Để ký: n, k, b, g_1 đến g_m	$(k+b)X_\alpha$ (× g_1 đến g_m không đáng kể)
Tính toán chữ ký: $W^* = S^{2^{k+b}} \times \prod_{i=1}^{m}(g_i^{2^b})^{R_i} \bmod n$	$(0{,}75(k+b))M_\alpha$
Tổng cộng	

B.2.4.4 GPS1

Để tạo ra coupon không có CRT: n	α bit
Tính toán khởi tạo: $W = 2^r \bmod n$	$(\lvert r\rvert - 1)X_\alpha$
Tổng cộng ($\lvert r\rvert = 2\lvert H\rvert + 80$)	$(1{,}5\lvert H\rvert + 60)M_\alpha$
Để tạo ra coupon với CRT: p_1, p_2, Cr	1,5 α bit
Tính toán khởi tạo: $W_i = 2^r \bmod p_i$	$2(\lvert r\rvert - 1)X_\pi + ChC$
$\lvert r\rvert < 0{,}5 \times \lvert n\rvert$ và $ChC \approx 1{,}5 \times M_\pi$	
Tổng cộng ($M_\pi \approx M_\alpha/4$)	$(0{,}75\,\lvert H\rvert + 30)M_\alpha$
Coupon và thành phần sử dụng coupon: Q	$\lvert H\rvert$ bit + ($\lvert H\rvert$ bit với mỗi coupon)
Phần thứ hai chữ ký: $S = r - R \times Q$	$0{,}5\,(\lvert R\rvert/\alpha)(\lvert Q\rvert/\alpha)M_\alpha$
Sử dụng coupon ($\lvert R\rvert = \lvert H\rvert$ và $\lvert Q\rvert = \lvert H\rvert$)	$(0{,}5(\lvert H\rvert/\alpha)^2 M_\alpha$
Để kiểm tra: n và G	2α bit
Tính toán chữ ký: $W^* = 2^S \times G^R \bmod n$	$(\lvert S\rvert - 1)X_\alpha + (HW(R) - 1)M_\alpha$
Tổng cộng ($HW(R) = \lvert H\rvert/2$ và $\lvert S\rvert = 2\lvert H\rvert + 80$)	$(2\lvert H\rvert + 60)M_\alpha$

B.2.4.5 GPS2

Để tạo ra coupon không có CRT: n và v	α bit (v không đáng kể)
Tính toán khởi tạo: $W_j = 2^{r \times v} \bmod n$	$(\lvert r\rvert + \lvert v\rvert)X_\alpha$
Tổng cộng ($\lvert r\rvert = \alpha + \lvert H\rvert + 80$)	$0{,}75\,(\alpha + 2\lvert H\rvert + 80)M_\alpha$
Để tạo ra coupon với CRT: p_1, p_2, Cr, v	1,5 α bit (v không đáng kể)
Tính toán khởi tạo: $W_i = 2^{r \times v \bmod p_i - 1} \bmod p_i$	$2(\pi - 1)X_\pi + ChC$
$2 \times \pi = \alpha$ và $ChC \approx 1{,}5 \times M_\pi$	$0{,}75\,\alpha\,M_\pi$
Tổng cộng ($M_\pi \approx M_\alpha/4$)	$(3/16)\alpha M_\alpha$
Coupon và thành phần sử dụng coupon: Q	α bit + (β bit với mỗi coupon)
Phần thứ hai chữ ký: $S = r - R \times Q$	$0{,}5\,(\lvert R\rvert/\alpha)(\lvert Q\rvert/\alpha)M_\alpha$
$\lvert R\rvert = \lvert H\rvert$; $\lvert Q\rvert = \alpha$.	$(0{,}5(\lvert H\rvert/\alpha)M_\alpha$
Để kiểm tra: n và v	α bit (v không đáng kể)

Tính toán chữ ký: $W^* = 2^{R+v\times S} \bmod n$	$(\lvert S \times v\rvert - 1)X_\alpha + (HW(S \times v) - 1)M_\alpha$
Tổng cộng ($\lvert S \times v\rvert = \alpha + \lvert H\rvert + 80, HW(S \times v) = (\alpha + \lvert H\rvert + 80)/2$)	$1{,}25(\alpha + \lvert H\rvert + 80)M_\alpha$

B.2.4.6 ESIGN

Để ký: v, p_1 và p_2 ($\lvert p_1\rvert = \lvert p_2\rvert = \alpha/3$)	0,67 α bit (v không đáng kể)
$p_1 \times p_2$	0,5 M_π
$\times p_2)$	M_π
$y = r^v mod\ (p_1 \times p_2 \times p_2)$	$2(\lvert v\rvert - 1)X_\alpha + 2(HW(v) - 1)M_\alpha$
$z = \left((r\ mod\ p_2) \times \left(v \times (y\ mod\ p_2)\right)^{-1}\right) mod\ p_2$	$2(4\ M_\pi + I_\pi)$
$S = \left\lfloor (2^{2\times\frac{\alpha}{3}} \times F - y)/(p_1 \times p_2)\right\rfloor$	2 M_α
$\times z mod p_2$	2 M_π
$\times p_1 \times p_2 + r\ mod (p_1 \times p_2 \times p_2)$	2 M_α
Tổng cộng (giả sử $I_\pi = 10\ M_\pi$ và $M_\alpha = 9\ M_\pi$)	$(1{,}5\ \lvert v\rvert + 2HW(v) + 4)M_\alpha$
Để kiểm tra: n và v	α bit (v không đáng kể)
Tính toán chữ ký: $S^v mod\ n$	$(\lvert v\rvert - 1)X_\alpha + (HW(v) - 1)M_\alpha$
Tổng cộng	$(0{,}75\lvert v\rvert + HW(v) - 1{,}75)M_\alpha$

B.2.4.7 Tổng hợp so sánh

Bảng B.2 tổng hợp các so sánh được trình bày cụ thể từ B.2.4.1 đến B.2.4.6

Bảng B.2 – Tổng hợp so sánh

	Cơ chế ký			Cơ chế kiểm tra	
	CRT	Lưu trữ (bit)	Độ phức tạp (M_α)	Lưu trữ (bit)	Độ phức tạp (M_α)
RSA/RW	Không	2 α	1,25 α	α	RSA: $0{,}75\lvert v\rvert + HW(v) - 1{,}75$
	Có	2,5 α	0,3125 α		RW: 0,75
GQ1	Không	2 α	$2(\lvert v\rvert - 1) \times t + t \times (HW(v) - 1{,}75)$	α	$1{,}25(\lvert v\rvert - 1) \times t + t \times (HWv - 1)$
GQ2	Không	$(m + 1)\ \alpha$	$0{,}5k(m + 3) + 0{,}75(b - 1)$	α	$0{,}75\ (k + b)$
	Có	$(m + 1{,}5)\ \alpha$	$0{,}25k(m + 3) + 0{,}375(b + 1)$		
GPS1-P	Không	α	$1{,}5\ \lvert H\rvert + 60$	$2 \times \alpha$	$2\lvert H\rvert + 60$
	Có	1,5 α	$0{,}75\ \lvert H\rvert + 30$		

GPS1-C	Không	α + ($\|H\|$ *cho mỗi coupon*)	$0{,}5\,\|H\|/\alpha$			
GPS2-P	Không	α	$0{,}75\,(\alpha + 2(\|v\| - 1)) + 60$		α	$1{,}25(\alpha + \|H\|) + 100$
	Có	$1{,}5\,\alpha$	$0{,}1875\,\alpha$			
GPS2-C	Không	α + ($\|H\|$ *cho mỗi coupon*)	$0{,}5\|H\|/\alpha$			
ESIGN	Không	$0{,}67\,\alpha$	$1{,}5\|v\| + 2HW(v) + 4$		α	$0{,}75\|v\| + HW(v) - 1{,}75$
- Kỹ thuật CRT không liên quan đến việc tạo ra chữ ký GQ1 và ESIGN. - Quá trình tạo chữ ký GPS1 và GPS2 gồm hai bước: quá trình tạo coupon, ký hiệu là P và sử dụng coupon, ký hiệu là C. Coupon được tạo ra trước bằng một thiết bị khác.						

B.2.4.8 Độ phức tạp tính toán với các kích thước mô-đun khác nhau

Với $|H| = 160$ (ví dụ: RIPEMD-160 và SHA-1), việc so sánh sử dụng các giá trị sau.

RSA - $v = 2^{16} + 1, \varepsilon = 160$ và $\tau = 0$ (tức là $|v| = 17, HW(v) = 2$, một giá trị salt 160 bit và không có giá trị trailer)

RW - $v = 2, \varepsilon = 160$ và $\tau = 0$ (tức là $|v| = 2, HW(v) = 1$, một giá trị salt 160 bit và không có giá trị trailer)

GQ1 - $v = 2^{80} + 13$ và $t = 1$ (tức là $(|v| - 1) \times t = 80$ và $HW(v) = 4$)

GQ2 - $b = 1, m = 10$ và $k = 8$ (các số cơ sở = 10 số nguyên tố đầu tiên: 2 đến 29 và $k \times m = 80$)

GPS1 - $g = 2(|R| = |Q| = |H| = 160, |G| = \alpha$ và $|r| = 2|H| + 80 = 400)$

GPS2 - $g = 2, v = 2^{160} + 7$ (tức là $|R| = 160, |Q| = \alpha$ và $|r| = \alpha + |H| + 80 = \alpha + 240$)

ESIGN - $v = 2^{10}$ (tức là $|v| = 11$ và $HW(v) = 1$)

Bảng B.3 so sánh độ phức tạp tính toán đối với các kích thước mô-đun khác nhau: $\alpha = 1024, 1536$ và 2048. Đơn vị tính của độ phức tạp tính toán là M_{1024} ($M_{1536} \approx 2{,}25\, M_{1024}$; $M_{2048} \approx 4\, M_{1024}$).

Sử dụng các cơ chế quy định cho GQ1, GQ2, GPS1 và GPS2 để xác thực, bên kiểm tra chuyển giao một giá trị thách đố và đợi hồi đáp trong độ trễ giới hạn quy định trong phụ lục F. Bảng B.3 bao gồm xác thực với kích thước của phần đầu tiên của chữ ký sau đây, ký hiệu là R. Thì giá trị hồi đáp không phải là chữ ký: đó là chứng nhận "không thể chuyển nhượng".

- Với GQ1, $m = 1$ và $v = 2^{16} + 1$ ($|R| = 16$).
- Với GQ2, $b = 1, k = 8$ và $m = 2$ ($|R| = 16$).

Bảng B.3 – Độ phức tạp tính toán đối với các kích thước mô-đun khác nhau

		Cơ chế ký (M_{1024})				Cơ chế kiểm tra (M_{1024})		
	CRT	$\alpha = 1024$	$\alpha = 1536$	$\alpha = 2048$		$\alpha = 1024$	$\alpha = 1536$	$\alpha = 2048$
RSA/RW	Không	1028	4320	10240		RSA 13	29,25	52
	Có	320	1080	2560		RW 0,75	1,69	3
GQ1	Không	162,25	365,06	649		103	231,75	412
Xác thực	*Có*	*34,25*	*77,06*	*137,00*		*22,25*	*50,06*	*89,00*
GQ2	Không	52,00	117,00	208		6,75	15,19	27
	Có	26,75	60,19	107				
Xác thực	*Có*	*10,75*	*24,19*	*43*		*6,75*	*15,19*	*27*
GPS1-P	Không	300	675	1200		380	855	1520
	Có	150	337,5	600				
GPS1-C	Không	0,012						
GPS2-P	Không	1068	3267	7344		1580	4995	11440
	Có	192	648	1536				
GPS2-C	Không	0,078	0,117	0,156				
ESIGN	Không	22,5	50,63	90		7,5	16,88	30

Phụ Lục C
(Tham khảo)
Các ví dụ

C.1 Lược đồ RSA-PSS

C.1.1 Thông điệp với giá trị salt

Thành phần dữ liệu để ký/kiểm tra – Độ lớn của mỗi thừa số nguyên tố là 512 bit. Độ lớn của số mô-đun là 1024 bit. Số mũ kiểm tra là $v = 3$ (chia hết $p_1 - 1$ hoặc $p_2 - 1$).

$p_1 =$ CC109249 5D867E64 065DEE3E 7955F2EB C7D47A2D 7C995338 8F97DDDC 3E1CA19C
35CA659E DC3D6C08 F64068EA FEDBD911 27F9CB7E DC174871 1B624E30 B857CAAD

$p_2 =$ D8CD81F0 35EC57EF E8229551 49D3BFF7 0C53520D 769D6D76 646C7A79 2E16EBD8
9FE6FC5B 6060BD97 8ED64A90 59C5B039 98A0E94C 86D78B85 BA37B5AF D987505F

$s =$ 1CCDA20B CFFB8D51 7EE96668 66621B11 822C7950 D55F4BB5 BEE37989 A7D17312
E326718B E0D62CCB 11415F78 B36BE2E6 0D599D4E 41346C82 D845498A 81B2F663
2FD7D1CC EFCABF74 17350238 109EC289 D5382762 B77A1C99 96DD1D2B 71A52FAF
52ABA9DE D19F3F5D 5D71D054 73EC9C79 92D84128 0BAC72B8 7BF51EB1 CCB65C87

$n =$ ACD1CC46 DFE54FE8 F9786672 664CA269 0D0AD7E5 003BC642 7954D939 EEE8B271
52E6A947 45050CC2 67883CD4 34875164 5019AFD5 873A8B11 119FB93F 0A31C654
C3ECFF07 3233530C 79BF90E0 26E2421D D378B88B 40136C48 7D33075A 1612AB90
C5B75D33 2659A5D0 B5C19576 102D3424 31AC3BBB A8F98449 BD58BC0B 5E254633

Chữ ký – Thông điệp là một xâu gồm 114 octet. Giá trị salt là một xâu gồm 20 octet.

$M =$ 859EEF2F D78ACA00 308BDC47 1193BF55 BF9D78DB 8F8A672B 484634F3 C9C26E64
78AE1026 0FE0DD8C 082E53A5 293AF217 3CD50C6D 5D354FEB F78B2602 1C25C027
12E78CD4 694C9F46 9777E451 E7F8E9E0 4CD3739C 6BBFEDAE 487FB556 44E9CA74
FF77A53C B729802F 6ED4A5FF A8BA1598 90FC

$E =$ E3B5D5D0 02C1BCE5 0C2B65EF 88A188D8 3BCE7E61

Với SHA-1 và PSS, chuyển đổi một thông điệp (114 octet), một giá trị salt (20 octet) và một giá trị trailer ("BC") thành một giá trị đặc trưng (1024 bit).

$F =$ 66E4672E 836AD121 BA244BED 6576B867 D9A447C2 8A6E66A5 B87DEE7F BC7E65AF
5057F86F AE8984D9 BA7F969A D6FE02A4 D75F7445 FEFDD85B 6D3A477C 28D24BA1
E3756F79 2DD1DCE8 CA94440E CB5279EC D3183A31 1FC896DA 1CB39311 AF37EA4A
75E24BDB FD5C1DA0 DE7CECDF 1A896F9D 8BC816D9 7CD7A2C4 3BAD546F BE8CFEBC

$S = G^s mod n$

$S =$ 0F624406 FC3A216B 23D44ECF F430C05A 455B8218 E22FE47B 1FEA060C 5A9CB2DE
A6981717 80B5E60C 50A567A5 58EF47B5 FE28AF9B E029611C 85A93345 9B0E610A
C64F45CC C1263A10 67E5BFC0 105BBFBC 9225A460 8385A417 EB80587B 4702C9F9
381658A7 72739BA8 2DA018E1 4AAE564C 0A749A05 D0C1E61C 93FDF777 6D824BE6

Kiểm tra – $G^* = S^v mod n$. $F^* = G^*$.

F^* = 66E4672E 836AD121 BA244BED 6576B867 D9A447C2 8A6E66A5 B87DEE7F BC7E65AF
5057F86F AE8984D9 BA7F969A D6FE02A4 D75F7445 FEFDD85B 6D3A477C 28D24BA1
E3756F79 2DD1DCE8 CA94440E CB5279EC D3183A31 1FC896DA 1CB39311 AF37EA4A
75E24BDB FD5C1DA0 DE7CECDF 1A896F9D 8BC816D9 7CD7A2C4 3BAD546F BE8CFEBC

HH^* = DF1A896F 9D8BC816 D97CD7A2 C43BAD54 6FBE8CFE

Để khôi phục lại xâu trung gian, một giá trị mặt nạ gồm 864 bit (=1024 – 160) được xây dựng từ HH^* và XOR với các bit trái nhất của F^*. Giá trị salt được khôi phục lại là một xâu gồm 20 octet. Giá trị trailer được khôi phục lại là "BC".

00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 000001E3 B5D5D002 C1BCE50C
2B65EF88 A188D83B CE7E61DF 1A896F9D 8BC816D9 7CD7A2C4 3BAD546F BE8CFEBC

Sau đó, một xâu gồm 384 bit (64 bit 0, 160 bit của h(M) và 160 bit của E^*) được băm thành HH.

E^* = E3B5D5D0 02C1BCE5 0C2B65EF 88A188D8 3BCE7E61

HH = DF1A896F 9D8BC816 D97CD7A2 C43BAD54 6FBE8CFE

C.1.2 Thông điệp không có giá trị salt

Các thành phần dữ liệu để ký/kiểm tra – Ví dụ sử dụng các thành phần giống như trong C.1.1.

Chữ ký – Thông điệp là một xâu gồm 114 octet. Giá trị salt rỗng.

M = 859EEF2F D78ACA00 308BDC47 1193BF55 BF9D78DB 8F8A672B 484634F3 C9C26E64
78AE1026 0FE0DD8C 082E53A5 293AF217 3CD50C6D 5D354FEB F78B2602 1C25C027
12E78CD4 694C9F46 9777E451 E7F8E9E0 4CD37739C 6BBFEDAE 487FB556 44E9CA74
FF77A53C B729802F 6ED4A5FF A8BA1598 90FC

Với SHA-1 và PSS, chuyển đổi một thông điệp (114 octet), một giá trị salt (rỗng) và một giá trị trailer ("BC") thành một giá trị đặc trưng (1024 bit).

F = 2DDA5328 280470C5 AFBBF866 78F0E0C6 5B473939 BF146088 B70009A3 8A8C8E25
3BDF02F3 B3DE52E9 364CACAC 3196F828 D5CDCF83 F9529F70 DB26F641 FC112E4C
11ACC6F0 15FF3C57 74C27775 96042A36 81923E5F 7A636D16 EEA8F881 3775E1A8
FB94ED45 9292E062 0AB94764 8E5FA0D7 5B53051C C87F4ECF E350AB8E 4DADABBC

$S = G^s modn$

S = 81A9AA0C A1D227C5 E6FDB537 B7C897D5 D96A6B24 B8D1EAA0 A4673B05 D6D98FF6
7045161A 28BF464F B72F884B 23AB3ED0 D27F80A9 0BBF2365 2A023B00 8E997933
D08B3914 453CDF10 28566F21 F2A88C37 2A750B0E 1E962656 9571C6AF 30359BA4
F9A10764 C69CBD2F 19461CD9 4A21337E 5B6AD86F EF65FDFE 1945802D 96FF4B51

Kiểm tra – $G^* = S^v modn. F^* = G^*$.

F^* = 2DDA5328 280470C5 AFBBF866 78F0E0C6 5B473939 BF146088 B70009A3 8A8C8E25
3BDF02F3 B3DE52E9 364CACAC 3196F828 D5CDCF83 F9529F70 DB26F641 FC112E4C
11ACC6F0 15FF3C57 74C27775 96042A36 81923E5F 7A636D16 EEA8F881 3775E1A8
FB94ED45 9292E062 0AB94764 8E5FA0D7 5B53051C C87F4ECF E350AB8E 4DADABBC

$HH^* =$ 648E5FA0 D75B5305 1CC87F4E CFE350AB 8E4DADAB

Để khôi phục lại xâu trung gian, một giá trị mặt nạ gồm 864 bit (=1024 – 160) được xây dựng từ HH^* và XOR với các bit trái nhất của F^*. Giá trị salt được khôi phục lại là một xâu gồm 20 octet. Giá trị trailer được khôi phục lại là "BC".

```
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 00000164 8E5FA0D7 5B53051C C87F4ECF E350AB8E 4DADABBC
```

Thì, một xâu gồm 224 bit (64 bit 0 và 160 bit của $h(M)$) được băm thành HH.

$HH =$ 648E5FA0 D75B5305 1CC87F4E CFE350AB 8E4DADAB

C.1.3 Thông điệp rỗng không có giá trị salt

Thành phần dữ liệu để ký/kiểm tra – Độ lớn của mỗi thừa số nguyên tố là 512 bit. Độ lớn của số mô-đun là 1024 bit. Số mũ kiểm tra là $v = 3$ (chia hết $p_1 - 1$ hoặc $p_2 - 1$).

$p_1 =$ FB961451 995C82F9 527CAAAF B3FB4254 6D00A01D 8B2BDE3D 2E7B8F7D 0C9E781E
B7FABFC8 E86E9F6D ACE3435A 9D043A99 93F3E473 D93FA888 D3577906 77A94931

$p_2 =$ FF0EAFCA 70585166 A8CD8E90 36E75290 2F32B863 068016B6 A89F2EA3 418882EF
6F570122 F92D2E9B EFFF7329 1818F251 BF095D6E 208F93CD CEF4767A 568AB241

$s =$ CA71B48C DF4A1342 5E1BAB87 9F471638 92AEB277 A9CBC369 B1CAD109 3C93FE22
33267EC0 805A7693 F6A506D0 F9723F6B 1A6F755A ECB0B7DE 1F440102 94186936
316AAC4B F39B37BF 6105DFA0 AEA60B82 C17306F2 179F2ED4 704D5A6F BCB141C0
C9380F5A 500823CE 67F8ED81 7F8A5100 59E9541B 498C91F4 1ABE8C10 6220E72B

$n =$ FAA8ED34 EEF1CE38 D29814B6 EEAA154D C060BB37 EB1A51E8 AB0398DD ADDFD334
CB9BE20C 087B1DDF 1F78A397 62B5F20A 7A730086 30913CD2 EE60183D E249DD16
9CA4EB3A E0420E51 13D73050 4A73A926 BEFBFF32 C89858DE 5E5B3899 FEC52521
04933163 625F2963 5AB8FAA7 AA14C4F3 C0DD2470 DEFCEB39 2429110A 0149A771

Chữ ký - Với SHA-1 và PSS, chuyển đổi một thông điệp (rỗng), một giá trị salt (rỗng) và một giá trị trailer (một octet có giá trị bằng "BC") thành một giá trị đặc trưng (1024 bit).

$F =$ 7CCB5422 2079C84C 343D0AB1 6307273B 36359229 BD3DFDEC A9FE8054 AD1EF319
44758A67 3B7C70C2 FACB6FE9 12690EE2 6DF58975 585A78C2 723F0C71 50535C80
8F0868F6 CA94F36C FB079FBB 9126286D 5EECA3CA ACA12593 033A0D64 136A7A72
D605080A 6CF68B6D DA0AE6A3 5D1688A6 0AC69FD5 3E44428B FD380E94 DB9176BC

$S = G^s mod n$

$S =$ F9DD9F72 FAB4AFFC ED3B0538 C5848B27 756AC50C B2890F4C BC268D96 C5E91EE8
8F3B058F 2EF6585F EF5323CA 4E2C308C C6140CF5 F5357960 5B3BF0CC 621082EB
77F4A42D 3567355E AA151FB4 652BAFFE 58A4B310 7A064669 FD4177C8 D79F5DE5
EEC562FF A2D0F5D9 C409AEA0 D5B9F8DF 493AF2F1 8F91D828 CE32C4CC 35C13113

Kiểm tra – $G^* = S^v mod n$.

G^* = 7CCB5422 2079C84C 343B0AB1 63072738 36359229 BD3DFDEC A9FE8054 AD1EF319
44758A67 3B7C70C2 FACB6FE9 12690EE2 6DF58975 585A78C2 723F0C71 50535C80
8F0868F6 CA94F36C FB079FBB 9126286D 5EECA3CA ACA12593 033A0D64 136A7A72
D605080A 6CF68B6D DA0AE6A3 5D1688A6 0AC69FD5 3E44428B FD380E94 DB9176BC

HH^* = A35D1688 A60AC69F D53E4442 8BFD380E 94DB9176

Để khôi phục lại xâu trung gian, một giá trị mặt nạ gồm 856 bit (=1024 – 160 – 8) được xây dựng từ HH^* và XOR với các bit trái nhất của F^*. Giá trị salt được khôi phục lại là rỗng.

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

Thì, một xâu gồm 224 bit (64 bit 0 và 160 bit của $h(\emptyset)$) được băm thành HH.

HH = A35D1688 A60AC69F D53E4442 8BFD380E 94DB9176

C.2 Lược đồ RW-PSS

C.2.1 Thông điệp với giá trị salt

Thành phần dữ liệu để ký/kiểm tra – Độ lớn của mỗi thừa số nguyên tố là 512 bit (một thừa số nguyên tố đồng dư với 3 *mod* 8 và thừa số còn lại đồng dư với 7 *mod* 8). Độ lớn của số mô-đun là 1024 bit.

p_1 = DBB3CB4C 375C0ECD 2FD300DB 4F085472 93CA004C EDD2019C E79CA08A 15EEFB25
DD3BAF98 183B0C2F 01D7F8B4 931856F6 DD3EBA17 7D763C03 E1DCEABC D803BE33

p_2 = EEAA4A53 47999FE7 6FB78760 64BBEC66 CB409A77 39EF5A59 06613DC3 7225D41D
2BEB1F9F 5EC77A85 38767A87 BB7015D6 07FF26DE 61282753 9306BA1C FFF093A7

s = 199A6985 E9B2BFF5 A2841CCC D80FC73A 28A14266 0987EB12 3DBCAEB2 B8EE546D
2356A3A5 7D9C28ED 71E455C4 466CBE30 7787DC5A 9959B747 5A189A8F 038A4741
E4B10153 BE08C26E 4401F7AB 6E7E9609 2CAF07C0 870B13B6 4F669667 3029EC2C
77AABC39 7FA528A2 45D7073C E69CC9BD CD7BEF91 599DCA48 4000C0BD 8AB0814E

n = CCD34C2F 4D95FFAD 1420E666 C07E39D1 450A1330 4C3F5891 EDE57595 C772A369
1AB51D2B ECE1476B 8F22AE22 3365F183 BC3EE2D4 CACDBA3A D0C4D478 1C523A10
EFE6203D 6F3BC226 BF9A4597 27B8F122 C482D8C8 6019F9A8 69329187 096430A6
C67CB103 742BCBC6 6906AD23 836EBABB 511D5D80 AB8CB599 74E9AAC6 2D785C45

Chữ ký – Thông điệp là một xâu gồm 114 octet. Giá trị salt là một xâu gồm 20 octet.

M = 859EEF2F D78ACA00 308BDC47 1193BF55 BF9D78DB 8F8A672B 484634F3 C9C26E64
78AE1026 0FE0DD8C 082E53A5 293AF217 3CD50C6D 5D354FEB F78B2602 1C25C027
12E78CD4 694C9F46 9777E451 E7F8E9E0 4CD3739C 6BBFEDAE 487FB556 44E9CA74
FF77A53C B729802F 6ED4A5FF A8BA1598 90FC

E = E3B5D5D0 02C1BCE5 0C2B65EF 88A188D8 3BCE7E61

Với SHA-1 và PSS, chuyển đổi một thông điệp (114 octet), một giá trị salt (20 octet) và một giá trị trailer (một octet có giá trị bằng "BC") thành một giá trị đặc trưng (1024 bit).

F = 66E4672E 836AD121 BA244BED 6576B867 D9A447C2 8A6E66A5 B87DEE7F BC7E65AF

5057F86F AE8984D9 BA7F969A D6FE02A4 D75F7445 FEFDD85B 6D3A477C 28D24BA1
E3756F79 2DD1DCF8 CA94440E CB5279EC D3183A31 1FC896DA 1CB39311 AF37EA4A
75E24BDB FD5C1DA0 DE7CECDF 1A896F9D 8BC816D9 7CD7A2C4 3BAD546F BE8CFEBC

Vì $(F|n) = -1, G = F/2$, nên $(G|n) = +1$. Do đó $S = G^S mod n$.

$G =$ 33723397 41B56890 DD1225F6 B2BB5C33 ECD223E1 45373352 DC3EF73F DF3F32D7
A82BFC37 D744C26C DD3FCB4D 6B7F0152 6BAFBA22 FF7EEC2D B69D23BE 146925D0
F1BAB7BC 96E8EE74 654A2207 65A93CF6 698C1D18 8FE44B6D 0E59C988 D79BF525
3AF125ED FEAE0ED0 6F3E766F 8D44B7CE C5E40B6C BE6BD162 1DD6AA37 DF467F5E

$S =$ 8A505E24 FCC61832 03636262 C6AD70F5 3AC1E5CE DC714F59 ED3693B1 F2332442
FD5D2FF1 2C8DBF9B 942A6A46 C6C63C1D 09C2D316 FF605081 19B19F3E 52F6A2BD
D20A6F20 F217C9AD 0F1E496B 70529DA9 1AD7879A F912FB99 ABD387EF AD6FE54C
72FF2FCD 80069BE0 2614AA1D 7C4FE2FF AC70D936 5A81F03B C7F1D82F 733B5E12

Kiểm tra – $G^* = S^2 mod n$.

$G^* =$ 33723397 41B56890 DD1225F6 B2BB5C33 ECD223E1 45373352 DC3EF73F DF3F32D7
A82BFC37 D744C26C DD3FCB4D 6B7F0152 6BAFBA22 FF7EEC2D B69D23BE 146925D0
F1BAB7BC 96E8EE74 654A2207 65A93CF6 698C1D18 8FE44B6D 0E59C988 D79BF525
3AF125ED FEAE0ED0 6F3E766F 8D44B7CE C5E40B6C BE6BD162 1DD6AA37 DF467F5E

Vì G^* đồng dư với 6 mod 8, $F^* = 2G^*$.

$F^* =$ 66E4672E 836AD121 BA244BED 6576B867 D9A447C2 8A6E66A5 B87DEE7F BC7E65AF
5057F86F AE8984D9 BA7F969A D6FE02A4 D75F7445 FEFDD85B 6D3A477C 28D24BA1
E3756F79 2DD1DCE8 CA94440E CB5279EC D3183A31 1FC896DA 1CB39311 AF37EA4A
75E24BDB FD5C1DA0 DE7CECDF 1A896F9D 8BC816D9 7CD7A2C4 3BAD546F BE8CFEBC

$HH^* =$ DF1A896F 9D8BC816 D97CD7A2 C43BAD54 6FBE8CFE

Để khôi phục lại xâu trung gian, một giá trị mặt nạ gồm 856 bit (=1024 – 160 – 8) được xây dựng từ HH^* và XOR với các bit trái nhất của F^*. Giá trị salt được khôi phục lại là một xâu gồm 20 octet.

00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 000001E3 B5D5D002 C1BCE50C
2B65EF88 A188D83B CE7E61DF 1A896F9D 8BC816D9 7CD7A2C4 3BAD546F BE8CFEBC

Thì, một xâu gồm 384 bit (64 bit 0 và 160 bit của $h(M)$ và 160 bit của E^*) được băm thành HH.

$E^* =$ E3B5D5D0 02C1BCE5 0C2B65EF 88A188D8 3BCE7E61

$HH =$ DF1A896F 9D8BC816 D97CD7A2 C43BAD54 6FBE8CFE

C.2.2 Thông điệp không có giá trị salt

Các thành phần dữ liệu để ký/kiểm tra – Ví dụ sử dụng các thành phần giống như trong C.2.1.

Chữ ký – Thông điệp là một xâu gồm 114 octet. Giá trị salt rỗng.

$M =$ 859EEF2F D78ACA00 308BDC47 1193BF55 BF9D78DB 8F8A672B 484634F3 C9C26E64
78AE1026 CFE0DD8C 082E53A5 293AF217 3CD50C6D 5D354FEB F78B2602 1C25C027
12E78CD4 694C9F46 9777E451 E7F8E9E0 4CD3739C 6BBFEDAE 487FB556 44E9CA74

FF77A53C B729802F 6ED4A5FF A8BA1598 90FC

Với SHA-1 và PSS, chuyển đổi một thông điệp (114 octet), một giá trị salt (rỗng) và một giá trị trailer (một octet có giá trị bằng "BC") thành một giá trị đặc trưng (1024 bit).

S = 2DDA5328 280470C5 AFBBF866 78F0E0C6 5B473939 BF146088 B70009A3 8A8C8E25
3BDF02F3 B3DE52E9 364CACAC 3196F828 D5CDCF83 F9529F70 DB26F641 FC112E4C
11ACC6F0 15FF3C57 74C27775 96042A36 81923E5F 7A636D16 EEA8F881 3775E1A8
FB94ED45 9292E062 0AB94764 8E5FA0D7 5B53051C C87F4ECF E350AB8E 4DADABBC

Vì $(F|n) = +1, G = F$, nên $(G|n) = +1$. Do đó $S = G^s mod n$.

F = A110B935 D2589D74 74ADDD01 D9397699 D34DCA6F 10FF7547 A18CA4CF 16BD845A
247EEA0E CAE8E452 F4E3942A 3D729927 35645276 E51B2C84 2499B71A 93398E1A
06F91686 B4CE2883 D4227E36 E9EDDC39 FED100BA 941F22D5 336A9237 C9CA808B
85BD195D 758F7766 51B38B29 B6566F8C A6D43A20 088DE73D 3C324E7F A3B1F3AF

Kiểm tra – $G^* = S^2 mod n$.

G^* = 9EF8F907 25918EE7 6464EE00 478D590A E9C2D9F6 8D2AF809 36E56BF2 3CE61543
DED61A38 3902F482 58D60176 01CEF95A E6711350 D1761AC9 F59DDE36 20410BC4
DE39594D 593C85CF 4AD7CE21 91B4C6EC 42F09A68 E5B68C91 7A899905 D1EE4EFD
CAE7C3BD E198EB64 5E4D65BE F50F19E3 F5CA5863 E30D66C9 9198FF37 DFCAB089

Vì G^* đồng dư với 1 mod 8, $F^* = n - G^*$.

F^* = 2DDA5328 280470C5 AFBBF866 78F0E0C6 5B473939 BF146088 B70009A3 8A8C8E25
3BDF02F3 B3DE52E9 364CACAC 3196F828 D5CDCF83 F9529F70 DB26F641 FC112E4C
11ACC6F0 15FF3C57 74C27775 96042A36 81923E5F 7A636D16 EEA8F881 3775E1A8
FB94ED45 9292E062 0AB94764 8E5FA0D7 5B53051C C87F4ECF E350AB8E 4DADABBC

HH^* = 648E5FA0 D75B5305 1CC87F4E CFE350AB 8E4DADAB

Để khôi phục lại xâu trung gian, một giá trị mặt nạ gồm 856 bit (=1024 – 160 – 8) được xây dựng từ HH^* và XOR với các bit trái nhất của F^*. Giá trị salt được khôi phục lại là rỗng.

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 00000164 8E5FA0D7 5B53051C C87F4ECF E350AB8E 4DADABBC

Thì, một xâu gồm 224 bit (64 bit 0 và 160 bit của $h(M)$) được băm thành HH.

HH = 648E5FA0 D75B5305 1CC87F4E CFE350AB 8E4DADAB

C.2.3 Thông điệp rỗng với giá trị salt

Các thành phần dữ liệu để ký/kiểm tra – Độ lớn của mỗi thừa số nguyên tố là 512 bit (một thừa số nguyên tố đồng dư với 3 *mod* 8 và thừa số còn lại đồng dư với 7 *mod* 8). Độ lớn của số mô-đun là 1024 bit.

p_1 = C41DB9CC D8777062 2BEA8836 1E49AFA2 B5B6CBD0 28479585 472150A1 96C65E89
C2114580 FDE60F6B E12CA9DD A370A3EA 74D33B52 8EB791A9 0FD52818 3D8F612F

p_2 = F69AD66B F97E4CCC B4A76FD3 1F43871D C71100CA F9256C3D BE98CC23 BEC06324
A2282D3C CFCAF00B 0E7492C0 1FB19CE5 0F73EEFD 1A08B0AE 6756E7DF 5670D69B

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
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 E6721FC4 066D37A8 9BF072C9 7BADB26C F6B29633 043DB474 6F9D2175
```

Chữ ký – Thông điệp là một xâu rỗng. Giá trị salt là một xâu gồm 160 bit.

```
E =     61DF870C 4890FE85 D6E3DD87 C3DCE372 3F91DB49
```

Với RIPEMD và PSS, chuyển đổi một thông điệp (rỗng), một giá trị salt (20 octet) và một giá trị trailer (một octet có giá trị bằng "BC") thành một giá trị đặc trưng (1024 bit).

```
F =     73FEAF13 EB12914A 43FE6350 22BB4AB8 188A8F3A BD8D8A9E 4AD6C355 EE920359
        C7F237AE 36B1212F E947F676 C68FE362 247D27D1 F298CA93 02EB21F4 A64C26CE
        44471EF8 C0DFE1A5 4606F0BA 8E63E87C DACA993B FA62973B 567473B4 D38FAE73
        AB228600 934A9CC1 D3263E63 2E21FD52 D2B95C5F 7023DA63 DE9509C0 1F6C7BBC
```

Vì $(F|n) = -1, G = F/2$, nên $(G|n) = +1$. Do đó $S = G^s mod n$.

```
G =     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

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
```

Kiểm tra – $G^* = S^2 mod n$.

```
G* =    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
```

Vì G^* đồng dư với 6 mod 8, $F^* = 2G^*$.

```
F* =    73FEAF13 EB12914A 43FE6350 22BB4AB8 188A8F3A BD8D8A9E 4AD6C355 EE920359
        C7F237AE 36B1212F E947F676 C68FE362 247D27D1 F298CA93 02EB21F4 A64C26CE
        44471EF8 C0DFE1A5 4606F0BA 8E63E87C DACA993B FA62973B 567473B4 D38FAE73
        AB228600 934A9CC1 D3263E63 2E21FD52 D2B95C5F 7023DA63 DE9509C0 1F6C7BBC

HH* =   632E21FD 52D2B95C 5F7023DA 63DE9509 C01F6C7B
```

Để khôi phục lại xâu trung gian, một giá trị mặt nạ gồm 856 bit (=1024 – 160 – 8) được xây dựng từ HH^* và XOR với các bit trái nhất của F^*. Giá trị salt được khôi phục lại là một xâu gồm 20 octet.

```
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
```

Thì, một xâu gồm 384 bit (64 bit 0, 160 bit của $h(\emptyset)$ và 160 bit của E^*) được băm thành HH.

```
E* =    61DF870C 4890FE85 D6E3DD87 C3DCE372 3F91DB49

HH =    632E21FD 52D2B95C 5F7023DA 63DE9509 C01F6C7B
```

C.3 Lược đồ GQ1

Thành phần dữ liệu để ký/kiểm tra – Độ lớn của mỗi thừa số nguyên tố là 512 bit. Độ lớn của số mô-đun là 1024 bit. Số mũ kiểm tra là $v = 2^{80} + 13$ (một số nguyên tố chia hết $p_1 - 1$ hoặc $p_2 - 1$).

$p_1 =$ D716BEA5 9AC10B1C B5CFD57D 0204C349 52240F8E 9BDD319D 4F5ADD0C D9478B7E
AF96558F 85A74A20 B6664136 DD589F35 CFF94287 1B3298BE 40ED2C86 899186E9

$p_2 =$ FBB4E01A A4BF2952 CE9BEDD7 0EEB1EC2 51CD63D1 0BD4332F 3A822FC4 4065FBC6
0197A2F7 0C969BCA 54BF79C6 6D9A2907 C06794F6 EF40CABB B45079DD 9BEBA6F9

$n =$ D37B4534 B4B788AE 23E1E471 9A395BBF F8A98EDB DCB39923 06C513AA A95E9A33
5221998C 20CD1344 CA50C591 93B84437 FFC1E91E 5EBEF958 76158751 02A7E836
24DA4F72 CAF28D1D F4296523 46D6F2C3 E17C6528 8790F6F6 D9783521 6B49F593
2728A967 D6D36561 621FF38D FC185DFA 5A160962 E7C8E087 CE90897B 16EA4EA1

$v =$ 10000 00000000 0000000D

$u =$ 08943E6A 64EE957A 4414AD43 2353F3E5 8DC47A64 207C07B2 A43F9C89 A4C36E25
EC66D68C A67B5931 07C612A2 B13C6AED 06AC073B BC625197 DCD66B0D B0B5C608
68E87A65 CDAC3207 78EE13F0 D7A3CC06 6F49C57E 0F91B31A BDFE911B DF85465C
A917203E 53625392 7711BDAA 035AC27C 828E69B2 33FD26A9 AB107875 9D47D3D3

Chuỗi dữ liệu định danh là một xâu bit biểu diễn "Alex Ample".

$$Id = 416C\ 6578\ 2041\ 6D70\ 6C65$$

Với SHA-1 và PSS, chuyển đổi dữ liệu định danh thành một giá trị đặc trưng (1024 bit).

$G =$ 3E641A22 D0D0747D 4ACC7188 4D3DFF2B 2ADFDC17 03B5A74E FD8333AB 8C4377BB
2A9B48E7 07F73409 ABFBCD2D ED69F52B 16A145CE 062FE6BD 712C1952 110DFB23
16C5F3F3 21922ED3 75A4DEB8 C41FA79B CAD86B0E A0D8FF02 C9D0D591 1BFF1E87
DBCF073F 71F18C08 EB944AE8 4883A1E1 3FB1DEA1 23B5B1EF EA2A9263 5BD5D88F

$Q = G^u mod n$

$Q =$ 3BED38CE BB1219BC 068774E0 E2655CDE F67FE547 BCF2D9FA 9FE167B1 E63B2F10
1A1483D3 8A8F24ED E365A3E4 4F4F10AD ECEA7B30 D042C14C 162477B8 184AE6CF
AA78441B 1FDFB0B2 23ABCD52 8B61F313 D859FCF9 C26FCAF9 E4D9DA9B A83E9D2F
DA041E8C CBF90056 C31D654B 546C1A7F 6729A8DD 8E68512F 39E3B6F0 7959CE61

Chữ ký – $W = r^v mod n$.

$r =$ 487CDB00 41BEED03 23FDD3DE C8542584 FA0E6CB9 90FAD587 8DB34E9B EDDC95B6
5D22790C 108E2184 07ED7F7D 686657BA B5A28EF8 1C2E2498 5B56E37D 9934E195
A38A835C C02CEE8E BA2F56C8 7663E332 976F5A37 20DACA12 0BCD3DF0 AEF6FD78
582EBFCE E6D05E06 172A871E AB0E8F5F C22DDB60 0F541B87 CF8E1473 58374406

$W =$ 649A17DF 13BE8088 55E154B0 E6698DEC 528A26FD 447CC267 CF040FCE C262D0EE
8B9BFCF4 C1053A4B 997755B4 8A207E83 AB16C84D 7137BC60 0FC50DDD 6E12C4FA
F0E2429C AACDDE3A 2C2D15F6 6D57E9A6 9389DAC2 D96A4D1E A34C1DD9 4E067D4C
AA8D8E7D 13F71B0B 6CFED133 8A42F6E7 94A81579 FA374E21 90B318B6 21139691

Thông báo là một xâu gồm 57 ký tự mã ASCII, hay là 448 bit.

M = abcdbcdecdefdefgefghfghighijhijkijkljklmklmnlmnomnopnopqo

= 61626364 62636465 63646566 64656667 65666768 66676869 6768696A 68696A6B
696A6B6C 6A6B6C6D 6B6C6D6E 6C6D6E6F 6D6E6F70 6E6F7071 6F

Với SHA-1 được sử dụng theo biến thể băm đầu tiên, tính toán $H = h(W||M)$.

H = 99394F1D 15924C03 74CF5DA4 85FCB2EC F5303F7F

Vì $v = 2^{80} + 13$, phần đầu tiên của chữ ký là một xâu gồm 80 bit.

R = 99394F1D 15924C03 74CF

S = 80C7274C D9F23290 3A6423D9 327156F6 9743EAEF 03E1EFED FDA8474C 97F6570D
9EF53C6C E2AE2BA6 8D01FFF9 AA820682 14BCD775 B95CC297 DDC38A63 741AB316
6B58275E 0FB728D2 6DB18A2C 3F14B621 CF3863F8 648B3149 FE896348 BE73D37E
2F06E6E2 6C84C044 984C09C6 58300B58 EC2383E3 B0A1F139 0D62B772 A69F37B5

Kiểm tra – Chuỗi dữ liệu định danh là một xâu bit biểu diễn "Alex Ample".

Id = 416C 6578 2041 6D70 6C65

Với SHA-1 và PSS, chuyển đổi dữ liệu định danh thành một giá trị đặc trưng (1024 bit).

G = 3E641A22 D0D0747D 4ACC7188 4D3DFF2B 2ADFDC17 03B5A74E FD8333AB 8C4377BB
2A9B48E7 07F73409 ABFBCD2D ED69F52B 16A145CE 062FE6BD 712C1952 110DFB23
16C5F3F3 21922ED3 75A4DEB8 C41FA79B CAD86B0E A0D8FF02 C9D0D591 1BFF1E87
DBCF073F 71F18C08 EB944AE9 4883A1E1 3FB1DEA1 23B5B1EF EA2A9263 5BD5D88F

$W^* = S^v \times G^R mod n$

W^* = 649A17DF 13BE8088 55E154B0 E6698DEC 528A26FD 447CC267 CF040FCE C262D0EE
8B9BFCF4 C1053A4B 997755B4 8A207E83 AB16C84D 7137BC60 0FC50DDD 6E12C4FA
F0E2429C AACDDE3A 2C2D15F6 6D57E9A6 9389DAC2 D96A4D1E A34C1DD9 4E067D4C
AA8D8E7D 13F71B0B 6CFED133 8A42F6E7 94A81579 FA374E21 90B318B6 21139691

H^* = 99394F1D 15924C03 74CF5DA4 85FCB2EC F5303F7F

R^* = 9939 4F1D1592 4C0374CF

C.4 Lược đồ GQ2

C.4.1 Ví dụ thứ nhất: $b > 1$ và $m = 10$

Thành phần dữ liệu để ký/kiểm tra – Độ lớn của mỗi thừa số nguyên tố là 512 bit. Độ lớn của số mô-đun là 1024 bit.

p_1 = EBF36016 972BFE86 E5FA0D25 21E852A8 D8D28681 973F9439 9E06DA9B AFB5B9AA
2823FD4B 6788C807 5B9581B5 2E8343F8 AC469E00 37149F01 15404132 E99EDF91

p_2 = F5ACDA1A 3C03EB5D 211AB7D1 6BDC15D8 AA624EFB 1C5CAE72 78B39C6A 86811C74
B1FE14C8 5BC9B189 7D25C467 84551316 D90C92FF B0ED7312 400E0C54 87A5DDE5

n = E26F3B7F 9BB6527A 98C545CC 3AACDE35 234D51B7 199F409A 102EBA25 88C9A15D
4B8937A5 BAD6A5BF 7CE79F28 C95973F4 315B2C13 78BA6783 CCCE8CFE 1A45CEEA
0129B046 9A6820D4 637A5BF3 25E80B82 AFB6F274 10F9D46C 7057066C 40AF0383
BD14EDE6 21DB0B27 EF03596E 6111DDD5 7373B2CA DCC8E18A EE50C918 B19329B5

Vì $b_1 = 4$ và $b_2 = 2$, tham số thay thế là $b = 4$. Tham số an toàn là $k = 8$.

$u_1 =$ 03F315E6 C0CDCB85 B00F7C82 541E4C8A 35891E22 61511F72 2AE62B5E C523F1B8
9A260238 681EA921 278773A8 D164507E 449A3A9B 0EEC075D 5BA41057 632B19CC

$u_2 =$ 0AB0F9AD CC449BAA 1984CDA7 D9159FE3 61CA2F37 E587F887 7348B0FA 92C27661
040EF29F 881E92FD DFB638C0 113E43C8 AA8A1015 A88F1555 F7519C81 5DB733DC

Có 10 số cơ sở ($m = 10$) là $g_1 = 2, g_2 = 3, g_3 = 5 \ldots g_9 = 23, g_{10} = 29$.

$Q_{1,1} =$ 82BBA646 0DE18D07 5DE2E587 21B39EB8 DE519421 6D708F55 AD6F4931 5C5B0855
CBC2998E EFD22770 C86C1D1E 5D86262B 993BA8C1 3B68F1C4 470AA1EC 423AC707

$Q_{1,2} =$ BE7E88FC A3C077CE 99470064 720AFBD1 85EE2F86 BE030D41 CD7963E2 3F6E8F60
AF6E27B9 DADBA151 6CF69B16 689B9B79 B6551C33 31EB9306 EE5A6941 C3510295

$Q_{2,1} =$ B14DE96C 2535745A A34B3383 1851EE0D 3FB2BE8E F35481C4 F70D2C83 9A764413
837CB60F 95C48BB7 9CDA14EB A6BCC2A0 E0534B98 EF31EF9F 2728BD4A 53BAA0AF

$Q_{2,2} =$ 1F63D720 C208381A 5018521A 7A94C3B4 C9391194 CB89A591 811985DE 8D577EA4
FCF1006A 6565450C 765FB060 BE850F6B 6591058A 2EEB4EF6 1E037196 A1F6865B

$Q_{3,1} =$ 3CCA59F0 2BF22ECA F41715C0 EB63B927 57311919 E35C111A FD30B233 0AEF9E24
4ED0258E 9C5D2D88 A3EFBFB1 C8748ADD 806477F1 2557D27A 6E57ADEE A8C852CE

$Q_{3,2} =$ DF5A0BB8 2A12B2AD 53997661 D8B3A0DA D597A0BD 2B45E6FF E1B86C85 74F3066A
A73566CB 65C57F74 2B172459 A7827489 BE751387 8F315CF8 1AD7FC58 A4A4DFE5

$Q_{4,1} =$ 91158AEF FA55FEBE AECF276F D9901100 74F047C0 600D14FC 8214307E 5F54B5E2
B932774A 7A8AD32D A99CDA71 AEA9B497 CC25F7F7 FE4EDCA3 F1E31788 EF5DDE13

$Q_{4,2} =$ 44AB0D39 BC94C14D 84094C96 DC39A55C 5C93E34B BAB404BD D3AC7CE6 20D27F2F
3E18D74A 59947BE4 44A65B15 5C34A5D5 23BEE51D 23222E47 D7DE3853 F1C28AD7

$Q_{5,1} =$ 504A973C 7D80D257 254D57A5 23C66FF5 17BA0459 2BE2905A 4470C934 895CF339
A24DE8C7 9915773B A6D5BABD C94FA867 793709C2 DBC86441 4FC5DB1D A06B98C3

$Q_{5,2} =$ 7C280A6B 5C863BBF A7067371 468F580B 12EA4E02 C7EBBFF1 06425C64 5FA1202B
215C623C A860A064 4717409C 4DE7C025 C2C54C76 115713BA EF38A13D 3519D8FC

$Q_{6,1} =$ 9E28C724 D91C36A1 E698147F C63DAB2C 1DC614B1 2AAB9815 32B5A48F 14294A70
15A02AAF D214E899 283D0C87 FEB6C22C 04A1684E 66746A18 E15E094A 33CE2916

$Q_{6,2} =$ D1B7089C 2B0DB77D A2C30284 F838D625 1BD12E80 DCBE42CF 62C17FD1 D46B67BB
A0856BDB 242CDA05 675FE38E 1D2D2B3D 7411C4DF 5D2693F2 51150984 A9D98825

$Q_{7,1} =$ 90371C09 3331C592 54A4D9F7 CD1D87F6 392D50AD 2927DF6D B0069020 C11DE222
DA979513 EE070AF2 8DEF161E 970771A7 92EBD088 BC035804 5BD90D3B 36FB43BF

$Q_{7,2} =$ 7B073391 164F2A6D 428FAC6C 7641D332 EE990071 2D3B7736 DE04A6FB BAB717B8
031E7E02 1E87A4E0 1EF97F5C 37EA95A6 8E00C9B8 75133CFE 676F0FF4 71D27C99

$Q_{8,1} =$ 1F8AC869 116F8959 65E15081 70E4F943 C4319C3A 86B39FEA 26D51C3D B45C25DC
70DA1286 18E64E76 93E56D98 52F1774E 1A211794 4EE4749B E5EFCD58 EC8E4704

$Q_{8,2} =$ D161EE9D C0955D92 45A09B09 A6296EB4 7CE3798B E799C6BE 1BA43FC4 69837579
A8EF1710 1D706BDB 533757EC 55057767 193BF413 01240301 48EBA7A1 66F94152

$Q_{9,1}$ = 7F6D3D4E EA6D838A CA90E050 0BC11F77 B7B2A7A2 9A0B2D70 FE335817 2D5C71AA
A26A78AF 0EA3C2E9 A24F4809 A7F9D816 297F99E6 4D83F965 29E3EFC5 8AC2D425

$Q_{9,2}$ = 7092A2EF 08527BAA BE0344DC A7D5DDA7 E7B09C61 115D6041 51058F5A 151A2244
972DAFC7 796186FB 7D36416C 0ECE7B65 DA96EEDF 9C086E29 BA468733 59650F97

$Q_{10,1}$ = 4833DAC9 2EFC0A52 F44C33D2 98B4604C A12C33EE 7B122FF9 D079A745 1670096E
21E65D78 DDBBE50A 39CCB146 6D807CCF 3795D5CA 7D846115 00BCFF24 B8B02457

$Q_{10,2}$ = 57AD0549 0C3CBC49 446296D2 FBB05666 72E160A0 7FD80DBF BD2A1A5A 6D6932EA
FEAA46B1 84B3E43A 869A4AAE E8A56015 18789A7D 42273083 944B52C5 20787136

Chữ ký – $W_i = r_i^{2^{k+b}} mod p_i$.

r_1 = 958FE0FE 77561815 FCCE3499 D2AA78C6 701CB4DF 3EAEF982 160F9254 592C63ED
D4692A99 336020DA 4427AD2A 5845CFDD 0153CEB3 6507C76A 9473DAC1 A764E4C2

r_2 = ED1F46C6 B0143F7F A70DC68C 0E8E4324 5F22CE6C BC811A7C E90D7B0C 0D828256
C479922A C1B1CD6E 52DD82F3 75B90D0C 9EA6FD45 34611F9C 2CE4EF1E DB7DB9B7

W = 202B4E86 A41BC533 50A20AB4 BAD183E4 1362321A 6EF33B89 162CA681 C993A94D
0F009CB3 4EFEBECB FB473A02 291888C8 A73D9B90 13D814BF AEFA104D 1B551E59
DFD8A626 C74F9F85 C047D5FF E580277D 14A13B84 537B421B 5E6F8F64 64334BA9
9092041F 9EADBAF1 3EA6246B 8A1E3275 31C41AE2 904FA368 BA980C56 356E4896

Thông điệp là một xâu gồm 57 ký tự ASCII, hay là 456 bit.

M = abcdbcdecdefdefgefghfghighijhijkijkljklmklmnlmnomnopnopqo

= 61626364 62636465 63646566 64656667 65666768 66676869 6768696A 68696A6B
696A6B6C 6A6B6C6D 6B6C6D6E 6C6D6E6F 6D6E6F70 6E6F7071 6F

Với SHA-1 được sử dụng theo biến thể băm đầu tiên, tính toán $H = h(W||M)$.

H = 6AD0F7A4 1C5F7A93 29CA5B49 AE8D7105 7010A69B

Phần thứ nhất của chữ ký là một xâu 80 bit, hay là 10 octet.

R_1=6A, R_2=D0, R_3=F7, R_4=A4, R_5=1C, R_6=5F, R_7=7A, R_8=93, R_9=29, R_{10}=CA

S_1 = 3E356475 F5020A7B 4DD75C90 FCB88994 CC147A08 9A0121B4 3B59119B 5AE46177
02C3672C B745A2A0 7BD11811 39771D77 FFD9ECD6 DBFED354 58E45185 9BA85073

S_2 = 097F6189 B060BE13 DF8CB226 1A72ED8B 29EDA213 90995927 019E9304 8AFC9720
1B9B2942 FECF94D1 893E176C C6DAA4ED 247EF7CD 5D19AE7D 59BCBB54 9518A45D

S = 6EADFEAF 25CFA808 B2BCEC31 6F6CD229 95E8599C 767F6A1F BAD1B2AD 86BE12FE
0CD1D5CB 1A09DB55 147E9D70 D7A13B6D 5A2BE45C 96E12695 D83328BF E0932757
A17EBC09 D0A49E92 BE539CE8 08D4F460 C588817C DD66AAAB 1A44794D 9E789943
E1A42021 AC22F0FC 56908E7E E9D0FB8C 04A6CE88 0F10F085 D72F786B D273EE12

Kiểm tra – $W^* = S^{2^{k+b}} \times (g_1^{2^b})^{R_1} \times \ldots (g_m^{2^b})^{R_m} mod n$

W^* = 202B4E86 A41BC533 50A20AB4 BAD183E4 1362321A 6EF33B89 162CA681 C993A94D
0F009CB3 4EFEBECB FB473A02 291888C8 A73D9B90 13D814BF AEFA104D 1B551E59
DFD8A626 C74F9F85 C047D5FF E580277D 14A13B84 537B421B 5E6F8F64 64334BA9
9092041F 9EADBAF1 3EA6246B 8A1E3275 31C41AE2 904FA368 BA980C56 356E4896

H^* = 6AD0F7A4 1C5F7A93 29CA5B49 AE8D7105 7010A69B

R^* = 6AD0 F7A41C5F 7A9329CA

C.4.2 Ví dụ thứ hai: $b = 1$ và $m = 4$

Thành phần dữ liệu để ký/kiểm tra – Độ lớn của mỗi thừa số nguyên tố là 512 bit. Độ lớn của số mô-đun là 1024 bit.

p_1 = EBF36016 972BFE86 E5FA0D25 21E852A8 D8D28681 973F9439 9E06DA9B AFB5B9AA
2823FD4B 6788C807 5B9581B5 2E8343F8 AC469E00 37149F01 15404101 12ECF827

p_2 = F5ACDA1A 3C03EB5D 211AB7D1 6BDC15D8 AA624EFB 1C5CAE72 78B39C6A 86811C74
B1FE14C8 5BC9B189 7D25C467 84551316 D90C92FF B0ED7312 400E0BA5 327E1DF3

n = E26F3B7F 9BB6527A 98C545CC 3AACDE35 234D51B7 199F409A 102EBA25 88C9A15D
4B8937A5 BAD6A5BF 7CE79F28 C95973F4 315B2C13 78BA6783 CCCE8C2C AC4BB5A4
FC439166 CAE4EE3B 4C8C9A58 CC18654A 87E1DD6E 2223DF5B D728EDA2 DB46D042
25E3DB20 0BF6F035 8ACA6C79 61D12407 A768CF6F B3824000 5B1C0A66 903DF805

Vì $b_1 = b_2 = 1$, tham số thay thế là $b = 1$. Tham số an toàn là $k = 20$. Do đó, số mũ kiểm tra là $v = 2^{21} = 2097152$ (ở hệ thập phân).

u_1 = 11411739 5367474A FC81C9AB 8C9E5F19 4B79E03E 9A85D9ED 690E5EF6 F67ABFCC
13732A9C 8A55D80D FE1D7137 0A3718BF B785B28B 5EAF213D 0F5A3FB7 E786B7C1

u_2 = 650DECFE AB2D5927 8AB00315 F1142953 632009EC 9344DB6A 74781226 FF34646B
941B28FA 28D58264 27A67783 D8084107 1394A798 DB25907F 7CF19802 3C092551

Có 4 số cơ sở ($m = 4$) là $g_1 = 2, g_2 = 3, g_3 = 5, g_4 = 7$.

$Q_{1,1}$ = 6B6C99CB 3C7BA9EC E455C0D9 75D97D24 BD8EFDA7 9C42B083 ED036C00 5F60D226
A458A073 1D4706AA 30C83CC9 E5B40772 4D30B963 4A82A7C6 8C3AD268 92ECD9A7

$Q_{1,2}$ = A2DBCAB3 9DC79BE3 0CEBEB77 6711016F 3F58CFE5 7511F1BE B0FE6858 C9CAA0D9
F77AD391 DE4B2348 54FDB389 A2919770 FE1BEBD6 266B2242 0113254F 3F2BF010

$Q_{2,1}$ = 2504EFC8 FF8B668D CECBA1CD 74C7385C B21F14EC A19D4169 1F6F79B7 99B67B9A
8460DACC 08D6D751 CEB4A936 F8048D43 7A5D7F53 D18551E4 EBE20773 782039A6

$Q_{2,2}$ = 2191936A E4032F92 E3D56BD1 837A50B5 26EAF3BE 8B21D9ED 9C31D966 FCA6EFA1
70BB5E6C 48947C34 276A56C9 3C3E60EE 1BCDCA60 3BF54BA8 7E06F299 9E926F66

$Q_{3,1}$ = 25E99A95 36B0A61F 611C677C 0BE33157 4CC00CCE 52E88139 EC4D8F6A 9AD417F4
9B37668D 0793EA8E DE220ADA C671A811 27599970 73869A53 632D6050 CE6534B8

$Q_{3,2}$ = 0E96C5C7 8762E342 3F1AC822 9611409B EF5AE5D7 FA68AF22 C6A94DC4 D202DFDE
0A3BDAF2 31B4C3E2 D8B4FFDC 06FD4D18 768E7829 00550B83 E75A7A8B 648E51A3

$Q_{4,1}$ = 31437E9C BB2A3FF9 32016F4B 1A3D0C77 7AD99519 085466AB 8D4010C7 32330887
C044CD80 3DEDE7B2 60321F13 CE4E0656 8C352155 3277EC9C CAF9132D 64EC3639

$Q_{4,2}$ = 7B4187E7 C0F8D801 D9CE9A35 CD2408B3 4B83AD55 1BB1A106 4AA62448 51B1861E
99EBA585 F182E835 42BD9ABE E5E40FCA 25A4395A C89001A4 E926D644 BC7163C1

Chữ ký – $W_i = r_i^{2^{k+b}} mod p_i$.

r_1 = 958FE0FE 77561815 FCCE3499 D2AA78C6 701CB4DF 3EAEF982 160F9254 592C63ED
D4692A99 336020DA 4427AD2A 5845CFDD 0153CEB3 6507C76A 9473DAC1 A764E4C2

r_2 = ED1F46C6 B0143F7F A70DC68C 0E8E4324 5F22CE6C BC811A7C E90D7B0C 0D828256
C479922A C1B1CD6E 52DD82F3 75B90D0C 9EA6FD45 34611F9C 2CE4EF1E DB7DB9B7

W = 82074289 8E8E9537 437D57D4 17184A82 06FEB795 F9CA167D 60BB7314 EB8F1360
5882C202 467DD2C0 F7F8D14B 87A7FB41 15D68D1C D6313C14 CA24DD84 E4F293F6
30AF2A90 EB122FD1 E113C184 DCB976AC FBCD0CA4 35EF6CDD E5F66F4C 06947B36
5E1E3B03 3D766C5B 8619B164 6470A0FA 961008A7 90CAA733 8F3119B1 C10263B8

Thông điệp là một xâu gồm 57 ký tự ASCII, hay là 456 bit.

M = abcdbcdecdefdefgefghfghighijhijkijkljklmklmnlmnomnopnopqo

= 61626364 62636465 63646566 64656667 65666768 66676869 6768696A 68696A6B
696A6B6C 6A6B6C6D 6B6C6D6E 6C6D6E6F 6D6E6F70 6E6F7071 6F

Với SHA-1 được sử dụng theo biến thể băm đầu tiên, tính toán $H = h(C||W||M)$.

H = DF96299E D7FFA63E 421B021D BBF1F0DF F9EFF729

Phần thứ nhất của chữ ký là một xâu 80 bit, hay là 4 xâu 20 bit.

R = DF96299E D7FFA63E 421B, tức là: R_1=DF962, R_2=99ED7, R_3=FFA63, R_4=E421B

S_1 = 7D41042D 3E007FF9 A0CFD957 FB31EDDE 5C38DB80 9031311B 87678442 CCF9B760
73198995 77A622C7 93E442C4 E4B00CB3 2AD7C919 8284D27D 10BBF60A 3D0C8943

S_2 = 8F21A259 C79DD004 9AE9F552 B9120E27 452B639F 5E32FD7D 1CC80F3F B63F91D2
B39DFE65 9B1ABEF0 B77ACB0F 24438FAE 86C2D492 0993E936 AC1F3E3B 87741ADB

S = 0C0C08F9 AAF3ACC0 13D4B871 C087B78F A971456C 0E6531DB FD40476B 9572B5D2
DF3080D0 032350A7 96976D07 A32323C0 CA64E943 786E5324 201A79AB FBEFD412
51A8C425 5C7BA61F 4583FBA4 C93E9332 90E9B89F 06FE1F6C DE65A020 E092131D
6CAF52E9 A7E0748D A63065FA 39E97AB7 A56C587E 1AF1E781 F1DA70BD D29CD81C

Kiểm tra – $W^* = S^{2^{k+b}} \times (g_1^{2^b})^{R_1} \times ... (g_m^{2^b})^{R_m} mod n$

W^* = 82074289 8E8E9537 437D57D4 17184A82 06FEB795 F9CA167D 60BB7314 EB8F1360
5882C202 467DD2C0 F7F8D14B 87A7FB41 15D68D1C D6313C14 CA24DD84 E4F293F6
30AF2A90 EB122FD1 E113C184 DCB976AC FBCD0CA4 35EF6CDD E5F66F4C 06947B36
5E1E3B03 3D766C5B 8619B164 6470A0FA 961008A7 90CAA733 8E3119B1 C10263B8

H^* = DF96299E D7FFA63E 421B021D BBF1F0DF F9EFF729

R^* = DF96299E D7FFA63E 421B

C.4.3 Ví dụ thứ ba: $b = m = 1$

Thành phần dữ liệu để ký/kiểm tra – Các thừa số nguyên tố và số mô-đun giống như trong C.4.2.

Vì $b_1 = b_2 = 1$, tham số thay thế là b = 1. Tham số an toàn là k = 80. Do đó, số mũ kiểm tra là $v = 2^{81}$.

u_1 = 35C918A9 00582D37 8FE446C5 6C90396A 545739DE E16BFE29 5C594FFD 32A42925
B36E7CE9 3C29725E 16A1E4FC 2C97313D F047890A 00C5CA74 30A8986D 14BC0B69

u_2 = 0D5D9A73 FE1370BE 6C5B7AFA 29CADD59 19B472B0 1E910CC2 5F9F7D89 8F9FBD77
37581681 EE2223DE 437CBDF2 9E84EE09 98C38FF9 BC02AFA6 0E3085DA 83ADD03A

Có 1 số cơ sở ($m = 1$) là $g_1 = 2$.

$Q_{1,1}$ = 5A8F0C99 74AA0B14 D3B10DFD B6E6B744 F9EF70C3 C048C232 AD051D03 01F471BC
CA8B82E1 C9888090 D5EE6942 70E86782 9AD31130 CC80C82F D1F61AC6 951E38F0

$Q_{1,2}$ = C7FDCB32 349A74E5 51200E87 245B8C1D C9E14192 68342A85 6DC5AA9F 58CE5B30
89F36C45 3CDDD494 1D14F696 61CE1EEA 71FED68E C8549FC1 1772AC8D 22F9322A

Chữ ký – $W_i = r_i^{2^{k+b}} mod p_i$.

r_1 = 958FE0FE 77561815 FCCE3499 D2AA78C6 701CB4DF 3EAEF982 160F9254 592C63ED
D4692A99 336020DA 4427AD2A 5845CFDD 0153CEB3 6507C76A 9473DAC1 A764E4C2

r_2 = ED1F46C6 B0143F7F A70DC68C 0E8E4324 5F22CE6C BC811A7C E90D7B0C 0D828256
C479922A C1B1CD6E 52DD82F3 75B90D0C 9EA6FD45 34611F9C 2CE4EF1E DB7DB9B7

W = D8B7FC73 E7D63980 40BD83D2 10C218E3 54E05104 A7F5F504 C504104D 45FE0EA6
829D5CFC 4FBAA8A6 291E86FE 78C5C8DE A32D58C2 D23831C4 5A3977B1 5A3AED68
2D7FCA9E 3C6AB4D9 BA502D7C B78D9BF4 FF4E1D1A 07462D0E 1E80A010 1232C74A
57481C4C ADFCBCDB DDD467D4 84829DB8 DF3D0F29 FCAB2A33 58C8EFE2 B22E541E

Thông điệp là một xâu gồm 57 ký tự ASCII, hay là 456 bit.

M = abcdbcdecdefdefgefghfghighijhijkijkljklmklmnlmnomnopnopqo

= 61626364 62636465 63646566 64656667 65666768 66676869 6768696A 68696A6B
696A6B6C 6A6B6C6D 6B6C6D6E 6C6D6E6F 6D6E6F70 6E6F7071 6F

Với SHA-1 được sử dụng theo biến thể băm đầu tiên, tính toán $H = h(W||M)$.

H = 6038021F 5173AD35 D0228511 1BC06E71 BE283E8C

Phần thứ nhất của chữ ký là một xâu 80 bit.

R = 6038021F 5173AD35 D022

S_1 = 3F6109BE 4C85B65E 052ECF16 466E5ED4 697EB562 22D9C28B D7AFCED9 84E2D6E0
8EA6CC1B 674AAD88 8A87E393 FF614842 2B4D222A 89FDD988 5D2B7BAA 0E2790F4

S_2 = 730249AC 01308B9B EE624AA0 B461462D 29F585F9 010BE04E D1A624F5 9B6350E7
65FE0C51 B89C7D30 8CEFE2C2 2EA84C82 C68E3228 47A33FFE 11B2EE4F D34DE163

S = DCBE96EA D4E23255 6DC10F5E 6657FC84 C76A291F 7D5EEE0B 0E3A3F21 D17BA34A
FA2EB265 A4AD8E99 94100F8A B676506C 2CBE826C C2CF5591 79FE4509 0C90BAB4
E4A29553 577D0CFD FAF8D2CB D502E501 0BDC5B29 05FA1092 6DF1C571 36CC580A
62F3FF2F 02D5AA5F 2EEED0F7 4CB5A612 E8E8CA51 5E5225E9 2B5F5BDA E6FED15E

Kiểm tra – $W^* = S^{2^{k+b}} \times (g_1^{2^b})^{R_1} mod n$

W^* = D8B7FC73 E7D63980 40BD83D2 10C218E3 54E05104 A7F5F504 C504104D 45FE0EA6
829D5CFC 4FBAA8A6 291E86FE 78C5C8DE A32D58C2 D23831C4 5A3977B1 5A3AED68
2D7FCA9E 3C6AB4D9 BA502D7C B78D9BF4 FF4E1D1A 07462D0E 1E80A010 1232C74A
57481C4C ADFCBCDB DDD467D4 84829DB8 DF3D0F29 FCAB2A33 58C8EFE2 B22E541E

H^* = 6038021F 5173AD35 D0228511 1BC06E71 BE283E8C

R^* = 6038021F 5173AD35 D022

C.5 Lược đồ GPS1

Thành phần dữ liệu để ký/kiểm tra – Độ lớn của mỗi thừa số nguyên tố là 512 bit. Độ lớn của số mô-đun là 1024 bit. Xâu bit bí mật bao gồm $|H| = 160$ bit. Số cơ sở là $g = 2$.

p_1=

EAB2E6E3 022960B7 28E00DDD B4439E87 067B9D2C C0C6DF4F AA2E7CC9 A65E6C3D 4D95ECE7 D983B3C4 EBE812C8 99F050F4 D5D231E0 9399CAB8 6ECFB654 02C0E4EB

p_2=

D115FD6E 87944C3F 407ED927 7D1178A3 A0C01A41 DD446EAC B89CC6BC 2FC01846 5D6C4E74 EDAD1C4E 17BFF8E7 882E3E07 C25AEFF3 3BD59EB1 62AD57B2 CA9717D3

n=

BFB03784 4B667442 37043AF8 16AD20C6 E719F8C0 E18E4A35 E3BFD9B4 7BF63F05 E08CCFDD B89763A2 D8EA6889 C6D17F73 39061A58 02981F10 6461F87E 3DA25C39 154C51A9 8263AE43 6686821D E53F2AFA A1C4CA6A 040D892C 39A33483 00D69532 E611379F 7C4B7659 95F1FAE4 FA3D33FA 60A71433 2B97422B F508B04C 0E2ACAB1

Q=

F2B65E4B 46BC211F 2A2909B5 77F9BF40 42B49595

G=

B4800C63 F655E640 028C05DB A59D5C4A B221CEB3 26EC5BD0 FB0E3961 28803C04 C40EE4A5 892FE494 85F639E5 429B58FC 1B77B412 AC08E848 AFFD6E39 58666FA7 F098F1BC 61153A9A 475E51EE 90A50F77 98C7068F 7B12A7D4 1891BFCC 9B21E186 13E41F1F A106AC57 1B670979 A9FD90D9 5A237208 8C2CAD54 C13CE112 42E1F912

Quá trình tạo coupon – Mỗi coupon (và mỗi phần thứ nhất của chữ ký) là một xâu gồm $|H| = 160$ bit; Do đó, mọi xâu bit ngẫu nhiên gồm $2|H| + 80 = 400$ bit.

r=

DD3B 0C9E9D3C 11F8A12C EF86B279 844FEFB9 1CA37E5E 4D953477 25A6E22E 48938CAF 145B0EE1 9923E1E8 63333BC5 AB37111E

W=

132F2236 49AB1067 1D06D167 D2815583 B075A639 D045009E 52B0E888 6046EEC5 52999E94 7E95EA97 F8C39073 24B3B1CD 8C638B15 012B2FD9 C8AA4BC6 80370FF1 5395986B C6E8DB17 7422974E 0C3F783A A549EE39 61E478E4 BB34CC0E 004D6CB6 72390C78 A26642ED 78828E77 ABEC813D 40F27174 EBAA1A10 5B60FFB1 36A471FD

T=

ACAE249E 1FC4322D D96E98C3 19C9DD28 7E126180

Quá trình sử dụng coupon – Thông điệp là một xâu 48 octet.

M=

F8D4764A 2B716EEB F31A5EC3 A2A214BF DEA62B53 C11A4D89 CA72E95C BCC15359 70786B89 0C3704E5 D7FE2D45 07719718

Với $T = h(W)$, với SHA-1 được sử dụng theo biến thể băm thứ ba, tính toán $R = h(h(W) \| M)$.

R=

2EB8ECA4 021955AC 113BC1C6 80058F99 4EEE51FD

S=

DD3B 0C9E9D3C 11F8A12C C33A9A18 99C00B59 79876097 FE059CC1 C7EC5D77 7AB96A70 5783AC00 72C36AEB 406839A5 24E517DD

Kiểm tra – $W' = G^R \times g^S mod\, n$

W'=

132F2236 49AB1067 1D06D167 D2815583 B075A639 D045009E 52B0E888 6C46EEC5 52999E94 7E95EA97 F8C39073 24B3B1CD 8C638B18 012B2FD9 C8AA4BC6 80370FF1 5395986B C6E8DB17 7422974E 0C3F783A A549EE39 61E478E4 BB34CC0E 004D6CB6 72390C78 A26642ED 78828E77 ABEC813D 40F27174 EBAA1A10 5B60FFB1 36A471FD

h(*W'*)=

ACAE249E 1FC4322D D96E98C3 19C9DD28 7E126180

R'=

2EB8ECA4 021955AC 113BC1C6 80058F99 4EEE51FD

C.6 Lược đồ GPS2

Thành phần dữ liệu để ký/kiểm tra – Độ lớn của mỗi thừa số nguyên tố là 512 bit. Độ lớn của số mô-đun là 1024 bit. Số mũ kiểm tra là $v = 2^{160} + 7$ (một số nguyên tố chia hết $p_1 - 1$ hoặc $p_2 - 1$). Số cơ sở là $g = 2$.

p_1 = C64CCAD5 257D396F C5C913B6 7DA871B2 93A2F18F B96DB409 10732E70 9C5B43BB
5CD2F846 080CD347 9D82CDA5 3138D667 AD1ABB51 F0969798 19D12064 C6BA2447

p_2 = C8A11B13 662D4910 E6950FD5 319C8DB0 9A569353 389ED9FE FB74291D C22ABBDF
8BE79413 030E4029 190DB076 4BCB7F6B C4CF5557 63C38E41 ECB6BFB1 2D5AFFB1

n = 9B68C9BB 35939E50 00D1EE1D BB8C398B DBE8EBEC 34A2DE5F C6683C06 E5C3D726
89A0D1AE AC6E2ED8 18063C75 4AE472D9 9814D17F F466CD99 49DB846E E0342555
F5259565 66E0B02F 88C01C3A 4A67B8EC 93B7CAF1 8B556218 EF87F670 9DCC0CDC
DA91CFA7 E8290D66 04EE08DC 08C20B5C 7FB9029E FD3537D8 C9766B89 0CCBCE17

v = 1 00000000 00000000 00000000 00000000 00000007

Q = 32872E7E E4C3DE36 4E6055DD FF82C082 F79B044B F577CE94 A88C99AE 0012EEA3
4FE81876 2A5F4791 76F23313 6FC8A4B3 89398CD8 2FBD0833 F599145D 7E3B3598
F2E016B9 649FB68D 23518763 A2F65A73 7302EFD5 90F0BEA9 DA4047FC 49A11B72

9AF6499E 56DA3DF2 A71DC422 FEE9DF17 280BC086 FCB2BCD2 15379E6E DA40D117

Quá trình tạo coupon – Mỗi coupon (và mỗi phần thứ nhất của chữ ký) là một xâu gồm $|H| = 160$ bit; Do đó, mọi xâu bit ngẫu nhiên gồm $\alpha + |H| + 80 = 1024 + 160 + 80 = 1264$ bit.

$r =$ 5DB7 023CC4A7 0D37412C 5FD64999 D19D86B4 42FE83BA D7263123 D8188DED
95D04097 64FDB882 9E170B9C D251C234 EC61ACA8 7439BCD3 5C62A1A6 7993AF09
913F2386 C4B77433 AFD5C0BC 2FD53E86 57FBCAF8 5E3CC341 C93AEFE5 ED4B8CB6
B31190F8 35F84100 36B15A7E 0594C11C 37639BCB F75F8C29 08248EE6 743DE34D
FE1284C0 D7745FFE A34433BC CDA50ECA 34BA2130 A6EF18D5 663DECD8 D554F2BD

$W =$ 09B4B9CA 5EAE564F A0A59D22 664E684B 31D84EF3 50FF3F69 55052C43 66F40312
AA9560E3 606A66ED 2CADFF94 4EFA26F2 14F4F155 16112B41 72BA7C28 BE863341
C1CF22E4 DDF29862 FADC1541 7E7258DD 708BEDD1 DBC4430B 1843E07A F433E411
09CAD9E3 E9FCDEAA E57C55FE E923F663 6F6C065B FA9FB98E B275FAA0 E3D7C719

$T =$ FBCD0E15 CB29E560 16C84F6A E0531651 BDA9393D

Quá trình sử dụng coupon – Thông điệp là một xâu 48 octet.

$M =$ 35063821 CEA6315C BB8D534C E6DB6476 9E2E9925 25E460D8 6DEC9A07 B234F6B7
1233128A 082E5B62 B2B8D500 5784340B

Vì $T = h(W)$, với SHA-1 được sử dụng theo biến thể băm thứ ba, tính toán $R = h(h(W)||M)$.

$R =$ 1444BA78 AC805FB6 0AB6FBAC DBFD38C0 683CAA45

$S =$ 5DB7 023CC4A7 0D37412C 5BD62941 84D74DE6 D98FC718 5E2AF30D 40660D86
3D7E8F4C B50F03A2 96B0D7F4 3575A3CF A598BE50 5A3198A6 0A66191E 3378C45E
92B8F961 6E3D1932 7A5768C5 9343E429 E7584550 BAE2551C 013F89CB BB844A39
E9B8C396 AA0E2F67 66EE15F7 D79B11CE 22A27FD4 C1AD8645 DF457264 8E0E2653
05BB5E3E 07717841 CB44F7A9 1D0A8E23 BA119FA0 05F8E075 655046A6 9D9F518A

Kiểm tra – $W^* = g^{v \times S+R} mod n$

$W^* =$ 09B4B9CA 5EAE564F A0A59D22 664E684B 31D84EF3 50FF3F69 55052C43 66F40312
AA9560E3 606A66ED 2CADFF94 4EFA26F2 14F4F155 16112B41 72BA7C28 BE863341
C1CF22E4 DDF29862 FADC1541 7E7258DD 708BEDD1 DBC4430B 1843E07A F433E411
09CAD9E3 E9FCDEAA E57C55FE E923F663 6F6C065B FA9FB98E B275FAA0 E3D7C719

$h(W^*) =$ FBCD0E15 CB29E560 16C84F6A E0531651 BDA9393D

$R^* =$ 1444BA78 AC805FB6 0AB6FBAC DBFD38C0 683CAA45

C.7 Lược đồ ESIGN-PSS

Thành phần dữ liệu để ký/kiểm tra – Độ lớn của mỗi thừa số nguyên tố là 768 bit. Đô lớn của số mô-đun là 2304 bit (= 3 × 768). Số mũ kiểm tra là $v = 1024$.

$p_1 =$ 8332D671 713A0DEA 71E9453A B323C499 2455D957 EF6985A5 3770AF04 E1C76529
A0BC855E CA025F9C 540CF0B5 3684EA5E 5777B647 17E78B99 1C2BACB6 9BEFED40
F414D805 A1594E56 90CE67F6 42C42714 7C94BA1F 2DC9ADF8 EACD114B 1723700F

$p_2 =$ FD3764F3 7B98DFE4 8E30B2C4 004E2D03 0A5F8018 2F943156 FE6E4B5F 16F902DA

D60E4730 30DEAB98 75F3D749 DE79C361 8874D195 4102DFE0 47637BAB 495C7DC2
912FDEB9 4B2D5ECA B798E90E C6E634B7 B4F1153B 4D9F4BD0 3C45CFC7 2600E549
n = 805C6554 66EEA57C A1798241 5AA1ACA7 DF54AB5C 17953109 9A08CF05 5D6BD99F
7E5D4FF8 95CB633B 3368DAC6 8C3FF751 1C5CCF45 6ADE1AA2 20558DAD 17D466DF
F0E7F3B9 3DDD6934 07A18A66 BC74CEB1 EBAC6901 4B6CE22A 78E70676 4CA5DE4D
196C7007 54CB46C7 30F77BC0 BC1955CC FB26DF7E 4C005DC7 B836ACC2 F04E696B
10578B6D 2CB993F3 4A01FB95 2727517F 4AC8499A 51829133 16B2FCAA 5C594C3E
9B8B24EC 313C8863 4B7BBFEF BFDAC7EB 689C79A8 6B5C4401 B7ECE53C AB9F2326
25C70842 2F5FE450 9631128D A2775427 0AF91FC9 B09300A0 E4339609 AA9A10B6
2F6812F1 91A3D598 177001A0 88DB58A4 AD2FEF5A 230735E0 0AEB8031 50403D11
51F15167 65BADA30 D57F2B4C B9438E59 551828F1 9704AAB5 4169F107 E66DAE3F

Chữ ký – Một xâu bit ngẫu nhiên gồm 1536 bit; sau đó tính toán hai số, ký hiệu là y (2304 bit) và z (768 bit).

r = 76A4D0DD 5B024775 2D546CA4 27B6E8BE 18BD2BA7 33842CB7 4399B33F CA7BFACA
346FCF34 77F20811 5576E1E1 BB6AF124 08633C3F B2918EAB 3A1645AE B58CFF4A
9265CC40 8F3F3AD6 8A4AE202 A11511D0 06BB0023 1C86E725 A39AF1A6 B1C83F2C
38716DD2 49C82A4F DC7BE305 2C78FFB4 887F7935 CE3932EE BAA8C80B 7491E0B6
38D5F816 3794EC9C 158F088E 1A93BEF8 93199AA1 E07BC11F 86FBCF75 76F28B89
261FE806 BAFF4451 83209223 807F5012 6D4C983C BE96C6DE 6ACBDC5F 9EF1D975
y = 7350F3FD 13A3E49B 4C7F83ED 334E45E6 28C9AA65 A2A9298C C6E52B23 FDB1AE1E
2197DA72 AE23AF02 9241408C DF5287BB 04CF88CC 871721ED D887A1BC A8E261F3
69A85E6C 77BF1A97 F511FD5B 4521C276 250C92E4 06954B0B 7FB59209 B8940FEB
6A20D4D6 FFEE125B 959E8F9F 2486AC2C 9F609561 363B7B7E 3FD93410 94C9D507
3C5075B3 71A41B98 D7E98778 D52922A2 319FED3B 88AF194D 841F9837 6F4B905C
E2835B36 1C226BEF B3B2DD84 C8A69B19 6AE5BB51 92B6DB42 7E75DD07 A3A2BDF2
8C6AFF24 482FDC8B 3592FF0A E130DA0C 513D9D75 31089919 6C94C114 10B90EE8
78FCAA83 02232BBF 17960B74 4E411455 4EB04652 C23B9D13 7F959E06 5499FCF4
7853786B EAF792D8 B8E76C92 6BC17587 346B2187 D7059CAD 9A01DF44 475FEC58
z = 037C592F 20A80F8C 9B296800 12F1D8A8 EDE80CDE 1A89AE4D 3E73014C 2ECA84AE
313C5A34 13388E16 E2EBE89F 42510A45 F68D0417 00EE31F3 F5E3340E BCD1D226
DBF0B6AA 7D5EABC0 57C90D78 618E2836 28D6EB9E 79D7CEF7 82D8CB35 E91F0CB9

Thông báo là một xâu gồm 3 octet. Giá trị salt là một xâu gồm 32 octet.

M = mã ASCII của "abc" = 616263

E = 3438C82B A8799E1F 1301BA2A 14BF2133 CFFF625B BD819493 9BC4107C F7384B9D

Với SHA-256 và PSS, biến đổi một thông điệp (3 octet), một giá trị salt (32 octet) và một giá trị trailer (rỗng) thành một giá trị đặc trưng (768 bit).

F = 422928E0 5A653BFA FE5F31C7 A92587DA 9827DDE7 17B787B2 3F4E7003 11BBBD9E
95C2F96E 1F974486 E20190F7 E752787F 7F6AF5C0 2571953F 68067250 57EBB19F
85DEF8CB 486F05F1 624AFFB3 F2E2131F E2405AAC 3C39A026 C08C6FA2 9EAD2C9D

S = 0B1747B4 75AEE7BC 9889BD75 6393B4699 81DD1592 4BD40195 067065C8 C3CB965F
1D167D2E 35372AE3 093F43EE 277895F8 47412B11 8E52C080 BB702E12 F20C9943
3C1E6E66 6A2C1C28 FEA43AD7 8E5B2723 808CB42F ED7ED057 64784324 5B01FAF7
02B7D4BD B8C6B73A 9E3DA80B EE3D79AF 0F86E200 D51E0F38 FAF3C1C3 1737D1E8
19085DF4 2A5F381E 7AFE9A16 E05F6BE2 160E4468 40F50418 0BE594B0 F9CB612F
78A86F06 31998E62 BFBFE6FE CBC2BCC5 5EACEE5E 3E677C37 D124AB42 1C10A841
F1EABA17 765D98AD 2B6CF806 1FE4945B AB694745 D6144BBB 38503CED 4FF1391E
22146E04 02B8DFB1 B0129CC8 35C4C5E2 BF908710 0AE77BA6 CE5CC469 8850FD02
8172AA31 062506D3 6C73C7A0 C39E5DD6 2B8D1C8F AC250E35 D9E78174 3268646B

Kiểm tra – $G^* = S^v mod n$ (biểu diễn bằng một xâu gồm 2304 bit, cũng ký hiệu là G^*). 768 bit trái nhất của G^* tạo ra F^*.

G^* = 422928E0 5A653BFA FE5F31C7 A92587DA 9827DDE7 17B787B2 3F4E7003 11BBBD9E
95C2F96E 1F974486 E20190F7 E752787F 7F6AF5C0 2571953F 68067250 57EBB19F
85DEF8CB 486F05F1 624AFFB3 F2E2131F E2405AAC 3C39A026 C08C6FA2 9EAD2C9D
2BD376CC A175B6B8 8ADBC508 5FD8FAAB F2F3953C D580826E 9A056E48 128D50AA
BF05B2DD DB018D42 A022F63E 77F6AE55 BF3078A3 887E5F7F 02676E9E 19AFCF57
4A3D53E1 95A31934 957E8B25 D81C004D D1E62BA5 F1D2795C 48BA90FD E3931D26
07F1B174 4FF7378A A4EF9E78 25D6A283 7E1632BE B9C15FB7 8A6686EB 9F988F42
6AF04BE6 E9304F2E 966E65A9 51C73506 E776834B F7915DD3 F65677ED 6451336A
3F6ECBAA A59C5997 13AF06FF F04EB5F7 F51D2D87 FB3413E9 A45DCB33 74862C4B

F^* = 422928E0 5A653BFA FE5F31C7 A92587DA 9827DDE7 17B787B2 3F4E7003 11BBBD9E
95C2F96E 1F974486 E20190F7 E752787F 7F6AF5C0 2571953F 68067250 57EBB19F
85DEF8CB 486F05F1 624AFFB3 F2E2131F E2405AAC 3C39A026 C08C6FA2 9EAD2C9D

Để khôi phục lại xâu trung gian, một giá trị mặt nạ gồm 512 bit được xây dựng từ 32 octet phải nhất của F^*, tức là HH^* XOR với 512 (= 768 – 256) bit trái nhất của F^*. Giá trị salt được khôi phục lại là một xâu gồm 32 octet.

E^* = 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000001
3438C82B A8799E1F 1301BA2A 14BF2133 CFFF625B BD819493 9BC4107C F7384B9D
3438C82B A8799E1F 1301BA2A 14BF2133 CFFF625B BD819493 9BC4107C F7384B9D

Thông điệp được băm thành một xâu gồm 256 bit, ký hiệu là H.

H = BA7816BF 8F01CFEA 414140DE 5DAE2223 B00361A3 96177A9C B410FF61 F20015AD

Một xâu gồm 576 bit (64 bit toàn 0 được nối với 256 bit của H và 256 bit của E^*) được băm thành HH.

HH = 85DEF8CB 486F05F1 624AFFB3 F2E2131F E2405AAC 3C39A026 C08C6FA2 9EAD2C9D

Phụ lục D
(Tham khảo)
Hai cơ chế định dạng khác nhau cho các lược đồ RSA/RW

D.1 Giới thiệu chung

Ngoài cơ chế quy định trong 6.4, hai cơ chế định dạng khác nhau thường được sử dụng với các lược đồ RSA và RW (xem bảng A.1). Cơ chế định dạng được quy định như sau:

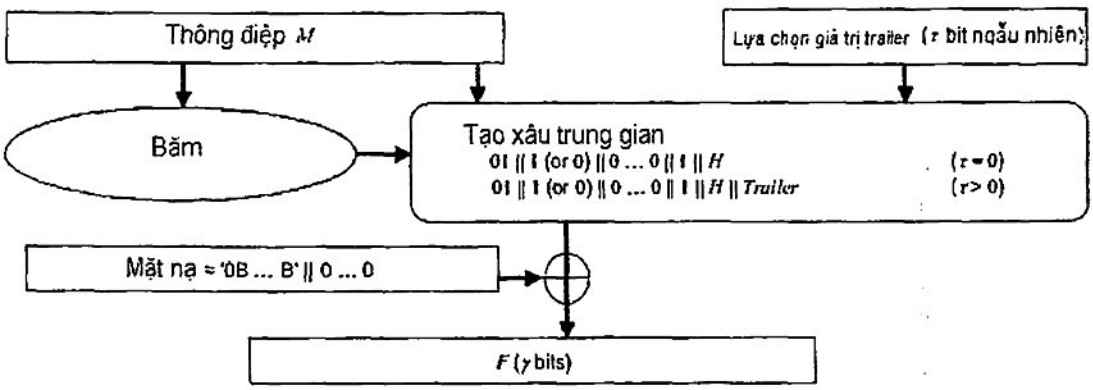
- Cơ chế D1 được trích dẫn từ ISO/IEC 9796-2:2002 [28].
- Cơ chế D2 được sử dụng rộng rãi trong các thẻ mạch tích hợp (xem ISO/IEC 7816 [24]).

Vì các cơ chế D1 và D2 bị ảnh hưởng bởi các va chạm trên hàm băm [21, 22], nên có thể bị loại bỏ khỏi bản sửa tiếp theo của tiêu chuẩn này.

Do không có tấn công nào được ghi nhận lên PSS, nên tiêu chuẩn này khuyến nghị sử dụng các biến thể của PSS (xem 6.4, 7.4 và 11.4).

D.2 Cơ chế định dạng D1

Quá trình chuyển đổi thông điệp M sử dụng một tham số (τ biểu thị độ dài bit của giá trị trailer) thành một giá trị đặc trưng gồm γ bit, ký hiệu là F. Hình D.1 minh họa cơ chế trên.



Hình D.1 – Cơ chế định dạng D1

1) Các giá trị tùy chọn như sau.

- Tùy chọn $\tau = 8$. Giá trị trailer là một octet đơn lẻ, đặt bằng "BC".
- Tùy chọn $\tau = 16$. Giá trị trailer là hai octet liên tiếp: octet phải nhất đặt bằng "CC", octet trái nhất xác định hàm băm sử dụng. Octet trái nhất được biểu diễn như sau.

- Khoảng từ "00" đến "7F" dành cho ISO/IEC JTC 1 SC27; ISO/IEC 10118 quy định một định danh duy nhất trong phạm vi đó đối với mỗi hàm băm tiêu chuẩn, ví dụ: "31" chỉ hàm đầu tiên trong phần 3, có tên gọi là RIPEMD-160 và "33" chỉ hàm thứ ba trong phần 3, có tên gọi là SHA-1.
- Khoảng từ "80" đến "FF" dành cho trường hợp đặc biệt.

$$Trailer = Định\ danh\ hàm\ băm\ ||"CC"$$

CHÚ THÍCH Một số nghiên cứu [12] đặt câu hỏi về ưu điểm của việc sử dụng định danh như trên trong giá trị trailer.

2) Băm M thành một xâu bit, ký hiệu là H.

$$H = h(M)$$

3) Tạo một xâu trung gian gồm γ bit được nối từ trái sang phải:

- Hai bit, đặt bằng 01;
- Một bit, đặt bằng 0 nếu M rỗng, bằng 1 trong các trường hợp còn lại;
- $\gamma - \tau - |H| - 4$ bit 0;
- H;
- Giá trị trailer (τ bit).

4) Tạo ra F bằng cách xử lý xâu trung gian trong các khối liên tiếp gồm bốn bit từ trái sang phải.

- Giữ lại khối trái nhất không bị thay đổi trong đó bit thứ tư trái nhất bắt đầu đệm.
    - Nếu đặt bằng 1, thì không đệm: giữ lại tất cả các bit liên tiếp không bị thay đổi.
    - Nếu đặt bằng 0, thì
        - Thay thế các khối liên tiếp có giá trị "0" bằng một khối "B", tức là 1011;
        - XOR khối tiếp theo đầu tiên không bằng "0" với "B", tức là 1011;
        - Giữ lại tất cả các bit tiếp theo không thay đổi.

CHÚ THÍCH Nếu $|n|$ và $|H|$ chia hết cho 4, thì giá trị đặc trưng là:

$$F = 6\text{B BB} \ldots \text{BB BA}||H||Trailer$$

5) Trả về F.

Kiểm tra một giá trị đặc trưng được khôi phục lại gồm γ bit, ký hiệu là F^*, đối với thông điệp M và giá trị tùy chọn τ sử dụng (trong khóa kiểm tra hoặc các tham số miền, hoặc là giá trị mặc định).

1) Kiểm tra giá trị trailer như sau.

- Nếu octet phải nhất của F^* bằng "BC", thì giá trị tùy chọn được khôi phục lại là $\tau^* = 8$.

- Nếu octect phải nhất của F^* bằng "CC" và octet bên trái của "CC" xác định hàm băm đang sử dụng, thì giá trị tùy chọn được khôi phục lại là $\tau^* = 16$.

- Loại bỏ trong các trường hợp còn lại (không thể biểu diễn giá trị trailer) và khi τ^* khác với τ.

2) Giữ lại bốn bit trái nhất không thay đổi.

- Loại bỏ nếu bit đầu tiên trái nhất bằng 1 hoặc bit tiếp theo bằng 0. Hai bit trái nhất bằng 01.

- Loại bỏ nếu bit thứ ba trái nhất bằng 1 nếu M rỗng hoặc bằng 0 nếu M không rỗng.

- Loại bỏ nếu bit thứ tư trái nhất bằng 1. Không có bit giới hạn.

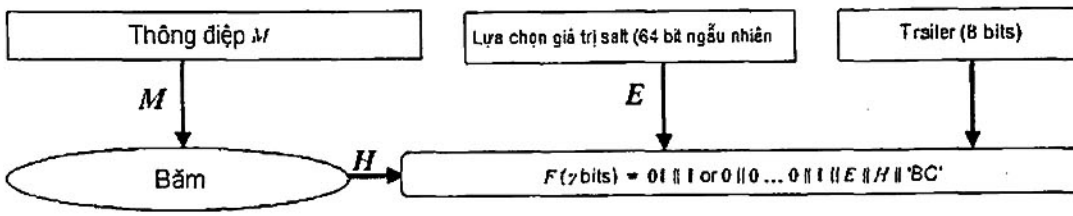
    - Mỗi khối liên tiếp gồm 4 bit có giá trị bằng 1011 (tức là "B") được đệm.
    - Khối tiếp theo đầu tiên không bằng "B" được XOR với "B" để khôi phục lại giá trị ban đầu của nó trong đó bit giới hạn là bit đầu tiên bằng 1 tính từ trái sang.
        - Nếu còn lại $|H|$ bit giữa bit giới hạn và giá trị trailer, thì tạo ra một xâu bit, ký hiệu là H^*.
        - Ngược lại, loại bỏ.

3) Băm M thành một xâu bit, ký hiệu là H.

4) Chấp nhận hoặc loại bỏ phụ thuộc vào H và H^* giống nhau hay khác nhau.

D.3 Cơ chế định dạng D2

Quá trình chuyển đổi thông điệp M thành một giá trị đặc trưng gồm γ bit, ký hiệu là F. Hình D.2 minh họa cơ chế này.



Hình D.2 – Cơ chế định dạng D2

1) Lựa chọn một xâu 64 bit ngẫu nhiên mới. Đó là giá trị salt, ký hiệu là E.
2) Băm M thành một xâu bit, ký hiệu là H.

$$H = h(M)$$

3) Tạo ra F bằng cách nối từ trái sang phải như sau:
- Hai bit bằng 01,
- Một bit, bằng 0 nếu M rỗng và bằng 1 trong trường hợp ngược lại,
- $\gamma - |H| - 76$ bit đệm bằng 0,
- Một bit giới hạn bằng 1,
- Giá trị salt E (64 bit),
- H, và
- Một giá trị trailer (một octet bằng "BC").

4) Trả về F.

Kiểm tra giá trị đặc trưng đã được khôi phục lại gồm γ bit, ký hiệu là F^* đối với thông điệp M.

1) Loại bỏ nếu bit trái nhất bằng 1 hoặc bit tiếp theo bằng 0. Hai bit trái nhất bằng 01.
2) Loại bỏ nếu bit tiếp theo bằng 1 nếu M rỗng hoặc bằng 0 nếu M không rỗng.
3) Loại bỏ nếu bit bất kỳ trong $|n| - |H| - 76$ bit tiếp theo (các bit đệm) bằng 1.
4) Loại bỏ nếu bit tiếp theo (bit giới hạn) bằng 0.
5) 64 bit tiếp theo (giá trị salt) tạo ra một xâu bit có giá trị bất kỳ.
6) $|H|$ bit tiếp theo (mã băm được khôi phục lại) tạo ra một xâu bit, ký hiệu là H^*.
7) Loại bỏ nếu 8 bit tiếp theo (giá trị trailer) không bằng "BC".
8) Băm M thành một mã băm, ký hiệu là H.
9) Chấp nhận hoặc loại bỏ phục thuộc vào H và H^* giống nhau hoặc khác nhau.

Phụ lục E
(Tham khảo)
Cho phép khôi phục lại thông điệp đối với các cơ chế kiểm tra RSA/RW

E.1 Cơ chế kiểm tra RSA/RW

Một số sản phẩm xử lý các thông điệp được ký tuân thủ tiêu chuẩn ISO/IEC 9796-2:2002 [28] cùng với các thông điệp được ký theo cơ chế ký quy định trong 6.2.

CHÚ THÍCH ISO/IEC 9796-2 [28] quy định các lược đồ chữ ký cho phép khôi phục lại thông điệp. Do đó, thông điệp M được tách thành một phần có thể khôi phục, ký hiệu là M_1 và một phần không thể khôi phục, ký hiệu là M_2.

- Có thể khôi phục hoàn toàn nếu M là phần duy nhất có thể khôi phục lại: $M_1 = M$ và M_2 rỗng.

- Chỉ khôi phục được một phần nếu M là kết quả của phép nối giữa phần có thể khôi phục lại và phần không thể khôi phục lại: $M = M_1||M_2$ (Do đó $|M_1| > 0$ lớn nhất có thể còn $|M_2| > 0$ là bội số của 8).

Kiểm tra giá trị đặc trưng được khôi phục lại gồm γ bit, ký hiệu là F^* đối với phần không thể khôi phục lại của thông điệp, ký hiệu là M_2 và một cơ chế định dạng đang sử dụng (PSS hoặc D1), với các giá trị tùy chọn đã dùng (cơ chế và giá trị tùy chọn có trong khóa kiểm tra hoặc các tham số miền hoặc là giá trị mặc định).

Nếu cơ chế định dạng sử dụng là PSS được quy định trong 6.4 hoặc trong ISO/IEC 9796-2:2002 [28], thì xử lý F^* với các giá trị tùy chọn ε và τ như sau.

1) Kiểm tra giá trị trailer như sau.

- Nếu octet phải nhất của của F^* bằng "BC", thì giá trị tùy chọn được khôi phục lại là $\tau^* = 8$.

- Nếu octect phải nhất của F^* bằng "CC" và octet bên trái của "CC" xác định hàm băm đang sử dụng, thì giá trị tùy chọn được khôi phục lại là $\tau^* = 16$.

- Loại bỏ trong các trường hợp còn lại (không thể biểu diễn giá trị trailer) và khi τ^* khác với τ.

2) Tách F^* thành một xâu được tạo mặt nạ gồm $\gamma - \tau - |H|$ bit và một xâu gồm $|H|$ bit, ký hiệu là HH^* tính từ trái sang phải (sau đó là giá trị trailer τ bit).
3) Tạo ra một giá trị mặt nạ $\gamma - \tau - |H|$ bit từ HH^* theo bước 3 của 6.4.
4) Thực hiện XOR giá trị mặt nạ với xâu được tạo mặt nạ để tạo ra một xâu trung gian đã được khôi phục lại trong đó bit giới hạn là bit đầu tiên bằng 1 tính từ trái sang.

- Nếu còn lại ε bit bên phải của bit giới hạn trong xâu trung gian đã được khôi phục lại, thì nó tạo ra một xâu bit, ký hiệu là E^* (M_1^* rỗng).

- Nếu số bit còn lại ít hơn, thì loại.

- Nếu số bit còn lại nhiều hơn, thì tạo ra một xâu bit, ký hiệu là M_1^*, sau đó là một xâu gồm ε bit, ký hiệu là E^*. Trường hợp $|M_1^*| > 0$ và $|M_2| > 0$, nếu bit giới hạn là một trong chín bit trái nhất của F^* và nếu $|M_2|$ là bội số của 8, thì tiếp tục; Ngược lại thì loại.

5) Băm M_2 thành một xâu bit, ký hiệu là H. Từ trái sang phải, nối 8 octet biểu diễn $|M_1^*|, M_1^*, H$ và E^*. Băm xâu vừa nối thành một xâu bit, ký hiệu là HH.

$$H = h(M_2)\ HH = h(|M_1^*|(8\ octets) \parallel M_1^* \parallel H \parallel E^*)$$

6) Chấp nhận hoặc loại bỏ phụ thuộc vào HH và HH^* giống nhau hoặc khác nhau. Ngoài ra, trong trường hợp chấp nhận, nếu $|M_1^*| > 0$, trả về M_1^* chính là phần đã được khôi phục lại.

Nếu cơ chế định dạng sử dụng là PSS được quy định trong 6.4 hoặc trong ISO/IEC 9796-2:2002 [28], thì xử lý F^* với giá trị tùy chọn τ như sau.

1) Kiểm tra giá trị trailer như sau.

- Nếu octet phải nhất của của F^* bằng "BC", thì giá trị tùy chọn được khôi phục lại là $\tau^* = 8$.

- Nếu octect phải nhất của F^* bằng "CC" và octet bên trái của "CC" xác định hàm băm đang sử dụng, thì giá trị tùy chọn được khôi phục lại là $\tau^* = 16$.

- Loại bỏ trong các trường hợp còn lại (không thể biểu diễn giá trị trailer) và khi τ^* khác với τ.

2) Giữ lại 4 bit trái nhất không thay đổi.

- Loại bỏ nếu bit trái nhất bằng 1 hoặc bit tiếp theo bằng 1. Hai bit trái nhất bằng 01.

- Loại bỏ nếu bit thứ ba trái nhất bằng 1 nếu M_2 rỗng hoặc bằng 0 nếu M_2 không rỗng.

- Bit thứ tư trái nhất bắt đầu loại bỏ giá trị đệm.

- Nếu bằng 1, đó là bit giới hạn. $\gamma - \tau - 4$ bit tiếp theo tạo ra một xâu gồm $\gamma - \tau - |H| - 4$ bit, ký hiệu là M_1^*, tiếp theo là một xâu gồm $|H|$ bit, ký hiệu là H^*.
- Nếu bằng 0, thì mỗi khối tiếp theo gồm 4 bit có giá trị bằng 1011 (tức là "B") được đệm. Khối thứ nhất tiếp theo không có giá trị bằng "B" sẽ được XOR với "B" để khôi phục lại giá trị ban đầu của nó trong đó bit giới hạn là bit đầu tiên bằng 1 tính từ trái sang.
  - Nếu còn lại $|H|$ bit giữa bit giới hạn và giá trị trailer, thì được ký hiệu là H^* (M_1^* rỗng).
  - Nếu số bit còn lại ít hơn, thì loại bỏ.
  - Nếu số bit còn lại nhiều hơn, tạo ra phần được khôi phục lại sau đó là một xâu gồm $|H|$ bit, tức là $M_1^*||H^*$. Trường hợp $|M_1^*| > 0$ và $|M_2| > 0$, nếu $|M_2|$ là bội của 8 và nếu bit giới hạn là một trong 11 bit trái nhất, thì tiếp tục; ngược lại thì loại.

3) Băm xâu $M_1^*||M_2$ thành một xâu bit $h(M_1^*||M_2)$, ký hiệu là H.

4) Chấp nhận hoặc loại bỏ phụ thuộc vào H và H^* giống nhau hoặc khác nhau. Ngoài ra, trong trường hợp chấp nhận, nếu $|M_1^*| > 0$, trả về phần đã được khôi phục lại M_1^*.

Phụ lục F
(Tham khảo)
Cho phép xác thực hai lần đối với các lược đồ GQ/GPS

F.1 Giới thiệu chung

Một số sản phẩm có một vài thay đổi nhỏ trong các cơ chế ký/kiểm tra GQ/QPS để thực hiện xác thực hai lần bằng một giá trị thách đố (một xâu bit, ký hiệu là C) do bên kiểm trả gửi, sau đó là giá trị hồi đáp (hai xâu bit, ký hiệu là R và S) do bên ký gửi.

Trong các lược đồ GQ1, GQ2, GPS1 và GPS2, độ dài của $|R|$ được quy định trong bảng B.1, giá trị hồi đáp là một chữ ký số tin cậy. Tuy nhiên, nếu $|R|$ nhỏ hơn hoặc bằng 48, thì trong trường hợp không bị trễ khi nhận được giá trị hồi đáp, thì bất cứ ai biết được các tham số miền và khóa kiểm tra của người ký đều có thể trả lời thách đố. Do đó, với $|R| \leq 48$, quá trình hồi đáp cho thấy rằng không thể tin được ai ngoại trừ bên kiểm tra, bên có thể kiểm tra được độ trễ khi nhận hồi đáp trong trường hợp sử dụng lược đồ trên.

- Mỗi lần xác thực, bên kiểm tra có phương pháp để lựa chọn các bit ngẫu nhiên nhằm tạo ra một giá trị thách đố mới, ký hiệu là C. Độ dài của giá trị thách đố là $|H|$ bit, do đó ưu thế thu được khi tính toán giá trị hồi đáp trước là không đáng kể.
- Trong các lược đồ GQ1 và GQ2, thay thế W bằng $C||W$.
- Trong các lược đồ GPS1 và GPS2, thay thế $h(W)$ bằng $C||h(W)$ (tức là, thay thế T bằng $C||T$).
- Mỗi lần xác thực, bên kiểm tra truyền các giá trị tạm thời của các tham số cố định để rút gọn $|R|$ xuống 48 bit hoặc ít hơn. Theo định nghĩa, một giá trị tạm thời thay thế giá trị cố định đối với lần xác thực hiện thời. Khi không có giá trị tạm thời cho một tham số cho trước, thì vẫn sử dụng giá trị cố định.
- Bên kiểm tra sẽ lựa chọn độ trễ cho quá trình hồi đáp theo từng lược đồ cụ thể được sử dụng.
- Về bên ký, các điều kiện an toàn cho quá trình hồi đáp phụ thuộc vào $|R|$. Tuy nhiên, các điều kiện an toàn cho quá trình tạo chữ ký chống lại xác thực động nằm ngoài phạm vi của tài liệu này.

CHÚ THÍCH: ISO/IEC 9798-3 [29] đặc tả các cơ chế xác thực sử dụng các kỹ thuật ký số: tạo ra chữ ký số, tức là chứng nhận được chuyển giao đến một bên thứ ba, ví dụ: bên phân xử. ISO/IEC 9798-5 [30] quy định các cơ chế xác thực sử dụng kỹ thuật tri thức không: Khi vượt qua cả ba, nó cung cấp chứng nhận không cần chuyển giao.

F.2 GQ1

Khi sử dụng cơ chế ký để xác thực, bên ký nhận được một giá trị thách đố (một xâu gồm $|H|$ bit, ký hiệu là C) và các giá trị tạm thời, ký hiệu là t' và k', sao cho $0 < t' \leq t, 0 < k' < |v|$ và $t' \times k' \leq 48$.

Trong bước 1 và 2, giá trị t' bị thay thế bởi t. Trong bước 3, xâu bit $C||W$ (gồm $|H| + t' \times |n|$ bit) bị thay thế bởi W và phần thứ nhất của giá trị hồi đáp, ký hiệu là R, là $t' \times k'$ bit trái nhất của H. Trong bước 4, giá trị t' và k' thay thế bởi t và $|v| - 1$.

Để sử dụng cơ chế kiểm tra trong xác thực, bên kiểm tra sẽ nhận được ột giá trị thách đố (một xâu bit ký hiệu là C) và các giá trị tạm thời, ký hiệu là t' và k', sao cho $0 < t' \leq t, 0 < k' < |v|$.

Trong bước 0 và 2, giá trị t' và k' thay thế bởi t và $|v| - 1$. Trong bước 3, xâu bit $C||W^*$(gồm $|H| + t' \times |n|$ bit) bị thay thế bởi W^* và phần thứ nhất của giá trị hồi đáp, ký hiệu là R^*, là $t' \times k'$ bit trái nhất của H^*.

Ví dụ về xác thực với $v = 2^{16} + 1$

Thành phần dữ liệu để ký/kiểm tra – Các thừa số nguyên tố và số mô-đun giống như các số sử dụng trong C.3.1. Số mũ kiểm tra là $v = 2^{16} + 1$ (là một số nguyên tố chia hết $p_1 - 1$ hoặc $p_2 - 1$).

$v =$ 10001

$u =$ 18384CCC 9C9A4CE6 61B06616 EF1A5CD4 436C9AD2 7A081D14 8E7ACD55 ED240B1D
AFCD2E8E 4676EA1B 259F02C3 79831FD7 F87BEB20 79EA1DF9 283BEEB5 83CBFA4B
5CAEF744 597550EB F85AE3D0 4CFC6F9F 26E035F0 E317D21F F3A241C7 92132BEC
633560E2 C9B5A3E5 88104BE0 61535C3E E4EC7220 838B3E01 53277B9F C5EA5137

Chuỗi dữ liệu định dang là một xâu bit biểu diễn "Alex Ample".

$$Id = 416C\ 6578\ 2041\ 6D70\ 6C65$$

Với SHA-1 và PSS, chuyển đổi dữ liệu đạnh thanh thành một giá trị đặc trưng (1024 bit).

$G =$ 3E641A22 D0D0747D 4ACC7188 4D3DFF2B 2ADFDC17 03B5A74E FD8333AB 8C4377BB
2A9B48E7 07F73409 ABFBCD2D ED69F52B 16A145CE 062FE6BD 712C1952 110DFB23
16C5F3F3 21922ED3 75A4DEB8 C41FA79B CAD86B0E A0D8FF02 C9D0D591 1BFF1E87
DBCF073F 71F18C08 EB944AE8 4883A1E1 3FB1DEA1 23B5B1EF EA2A9263 5BD5D88F

$Q = G^u mod\ n$

$Q =$ 24B9559A 80BD4D89 B9802A14 36DA3BDF 8DDF8DC3 993DEB1F A7EE0B4D B9F2EFFC
3003722C 9217CE8F BFEB962A 39B32DED F02C25CF 02702195 7A103024 15A7D59A
133A2B06 840B1DCA 10445287 FF875EAD D7EAFC8B 12B7C7E3 E05375C5 4D2369B7
9DFCEC0F 9235ADB3 16427D66 70D9422D 39C4F32B E1A406B5 E26736E1 F68E3682

Chữ ký – $W = r^v mod\ n$

$r =$ 487CDB00 41BEED03 23FDD3DE C8542584 FA0E6CB9 90FAD587 8DB34E9B EDDC95B6
5D22790C 108E2184 07ED7F7D 686657BA B5A28EF8 1C2E2498 5B56E37D 9934E195
A38A835C C02CEE8E BA2F56C8 7663E332 976F5A37 20DACA12 0BCD3DF0 AEF6FD78
582EBFCE E6D05E06 172A871E AB0E8F5F C22DDB60 0F541B87 CF8E1473 58374406

$W =$ 411F7E73 D995AC63 BACAE1F2 F1BF8D03 4886E36C 5825BC31 BDB761E8 567B6762
9947B41C 56A2EC07 8D02B880 76451F4F 991892D2 2F291949 F6F462B5 9098D627
F473111C FD260FFD 4428DD0C 3D270B82 F09E51C3 CF9065BD 744F708C 5D5C08B8
39336472 208415CC 72EBF75D 5A339134 C21E68AD 7AE057AB 8B25B776 CFCE18D1

Thông điệp là một xâu gồm 57 ký tự ASCII, tức là 448 bit. Giá trị thách đố C là một xâu gồm 20 octet.

M = abcdbcdecdefdefgefghfghighijhijkijkljklmklmnlmnomnopnopqo

= 61626364 62636465 63646566 64656667 65666768 66676869 6768696A 68696A6B
696A6B6C 6A6B6C6D 6B6C6D6E 6C6D6E6F 6D6E6F70 6E6F7071 6F

C = E3B5D5D0 02C1BCE5 0C2B65EF 88A183D8 3BCE7E61

Với SHA-1 được sử dụng theo biến thể băm đầu tiên, tính toán $H = h(C||W||M)$.

H = CCD650CD 522F5F45 9EB5F5FA 07E60319 4BFB1B0D

Vì $v = 2^{16} + 1$, phần đầu tiên của giá trị hồi đáp là một xâu gồm 16 bit (một chữ ký yêu cầu $t = 5$).

R = CCD6

S = 2A2D2BDE 478C5B00 83E8299C F22FA88D 0D7EC1BD E027E36C A8EA37E3 14A02E04
AE9DA62F 35CF7525 B5FE32F1 62142AD7 5A65BB8B 15632028 53A19F73 7B3A488F
8A85A094 25CDFF26 D3F19612 40E6035B 4B037D8F B95D8A50 9DF0AA5C 88D0A8FC
3FC15A6D D2B51F2C F32A5F9A 3B89EC92 22D99241 7053C4F1 6DF2A57A D4CE06AF

Kiểm tra – Chuỗi dữ liệu định danh là một xâu bit biểu diễn "Alex Ample".

Id = 416C 6578 2041 6D70 6C65

Với SHA-1 và PSS, chuyển đổi dữ liệu định danh thành một giá trị đặc trưng (1024 bit).

G = 3E641A22 D0D0747D 4ACC7188 4D3DFF2B 2ADFDC17 03B5A74E FD8333AB 8C4377BB
2A9B48E7 07F73409 ABFBCD2D ED69F52B 16A145CE 062FE6BD 712C1952 110DFB23
16C5F3F3 21922ED3 75A4DEB8 C41FA79B CAD86B0E A0D8FF02 C9D0D591 1BFF1E87
DBCF073F 71F18C08 EB944AE8 4883A1E1 3FB1DEA1 23B5B1EF EA2A9263 5BD5D88F

$W^* = S^v \times G^R mod n$

W^* = 411F7E73 D995AC63 BACAE1F2 F1BF8D03 4886E36C 5825BC31 BDB761E8 567B6762
9947B41C 56A2EC07 8D02B880 76451F4F 991892D2 2F291949 F6F462B5 9098D627
F473111C FD260FFD 4428DD0C 3D270B82 F09E51C3 CF9065BD 744F708C 5D5C08B8
39336472 208415CC 72EBF75D 5A339134 C21E68AD 7AE057AB 8B25D776 CFCE18D1

H^* = CCD650CD 522F5F45 9EB5F5FA 07E60319 4BFB1B0D

R^* = CCD6

F.3 GQ2

CHÚ THÍCH Khi bắt đầu một phiên (ví dụ: kết nối tới một mạng nội bộ, xem trường hợp 2 trong lưu ý ở phần đầu của 7.1), người dùng cung cấp định danh của mình để có được khóa ký bí mật gắn với định danh. Nếu người dùng sử dụng một số mô-đun GQ2 mới cho mỗi phiên liên lạc, thì độ lớn của số mô-đun GQ2 sẽ nhỏ hơn độ lớn của số mô-đun (số mô-đun dài hạn) là cơ sở cho khóa ký bí mật của người dùng. Yêu cầu đặt ra là số mô-đun GQ2 (số mô-đun ngắn hạn) không thể bị phân tích thành thừa số nguyên tố với năng lực tính toán của các máy tính hiện thời trong khoảng thời gian của phiên liên lạc đó.

Sử dụng cơ chế ký để xác thực, bên ký nhận được một giá trị thách đố (một xâu gồm $|H|$ bit, ký hiệu là C) và các giá trị tạm thời, ký hiệu là m' và t', sao cho $0 < m' \leq m, 0 < t' \leq t$ và $k \times m' \times t' \leq 48$ (nhưng số k sẽ được giữ bí mật), Khi $m' < m$, m' số cơ sở và f thừa số nguyên tố sẽ thỏa mãn các ràng buộc được quy định trong 8.1.

Trong bước 1 và 2, giá trị t' được thay bằng t. Trong bước 3, xâu bit $C||W$ ($|H| + t' \times |n|$ bit) thay bằng W và phần đầu tiên của giá trị hồi đáp, ký hiệu là R sẽ là $k \times m' \times t'$ bit của H. Trong bước 4, các giá trị t' và m' được thay bằng t và m.

Sử dụng cơ chế kiểm tra để xác thực, bên kiểm tra sẽ được chuyển giao một giá trị thách đố (một xâu bit, ký hiệu là C) và các giá trị tạm thời, ký hiệu là m' và t', thỏa mãn $0 < m' \leq m$ và $0 < t' \leq t$.

Trong bước 0 và 1, các giá trị m' và t' được thay bằng m và t. Trong bước 2, xâu bit $C||W^*$ ($|H| + t' \times |n|$ bit) được thay bằng W^* và phần đầu tiên của giá trị hồi đáp, ký hiệu là R^*, sẽ là $k \times m' \times t'$ bit trái nhất của H^*.

Ví dụ về xác thực với $b = 1$ và $m = 1$

Thành phần dữ liệu để ký/kiểm tra – Các thừa số nguyên tố và số mô-đun giống như các số sử dụng trong C.4.2. Tham số thay thế là $b = 1$. Tham số an toàn là $k = 20$, với một số cơ sở ($m' = 1$), cụ thể là $g_1 = 2$ (các thừa số nguyên tố đều đồng dư với 3 mod 4, nhưng không đồng dư với nhau mod 8).

Chữ ký – $W_i = r_i^{2^{k+b}} \bmod p_i$

$r_1 =$ 958FE0FE 77561815 FCCE3499 D2AA78C6 701CB4DF 3EAEF982 160F9254 592C63ED
D4692A99 336020DA 4427AD2A 5845CFDD 0153CEB3 6507C76A 9473DAC1 A764E4C2

$r_2 =$ ED1F46C6 B0143F7F A70DC68C 0E8E4324 5F22CE6C BC811A7C E90D7B0C 0D828256
C479922A C1B1CD6E 52DD82F3 75B90D0C 9FA6FD45 34611F9C 2CE4EF1E DB7DB9B7

$W =$ 82074289 8F8E9537 437D57D4 17184A82 06FEB795 F9CA167D 60BB7314 EB8F1360
5882C202 467DD2C0 F7F8D14B 87A7FB41 15D68D1C D6313C14 CA24DD84 E4F293F6
30AF2A90 EB122FD1 E113C184 DCB976AC FBCD0CA4 35EF6CDD E5F66F4C 06947B36
5E1E3B03 3D766C5B 8619B164 6470A0FA 961008A7 90CAA733 8E3119B1 C10263B8

Thông điệp là một xâu gồm 57 ký tự ASCII, tức là 456 bit. Giá trị thách đố C là một xâu gồm 20 bit.

$M =$ abcdbcdecdefdefgefghfghighijhijkijkljklmklmnlmnomnopnopqo

$=$ 61626364 62636465 63646566 64656667 6566676B 66676869 6768696A 68696A6B
696A6B6C 6A6B6C6D 6B6C6D6E 6C6D6E6F 6D6E6F70 6E6F7071 6F

$C =$ E3B5D5D0 02C1BCE5 0C2B65EF 88A188D8 3BCE7E61

Với SHA-1 được sử dụng theo biến thể băm đầu tiên, tính toán $H = h(C||W||M)$.

$H =$ A0C43EEC A6494C3E DC275FC0 AF92ECA8 43EAAAA1

Phần đầu tiên của giá trị hồi đáp là một xâu gồm 20 bit (một chữ ký yêu cầu $t = 4$ hoặc $m = 4$).

$R =$ A0C43

$S_1 =$ 96F11063 9E192357 A1BD7629 859DF858 319CDEFB FA090047 E0702F80 026C5C3F
FD057EBD CC0DE71E B5ED9876 39BD62BC 9B7F4D1D 15E0A88A DC584E99 DB5D0A80

$S_2 =$ E94B3F98 48F80136 691DABFF C05C1412 D3ED8A69 50ACE88E 2949B59A 3F008C7D
34053431 C68DB88D 58056566 6687471E DD8B2C67 27BDC7E9 E4B79C87 F0283359

$S =$ B3066B24 1987017A D573EC3D 0FD90F19 F4AC8B88 FFEDE385 616345FC 312CFBF3
2F31E0C4 0EAA8420 F54495B4 27A29B42 07AE201B 670C9662 3FA1C0D5 E2CD1333

BC2D47EE E83EF91A D2DA3374 F237949A F81757D3 EDCFD5A4 41E5B287 E4C78A59
EBABAD3E 8A2EF108 E2279347 E67D5DB9 EFF09700 991E367C 737BC66D 07C16C55

Kiểm tra – $W^* = S^{2^{k+1}} \times (g^2)^R mod\ n$

W^* = 82074289 8E8E9537 437D57D1 17184A82 06FEB795 F9CA167D 60BB7314 EB8F1360
5882C202 467DD2C0 F7F8D14B 87A7FB41 15D68D1C D6313C14 CA24DD84 E4F293F6
30AF2A90 EB122FD1 E113C184 DCB976AC FBCD0CA4 35EF6CDD E5F66F4C 06947B36
5E1E3B03 3D766C5B 8619B164 6470A0FA 961008A7 90CAA733 8E3119B1 C10263B8

H^* = A0C43EEC A6494C3E DC275FC0 AF92ECA8 43EAAAA1

R^* = A0C43

F.4 GPS1

Sử dụng cơ chế ký để xác thực, bên ký nhận được một giá trị thách đố (một xâu gồm $|H|$ bit, ký hiệu là C) và một giá trị tạm thời, ký hiệu là k, do đó $0 < k < 48$.

Trong bước 3, xâu bit $C||T$ ($2|H|$ bit) được thay thế bằng T. Trong bước 3 và 4, k bit trái nhất của R được thay thế bằng R.

Sử dụng cơ chế kiểm tra để xác thực, bên kiểm tra sẽ được chuyển giao một giá trị thách đố (một xâu bit, ký hiệu là C) và một giá trị tạm thời, ký hiệu là k, do đó $0 < k \leq |H|$.

Trong bước 0, $|R| \neq k$ được thay thế bằng $|R| \neq |H|$. Trong bước 2, xâu bit $C||h(W^*)$ ($2|H|$ bit) được thay thế bằng $h(W^*)$. Trong bước 3 và 4, k bit trái nhất của R^* được thay thế bằng R^*.

F.5 GPS2

Sử dụng cơ chế ký để xác thực, bên ký nhận được một giá trị thách đố (một xâu gồm $|H|$ bit, ký hiệu là C) và một giá trị tạm thời, ký hiệu là k, do đó $0 < k < 48$.

Trong bước 3, xâu bit $C||T$ ($2|H|$ bit) được thay thế bằng T và phần đầu tiên của giá trị hồi đáp, ký hiệu là R, sẽ là k bit trái nhất của H.

Sử dụng cơ chế kiểm tra để xác thực, bên kiểm tra sẽ được chuyển giao một giá trị thách đố (một xâu bit, ký hiệu là C) và một giá trị tạm thời, ký hiệu là k, do đó $0 < k \leq |H|$.

Trong bước 0, $|R| \neq k$ được thay thế bằng $|R| \neq |H|$. Trong bước 2, xâu bit $C||h(W^*)$ ($2|H|$ bit) được thay thế bằng $h(W^*)$ và phần đầu tiên đã được khôi phục lại của giá trị hồi đáp, ký hiệu là R^* sẽ là k bit trái nhất của H^*.

Thư mục tài liệu tham khảo

[1] M. Bellare and P. Rogaway, *The exact security of digital signatures: How to sign with RSA and Rabin*, in Proc. Eurocrypt '96, U. Maurer, Ed., Lecture Notes in Computer Science, Vol. 1070, Advances in Cryptology, pp. 399-416, Berlin, Springer-Verlag, 1996

[2] J.-S. Coron, On the exact security of full domain hashing, in Proc. Crypto 2000, M. Bellare, Ed., Lecture Notes in Computer Science, Vol. 1880, Advances in Cryptology, pp. 229-235, Berlin, Springer-Verlag, 2000

[3] A. Fujioka, T. Okamoto and S. Miyaguchi, ESIGN, an efficient digital signature implementation for smart cards, in Proc. Eurocrypt '91, D.W. Davies, Ed., Lecture Notes in Computer Science, Vol. 547, Advances in Cryptology, pp. 446-457, Berlin, Springer-Verlag, 1992

[4] M. Gardner, A new kind of cipher that would take millions of years to break, Scientific American, Vol. 237-8, pp. 120-124, 1977

[5] M. Girault. Self-certified public keys, in Proc. Eurocrypt '91, D.W. Davies, Ed., Lecture Notes in Computer Science, Vol. 547, Advances in Cryptology, pp. 490-497, Berlin, Springer-Verlag, 1992

[6] M. Girault and J.C. Paillès. On-line / off-line RSA-like, Workshop on Cryptography and Coding 2003

[7] S. Goldwasser, S. Micali and C. Rackoff, The knowledge complexity ofinteractive proof systems, in SIAM Journal on Computing, Vol. 18, pp. 186-208, 1989

[8] S. Goldwasser, S. Micali and R.L. Rivest, A digital signature scheme secure against adaptive chosen-message attacks, in SIAM Journal on Computing, Vol. 17-2, pp. 491-531, April 1988

[9] L.C. Guillou and J.-J. Quisquater, A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory, in Proc. Eurocrypt '88, C.G. Günther, Ed., Lecture Notes in Computer Science, Vol. 330, Advances in Cryptology, pp. 123-128, Berlin, Springer-Verlag, 1988

[10] L.C. Guillou and J.-J. Quisquater, A paradoxical identity-based signature scheme resulting from zeroknowledge, in Proc. Crypto '88, Sh. Goldwasser, Ed., Lecture Notes in Computer Science, Vol. 403, Advances in Cryptology, pp. 216-231, Berlin, Springer-Verlag, 1988

[11] L.C. Guillou, M. Ugon and J.-J. Quisquater, Cryptographic authentication protocols for smart cards, in Computer Networks Magazine, Vol. 36, pp. 437-451, North Holland Elsevier Publishing, July 2001

[12] B. Kaliski, On hash function firewalls in signature schemes, in Proc. Cryptographers' Track RSA Conference 2002, B. Preneel, Ed, Lecture Notes in Computer Science, Vol. 2271, pp. 1-16, Berlin, Springer-Verlag, 2002

[13] D.E. Knuth, The Art of Computer Programming, Vol. 2. Addison-Wesley, 3rd edition, 1997

[14] A.K. Lenstra and E.R. Verheul, Selecting cryptographic key sizes, in Journal of Cryptology, Vol. 14-4, pp. 255-293, 2001

[15] A.J. Menezes, P.C. van Oorschot and S.A. Vanstone, Handbook of Applied Cryptography, Boca Raton, CRC Press, 1997

[16] A.M. Odlyzko, The future of integer factorization, in Cryptobytes, Vol. 1-2, pp. 5-12, Summer 1995

[17] G. Poupard and J. Stern, Security analysis of a practical "on the fly" authentication and signature generation, in Proc. Eurocrypt '98, K. Nyberg, Ed., Lecture Notes in Computer Science, Vol. 1403, Advances in Cryptology, pp. 422-436, Berlin, Springer-Verlag, 1998

[18] M.O. Rabin, Digital signatures and public-key functions as intractable as factorization, Technical Report MIT/LCS/TR-212, MIT Laboratory for Computer Science, January 1979

[19] R.L. Rivest, A. Shamir and L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, in Communications of the ACM, Vol. 21-2, pp. 120-126, 1978

[20] R. Silverman, A cost-based security analysis of symmetric and asymmetric key lengths, RSA Labs Bulletin, Vol. 13, April 2000 (revised November 2001)

[21] X. Wang and H. Yu, How to break MD5 and other hash-functions, in Proc. Eurocrypt '05, R. Cramer, Ed., Lecture Notes in Computer Science, Vol. 3494, Advances in Cryptology, pp. 19-35, Berlin, Springer-Verlag, 2005

[22] X. Wang, Y. Yin and H. Yu, Finding collisions in the full SHA-1, in Proc. Crypto '05, V. Shoup, Ed., Lecture Notes in Computer Science, Vol. 3621, Advances in Cryptology, pp. 17-36, Berlin, SpringerVerlag, 2005

[23] H.C. Williams, Some public-key crypto-functions as intractable as factorization, in Proc. Crypto '84, G.R. Blakley and D. Chaum, Eds., Lecture Notes in Computer Science, Vol. 196, Advances in Cryptology, pp. 66-70, Berlin, Springer-Verlag, 1985

[24] ISO/IEC 7816 (all parts), Identification cards — Integrated circuit cards

[25] ISO/IEC 8824-1:2002, Information technology — Abstract Syntax Notation One (ASN.1): Specification of basic notation

[26] ISO/IEC 8825-1:2002, Information technology — ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)

[27] ISO/IEC 9594-8:2005, Information technology — Open Systems Interconnection — The Directory: Public-key and attribute certificate frameworks

[28] ISO/IEC 9796 (all parts), Information technology — Security techniques — Digital signature schemes giving message recovery

[29] TCVN 11817-3 (ISO/IEC 9798-3), Công nghệ thông tin – Các kỹ thuật an toàn – Xác thực thực thể - Phần 3: Cơ chế sử dụng kỹ thuật chữ ký số.

[30] ISO/IEC 9798-5:2004, Information technology — Security techniques — Entity authentication — Part 5: Mechanisms using zero-knowledge techniques

[31] TCVN 7817 (ISO/IEC 11770) (tất cả các phần), Công nghệ thông tin – Các kỹ thuật an toàn – Quản lý khóa

[32] ISO/IEC 15945:2002, Information technology — Security techniques — Specification of TTP services to support the application of digital signatures

[33] ISO/IEC 18031:2005, Information technology — Security techniques — Random bit generation

[34] ISO/IEC 18032:2005, Information technology — Security techniques — Prime number generation

[35] TCVN 12214-3 (ISO/IEC 14888-3), Công nghệ thông tin – Các kỹ thuật an toàn – Chữ ký số kèm phụ lục – Phần 3: Cơ chế dựa trên logarit rời rạc