

TRƯỜNG ĐẠI HỌC SƯ PHẠM HÀ NỘI 2

KHOA CÔNG NGHỆ THÔNG TIN



NGUYỄN HƯƠNG TRÀ

**CHUẨN CHỮ KÝ SỐ DSA  
VÀ ỨNG DỤNG TRONG HÓA ĐƠN  
TIỀN ĐIỆN ĐIỆN TỬ QUẬN LONG BIÊN**

**KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC**

**Chuyên ngành: Khoa học máy tính**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM HÀ NỘI 2**

**KHOA CÔNG NGHỆ THÔNG TIN**



**NGUYỄN HƯƠNG TRÀ**

**CHUẨN CHỮ KÝ SỐ DSA  
VÀ ỨNG DỤNG TRONG HÓA ĐƠN  
TIỀN ĐIỆN ĐIỆN TỬ QUẬN LONG BIÊN**

**KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC**

**Chuyên ngành: Khoa học máy tính**

**Người hướng dẫn khoa học**

**TS. LƯU THỊ BÍCH HƯƠNG**

**HÀ NỘI – 2015**

## LỜI CẢM ƠN

Để hoàn thành được khóa luận này, trước hết em xin gửi lời cảm ơn sâu sắc nhất tới **TS. Lưu Thị Bích Hương** đã tận tình hướng dẫn, chỉ bảo, định hướng, đóng góp những ý kiến quý báu cho em trong suốt quá trình thực hiện.

Em xin chân thành cảm ơn các thầy, cô giáo trong khoa Công nghệ Thông tin, trường Đại học Sư phạm Hà Nội 2 đã quan tâm giảng dạy và giúp đỡ em trong suốt bốn năm học vừa qua cũng như trong thời gian em làm bài khóa luận này. Là sinh viên khoa Công nghệ Thông tin, em rất tự hào về khoa mình học, về thầy cô giáo của mình. Em xin kính chúc các thầy, các cô luôn mạnh khỏe, hạnh phúc và thành công. Chúc khoa Công nghệ Thông tin sẽ ngày một khang trang, vững mạnh, góp phần to lớn trong sự nghiệp đào tạo chuyên nghiệp của trường Đại học Sư phạm Hà Nội 2.

Lần đầu nghiên cứu khoa học, chắc chắn đề tài của em không tránh khỏi những thiếu sót, hạn chế. Vì vậy, em rất mong sự đóng góp ý kiến của các thầy cô giáo và các bạn để đề tài của em được hoàn thiện.

Cuối cùng, em xin cảm ơn tới gia đình, bạn bè của em, đã luôn luôn động viên, khích lệ tinh thần và tạo điều kiện tốt nhất cho em hoàn thành khóa luận này.

Hà Nội, tháng 05 năm 2015

Sinh viên

**Nguyễn Hương Trà**

## LỜI CAM ĐOAN

Tên em là: **Nguyễn Hương Trà**

Sinh viên: K37A – CNTT, trường Đại học Sư phạm Hà Nội 2.

Em xin cam đoan:

1. Đề tài “*Chuẩn ký số DSA và ứng dụng trong hóa đơn tiền điện tử quận Long Biên*” là kết quả tìm hiểu và nghiên cứu của riêng em, dưới sự hướng dẫn của TS. Lưu Thị Bích Hương.
2. Khóa luận hoàn toàn không sao chép từ các tài liệu có sẵn đã được công bố khác.
3. Kết quả không trùng với các tác giả khác.

Nếu sai em xin hoàn toàn chịu trách nhiệm.

Hà Nội, tháng 05 năm 2015

Người cam đoan

**Nguyễn Hương Trà**

## MỤC LỤC

MỞ ĐẦU .....	1
CHƯƠNG 1: CƠ SỞ LÝ THUYẾT .....	5
1.1. Chuẩn hàm băm an toàn.....	5
1.2. Các giải thuật hàm băm an toàn.....	6
1.2.1. SHA-1 .....	6
1.2.2. SHA-256 .....	8
1.2.3. SHA-512 .....	11
1.2.4. SHA-384 .....	14
1.3. Chữ ký số .....	14
1.3.1. Định nghĩa.....	14
1.3.2. Chức năng chữ ký số.....	15
1.3.3. Quá trình tạo chữ ký số.....	17
1.3.4. Quá trình thẩm định chữ ký số.....	18
CHƯƠNG 2: CHUẨN CHỮ KÝ SỐ DSA .....	19
2.1. Giới thiệu .....	19
2.2. Thuật toán chữ ký số DSA.....	21
2.2.1. Nguyên tắc hoạt động .....	21
2.2.2. Các tham số.....	22
2.2.3. Kích thước tham số và các hàm băm sử dụng .....	23
2.2.4. Các tham số miền.....	24
2.2.5. Cặp khoá .....	26
2.3. Tạo cặp khoá .....	27
CHƯƠNG 3: THIẾT KẾ VÀ CÀI ĐẶT ỨNG DỤNG.....	29
3.1. Phát biểu bài toán.....	29
3.2. Thiết kế bằng giải thuật DSA.....	32
3.2.1. Sơ đồ lớp của giải thuật băm .....	32
3.2.2. Sơ đồ tạo các tham số miền .....	34

3.2.3. Chức năng tạo khoá .....	43
3.2.4. Chức năng tạo và thẩm định chữ ký số.....	44
3.3. Thiết kế giao diện.....	45
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....	51
TÀI LIỆU THAM KHẢO.....	53

## MỤC LỤC HÌNH

Hình 1.1: Quá trình tạo chữ ký số.....	17
Hình 1.2: Quá trình thẩm định chữ ký số.....	18
Hình 2.1: Sơ đồ sử dụng giải thuật hàm băm trong giải thật chữ ký số .....	21
Hình 3.1: Quá trình mã hóa của SHA-1.....	34
Hình 3.2: Quá trình tạo chữ ký số và kiểm tra chữ ký số dùng DSA .....	45
Hình 3.3: Form giao diện chính .....	46
Hình 3.4: Form tạo hóa đơn .....	47
Hình 3.5: Form tạo khóa .....	48
Hình 3.6: Form giải mã thành công .....	48
Hình 3.7: Form tạo file ảnh JPG .....	49
Hình 3.8: Form kết quả nhận được .....	50

## MỞ ĐẦU

### 1. Lý do chọn đề tài

Trong sự phát triển của xã hội, kể từ khi có sự trao đổi thông tin, an toàn thông tin trở thành một nhu cầu gắn liền với cuộc sống con người. Đặc biệt trong thời đại mà thương mại điện tử đang phát triển thì việc có được các công cụ đầy đủ để đảm bảo cho sự an toàn trao đổi thông tin liên lạc là vô cùng cần thiết. Chữ ký số đã ra đời với nhiều tính năng ưu việt phục vụ cho việc đảm bảo an toàn thông tin. Nó chính là thông tin đi kèm theo dữ liệu nhằm mục đích xác định người chủ của dữ liệu đó. Trong kinh doanh, chữ ký số được hiểu như con dấu và chữ ký của một doanh nghiệp. Nó không những chỉ dùng trong việc kê khai thuế, mà người sử dụng có thể sử dụng trong tất cả các giao dịch điện tử với mọi tổ chức và cá nhân khác. Trong thời đại công nghệ thông tin hiện nay thì việc rút ngắn khoảng cách giữa không gian, thời gian luôn là một đòi hỏi cấp thiết trong công việc kinh doanh, chữ ký số ra đời đã giúp cho các doanh nghiệp tiết kiệm nhiều thời gian, công sức trong một số công việc giao dịch với ngân hàng, cơ quan hành chính...

Ngày nay, việc thanh toán bằng hóa đơn điện tử cho các giao dịch mua bán được xem là xu thế mới, là hướng đi phù hợp với thời đại công nghệ số và thương mại điện tử ngày càng phát triển trên thị trường nước ta và quốc tế. Hóa đơn điện tử là tập hợp các thông điệp dữ liệu điện tử về bán hàng, cung ứng dịch vụ; được khởi tạo, lập, gửi, nhận, lưu trữ và quản lý bằng phương tiện điện tử. Sử dụng hóa đơn điện tử thay thế cho hóa đơn giấy truyền thống sẽ góp phần hiện đại hóa công tác hạch toán kế toán, tiết kiệm chi phí, nâng cao tính bảo mật của doanh nghiệp. Đây cũng là biện pháp hữu hiệu hỗ trợ ngành thuế từng bước ngăn chặn và kiểm soát việc sử dụng hóa đơn giả để trốn thuế. Thời gian gần đây, các công ty điện lực nước ta cũng đang từng bước chuyển từ việc thanh toán bằng hóa đơn giấy sang việc thanh toán bằng hóa đơn điện tử. Việc sử dụng hóa đơn điện tử trong giao dịch mua bán điện cho phép người bán tiết kiệm được chi phí in ấn, thuận tiện trong bảo quản,



lưu trữ và hạch toán kế toán, đối chiếu dữ liệu, quản trị kinh doanh của doanh nghiệp, kê khai, nộp thuế, quá trình thanh toán nhanh hơn đồng thời góp phần bảo vệ môi trường. Khách hàng có thể cập nhật qua phương tiện điện tử thông tin thông báo cước và lựa chọn bất kỳ hình thức thanh toán tiền điện nào và sẽ nhận được biên nhận sau khi thanh toán.

Tuy nhiên việc thanh toán bằng hóa đơn điện tử ở nước ta mới chỉ được áp dụng thí điểm ở những thành phố lớn. Các tổng công ty điện lực đang tích cực chuẩn bị mọi điều kiện để triển khai toàn bộ hóa đơn điện tử trong kinh doanh điện trên cả nước từ năm 2015. Và để làm được điều đó, ngành điện lực đòi hỏi một hệ thống các phần mềm điện tử phục vụ cho quá trình thanh toán. Trong đó, việc ký số hóa đơn là một khâu quan trọng bởi nó chính là việc xác nhận khách hàng đã thanh toán hay chưa. Từ những yếu tố đó, em đã chọn đề tài “*Chuẩn ký số DSA và ứng dụng trong hóa đơn tiền điện điện tử quận Long Biên*” làm đề tài khóa luận tốt nghiệp của mình.

## **2. Mục đích nghiên cứu**

Nghiên cứu về các giải thuật băm được sử dụng trong việc mã hóa và bảo mật thông tin như: SHA-1, SHA- 256, SHA- 512, SHA- 384. Từ đó áp dụng vào việc tạo chữ ký số cho hóa đơn tiền điện.

## **3. Nhiệm vụ nghiên cứu**

- Tìm hiểu các giải thuật băm an toàn và chuẩn chữ ký số DSA.
- Xây dựng ứng dụng về chữ ký số trong hóa đơn điện tử tiền điện tại Điện lực Long Biên.

## **4. Đối tượng và phạm vi nghiên cứu**

Đối tượng nghiên cứu của khóa luận là các hàm băm an toàn, chuẩn chữ ký số DSA và việc ký số hóa đơn tiền điện điện tử.

Phạm vi nghiên cứu: Chuẩn ký số DSA và in hóa đơn dạng JPG của hóa đơn tiền điện quận Long Biên.

## **5. Ý nghĩa khoa học và thực tiễn**

Kết quả nghiên cứu của khóa luận có ý nghĩa trong quá trình phát triển hạ tầng công nghệ thông tin của nước ta. Góp phần đẩy mạnh ứng dụng công nghệ thông tin trong kinh doanh và dịch vụ khách hàng. Không những thế, việc áp dụng hoá đơn điện tử phù hợp với các công nghệ tiên tiến đang được áp dụng trên thế giới như Mobile Banking, Internet Banking, SMS Banking,...

Trước mắt, hóa đơn điện tử mang lại lợi ích cho cơ quan thuế, cơ quan thuế không kê khai hóa đơn bằng giấy như trước đây. Khi thực hiện áp dụng hóa đơn điện tử, việc kê khai được thực hiện qua mạng internet bằng các phần mềm xác thực tính đúng đắn của hóa đơn và đảm bảo được nhanh chóng và chính xác, giảm việc lưu trữ hóa đơn bằng giấy.

Về mặt xã hội, hóa đơn điện tử giúp giảm thanh toán tiền mặt, góp phần bảo vệ môi trường so với trước đây sử dụng hóa đơn giấy. Nó cũng giúp nâng cao năng lực cạnh tranh, hiện đại hoá quản trị doanh nghiệp, giúp doanh nghiệp kê khai thuế nhanh chóng, kết nối trực tiếp đến hệ thống kế toán.

## **6. Phương pháp nghiên cứu**

### *a- Phương pháp nghiên cứu lý luận*

Nghiên cứu qua việc đọc sách, báo và các tài liệu liên quan nhằm xây dựng cơ sở lý thuyết của khóa luận và các biện pháp cần thiết để giải quyết các vấn đề của khóa luận.

### *b- Phương pháp chuyên gia*

Tham khảo ý kiến của các chuyên gia để có thể thiết kế chương trình phù hợp với yêu cầu thực tiễn. Nội dung xử lý nhanh đáp ứng được yêu cầu ngày càng cao của người sử dụng.

### *c- Phương pháp thực nghiệm*

Thông qua quan sát thực tế, yêu cầu của cơ sở, những lý luận được nghiên cứu và kết quả đạt được qua những phương pháp trên.

## **7. Cấu trúc khóa luận**

Ngoài phần lời cảm ơn, mở đầu, kết luận và hướng phát triển, tài liệu tham khảo, khóa luận có những nội dung sau:

**Chương 1: Cơ sở lý thuyết** - Tập trung nghiên cứu khái quát lý thuyết cơ sở của hàm băm được sử dụng trong việc ký số. Trong đó sẽ có cái nhìn tổng quan về chữ ký số và các giải thuật được sử dụng. Sau đó, khóa luận sẽ nêu ra phương pháp tiếp cận và thực hiện.

**Chương 2: Chuẩn chữ ký số DSA** - Chương này nghiên cứu về cơ sở hạ tầng cơ bản để tạo chữ ký số. Trong đó nghiên cứu cụ thể về nguyên tắc hoạt động, cách tạo cặp khóa và tạo chữ ký số của chuẩn ký số DSA.

**Chương 3: Thiết kế và cài đặt ứng dụng** – Phát biểu chi tiết bài toán mà khóa luận đề ra từ đó đi đến thiết kế giải thuật DSA và xây dựng được ứng dụng chữ ký số cho hóa đơn tiền điện điện tử tại Điện lực Long Biên.

## CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

### 1.1. Chuẩn hàm băm an toàn

Chuẩn hàm băm an toàn SHS-FIPS PUB 180 (Secure Hash Standard) được NIST đưa ra lần đầu vào 11/5/1993. Đây là một giải thuật hàm băm có tên SHA-1. Và phiên bản thứ 2 của chuẩn hàm băm an toàn là FIPS PUB 180-2, được đưa ra vào ngày 1/8/2002.

Trong FIPS PUB 180-2 chỉ rõ 4 giải thuật hàm băm an toàn là SHA-1, SHA-256, SHA-384, và SHA-512. Cả 4 giải thuật này đều là các hàm băm một chiều có thể xử lý thông điệp để tạo ra một thông điệp thu gọn của thông điệp ban đầu. Các giải thuật này đều đảm bảo được tính toàn vẹn của thông điệp: bất cứ sự thay đổi nào với thông điệp  $M$  thì đều dẫn tới sự thay đổi của thông điệp rút gọn  $H(M)$ - với một xác suất là rất lớn. Đặc điểm này rất có ích trong việc tạo và xác nhận chữ ký số cũng như trong việc tạo ra các số ngẫu nhiên.

Mỗi một giải thuật đều gồm hai bước: bước tiền xử lý và bước tính toán băm. Bước tiền xử lý bao gồm các công việc như độn tin, chia khối, thiết lập giá trị khởi tạo dùng cho tính toán băm. Bước tính toán băm sẽ tạo ra một danh sách thông định sẵn từ bản tin độn và sử dụng cơ chế đó với các hàm, hằng và các phép toán để tạo ra một chuỗi các giá trị băm. Giá trị băm cuối cùng được tạo bởi bước tính toán băm sẽ chính là thông điệp rút gọn.

Bốn giải thuật này khác nhau chủ yếu ở số bit bảo mật. Số bit bảo mật có liên quan trực tiếp tới độ dài của thông điệp rút gọn. Khi một giải thuật hàm băm được sử dụng cùng với một giải thuật khác thì ta cần phải chỉ ra hàm băm cần sử dụng có số bit bảo mật là bao nhiêu. Ví dụ, nếu một thông điệp được ký bằng giải thuật chữ ký số với 128 bit bảo mật thì giải thuật chữ ký số đó có thể yêu cầu sử dụng một hàm băm an toàn có 128 bit bảo mật (ví dụ SHA-256).

Ngoài ra, các giải thuật khác nhau ở kích thước của khối và từ dữ liệu được dùng trong quá trình băm. Bảng dưới đây sẽ trình bày các đặc điểm cơ bản của các giải thuật hàm băm an toàn

**Bảng 1.1: Đặc điểm của các giải thuật hàm băm an toàn**

Giải thuật	Kích thước thông điệp (bít)	Kích thước khối (bít)	Kích thước từ (bít)	Kích thước thông điệp thu gọn (bít)	Bít an toàn (bít)
SHA-1	$< 2^{64}$	512	32	160	80
SHA-256	$< 2^{64}$	512	32	256	128
SHA-384	$< 2^{128}$	1024	64	384	192
SHA-512	$< 2^{128}$	1024	64	512	256

## 1.2. Các giải thuật hàm băm an toàn

### 1.2.1. SHA-1

SHA-1 có thể được dùng để tính băm cho các thông điệp  $M$  có độ dài là  $l$  bít với  $0 \leq l < 2^{64}$ . Giải thuật sử dụng:

1. Một danh sách thông định sẵn gồm 80 từ 32 bít.
2. Năm biến làm việc có độ dài 32 bít.
3. Một giá trị băm bao gồm 5 từ 32 bít. Kết quả cuối cùng của giải thuật SHA-1 là một thông điệp rút gọn 160 bít.

Các từ của danh sách thông định sẵn được đánh nhãn là  $W_0, W_1, \dots, W_{79}$ . Năm biến làm việc là  $a, b, c, d$  và  $e$ . Các từ của giá trị băm được gán nhãn là  $H_0^{(i)}, H_1^{(i)}, \dots, H_4^{(i)}$  trong đó giá trị băm khởi tạo  $H^{(0)}$  liên tiếp được thay thế bởi các giá trị băm trung gian  $H^{(i)}$  cho đến giá trị băm cuối cùng  $H^{(N)}$ . Ngoài ra SHA-1 còn sử dụng một biến trung gian  $T$  có kích thước bằng một từ 32 bít.

## Qui trình tiền xử lý của SHA-1

1. Độ tin cho bản tin  $M$ .
2. Phân bản tin  $M$  ra thành  $N$  khối 512 bit  $M^{(1)}, M^{(2)}, \dots, M^{(N)}$ .
3. Thiết lập giá trị khởi tạo  $H^{(0)}$ .

## Qui trình tính toán băm của SHA-1

SHA-1 được thực hiện trên phép cộng modulo  $2^{32}$  trên từng từ 32 bit của khối đầu vào. Sau khi tiền xử lý, mỗi khối tin  $M^{(1)}, M^{(2)}, \dots, M^{(N)}$  được xử lý lần lượt theo các bước sau:

For  $i = 1$  to  $N$  do

{

1. Chuẩn bị danh sách thông định sẵn  $\{W_t\}$ :

$$W_t = \begin{cases} M_t^{(i)} & \text{nếu } 0 \leq t \leq 15 \\ \text{ROTL}^1(W_{t-3}) \oplus W_{t-8} \oplus W_{t-16} & \text{nếu } 16 \leq t \leq 79 \end{cases}$$

2. Khởi tạo 5 biến hoạt động  $a, b, c, d$  và  $e$  ứng với giá trị băm thứ  $(i-1)$ .

$$a = H_0^{(i-1)}$$

$$b = H_1^{(i-1)}$$

$$c = H_2^{(i-1)}$$

$$d = H_3^{(i-1)}$$

$$e = H_4^{(i-1)}$$

3. For  $t = 0$  to  $79$  do

{

$$T = \text{ROTL}^5(a) + f_t(b, c, d) + e + K_t + W_t$$

$$e = d$$

$$d = c$$

$$c = \text{ROTL}^3(b)$$

$$b = a$$

$$a = T$$

}

4. Tính giá trị băm trung gian thứ  $i$  -  $H^{(i)}$

$$H_0^{(i)} = a + H_0^{(i-1)}$$

$$H_1^{(i)} = b + H_1^{(i-1)}$$

$$H_2^{(i)} = c + H_2^{(i-1)}$$

$$H_3^{(i)} = d + H_3^{(i-1)}$$

$$H_4^{(i)} = e + H_4^{(i-1)}$$

}

Sau khi lặp lại các bước từ (1) tới (4) là  $N$  lần, thông điệp rút gọn 160 bit của  $M$  sẽ là  $H_0^{(N)} \parallel H_1^{(N)} \parallel \dots \parallel H_4^{(N)}$ .

### 1.2.2. SHA-256

SHA-256 có thể được dùng để tính băm cho các thông điệp  $M$  có độ dài là  $l$  bit với  $1 \leq l \leq 2^{64}$ . Giải thuật SHA-256 sử dụng:

1. Một danh sách thông định sẵn gồm 64 từ 32 bit.

2. Tám biến hoạt động có độ dài 32 bit.

3. Một giá trị băm gồm 8 từ 32 bit. Kết quả cuối cùng của giải thuật SHA-256 là thông điệp rút gọn dài 256.

Các từ của danh sách thông định sẵn được gán nhãn là  $W_0, W_1, \dots, W_{63}$ . Tám biến hoạt động đặt tên là  $a, b, c, d, e, f, g$  và  $h$ . Các từ của giá trị băm

được gán nhãn là  $H_0^{(i)}, H_1^{(i)}, \dots, H_7^{(i)}$  trong đó giá trị băm khởi tạo  $H^{(0)}$  sẽ được thay thế liên tiếp bằng các giá trị băm trung gian  $H^{(i)}$  cho đến giá trị băm cuối cùng là  $H^{(N)}$ . SHA-256 còn sử dụng hai biến trung gian  $T_1$  và  $T_2$  có độ lớn là 32 bit.

### Quy trình tiền xử lý của SHA-256

1. Độ tin cho bản tin  $M$ .
2. Phân bản tin  $M$  ra thành  $N$  khối 512 bit  $M^{(1)}, M^{(2)}, \dots, M^{(N)}$ .
3. Thiết lập giá trị khởi tạo  $H^{(0)}$ .

### Quy trình tính toán băm của SHA-256

SHA-256 được thực hiện trên phép cộng modulo  $2^{64}$  trên từng từ 32 bit của khối đầu vào. Sau bước tiền xử lý, từng khối tin  $M^{(1)}, M^{(2)}, \dots, M^{(N)}$  sẽ được xử lý tuần tự theo các bước sau:

For  $i = 1$  to  $N$  do

{

1. Chuẩn bị danh sách thông định sẵn,  $\{W_t\}$ :

$$W_t = \begin{cases} M_t^{(i)} & \text{nếu } 0 \leq t \leq 15 \\ \sigma_1^{(256)}(W_{t-2}) + W_{t-7} + \sigma_0^{(256)}(W_{t-15}) + W_{t-16} & \text{nếu } 16 \leq t \leq 63 \end{cases}$$

2. Khởi tạo 8 biến hoạt động  $a, b, c, d, e, f, g$  và  $h$  với giá trị băm thứ  $(i-1)$

$$a = H_0^{(i-1)}$$

$$b = H_1^{(i-1)}$$

$$c = H_2^{(i-1)}$$

$$d = H_3^{(i-1)}$$

$$e = H_4^{(i-1)}$$

$$f = H_5^{(i-1)}$$



$$g = H_6^{(i-1)}$$

$$h = H_7^{(i-1)}$$

3. For  $t = 0$  to 63 do

{

$$T_1 = h + \sum_1^{256}(e) + ch(e, f, g) + K_t^{\{256\}} + W_t$$

$$T_2 = \sum_0^{256}(e) + ch(e, f, g) + K_t^{\{256\}} + W_t$$

$$h = g$$

$$g = f$$

$$f = e$$

$$e = d + T_1$$

$$d = c$$

$$c = b$$

$$a = T_1 + T_2$$

}

4. Tính toán giá trị băm trung gian thứ  $i$  ( $H^{(i)}$ )

$$H_0^{(i)} = a + H_0^{(i-1)}$$

$$H_1^{(i)} = b + H_1^{(i-1)}$$

$$H_2^{(i)} = c + H_2^{(i-1)}$$

$$H_3^{(i)} = d + H_3^{(i-1)}$$

$$H_4^{(i)} = e + H_4^{(i-1)}$$

$$H_5^{(i)} = g + H_5^{(i-1)}$$

$$H_6^{(i)} = h + H_6^{(i-1)}$$

$$H_7^{(i)} = f + H_7^{(i-1)}$$

}

Sau khi lặp lại  $N$  lần các bước từ (1) tới (4) thì ta thu được thông điệp rút gọn 256 bit của thông điệp  $M$  là:  $H_0^{(N)} // H_1^{(N)} // H_2^{(N)} // H_3^{(N)} // H_4^{(N)} // H_5^{(N)} // H_6^{(N)} // H_7^{(N)}$

### 1.2.3. SHA-512

Trong phần này, em xin trình bày giải thuật SHA-512 trước giải thuật SHA-384 bởi vì giải thuật SHA-384 giống hệt giải thuật SHA-512, chỉ có giá trị băm khởi tạo là khác và giá trị băm cuối cùng được cắt lấy 384 bit chứ không phải lấy toàn bộ 512 bit như giải thuật SHA-512 để tạo ra thông điệp rút gọn. Mỗi giải thuật băm an toàn đều có các phương pháp tính luân phiên để tạo ra kết quả.

SHA-512 có thể được dùng để tính băm cho các thông điệp  $M$  có độ dài là  $l$  bit với  $1 \leq l \leq 2^{128}$ . Giải thuật SHA-512 sử dụng:

1. Một danh sách thông định sẵn gồm 80 từ 64 bit.
2. Tám biến hoạt động có độ dài 64 bit.
3. Một giá trị băm gồm 8 từ 64 bit. Kết quả cuối cùng của giải thuật SHA-512 là thông điệp rút gọn dài 512 bit.

Các từ của danh sách thông định sẵn được gán nhãn là  $W_0, W_1, \dots, W_{79}$ . Tám biến hoạt động đặt tên là  $a, b, c, d, e, f, g$  và  $h$ . Các từ của giá trị băm được gán nhãn là  $H_0^{(i)}, H_1^{(i)}, \dots, H_7^{(i)}$  trong đó giá trị băm khởi tạo  $H^{(0)}$  sẽ được thay thế liên tiếp bằng các giá trị băm trung gian  $H^{(i)}$  cho đến giá trị băm cuối cùng là  $H^{(N)}$ . SHA-512 cũng sử dụng hai biến trung gian  $T_1$  và  $T_2$  nhưng có độ lớn là 64 bit.

### Quy trình tiền xử lý của SHA-512

1. Độn tin cho bản tin  $M$ .
2. Phân bản tin  $M$  ra thành  $N$  khối 1024 bit  $M^{(1)}, M^{(2)}, \dots, M^{(N)}$ .
3. Thiết lập giá trị khởi tạo  $H^{(0)}$ .

## Qui trình tính toán băm của SHA-512

SHA-512 được thực hiện trên phép cộng modulo  $2^{64}$  trên từng từ 64 bit của khối đầu vào. Sau bước tiền xử lý, từng khối tin  $M^{(1)}, M^{(2)}, \dots, M^{(N)}$  sẽ được xử lý tuần tự theo các bước sau:

For  $i = 1$  to  $N$  do

{

1. Chuẩn bị danh sách thông định sẵn,  $\{W_t\}$ :

$$W_t = \begin{cases} M_t^{(i)} & \text{nếu } 0 \leq t \leq 15 \\ \sigma_1^{(256)}(W_{t-2}) + W_{t-7} + \sigma_0^{(256)}(W_{t-15}) + W_{t-16} & \text{nếu } 16 \leq t \leq 79 \end{cases}$$

2. Khởi tạo 8 biến hoạt động  $a, b, c, d, e, f, g$  và  $h$  với giá trị băm thứ (i-1)

$$a = H_0^{(i-1)}$$

$$b = H_1^{(i-1)}$$

$$c = H_2^{(i-1)}$$

$$d = H_3^{(i-1)}$$

$$e = H_4^{(i-1)}$$

$$g = H_5^{(i-1)}$$

$$h = H_6^{(i-1)}$$

$$f = H_7^{(i-1)}$$

3. For  $t = 0$  to 79 do

{

$$T_1 = h + \Sigma_1^{512}(e) + ch(e, f, g) + K_t^{\{512\}} + W_t$$

$$T_2 = \Sigma_0^{512}(e) + ch(e, f, g) + K_t^{\{512\}} + W_t$$

$$h = g$$

$$g = f$$

$$f = e$$

$$e = d + T_1$$

$$d = c$$

$$c = b$$

$$a = T_1 + T_2$$

}

4. Tính toán giá trị băm trung gian thứ  $i$  ( $H^{(i)}$ )

$$H_0^{(i)} = a + H_0^{(i-1)}$$

$$H_1^{(i)} = b + H_1^{(i-1)}$$

$$H_2^{(i)} = c + H_2^{(i-1)}$$

$$H_3^{(i)} = d + H_3^{(i-1)}$$

$$H_4^{(i)} = e + H_4^{(i-1)}$$

$$H_5^{(i)} = g + H_5^{(i-1)}$$

$$H_6^{(i)} = h + H_6^{(i-1)}$$

$$H_7^{(i)} = f + H_7^{(i-1)}$$

}

Sau khi lặp lại  $N$  lần các bước từ (1) tới (4) thì thu được thông điệp rút gọn 512 bit của thông điệp  $M$  là:  $H_0^{(N)} \parallel H_1^{(N)} \parallel H_2^{(N)} \parallel H_3^{(N)} \parallel H_4^{(N)} \parallel H_5^{(N)} \parallel H_6^{(N)} \parallel H_7^{(N)}$ .

### 1.2.4. SHA-384

SHA-384 có thể được dùng để tính băm cho các thông điệp  $M$  có độ dài  $l$  bit với  $0 \leq l \leq 2^{128}$ . Giải thuật này được định nghĩa tương tự với giải thuật SHA-512 trừ hai điểm khác biệt duy nhất sau:

1. Giá trị băm khởi tạo  $H^{(0)}$  của SHA-384 khác với giá trị khởi tạo  $H^{(0)}$  của SHA-512.

2. Thông điệp rút gọn của SHA-384 có độ dài là 384 bit chứ không phải là 512 bit như của SHA-512. Bởi vì ở khâu cuối cùng, nó chặt bớt 128 bit bên phải của giá trị băm cuối cùng  $H^{(N)}$ , và chỉ giữ lại 384 bit trái nhất của giá trị băm cuối cùng đó để tạo ra thông điệp rút gọn là:  $H_0^{(N)} // H_1^{(N)} // H_2^{(N)} // H_3^{(N)} // H_4^{(N)} // H_5^{(N)}$  thay vì là giữ nguyên toàn bộ tất cả các bit (hay các từ) của  $H^{(N)}$  như SHA-512.

## 1.3. Chữ ký số

### 1.3.1. Định nghĩa

Một sơ đồ chữ ký số là bộ 5 (P, A, K, S, V) thỏa mãn các điều kiện sau:

- + P: là tập hữu hạn các bức điện có thể.
- + A: là tập hữu hạn các chữ ký có thể.
- + K: không gian khoá, là tập hữu hạn các khoá có thể.
- + Với mỗi  $K \in K$  tồn tại một thuật toán ký  $Sig_K \in S$  và một thuật toán xác minh  $Ver_K \in V$ .

Mỗi  $Sig_K: P \rightarrow A$  và  $Ver_K: P \times A \rightarrow \{TRUE, FALSE\}$  là những hàm sao cho mỗi bức điện  $x$  thuộc P và mỗi bức điện  $y \in A$  thỏa mãn phương trình sau đây:

$$Ver(x, y) = \begin{cases} TRUE & \text{nếu } y = Sig(x) \\ FALSE & \text{nếu } y \neq Sig(x) \end{cases}$$

Với mỗi  $K \in K$ , hàm  $\text{Sig}_K$  và  $\text{Ver}_K$  là các hàm thời gian đa thức.  $\text{Ver}_K$  sẽ là hàm công khai còn  $\text{Sig}_K$  là hàm bí mật. Gọi Alice là người gửi còn Bob là người nhận. Không thể dễ dàng tính toán để giả mạo chữ ký của Bob trên bức điện  $x$ . Nghĩa là với  $x$  cho trước, chỉ có Bob mới có thể tính được chữ ký  $y$  để  $\text{Ver}(x, y) = \text{True}$ . Một sơ đồ chữ ký không thể an toàn vô điều kiện vì một người tò mò nào đó có thể kiểm tra tất cả các chữ số  $y$  có thể trên bức điện  $x$  nhờ dùng thuật toán  $\text{Ver}$  công khai cho đến khi anh ta có thể tìm thấy một chữ ký đúng. Vì thế, nếu có đủ thời gian anh ta luôn luôn có thể giả mạo chữ ký của Bob. Như vậy, giống như trường hợp hệ thống mã hoá công khai, mục đích đặt ra là tìm các sơ đồ chữ ký số an toàn về mặt tính toán.

### ***1.3.2. Chức năng chữ ký số***

#### ***1.3.2.1. Khả năng nhận thức***

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

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

#### 1.3.2.2. Tính toàn vẹn

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

#### 1.3.2.3. Tính không thể phủ nhận

Trong giao dịch, một bên có thể từ chối nhận một văn bản nào đó là do mình gửi. Để ngăn ngừa khả năng này, bên nhận có thể yêu cầu bên gửi phải gửi kèm chữ ký số với văn bản. Khi có tranh chấp, bên nhận sẽ dùng chữ ký này như một chứng cứ để bên thứ ba giải quyết. Tuy nhiên, khóa bí mật vẫn có thể bị lộ và tính không thể phủ nhận cũng không thể đạt được hoàn toàn.

Vậy làm thế nào để đảm bảo các tính chất trên? Ở đây, bài toán sẽ sử dụng mã hóa để thực hiện việc tạo chữ ký điện tử. Một số thuật toán sau được sử dụng trong việc tạo ra chữ ký điện tử:

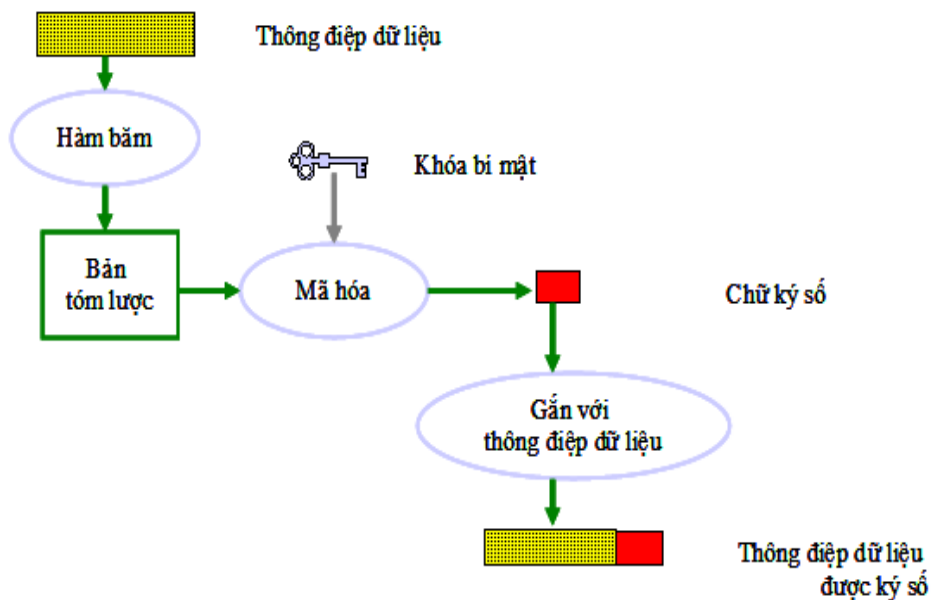
- Full Domain hash, RSA- PSS,... dựa trên RSA
- DSA
- ECDSA
- ElGamal Signature Scheme
- Undeniable Signature
- SHA (thông thường là SHA-1) với RSA

### 1.3.3. Quá trình tạo chữ ký số

- Dùng giải thuật băm để tính message digest (MD) của thông điệp cần truyền đi. Kết quả ta được một message digest.

- Sử dụng khóa bí mật của người gửi để mã hóa message digest thu được ở bước 1. Tiếp theo, bước này sẽ dùng giải thuật DSA. Kết quả thu được gọi là digital signature (DS) của thông điệp ban đầu. Công việc này gọi là “ký” vào thông điệp. Sau khi đã ký vào thông điệp, mọi sự thay đổi trên thông điệp sẽ bị phát hiện trong giai đoạn kiểm tra. Ngoài ra, việc ký này đảm bảo người nhận tin tưởng thông điệp này xuất phát từ người gửi chứ không phải là ai khác.

- Gộp digital signature vào thông điệp ban đầu và gửi đến người nhận. Dùng giải thuật băm để thay đổi thông điệp cần truyền đi, kết quả thu được một message digest, dùng giải thuật băm SHA-1 và message digest sẽ có chiều dài 160 bit.

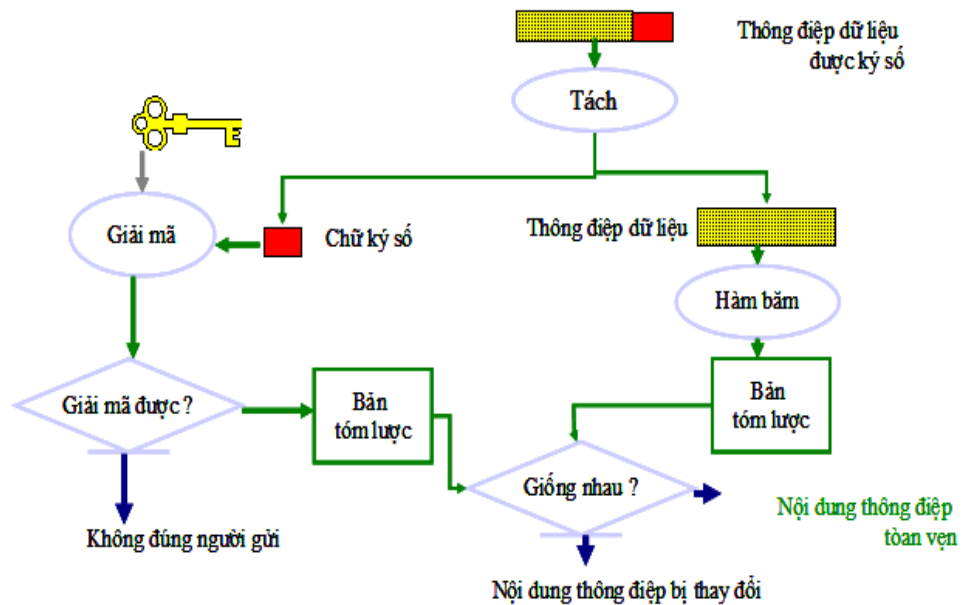


Hình 1.1: Quá trình tạo chữ ký số



#### 1.3.4. Quá trình thẩm định chữ ký số

- Tách message ban đầu và chữ ký số. Dùng khóa công khai của người gửi (khóa này được thông báo đến mọi người) để giải mã chữ ký số của thông điệp.
- Dùng giải thuật SHA-1 băm thông điệp ban đầu.
- So sánh 2 chuỗi băm kết quả thu được ở 2 bước trên. Nếu trùng nhau, kết luận thông điệp này không bị thay đổi trong quá trình truyền và thông điệp này là của người gửi.



Hình 1.2: Quá trình thẩm định chữ ký số

## CHƯƠNG 2: CHUẨN CHỮ KÝ SỐ DSA

### 2.1. Giới thiệu

Để nâng cấp việc sử dụng thương mại điện tử của quốc gia và trong giao dịch, Viện tiêu chuẩn và công nghệ quốc gia Hoa kỳ (NIST) đã đưa ra chuẩn xử lý thông tin FIPS 186 là chuẩn chữ ký số (DSS- Digital Signature Standard) vào ngày 19/5/1994 và được chấp nhận từ ngày 1/12/1994. Phiên bản đầu tiên của chuẩn chữ ký số là FIPS PUB 186, phiên bản tiếp theo là FIPS PUB 186-1 được đưa ra vào ngày 15/12/1998, phiên bản thứ 3 là FIPS PUB 186-2 đưa ra vào ngày 27/1/2000 và phiên bản mới nhất hiện nay là FIPS PUB 186-3 được công bố vào tháng 3/2006.

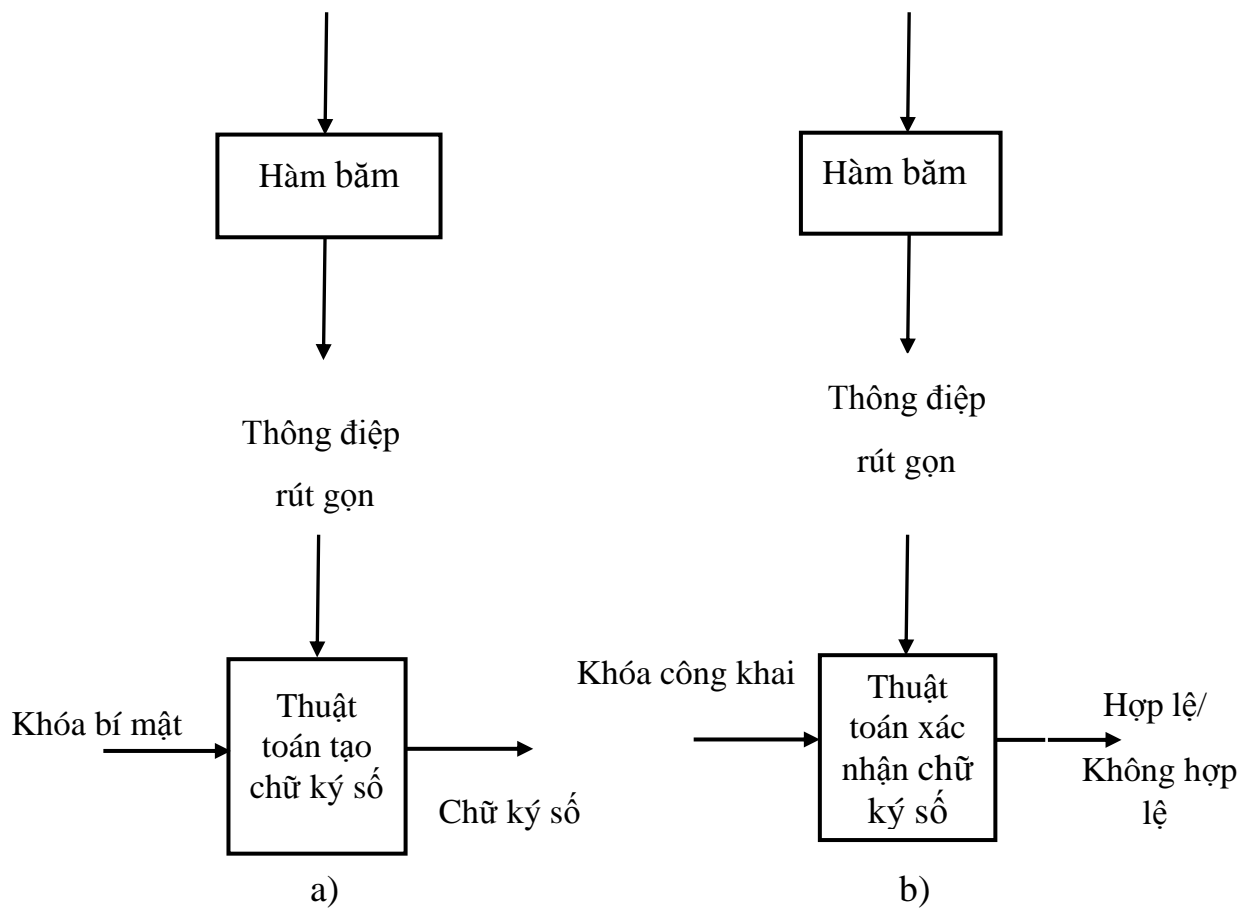
Thuật toán chữ ký số – Digital Signature Algorithm (DSA) – là 1 tiêu chuẩn xử lý thông tin cơ bản (Federal Information Processing Standard – FIPS) cho các chữ ký số. DSA được bao hàm bên trong bằng sáng chế số 5.231.668 được chính phủ Hoa Kỳ cấp vào ngày 26/7/1991 và được đóng góp bởi David W. Kravitz, một cựu nhân viên của NSA. Bằng sáng chế này đã được trao cho chính phủ Hoa Kỳ, đại diện bởi Bộ trưởng thương mại, tại thủ đô Washington, và NIST đã làm cho bằng sáng chế này trở thành miễn phí trên toàn thế giới. Claus P. Schnorr cho rằng bằng sáng chế số 4.995.082 của ông đã bao hàm luôn DSA; tuyên bố này vẫn đang được tranh cãi. DSA là 1 biến thể của sơ đồ chữ ký ElGamal.

Bảng dưới đây so sánh sự khác nhau giữa các phiên bản đã có của DSS.

**Bảng 2.1: Sự khác nhau giữa các phiên bản DSS**

Phiên bản	Giá trị cặp (L, N) (bít)	Nội dung	Chuẩn hàm băm sử dụng
FIPS PUB 186	$L \in [512, 1024]$ $N \in [159, 160]$	Giải thuật DSA	Chưa có
FIPS PUB 186-1	$L \in [512, 1024]$ $N \in [159, 160]$	Giải thuật DSA	FIPS PUB 180-1
FIPS PUB 186-2	$L \in [512, 1024]$ $N \in [159, 160]$	Giải thuật DSA Giải thuật ECDSA	FIPS PUB 180-1
FIPS PUB 186-3	$L = 1024, N = 160$ $L = 2048, N = 224$ $L = 3072, N = 256$	Giải thuật DSA Giải thuật ECDSA	FIPS PUB 180-2

Phần tiếp theo sẽ trình bày về giải thuật DSA trong phiên bản FIPS PUB 186-3. Một hàm băm sẽ được sử dụng tương ứng trong giải thuật tạo và xác thực chữ ký, nhằm giảm bớt độ dài thông điệp ký. Sơ đồ sau thể hiện việc sử dụng hàm băm trong giải thuật tạo và xác nhận chữ ký số.



Hình 2.1: Sơ đồ sử dụng giải thuật hàm băm trong giải thuật chữ ký số

a) Tạo chữ ký số

b) Xác nhận chữ ký số

## 2.2. Thuật toán chữ ký số DSA

### 2.2.1. Nguyên tắc hoạt động

DSA dùng một hệ thống khoá chung không đảo ngược, trên cơ sở sự tiếp cận của ElGamal, được sửa đổi bởi Schnorr [SCH1]. Sự an toàn của nó phụ thuộc vào mức độ phức tạp của việc tính toán các loga rời rạc. Có:

$$y = g^x \bmod p$$

Trong đó:  $p$  là một số nguyên tố và  $g = j^{[(p-1)/q]} \bmod p$  với  $j$  là bất kỳ một số nguyên dương ngẫu nhiên sao cho  $1 < j < p$  để:  $j^{[(p-1)/q]} \bmod p > 1$ . Tính  $y$  với  $g$ ,  $x$ , và  $p$  đã cho, nhưng rất khó để tính  $x$ , khi cho  $y$ ,  $g$ , và  $p$ . Điều này đưa ra một nền tảng cho hệ thống khoá chung trong đó  $x$  là một khoá riêng và  $y$  là một khoá chung. Hệ thống sử dụng 3 số nguyên  $p$ ,  $q$ , và  $g$  có thể được tạo

chung và phổ biến cho các nhóm người sử dụng,  $p$  là một môđun nguyên, nằm trong khoảng 512 đến 1024 bit,  $q$  là một số chia nguyên 160 bit.

Để người gửi đưa ra, khoá riêng  $x$  được chọn một cách ngẫu nhiên, với  $1 < x < q$ . Khoá chung  $y$  được tính như ở trên. Để ký hiệu một tin nhắn mà có điện báo  $h$ , người sử dụng chọn một số nguyên ngẫu nhiên  $k$  (với  $0 < k < q$ ) và sử dụng khoá riêng  $x$  để tính  $(r, s)$ . Hai số  $r, s$  được tính như sau:

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

$$s = (k^{-1}(h + x^r)) \bmod q$$

trong đó  $k^{-1}$  là nghịch đảo của  $k \bmod q$ ; ví dụ,  $(k^{-1}) \bmod q = 1$  và  $0 < k^{-1} < q$ . Một cặp giá trị  $(r, s)$  tạo thành phụ lục chữ ký cho tin nhắn.

Để phân loại một chữ ký đã nhận  $(r', s')$  kèm theo một tin nhắn với điện báo  $h'$ , người nhận đầu tiên kiểm tra  $0 < r' < q$  và  $0 < s' < q$ . Nếu một trong hai điều kiện này bị sai, chữ ký đó sẽ bị loại. Ngoài ra, người nhận sau đó tính từ  $s'$  và  $h'$  ra giá trị  $v$ . Để chữ ký được phân loại chính xác, giá trị này cần phải giống như là giá trị  $r'$  đã được gửi trong chữ ký. Công thức tính  $v$  như sau:

$$w = (s')^{-1} \bmod q$$

$$u^1 = ((h')^w) \bmod q$$

$$u^2 = ((r')^w) \bmod q$$

$$v = ((g^{u^1} y^{u^2}) \bmod p) \bmod q$$

### 2.2.2. Các tham số

Chữ ký số DSA được tính toán dựa trên một tập các tham số miền, khóa bí mật, số bí mật của mỗi thông điệp, dữ liệu để ký và một hàm băm. Việc xác nhận chữ ký số cũng dựa trên tập các tham số miền và khóa công khai này, dựa trên dữ liệu dùng để xác nhận và hàm băm đã dùng để tạo ra chữ ký. Các tham số này là:

- $p$ : là một số nguyên tố trong đó  $2^{L-1} < p < 2^L$  với  $L$  là độ dài bit của biến  $p$ .
- $q$ : là một ước số nguyên tố của  $p-1$  trong đó  $2^{N-1} < p < 2^N$  với  $N$  là chiều dài của biến  $q$  (tính theo bit).
- $g$ : là căn bậc  $q$  của  $h$  theo mod  $p$ . Dễ dàng tính được  $g$  như sau:  
 $g = h^{(p-1)/q} \bmod p$ . Với  $1 < h < p-1$
- $x$ : là khóa bí mật.  $x$  là một số nguyên dương được tạo ra một cách ngẫu nhiên sao cho  $0 < x < q-1$ .
- $y$ : là khóa công khai, trong đó  $y = g^x \bmod p$ .
- $k$ : là số bí mật và là duy nhất với mỗi một tin nhắn.  $k$  là số nguyên được tạo ra ngẫu nhiên sao cho  $0 < k < q$ .

### 2.2.3. Kích thước tham số và các hàm băm sử dụng

Chuẩn này có đưa ra một số các lựa chọn cho cặp  $(L, N)$  như sau:

$$(L = 1024, N = 160), (L = 2048, N = 224) \text{ và } (L = 3072, N = 256)$$

Quá trình tạo chữ ký số sẽ sử dụng tới một hàm băm. Độ an toàn của hàm băm này phải bằng hoặc lớn hơn độ an toàn của cặp  $(L, N)$ . Người ta khuyến cáo rằng độ an toàn của cặp  $(L, N)$  và của hàm băm là như nhau trừ khi có một sự thỏa thuận giữa các bên tham gia nhằm sử dụng một hàm băm mạnh. Những hàm băm có độ an toàn nhỏ hơn độ an toàn của cặp  $(L, N)$  sẽ không được sử dụng. Nếu đầu ra của hàm băm có chiều dài là  $T > N$  bit thì  $N$  bit trái nhất của nó sẽ được sử dụng (thay cho  $T$  bit đầu ra như thường lệ) trong mọi tính toán của qui trình tạo và xác nhận chữ ký số. Mỗi cặp  $(L, N)$  được lựa chọn sao cho nó có thể bảo vệ được thông tin trong toàn bộ quãng thời gian sống của thông tin. Ví dụ nếu chữ ký số được tạo ra năm 2007 cho thông tin cần được bảo vệ trong 5 năm thì cặp  $(L, N)$  được chọn phải đủ lớn để có thể bảo vệ được thông tin trong suốt 5 năm đó.

Sau đây là bảng lựa chọn hàm băm cho từng cặp  $(L, N)$ .

**Bảng 2.2: Lựa chọn hàm băm cho cặp (L, N)**

Cặp L, N (bít)	Bit an toàn (bít)	Hàm băm
L = 1024 N = 160	80	SHA-1 SHA-256 SHA-384 SHA-512
L = 2048 N = 224	112	SHA-256 SHA-384 SHA-512
L = 3072 N = 256	128	SHA-256 SHA-384 SHA-512

**2.2.4. Các tham số miền**

Giải thuật DSA qui định cặp khóa được sử dụng cho quá trình tạo và xác nhận chữ ký số phải được tạo ra dựa trên tập các tham số miền của DSA. Các tham số miền này có thể được một nhóm người hoặc cả cộng đồng biết tới. Người sử dụng tập tham số này sẽ phải đảm bảo tính hợp lệ của chúng trước khi sử dụng chúng. Mặc dù các tham số này có thể là công khai nhưng chúng vẫn cần phải được quản lý nhằm bảo vệ sự phù hợp tương ứng giữa cặp khóa và các tham số miền cho tất cả các bên sử dụng cặp khóa.

Với chuẩn DSA, tham số miền gồm có các số nguyên  $p$ ,  $q$ ,  $g$  và *domain\_parameter\_seed* - tham số gốc, *counter* - bộ đếm (nếu có yêu cầu) đã sử dụng để tạo ra  $p$ ,  $q$ . Như vậy bộ tham số miền đầy đủ sẽ là  $(p, q, g, \{ \text{domain\_parameter\_seed}, \text{counter} \})$ .

Việc phát sinh tham số được thực hiện lần lượt như sau:

- Chọn 1 hàm băm mật mã đã được phê duyệt  $H$ . Trong DSA ban đầu,  $H$  luôn luôn là SHA-1, nhưng các hàm băm mạnh hơn SHA-2 đã được cho

phép sử dụng trong DSA gần đây. Kết quả xuất ra của hàm băm có thể được cắt lại bằng với kích thước của cặp khóa.

- Quyết định 1 chiều dài khóa  $L$  và  $N$ . Đây là thước đo chính cho độ bảo mật của khóa. DSA ban đầu giới hạn  $L$  là bội số của 64 sao cho  $512 < L < 1024$ . NIST 800-57 đề nghị chiều dài là 2048 (hoặc 3072) cho các khóa với vòng đời bảo mật mở rộng vượt qua 2010 (hoặc 2030), sử dụng  $N$  dài hơn tương ứng. FIPS 186-3 chỉ rõ các cặp chiều dài của  $L$  và  $N$  là (1024, 160), (2048, 224) và (3072, 256).

- Chọn số nguyên tố  $q$  có độ lớn  $N$  bit.  $N$  phải nhỏ hơn hoặc bằng chiều dài kết quả băm ở trên.

- Chọn số nguyên tố  $p$  có độ lớn  $L$  bit sao cho  $(p-1) \bmod q = 0$ .

- Chọn  $g$  với công thức sau:  $g = h^{(p-1)/q} \bmod p$  với  $g > 1$ , với  $h$  là 1 số bất kỳ thỏa mãn  $1 < h < p-1$ . Nếu  $g = 1$ , thử lại với  $h$  khác. Đa số các giá trị của  $h$  đều dẫn đến 1 giá trị  $g$  khả dụng; thường thì  $h=2$  được sử dụng phổ biến.

Các tham số thuật toán  $(p, q, g)$  có thể được chia sẻ giữa các người dùng khác nhau trong hệ thống.

### **Tạo các tham số miền**

Các tham số miền có thể do một bên ủy nhiệm thứ 3 hoặc do một thực thể nào đó tạo ra. Việc xác nhận đảm bảo tính hợp lệ của chúng sẽ được tiến hành trước khi tạo khóa cũng như trước khi tạo và xác chữ ký số.

Qui trình tạo  $p, q$  có đầu vào là các giá trị  $L, N$  - là độ dài tương ứng của các số  $p, q$  cần tạo ra và có đầu ra là giá trị của  $p, q$  cũng như các biến *domain\_parameter\_seed*, *counter* (nếu cần).

### **Quản lý các tham số miền**

Bởi vì cặp khóa được tạo ra dựa trên tập các tham số miền nên giữa chúng có một sự gắn kết chặt chẽ. Chính vì vậy, các tham số miền sau khi



được tạo ra cần phải được quản lý để tránh việc chỉnh sửa bất hợp pháp cho đến khi chúng không còn được sử dụng nữa.

### **2.2.5. Cặp khóa**

Mỗi một người ký đều sở hữu một cặp khóa bao gồm khóa bí mật  $x$  và khóa công khai  $y$ , chúng có quan hệ toán học qua lại lẫn nhau. Trong đó khóa bí mật chỉ sử dụng khi tạo chữ ký số còn khóa công khai vẫn tiếp tục được sử dụng khi chữ ký số vẫn còn cần xác nhận.

#### **Tạo cặp khóa.**

Cặp khóa  $(x, y)$  được tạo ra dựa trên tập các tham số miền  $(p, q, g \{ \text{domain\_parameter\_seed, counter} \})$  và giải thuật tạo khóa được trình bày cụ thể trong phần sau.

#### **Quản lý cặp khóa.**

Quản lý cặp khóa là một công việc thiết yếu và quan trọng. Việc sử dụng chữ ký số có được an toàn hay không là phụ thuộc vào việc quản lý cặp khóa, cụ thể như sau:

- Các tham số miền cần phải được đảm bảo trước khi tạo cặp khóa cũng như trước khi xác nhận và kiểm chứng chữ ký số.
- Mỗi cặp khóa cần phải được gắn liền với một tập các tham số miền nhất định mà dựa trên đó, cặp khóa được tạo ra.
- Cặp khóa chỉ được sử dụng để tạo và xác nhận chữ ký số bằng cách sử dụng các tham số miền gắn kết với nó.
- Khóa bí mật chỉ được sử dụng để tạo ra chữ ký số và sau đó, nó phải được giữ bí mật. Còn khóa công khai chỉ được sử dụng để xác nhận chữ ký số và được công khai cho mọi người biết.
- Bên định ký cần phải chắc chắn sở hữu khóa bí mật trước hoặc lúc dùng nó để tạo ra chữ ký số.

- Khóa bí mật cần phải được bảo vệ tránh các truy cập, giả mạo và chỉnh sửa bất hợp pháp.
- Khóa công khai cần phải được bảo vệ tránh việc chỉnh sửa bất hợp pháp (bao gồm cả trường hợp thay thế).
- Người xác nhận cần phải chắc chắn về quan hệ ràng buộc giữa khóa công khai, các tham số miền của nó và người sở hữu cặp khóa.
- Người xác nhận cần phải có được khóa công khai theo một cách thức đáng tin cậy.
- Người xác nhận cũng cần phải đảm bảo được rằng người tuyên bố là mình đã ký lên thông điệp phải là người sở hữu cặp khóa và rằng người sở hữu này phải nắm giữ khóa bí mật đã được dùng để tạo ra chữ ký số vào thời điểm chữ ký được tạo ra (khóa bí mật phải gắn chặt với khóa công khai mà sẽ được dùng để xác nhận chữ ký số).
- Người ký và người xác nhận cần phải đảm bảo chắc chắn về tính hợp lệ của khóa công khai.

### 2.3. Tạo cặp khoá

Trong phương pháp này, một số ngẫu nhiên  $N$  bit sẽ được tạo ra (bằng với số bit của khóa  $x$  cần tạo) và sau đó, số ngẫu nhiên này sẽ được kiểm tra để xác định xem liệu nó có thể tạo ra một giá trị  $x$  nằm trong khoảng xác định của nó hay không. Nếu  $x$  nằm ngoài khoảng xác định thì một số ngẫu nhiên khác sẽ lại được sinh ra và qui trình sẽ được lặp đi lặp lại cho đến khi tìm được một giá trị  $x$  chấp nhận được. Qui trình tạo khóa sẽ như sau:

Input:  $p, q, g$

Output: cặp khóa  $(x, y)$  với  $x$  là khóa bí mật còn  $y$  là khóa công khai

Qui trình:

1.  $N = \text{len}(q); L = \text{len}(p)$ .
2. If  $(L, N)$  is invalid then return *ERROR, Invalid\_x, Invalid\_y*.

3. Tạo một chuỗi bit ngẫu nhiên có độ lớn  $N$  bit.
4. Chuyển đổi chuỗi bit vừa tạo được thành số nguyên  $c$ .
5. If  $(c > q-1)$ , then goto step 4.
6.  $x = (c \bmod (q-1)) + 1$ .
7.  $y = g^x \bmod p$ .
8. Return *SUCCESS*,  $x$ ,  $y$ .

Thuật toán tạo khóa được thực hiện theo trình tự sau:

- Chọn số nguyên tố 160 bit  $q$ .
- Chọn 1 số nguyên tố  $L$  bit  $p$ , sao cho  $p = q * z + 1$  với số nguyên  $z$  nào đó,  $512 \leq L \leq 1024$ ,  $L$  chia hết cho 64.
- Chọn  $h$ , với  $1 < h < p - 1$  sao cho  $g = h^z \bmod p > 1$ . Trong đó  $z = (p-1)/q$ .
- Chọn  $x$  ngẫu nhiên, thỏa mãn  $0 < x < q$ .
- Tính giá trị  $y = g^x \bmod p$ .
- Khóa công khai là  $(p, q, g, y)$ . Khóa riêng là  $x$ .

Chú ý rằng  $(p, q, g)$  có thể dùng chung bởi nhiều người dùng trong hệ thống nếu muốn. FIPS 186-3 sử dụng SHA-224/256/384/512 như hàm băm,  $q$  với kích thước 224, 256, 384, và 512 bit,  $L$  nhận giá trị 2048, 3072, 7680, và 15360 tương ứng. Có các giải thuật hiệu quả để tính toán các biểu thức mũ và lấy phần dư khi chia cho số nguyên tố lớn  $h^z \bmod p$  và  $g^x \bmod p$ .

## CHƯƠNG 3: THIẾT KẾ VÀ CÀI ĐẶT ỨNG DỤNG

### 3.1. Phát biểu bài toán

Hiện nay, việc in hóa đơn theo mẫu của Bộ Tài Chính vẫn được in theo quy trình như sau:

- Dữ liệu cước hàng tháng sau khi được tính sẽ được tổng hợp lại theo từng mã khách hàng trên kho cơ sở dữ liệu Oracle.
- In hóa đơn cho các mã khách hàng theo từng khu vực và được sắp xếp theo thứ tự nhất định: Đơn vị, Mã đường thư, Số hóa đơn.
- Tạo file text hóa đơn (\*.txt) có cấu trúc theo từng khu vực (ví dụ: VT1THK04.TXT; VT2CHK04.TXT; VT3PCG04.TXT; ...)
- Bàn giao dữ liệu hóa đơn dạng text cho nhà máy in.
- Sau khi in, hóa đơn sẽ được trả về cho từng đơn vị quản lý bán hàng và được chuyển đến tay các đại lý thu thuê để đi thu tiền của khách hàng.

Do số lượng hóa đơn hàng tháng in là lớn, lên đến gần một triệu khách hàng trong một tháng, việc phát sinh sai sót là khó tránh khỏi. Thực tế cho thấy, khi khách hàng đến thanh toán tiền điện, với số lượng giấy rất lớn nên việc tìm kiếm rất khó khăn. Đối với hệ thống thanh toán này còn một số hạn chế như sau:

- Tính cước sai cho khách hàng.
- Tổng hợp cước sai cho khách hàng.
- Bàn giao dữ liệu in hóa đơn sai tháng cần in.

Những sai sót này lại thường chỉ được phát hiện khi hóa đơn đã đến được tay khách hàng. Điều này sẽ làm tốn rất nhiều chi phí công in ấn, giấy mực cũng như nhân công để đi thu hồi lại các ấn phẩm đã in sai. Vì vậy, Điện lực Long Biên cũng như các công ty Điện lực khác gặp rất nhiều khó khăn trong quá trình thanh toán và quản lý hóa đơn.

Từ những hạn chế gặp phải, thực tế yêu cầu phải đưa ra một cách thức thanh toán mới để khắc phục những điều trên. Đó chính là việc chuyển từ thanh toán tiền điện bằng hóa đơn giấy tự in sang thanh toán bằng hóa đơn điện tử. Việc áp dụng hóa đơn điện tử vào hoạt động bán điện là cần thiết do nhiều lợi ích cụ thể mang lại cho doanh nghiệp và người sử dụng điện như sau:

- Trước mắt, hóa đơn điện tử mang lại lợi ích cho cơ quan thuế, cơ quan thuế không kê khai hóa đơn bằng giấy như trước đây. Khi thực hiện áp dụng hóa đơn điện tử, việc kê khai được thực hiện qua mạng Internet bằng các phần mềm xác thực tính đúng đắn của hóa đơn và đảm bảo được nhanh chóng và chính xác, giảm việc lưu trữ hóa đơn bằng giấy như trước đây.

- Giảm chi phí in hóa đơn giấy và chi phí gửi, bảo quản, lưu trữ hoá đơn... so với sử dụng hoá tự in. Do đó, giúp doanh nghiệp tiết kiệm chi phí, nâng cao hiệu quả sản xuất kinh doanh, tăng năng lực cạnh tranh.

- Thuận tiện cho việc hạch toán kế toán; đối chiếu dữ liệu. Quá trình xử lý nhanh hơn và rẻ hơn vì thông tin trên hóa đơn điện tử được liên kết với các hệ thống quản lý thông tin khách hàng, hệ thống kế toán.

- Quá trình thanh toán nhanh hơn do việc lập, gửi và nhận hóa đơn được thực hiện nhanh hơn thông qua các phương tiện điện tử và không phải gửi hóa đơn qua đường bưu điện hoặc nhà vận chuyển. Xét về mặt thanh toán, hóa đơn tự in có 2 hình thức thanh toán là thanh toán qua hóa đơn giá trị gia tăng từ thu ngân viên trực tiếp và qua ngân hàng. Còn đối với hóa đơn điện tử có 5 hình thức thanh toán là thanh toán qua biên nhận thanh toán, SMS, Email, POS và Website chăm sóc khách hàng.

- Góp phần hiện đại hoá công tác hạch toán kế toán, quản trị doanh nghiệp để phù hợp hơn với xu thế kinh doanh ngày càng phát triển trên thị trường quốc tế hiện nay.

- Khách hàng có thể truy cập vào Website của bên bán để xem và tải hóa đơn khi cần. Do đó, khách hàng không phải lưu trữ, bảo quản hóa đơn,

tránh rủi ro mất hóa đơn. Về gửi hóa đơn điện tử cho người mua, công ty điện lực chuyển toàn bộ hóa đơn điện tử của khách hàng lên website để khách hàng có thể tra cứu bằng cách truy cập vào cổng thông tin chăm sóc khách hàng của công ty điện lực để nhận và tải hóa đơn điện tử, hoặc đơn vị kế toán đăng ký nhận hóa đơn qua email thì công ty sẽ thực hiện gửi hóa đơn qua mail cho khách hàng.

Nhận thấy được những lợi ích mà hóa đơn điện tử mang lại, hầu hết các doanh nghiệp ở nước ta đang dần chuyển sang hình thức thanh toán hiện đại này. Ngành điện lực cũng đang nỗ lực để áp dụng việc thanh toán bằng hóa đơn điện tử trên khắp cả nước. Tuy nhiên, do điều kiện về công nghệ thông tin chưa đáp ứng được nên mới chỉ được áp dụng ở những quận huyện và thành phố lớn.

Đối với thành phố Hà Nội, Công ty điện lực Long Biên là một trong những doanh nghiệp đi tiên phong cho việc thanh toán mới này. Vì thế bước đầu chuyển đổi gặp không ít những khó khăn. Khó khăn đáng nói nhất chính là vấn đề xây dựng hệ thống công nghệ thông tin phục vụ cho việc ký số hóa đơn tiền điện bởi lẽ đây là phương pháp thanh toán mới chưa được áp dụng rộng rãi. Cụ thể, một hệ thống quản lý của doanh nghiệp bao gồm rất nhiều hệ thống nhỏ. Trong hệ thống quản lý thanh toán tiền điện thì hệ thống nhỏ ký số hóa đơn là quan trọng nhất bởi nó xác nhận khách hàng đã thanh toán hay chưa. Trong khi đó, nguồn nhân lực công nghệ thông tin chưa đáp ứng được về chất lượng; hạ tầng công nghệ thông tin và mức độ áp dụng hệ thống ứng dụng công nghệ thông tin chưa đồng đều, một số hệ thống chưa được nâng cấp theo kịp với yêu cầu thực tế. Nhưng ngược lại giải pháp thanh toán mới này sẽ nâng cao chất lượng dịch vụ khách hàng, góp phần tạo ấn tượng tốt đẹp của khách hàng về phong cách làm việc chuyên nghiệp. Do vậy, bài toán đặt ra là xây dựng một ứng dụng ký số hóa đơn cho doanh nghiệp này. Chức năng cơ bản của ứng dụng là doanh nghiệp dùng chứng chỉ của họ hoặc chứng chỉ

được cho phép ký để tạo ra file hóa đơn JPG dựa cơ sở dữ liệu khách hàng của công ty Điện lực Long Biên.

### 3.2. Thiết kế bằng giải thuật DSA

#### 3.2.1. Sơ đồ lớp của giải thuật băm

Khối dữ liệu đầu vào  $x$  có chiều dài hữu hạn tùy ý sẽ được phân thành các khối con liên tiếp có chiều dài cố định  $r$ , giả sử được đánh số là  $x_1, x_2, \dots, x_m$ . Tuy nhiên do chiều dài của khối dữ liệu ban đầu  $x$  là tùy ý, do đó cần phải thêm vào dữ liệu ban đầu một số bit phụ sao cho tổng số bit của khối dữ liệu  $x'$  sau khi thêm vào sẽ là bội số của  $r$ . Ngoài ra khối bit thêm vào thường chứa một khối bit (có chiều dài cố định, thường là 64 bit) xác định chiều dài thực sự của khối bit dữ liệu khi chưa thêm các bit phụ.

Tiếp theo, lần lượt cắt các khối con  $r$  bit từ khối mở rộng  $x'$ . Mỗi khối con  $r$  bit  $x_i$  lần lượt bước qua một hàm nén  $f$  của hàm băm  $h(x)$ . Tại bước thứ  $i$ , hàm nén  $f$  nhận dữ liệu đầu vào là  $x_i$  và kết quả trung gian của bước trước đó (bước  $i-1$ ) để tạo đầu ra là kết quả trung gian bước thứ  $i$ , được ký hiệu là  $H_i$ . Kết quả trung gian tại mỗi bước  $H_i$  là một chuỗi bit có độ dài cố định bằng  $n > 0$ .

Kết quả được ký hiệu  $IV$  là giá trị ban đầu (cho  $H_0$ ), thì quá trình lặp xử lý dãy các khối con  $x_1, x_2, \dots, x_m$  được mô tả:

$$H_0 = IV$$

$$H_i = f(H_{i-1}, x_i) \quad (i = 1, 2, \dots, m)$$

$$h(x) = g(H_m)$$

- Các biến  $H_i$  là các biến dây chuyền.
- Hàm  $g(x)$  lấy biến dây chuyền cuối cùng để tạo ra mã băm cuối cùng cần tìm. Trong hầu hết các thuật toán,  $g(x)$  thường được chọn là ánh xạ đồng nhất, tức là:  $g(H_m) = H_m$ .

- Khâu then chốt trong xây dựng hàm băm là thiết kế hàm nén  $f$ .
- Giá trị của hàm băm mật mã của một thông điệp được gọi là Message Digest (MD).

Ở đây, bài toán sử dụng hàm băm SHA-1. Giải thuật SHA-1 tính toán kết quả băm dài 160 bit đối với thông điệp có độ dài nhỏ hơn  $2^{64}$  bit. Giải thuật có độ dài của từ là 32 bit mỗi thanh. Hàm nén làm việc với khối thông điệp 512 bit, khối được chia thành 16 từ 32 bit biểu diễn bởi  $W_j$  với  $j = 1, \dots, 15$ .

Bên trong, hàm nén chia thành 80 bước liên tiếp. Một sự phân biệt nữa là việc chia vòng: có 4 vòng, mỗi vòng gồm 20 bước. Phép tính bước của SHA-1 theo mẫu sau:

$$E \leftarrow E + f_r(B, C, D) + A \ll 5 + W_j + U_r$$

$$B \leftarrow B \ll 30$$

Mỗi bước tính giá trị mới cho 2 trong 5 thanh ghi. Trong trường hợp này ta xét đến bước cập nhật giá trị cho thanh ghi  $E$  và cũng quay giá trị của thanh ghi  $B$  một khoảng 30 bit về bên trái. Phép tính cập nhật giá trị cho thanh ghi  $E$  phụ thuộc vào 4 thanh ghi còn lại và theo:

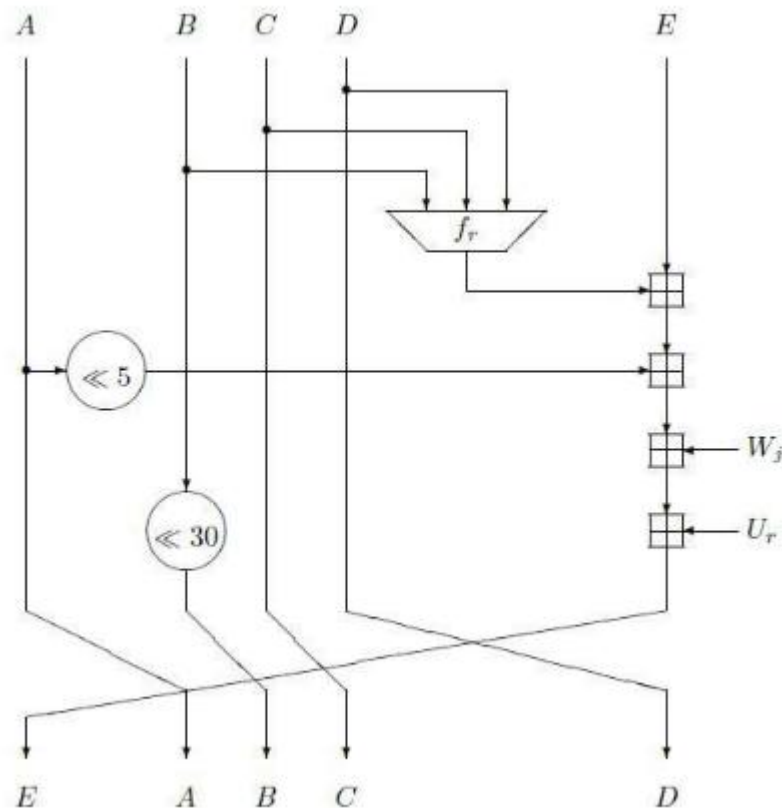
- Từ mang thông điệp  $W_j$  với  $j = \{0, 1, \dots, 79\}$ .
- Hàm Boolean  $f_r$  phụ thuộc vào vòng.
- Hằng số thêm vào  $U_r$  phụ thuộc vào vòng.

Hàm Boolean được sử dụng ở các vòng khác nhau trong hàm nén là hàm lựa chọn, đa số và exor. Hàm exor được sử dụng trong vòng 2 và 4. 16 từ đầu tiên  $W_j$  ( $j = 0, 1, \dots, 15$ ) bằng với khối lượng thông điệp đầu vào của hàm nén. 64 từ còn lại  $W_j$  ( $j = 16, \dots, 79$ ) được tính bằng thủ tục sau cho thông điệp mở rộng:

$$W_j = (W_{j-3} \text{ xor } W_{j-8} \text{ xor } W_{j-14} \text{ xor } W_{j-16}) \ll 1$$



Hình sau biểu diễn việc tính bước trong SHA-1. 5 bước liên tiếp cập nhật giá trị cho thanh ghi  $E, D, C, B, A$  tương ứng và cùng quay giá trị của thanh ghi  $B, A, E, D, C$  đi 30 bit vị trí sang bên trái. Sau 5 bước chuỗi biến được cập nhật hoàn chỉnh. Một vòng của hàm nén bao gồm 4 chuỗi của 5 bước. Mỗi thanh ghi được cập nhật 4 lần trong mỗi vòng và 16 lần trong mỗi hàm nén.



Hình 3.1: Quá trình mã hóa của SHA-1

Tuy nhiên sau 80 bước, hàm nén sử dụng phép toán feed- forward để thêm các giá trị khởi tạo vào giá trị cuối. Kết quả là chuỗi biến đầu ra của hàm nén, vì vậy hàm nén không bị nghịch đảo.

### 3.2.2. Sơ đồ tạo các tham số miền

Miền tham số của DSA là các số nguyên  $p, q, g$ . Trong đó:

- $p$  là số nguyên tố mà  $2^{L-1} < p < 2^L$ ,  $L$  là chiều dài bit của  $p$ .

- $q$  là số nguyên tố chia hết bởi  $(p-1)$ ,  $2^{N-1} < q < 2^N$ ,  $N$  là chiều dài bit của  $p$ .
- $g$  là một nhóm phụ của  $p$  và  $q$ , trong đó  $1 < g < p$ .

### Tạo số nguyên tố $p$ và $q$ bằng việc sử dụng hàm băm

Phương pháp này sử dụng một hàm băm phải có độ an toàn bằng hoặc lớn hơn độ an toàn của cặp  $(L, N)$ . Tuy nhiên người ta khuyến cáo rằng độ an toàn của hàm băm và của cặp  $(L, N)$  sẽ là như nhau trừ khi có một sự thỏa thuận giữa các bên tham gia nhằm sử dụng một hàm băm mạnh. Tham số gốc *domain\_parameter\_seed* có độ dài là *seedlen* bit, trong đó  $seedlen \geq N$ .

Qui trình này sẽ trả về cặp 2 số nguyên  $p$  và  $q$  có xác suất là nguyên tố rất cao. Để giúp cho người xác nhận có thể xác nhận được chính xác chúng thì giá trị của tham số *domain\_parameter\_seed* và *counter* sử dụng trong quá trình tạo  $p, q$  sẽ được trả về trong đó *domain\_parameter\_seed* và *counter* không cần thiết phải giữ bí mật. Đặt *Hash()* là hàm băm được lựa chọn phù hợp với cặp  $(L, N)$  và *outlen* là chiều dài đầu ra của hàm băm trong đó  $outlen \geq N$ .

Qui trình tạo  $p, q$  như sau:

Đầu vào:

1.  $L$  Chiều dài mong muốn của số nguyên tố  $p$ .
2.  $N$  Chiều dài mong muốn của số nguyên tố  $q$ .
3. *seedlen* Chiều dài mong muốn của tham số gốc trong đó  $seedlen \geq N$ .

Đầu ra:

1. *status* Trạng thái trả về của hàm tạo, trong đó *status* có thể là VALID hoặc INVALID. Nếu *status* = INVALID được trả về thì có nghĩa là không có giá trị nào của

các tham số đầu ra được trả về hoặc là giá trị của chúng không hợp lệ.

- |  |   |
|--|---|
| 2. $p, q$                                  | Các số nguyên tố tạo được $p, q$ .                          |
| 3. <i>domain_parameter_seed</i> (tùy chọn) | Là một giá trị khởi tạo được sử dụng để tạo ra $p$ và $q$ . |
| 4. <i>counter</i> : Bộ đếm (tùy chọn)      | Là giá trị đếm được tạo ra trong quá trình tạo $p, q$ .     |

Qui trình:

1. Kiểm tra cặp  $(L, N)$  có thuộc danh sách các cặp  $(L, N)$  được chấp nhận không. Nếu không thuộc thì trả về giá trị INVALID.

2. If ( $seedlen < N$ ), then return INVALID.

3.  $n = L / outlen - 1$ .

4.  $b = L - 1 - (n * outlen)$ .

5. Gán một chuỗi bất kì có độ dài  $seedlen$  bit cho tham số gốc.

6.  $U = \text{Hash}(\text{domain\_parameter\_seed}) \bmod 2^N$ .

7.  $q = U \vee 2^{N-1} \vee 1$ .

/\* Mục đích của bước này là gán bit đầu tiên (bit thứ  $N$ ) và bit cuối cùng (bit 0) của  $U$  thành 1 \*/

8.  $p$  is prime?

/\* Kiểm tra xem  $p$  có là nguyên tố không bằng cách sử dụng thuật toán kiểm tra tính nguyên tố mạnh hay không. \*/

9. If  $q$  is not a prime, then go to step 5.

/\* Nếu  $q$  không là nguyên tố thì quay lại bước 5 để gán cho tham số gốc một giá trị mới nhằm thu được một số  $q$  mới có thể là nguyên tố \*/

10.  $offset = 1$ . /\*tạo được  $q$  là nguyên tố rồi, bắt đầu tạo  $p$ \*/

11. For  $counter = 0$  to  $4095$  do /\*thử tới đa 4095 lần\*/

11.1 For  $j = 0$  to  $n$  do

$$V_j = \text{Hash}((domain\_parameter\_seed + offset + j) \bmod 2^{seedlen}).$$

$$11.2 \quad W = V_0 + (V_1 * 2^{outlen}) + \dots + (V_{n-1} * 2^{(n-1) * outlen}) + ((V_n \bmod 2^b) * 2^{n * outlen}).$$

/\* Mục đích của bước này là tạo ra một số  $W$  gồm đúng  $L$  bit.  $W$  có dạng như sau:  $W = V_n' V_{n-1} V_{n-2} \dots V_1 V_0$  trong đó  $V_n' = V_n \bmod 2^b$  có độ dài là  $b$  bit,  $V_{n-1}, \dots, V_0$  đều có độ dài là  $outlen$  bit.

Có thể hiểu là ta ghép liên tiếp các chuỗi  $V_0, \dots, V_{n-1}, V_n'$  để tạo ra  $W$  \*/

$$11.3 \quad X = W + 2^{L-1}.$$

/\* Vì  $W$  được tạo như trên nên chắc chắn  $W$  có độ dài là  $L$  bit, do đó số  $W$  sẽ thỏa mãn  $0 \leq W < 2^{L-1}$  nên ta có  $2^{L-1} \leq X < 2^L$

Mục đích của bước này là bước đầu tạo ra số  $L$  bit thỏa mãn điều kiện nằm trong khoảng  $[2^{L-1}, 2^L - 1]$  giống như  $p$  \*/

$$11.4 \quad c = X \bmod 2^q.$$

$$11.5 \quad p = X - (c - 1).$$

/\* Cách tạo  $c$  và  $p$  như trên dẫn tới  $p \equiv 1 \pmod{2^q}$ . Tức là ta có  $p-1$  chia hết cho  $2^q$  hay  $p-1$  chia hết cho  $q$ . Đây chính là một mối quan hệ giữa hai số nguyên tố  $p$  và  $q$ . \*/

11.6 If  $(p < 2^{L-1})$ , then go to step 11.9.

/\*nếu  $p$  không thỏa điều kiện ràng buộc là  $p \geq 2^{L-1}$  thì phải quay lại bước 11.9 để thay đổi giá trị của  $offset$  nhằm tạo ra một giá trị  $W$  mới ở bước

lặp tiếp theo nhằm mong muốn tạo ra được số một số  $p$  mới là nguyên tố thỏa mãn các điều kiện ràng buộc \*/

11.7 Sử dụng thuật toán kiểm tra tính nguyên tố mạnh để kiểm tra  $q$ .

11.8 Nếu  $p$  là nguyên tố thì trả về giá trị VALID và giá trị của  $p$ ,  $q$ , *domain\_parameter\_seed* và *counter*.

11.9  $offset = offset + n + 1$ .

/\* Khi  $p$  không là nguyên tố hoặc  $p$  không thỏa mãn điều kiện ràng buộc của nó thì thay đổi *offset* và bắt đầu vòng lặp mới từ 11.1 đến 11.8 nếu bộ đếm vẫn còn nhỏ hơn 4096 \*/

12. Go to step 5.

/\* Sau 4096 bước thử mà vẫn không tìm được  $p$  nào thì quay lại bước 5 để tạo lại từ đầu một số  $q$  nguyên tố mới và sau đó là tạo  $p$  \*/

### **Kiểm tra số nguyên tố $p$ và $q$ bằng việc sử dụng hàm băm**

Thuật toán kiểm chứng này được sử dụng để kiểm chứng các số nguyên  $p$ ,  $q$  được tạo ra từ thuật toán tạo các số nguyên tố nêu trong mục trên. Đầu vào của thuật toán là các giá trị  $p$ ,  $q$  cần kiểm chứng, *domain\_parameter\_seed* và *counter*. Hàm băm sử dụng chính là hàm băm đã dùng để tạo ra  $p$ ,  $q$  và đặt *outlen* là kích thước khối đầu ra.

Đầu vào:

- |                                 |   |
|---------------------------------|---|
| 1. $p$ , $q$                    | Là 2 số cần kiểm chứng tính nguyên tố.                  |
| 3. <i>domain_parameter_seed</i> | Là tham số gốc đã được dùng để tạo ra $p$ và $q$ .      |
| 4. <i>counter</i>               | Là bộ đếm được xác định trong quá trình tạo $p$ , $q$ . |

Đầu ra:

*status*           Trạng thái trả về của hàm trong đó nó có thể nhận 1 trong 2 giá trị là VALID hoặc INVALID.

Qui trình:

1.  $L = \text{len}(p)$ .

2.  $N = \text{len}(q)$ .

3. Kiểm tra cặp  $(L, N)$  có thuộc danh sách các cặp  $(L, N)$  được chấp nhận không. Nếu không thuộc thì trả về giá trị INVALID.

4. If  $(\text{counter} > 4095)$ , then return INVALID.

/\* Vì giá trị của bộ đếm counter sử dụng trong hàm tạo  $p, q$  là nhỏ hơn hoặc bằng 4095. \*/

5.  $\text{seedlen} = \text{len}(\text{domain\_parameter\_seed})$ .

6. If  $(\text{seedlen} < (N))$ , then return INVALID.

7.  $U = \text{Hash}(\text{domain\_parameter\_seed}) \bmod 2^N$ .

/\* Tạo  $U$  theo công thức đã sử dụng trong giải thuật tạo  $p, q$ . \*/

8.  $\text{computed\_q} = U \vee 2^{N-1} \vee 1$ .

/\* Tạo ra  $\text{computed\_q}$  theo công thức đã dùng để tạo  $q$ . \*/

9. If  $(\text{computed\_q} \neq q)$  or  $(\text{computed\_q}$  không là nguyên tố), then return INVALID.

/\* Sử dụng hàm kiểm tra tính nguyên tố mạnh để kiểm tra xem  $\text{computed\_q}$  có là nguyên tố không. Nếu  $\text{computed\_q}$  không phải là nguyên tố hoặc  $\text{computed\_q} \neq q$  thì chứng tỏ đã hoặc  $q$  không phải là nguyên tố hoặc dữ liệu đã có lỗi. \*/

10.  $n = L / \text{outlen} - 1$ .

11.  $b = L - 1 - (n * outlen)$ .

12.  $offset = 1$ .

13. For  $i = 0$  to  $counter$  do

/\* Vòng lặp for giống hệt trong giải thuật tạo. \*/

13.1 For  $j = 0$  to  $n$  do

$V_j = \text{Hash}((domain\_parameter\_seed + offset + j) \bmod 2^{seedlen})$ .

13.2  $W = V_0 + (V_1 * 2^{outlen}) + \dots + (V_{n-1} * 2^{(n-1) * outlen}) + ((V_n \bmod 2^b) * 2^{n*outlen})$ .

13.3  $X = W + 2^{L-1}$ .

13.4  $c = X \bmod 2^q$ .

13.5  $computed\_p = X - (c - 1)$ .

13.6 If  $(computed\_p < 2^{L-1})$ , then go to step 13.9

13.7 Kiểm tra  $computed\_p$  có là nguyên tố không.

13.8 If  $computed\_p$  là nguyên tố, then go to step 15.

13.9  $offset = offset + n + 1$ .

14. If  $((i \neq counter) \text{ or } (computed\_p \neq p) \text{ or } (computed\_p \text{ is not a prime}))$ , then return INVALID.

15. Return VALID.

### Tạo số $g$

Số  $g$  được tạo ra dựa trên các giá trị của  $p$ ,  $q$  và tham số  $domain\_parameter\_seed$  trả về từ thủ tục tạo  $p$ ,  $q$  tương ứng. Số  $g$  ở đây có thể kiểm chứng được bằng thủ tục. Phương pháp này có thể giúp tạo ra nhiều giá trị  $g$  cho cùng một cặp  $(p, q)$  xác định. Việc sử dụng các giá trị khác nhau

của  $g$  có thể hỗ trợ cho việc phân biệt khóa. Ví dụ, ta sử dụng số  $g$  được tạo ra với  $index = 1$  cho chữ ký số và với  $index = 2$  cho việc thiết lập khóa.

Đặt  $Hash()$  là hàm băm được lựa chọn cho cặp  $(L, N)$ . Qui trình tạo  $g$  sẽ như sau:

Đầu vào:

- |                              |  |
|------------------------------|--|
| 1. $p, q$                    | Là các số nguyên tố.   |
| 2. $domain\_parameter\_seed$ | Là tham số gốc được sử dụng trong thủ tục tạo $p, q$ .                                       |
| 3. $index$                   | Chỉ số được sử dụng để tạo $g$ . Chỉ số $index$ biểu diễn bởi một số nguyên 8 bit không dấu. |

Đầu ra:

- |             |  |
|-------------|--|
| 1. $status$ | Trạng thái trả về của hàm tạo, trong đó $status$ có thể nhận 1 trong 2 giá trị VALID và INVALID. |
| 2. $g$      | Giá trị của số $g$ tạo được.   |

Qui trình:

1. If ( $index$  is incorrect) then return INVALID.
2.  $N = \text{len}(q)$ ;
3.  $e = (p-1) / q$ ;
4.  $count = 0$ ;
5.  $count = count + 1$ ;
6. If  $count = 0$  then return INVALID.
7.  $U = domain\_parameter\_seed || \text{"ggen"} || index || count$ .
8.  $W = Hash(U)$ .
9.  $g = W^e \bmod p$ .



/\* Ta có:  $g = W^e \bmod p$

$$\Leftrightarrow g = \text{Hash}(U)^{(p-1)/q}$$

$$\Leftrightarrow g^q = \text{Hash}(U)^{p-1} \quad (1)$$

Mà  $0 < \text{Hash}(U) < p$  và  $p$  nguyên tố  $\Rightarrow (\text{Hash}(U), p) = 1$ .

$$\text{Theo Ferma, ta có được } (\text{Hash}(U))^{p-1} \equiv 1 \pmod{p} \quad (2)$$

Từ (1) và (2) ta có  $g^q \equiv 1 \bmod p$

Như vậy số  $g$  thỏa mãn tính chất là căn bậc  $q$  của 1 theo modulo  $p$ .

\*/

10. If  $(g < 2)$  then goto step 5.

11. Return VALID and the value of  $g$ .

### Kiểm tra số $g$

Thuật toán này được dùng để kiểm chứng giá trị  $g$  với  $g$  được tạo ra bởi thủ tục tạo  $g$  dựa trên các giá trị  $p, q$ , tham số *domain\_parameter\_seed* và chỉ số *index* thích hợp. Giả thiết rằng  $p$  và  $q$  đã được kiểm chứng trước đó rồi.

Các tham số gốc sẽ được lấy từ đầu ra của thủ tục tạo  $p, q$ . Đặt Hash() là hàm băm được chọn phù hợp với cặp  $(L, N)$ . Qui trình kiểm chứng sẽ như sau:

Đầu vào:

- |                                 |   |
|---------------------------------|---|
| 1. $p, q$                       | Các số nguyên tố.   |
| 2. <i>domain_parameter_seed</i> | Tham số gốc được dùng để tạo $p$ và $q$ .   |
| 3. <i>index</i>                 | Chỉ số được dùng để tạo ra số $g$ trong đó <i>index</i> được biểu diễn bởi số nguyên không dấu 8 bit. |

Đầu ra:

*status*

Trạng thái trả về của thủ tục, trong đó *status* nhận 1 trong 2 giá trị là VALID và INVALID.

Qui trình:

1. If (*index* is incorrect) then return INVALID.
2. If not ( $2 \leq g \leq (p-1)$ ) then return INVALID.
3. If ( $g^q \neq 1 \bmod p$ ) then return INVALID.
4.  $N = \text{len}(q)$ .
5.  $e = (p-1) / q$ .
6.  $count = 0$ .
7.  $count = count + 1$ .
8. If  $count = 0$  then return INVALID.
9.  $U = \text{domain\_parameter\_seed} \parallel \text{"ggen"} \parallel index \parallel count$ .
10.  $W = \text{Hash}(U)$ .
11.  $computed\_g = W^e \bmod p$ .
12. If ( $computed\_g < 2$ ) then goto step 7.
13. If ( $computed\_g = g$ ) then return VALID, else return INVALID.

### 3.2.3. Chức năng tạo khoá

Thuật toán DSA đòi hỏi cặp khóa chung và riêng sử dụng cho quá trình sinh và xác minh chữ ký số được sinh ra từ một tập các tham số riêng. Những tham số này có thể đại diện cho một nhóm người sử dụng và có thể là công khai. Một người sử dụng của một tập tham số sẽ phải có sự đảm bảo khi muốn sử dụng chúng. Tập tham số này có thể được sử dụng trong khoảng thời gian cố định.

#### Tạo khóa chung

- Tạo 1 số nguyên tố  $p$  đủ lớn chiều dài từ 512 - 1024 bit, là bội số chiều dài và độ lớn của 64.

- Số nguyên tố  $q$  sao cho  $q$  chia hết  $(p-1)$ , chiều dài 160 bit.
- Số nguyên  $g = h^{(p-1)/q} \bmod p$ . Trong đó:  $1 < h < (p-1)$  và  $g > 1$ .

#### **Tạo khóa bí mật $x$**

- Khóa bí mật là giá trị  $x$  sao cho:  $0 < x < q$ .

#### **Tạo khóa công khai $y$**

- Khóa công khai là số nguyên  $y$  thỏa mãn:  $y = g^x \bmod p$ .

#### **3.2.4. Chức năng tạo và thẩm định chữ ký số**

Các thành phần cần thiết của một hệ thống tạo và thẩm định chữ ký số là các thuật toán khoá công cộng và thuật toán băm mà sẽ được lựa chọn một cách cẩn thận theo yêu cầu. Số liệu gốc trước tiên sẽ được làm mới lại bằng cách sử dụng hàm một chiều Hash, sau đó được mật mã hoá bằng khoá bí mật của người gửi. Mục đích của thủ tục này là giảm thời gian mật mã hoá do các thuật toán không đối xứng thực thi chậm hơn nhiều các thuật toán đối xứng. Cả số liệu gốc lẫn chữ ký đều được gửi qua kênh thông tin không an toàn đến người nhận.

Để kiểm tra tính toàn vẹn của số liệu, người nhận rút khoá công cộng của người gửi từ chứng nhận điện tử của nó và sử dụng khoá công cộng này để giải mật mã chữ ký điện tử. Sau đó áp dụng hàm băm mà đã được người gửi sử dụng cho số liệu gốc nhận được. Các tóm tắt bản tin thu được sẽ được so sánh với nhau để thẩm định rằng bản tin không bị sửa đổi trong quá trình truyền dẫn và nó thực sự được gửi từ người gửi mong đợi. Chú ý rằng người nhận phải lấy chứng nhận số của người gửi từ một server chứng nhận, đây chính là bước kiểm tra giá trị chứng nhận.

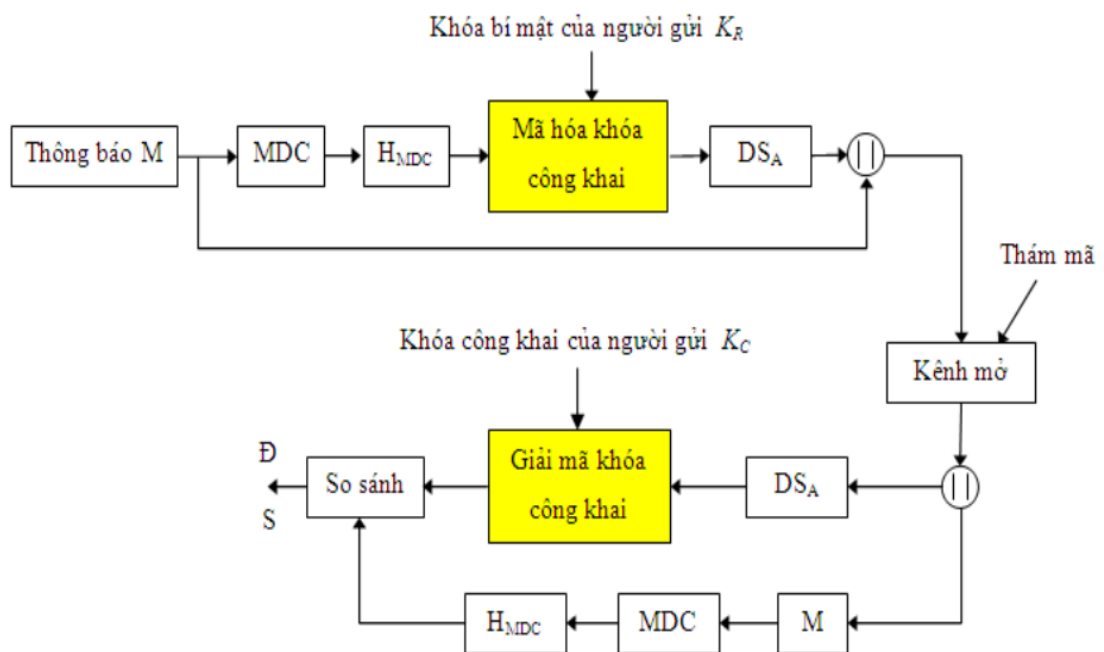
Quá trình ký:

- Băm thông báo  $M$  để có mã băm:  $H_{MDC} = H(M)$ .
- Mã hóa mã băm  $H_{MDC}$  bằng hệ mật DSA với khóa bí mật  $A(d_A)$ :  $DS_A = H(M)^{d_A} \bmod n_A$  được chữ ký số của  $DS_A$ .

- Ghép chữ ký số  $DS_A$  với thông báo:  $M // DS_A$  và truyền đi.

Quá trình thẩm định:

- Tách thông báo  $M$  và chữ ký số  $DS_A$ .
- Giải mã  $DS_A$  chữ ký số bằng khóa công khai của A( $e_A$ ) để thu được mã băm của bên phát.
- Tiến hành băm  $M$  để tạo mã băm độc lập:  $H_{MDC}$  và so sánh với mã băm bên phát.



Hình 3.2: Quá trình tạo chữ ký số và kiểm tra chữ ký số dùng DSA

### 3.3. Thiết kế giao diện


- Form giao diện chính



Hình 3.3: Form giao diện chính

- Form tạo hóa đơn

Chi tiết hóa đơn tiền điện

 **EVNHANOI** Mã hóa đơn : A10000

**Công ty Điện lực Long Biên**  
**THÔNG BÁO**  
**THANH TOÁN TIỀN ĐIỆN**

Từ ngày 22/04/2015 Đến ngày 22/05/2015

Thông tin khách hàng

Tên khách hàng : Nguyễn Văn An

Địa chỉ : Số 10, tổ 1- Phúc Lợi- LB- HN

Mã khách hàng : A100001

Số công tơ : A100005

	ĐNTT	Đơn giá	Thành tiền
	100	1002	100200
	50	1450	72500
	50	1720	86000
▶	100	2150	215000
*			

Tổng 300 473700



Thuế GTGT 10 (%) 47370.0

Tổng tiền 521070.0

Địa điểm : Long Biên Ngày lập hóa đơn 22/05/2015

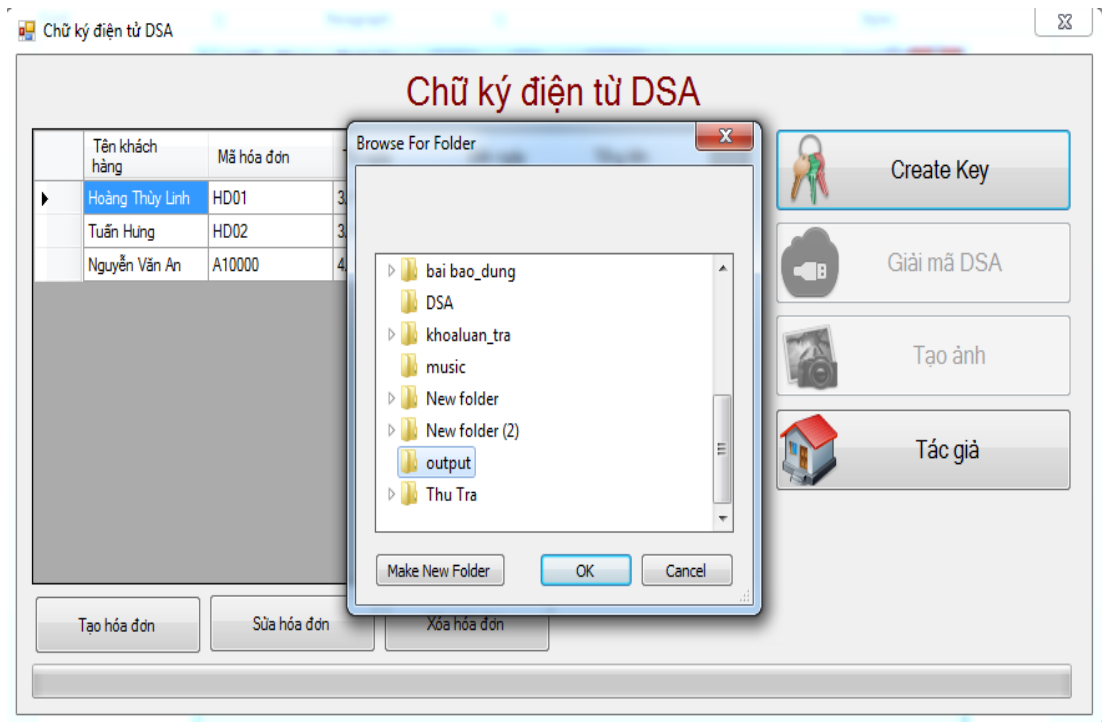
Giám đốc  
Điện thoại: 22154216

Lưu ý: Nếu có vấn đề cần giải đáp, vui lòng liên hệ  
Trung tâm hỗ trợ khách hàng (Tổng đài: 04.22222000)

 **Lưu hóa đơn**  **Hủy bỏ**

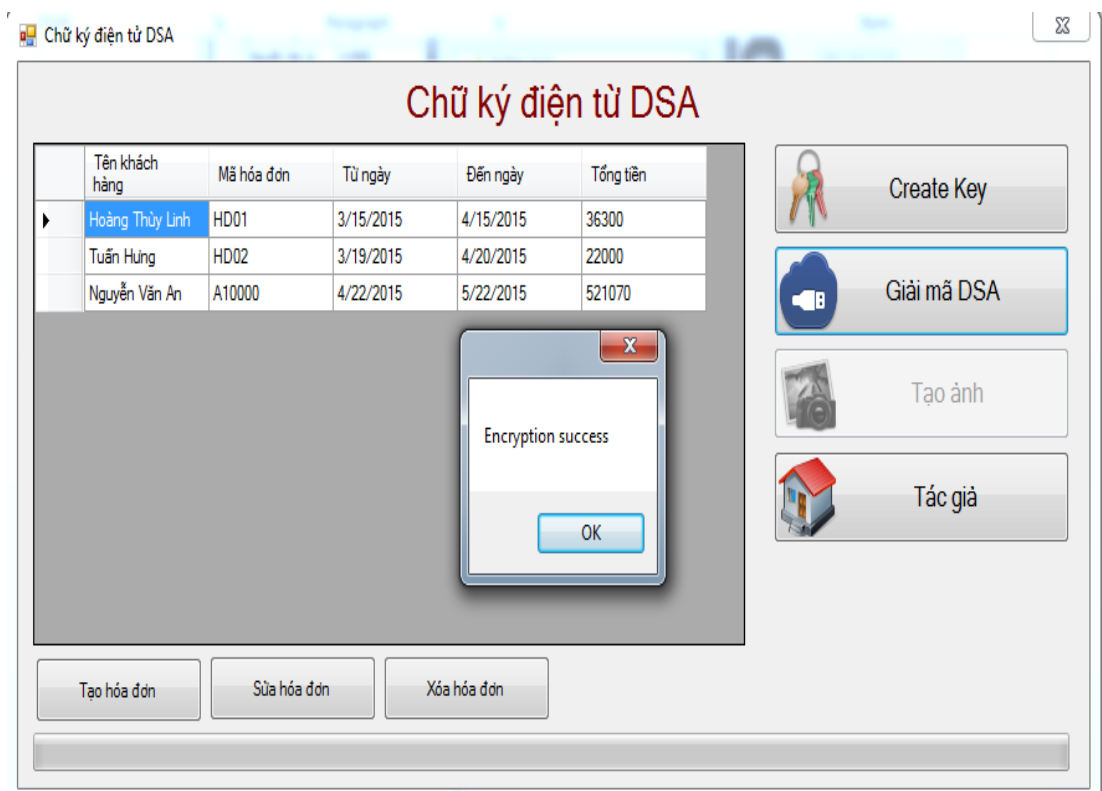
Hình 3.4: Form tạo hóa đơn

- Form tạo khóa



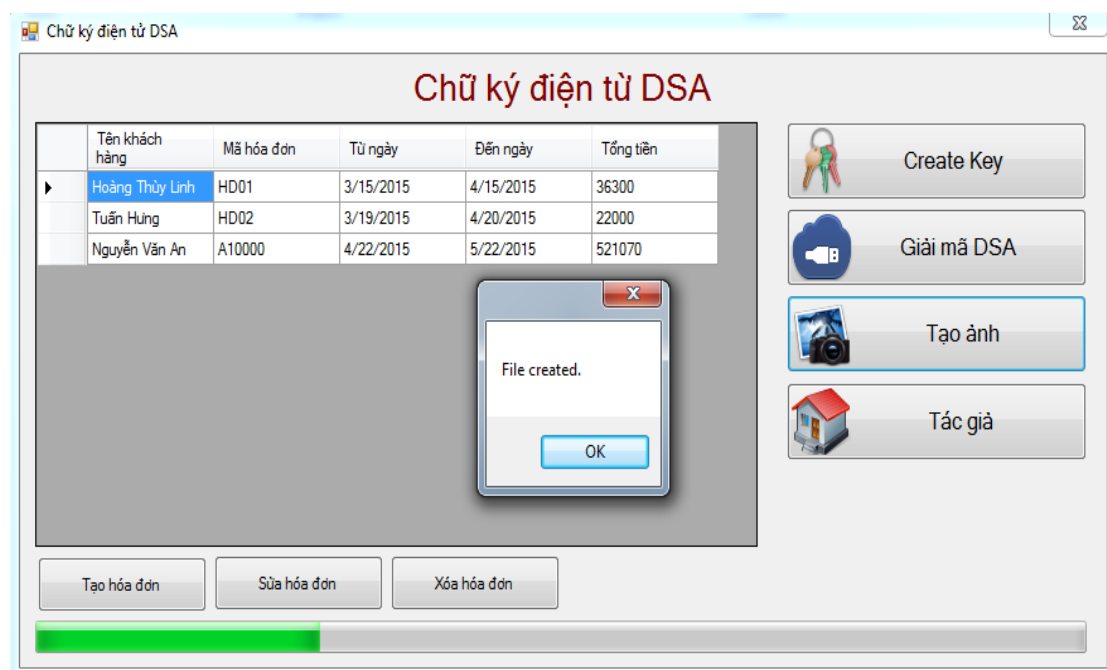
Hình 3.5: Form tạo khóa

- Form giải mã thành công



Hình 3.6: Form giải mã thành công

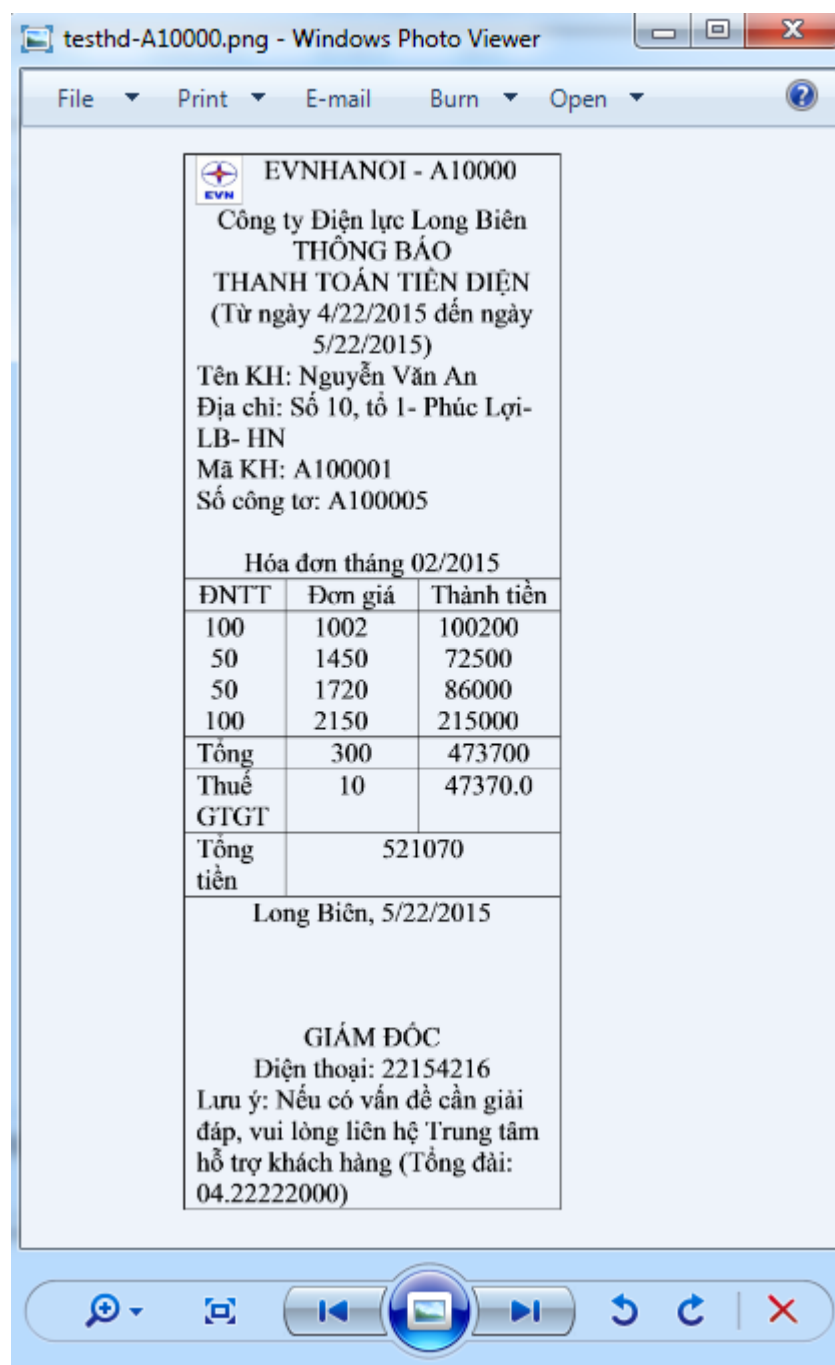
- Form tạo file ảnh JPG



Hình 3.7: Form tạo file ảnh JPG

- Form kết quả nhận được





Hình 3.8: Form kết quả nhận được

## KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 1. Kết quả đạt được

Hiện nay, hình thức sử dụng hóa đơn điện tử đã khá phổ biến và sẽ tiếp tục là mục tiêu hướng đến của các doanh nghiệp. Muốn hóa đơn điện tử phát triển thì buộc phải sử dụng chữ ký số. Với vai trò quan trọng trong việc đảm bảo an toàn thông tin cho các giao dịch điện tử, chữ ký số là phương tiện hữu hiệu để các doanh nghiệp tăng tính cạnh tranh trong thương mại điện tử nhất là trong bối cảnh hội nhập kinh tế thế giới như hiện nay.

Việc nghiên cứu và tìm hiểu về chữ ký số để đáp ứng nhu cầu xác thực thông tin và người dùng là rất cần thiết đặc biệt là trong các giao dịch điện tử.

Khóa luận đã nghiên cứu về việc ký số hóa đơn điện tử và đạt được những kết quả chính sau:

- Đặc điểm, quá trình xử lý và tính toán của các hàm băm an toàn: SHA-1, SHA-256, SHA-512, SHA-384.
- Những khái niệm, đặc điểm cơ bản của một hệ thống chữ ký số.
- Tư tưởng của thuật toán cấp phát khóa, sinh và kiểm tra chữ ký số DSA.
- Xây dựng một ứng dụng chữ ký số trong hóa đơn điện tử tiền điện nhằm phục vụ cho nhu cầu lưu trữ, tra cứu, tiết giảm chi phí in ấn.

### 2. Hướng phát triển

Ngoài những kết quả đã đạt được, khóa luận vẫn còn những nhược điểm cần khắc phục là:

- + Hóa đơn theo định dạng JPG nên dữ liệu khá lớn.
- + Quá trình xử lý dữ liệu còn chậm.

Do vậy, hướng phát triển của khóa luận là đi sâu nghiên cứu thêm về lý thuyết hàm băm và tìm hiểu các thuật toán tối ưu hơn trong việc ký số hóa đơn.

Về mặt triển khai thực hiện, khóa luận đã cài đặt chương trình tạo chữ ký số trên hóa đơn tiền điện, bước đầu đã thu được một số kết quả nhất định. Tuy nhiên do đặc thù của giải thuật là làm việc với dữ liệu có kích thước lớn (cặp số nguyên tố  $p, q$  có độ dài tối thiểu là 1024 và 160 bit kéo theo các phép toán làm việc với chúng sẽ mất nhiều thời gian, đặc biệt là phép tính mũ môđun) và số vòng lặp thử được dùng trong giải thuật khá nhiều khiến cho thời gian tạo chữ ký số bị tăng lên rất nhiều. Như vậy chi phí về mặt thời gian là khó khăn chính khi triển khai cài đặt chương trình. Vì vậy vấn đề đặt ra là làm sao để cải thiện được hiệu năng tính toán, giảm thiểu thời gian cũng như phải có sự đầu tư về mặt cơ sở hạ tầng, nâng cấp thiết bị mà hơn hết là tiến hành thực hiện chương trình trên các máy tính chuyên dụng có cấu hình mạnh. Nếu điều kiện cho phép đây sẽ là những mục tiêu hướng tới trong tương lai của chương trình.

Trong quá trình nghiên cứu, thực hiện khóa luận mặc dù đã cố gắng tập trung nghiên cứu và tham khảo nhiều tài liệu, báo cáo, tạp chí khoa học, nhưng do trình độ còn nhiều giới hạn nên khóa luận không thể tránh khỏi thiếu sót và hạn chế. Em rất mong được sự chỉ bảo đóng góp nhiều hơn nữa của các thầy, cô giáo và các bạn.

Một lần nữa, em xin chân thành gửi lời cảm ơn sự giúp đỡ của các thầy, cô giáo khoa Công nghệ Thông tin, đặc biệt là TS. Lưu Thị Bích Hương đã giúp đỡ em trong quá trình làm khóa luận tốt nghiệp này.

## TÀI LIỆU THAM KHẢO

### TIẾNG VIỆT:

1. Phan Đình Diệu (1999), *Lý thuyết mật mã và an toàn thông tin*, Đại học Quốc gia Hà Nội.
2. TS. Dương Anh Đức - ThS. Trần Minh Triết (2005), *Mã hóa và ứng dụng*, Khoa Công nghệ Thông tin, Trường Đại học Khoa học Tự nhiên, Đại học Quốc gia thành phố Hồ Chí Minh.
3. PGS.TS Hồ Thuần (2000), Giáo trình “*Lý thuyết mật mã và an toàn dữ liệu*”, Trường Đại học Bách Khoa Hà Nội.

### TIẾNG ANH:

4. Mohan Atreya, Ben Hammond, Stephen Paine, Paul Starrett, Stephen Wu (2002), *Digital Signatures*, RSA.
5. *Federal Information Processing Standards Publication 180-2 Specifications for the SECURE HASH STANDARD*, 2002.
6. R.Rivest (1992), *The MD5 Message Digest Algorithm*, MIT Laboratory for Computer Science and RSA DataSecurity, Inc.
7. FIPS (2004), *Announcing the Secure Hash Standard*.