NGHIÊN CỨU PHÁT TRIỂN CHỮ KÝ ĐIỆN TỬ DỰA TRÊN HỆ MẬT MÃ HÓA KHÓA CÔNG KHAI RSA

TS. Lê Xuân Trường[1]
Nguyễn Văn Trường[2]

TÓM TẮT

Chữ ký số là một hình thức được dùng để định danh người dùng trong các giao dịch điện tử hiện nay. Quy trình xây dựng chữ ký số thường dựa vào các thuật toán DSA, mà phổ biến nhất là hệ mật mã khóa công khai RSA. Độ bảo mật của RSA dựa vào độ khó khi phân tích một số nguyên dương lớn thành tích các thừa số nguyên tố và thường được gắn liền với các phép tính lũy thừa modulo với các số nguyên lớn từ 1024 đến 2048 bit. Trong bài báo này sẽ giới thiệu phương pháp thao tác tính toán với số nguyên lớn, nguyên tắc chọn e, d, n để giảm thiểu thời gian thực thi. Bên cạnh đó, Bài báo mô tả mô hình, quy trình hoạt động của 2 chuẩn chữ ký số PKCS#1v1.5 và RSA-PSS, giới thiệu công nghệ Sharepoint để quản lý thu thập chữ ký số.

Từ khóa: Mã hóa, thuật toán RSA, Montgomery, lũy thừa modulo, PKCS#1v1.5, RSA-PSS, thu thập chữ ký số trong sharepoint.

ABSTRACT

Nowadays in electronic transactions, the digital signature is a form to identify users. Construction processes of digital signature are usually based on algorithm DSA, of which the most popular is the RSA public key cryptography. The security of RSA bases on the difficulty when analyzing a large positive integer which generates prime factors and normally associates with the modular power exponentiation with large integer numbers from 1024 to 2048 bits. This paper will introduce methods to calculate large integer, the principle of selecting e, d, n to minimize execution time. Besides, the paper describes the model, the process of operation of 2 digital signature standard PKCS # 1v1.5 and RSA-PSS, as well as the Sharepoint technologies to manage digital signature collection.

Keywords: Crytography, RSA Algorithm, Montgomery, modular exponentiation, PKCS#1v1.5, RSA-PSS, Collect Signature SharePoint.

1. GIỚI THIỆU

Cùng với sự phát triển nhanh chóng của công nghệ thông tin, hầu hết các giao dịch, các văn bản tài liệu được lưu trữ và truyền qua mạng. Nếu ta sử dụng hình thức chữ ký truyền thống như trên sẽ nảy sinh ra nhiều vấn đề: có thể có những thông tin gian lận xuất phát từ bên gửi hay bên nhận, thông tin bị sửa đổi trong quá trình truyền đi, người gửi phủ nhận những thông tin gây bất lợi cho mình, thông tin bị rò rỉ,... Do đó đặt ra yêu cầu là thông tin được truyền toàn vẹn, và xác minh được thông tin người gửi. Một trong những giải pháp được ứng dụng để giải quyết vấn đề này là chữ ký điện tử mà hình thức phổ biến của nó là chữ ký số.

Chữ ký số cho phép ký tài liệu, văn bản và gồm tất cả các tính năng của chữ ký tay:

[1] *Phó trưởng khoa, Khoa Công nghệ thông tin, Trường Đại học Mở TP.HCM.*
[2] *Sinh viên Khoa Công nghệ thông tin, Trường Đại học Mở TP.HCM.*

- Khả năng xác minh
- Chống giả mạo
- Không thể sử dụng lại
- Không thể thay đổi
- Chống chối bỏ

Phương pháp nghiên cứu:

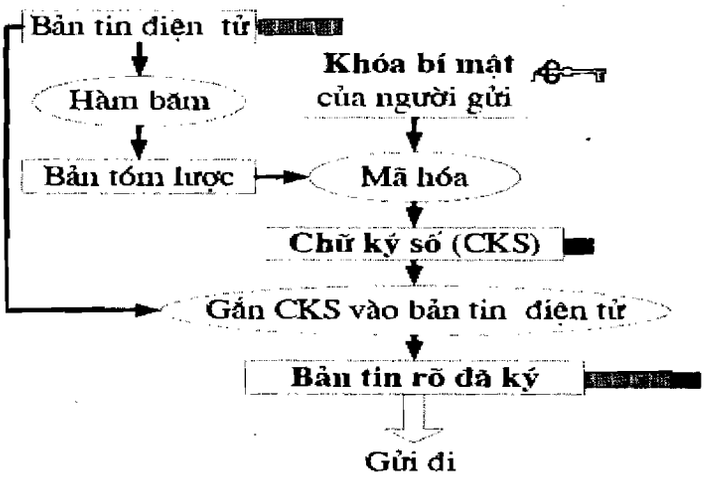
- Tìm hiểu mô hình hoạt động chữ ký số: phương pháp tạo, cách thức chứng thực, quản lý chữ ký số trong thực tế,...
- Tìm hiểu và nghiên cứu các mô hình, chuẩn chữ ký số ứng dụng, tích hợp trong các ứng dụng chứng thực: PKCS#1v1.5, RSA-PSS.
- Nghiên cứu thuật giải RSA: quy trình hoạt động, phương pháp tạo khóa, độ bảo mật và ứng dụng RSA với chữ ký số,...
- Nghiên cứu phát triển thuật toán RSA với ứng dụng số nguyên lớn.
- Tìm kiếm và phân tích các thuật giải gắn liền với RSA khi tạo chữ ký số: MD5, SHA, Miller-Rabin, Euclid, Euclid mở rộng, Fermat, Chinese-Remainder, modular exponentiation,...
- Tìm hiểu phương pháp tạo và xác nhận chữ ký trong các loại file: word, excel, pdf, open office,...
- Nghiên cứu xây dựng ứng dụng giả lập chứng chỉ số với thuật toán tính số nguyên lớn.

2. CƠ SỞ LÝ THUYẾT

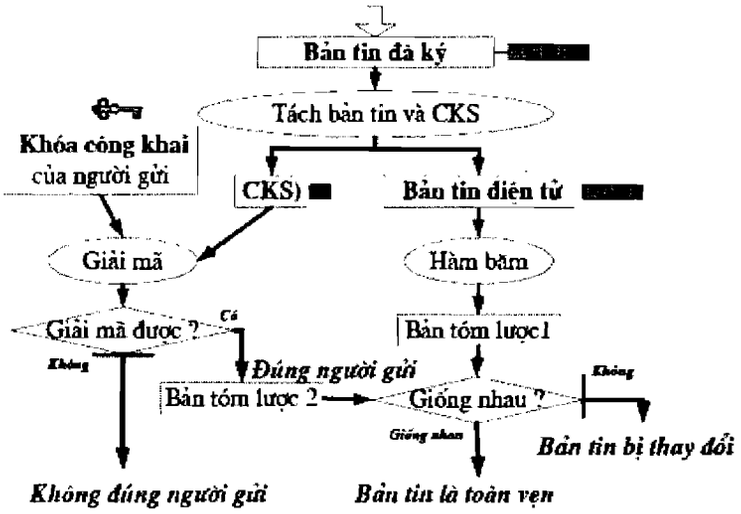
2.1. Chữ ký số

Chữ ký số (Digital signature) là một tập con của chữ ký điện tử, là mô hình sử dụng các kỹ thuật mật mã để gắn với mỗi người sử dụng một cặp khóa công khai - bí mật và qua đó có thể ký các văn bản điện tử cũng như trao đổi các thông tin mật. Khóa công khai thường được phân phối thông qua chứng thực khóa công khai.

Hình 2.1. Sơ đồ cách tạo chữ ký điện tử



Hình 2.2. Sơ đồ xác nhận chữ ký số



Phương thức hoạt động của chữ ký số bao gồm 3 thuật toán sau:

- Thuật toán tạo khóa: khởi tạo một cặp bao gồm khóa công khai và khóa bí mật
- Thuật toán ký số: sử dụng thuật toán băm để băm tài liệu sẽ ký điện tử thành tài liệu rút gọn (message digest), sau đó dùng khóa bí mật để ký điện tử lên tài liệu.
- Thuật toán kiểm tra chữ ký số: khóa công khai của người ký được dùng để giải mã chữ ký thành một thông điệp, giá trị này được so sánh với giá trị băm của tài liệu để xác định tính hợp lệ của chữ ký.

2.2. Hàm băm

Định nghĩa: Hàm băm H nhận đầu vào là một thông tin X và đầu ra là một chuỗi có độ dài cố định. Chuỗi này được gọi là bản thông tin tóm lược (message digest) của thông tin X, và kích thước của chuỗi tùy thuộc vào hàm băm sử dụng

Tính chất của hàm băm:

- Đầu vào là một chuỗi bất kỳ và sinh ra là một chuỗi có độ dài cố định .
- Là hàm một chiều.
- Có tính phi đụng độ cao: tức là rất khó để tìm ra $H(X) = H(X')$ với $X \neq X'$.

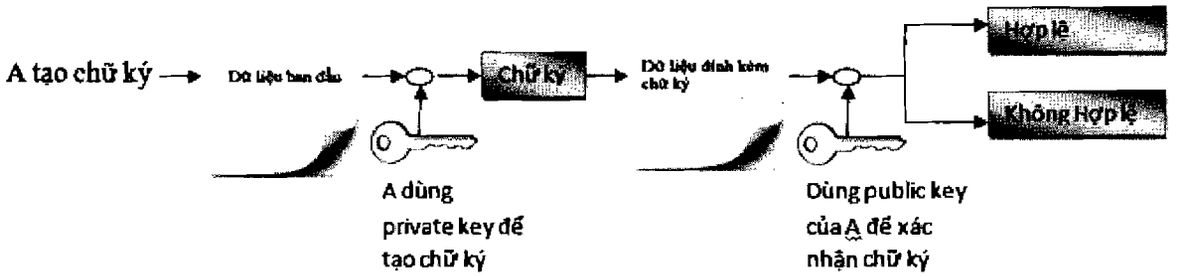
Một số loại hàm băm: MD5(128 bits), SHA1(160 bits), SHA2

- MD5("abcd") = E2FC714C4727EE9 395F324CD2E7F331F
- SHA1("abcd")=81FE8BFE87576C3 ECB22426F8E57847382917ACF

Hiện nay, việc xác định loại hàm băm phù hợp kết hợp với RSA cũng vô cùng quan trọng. Đã từ lâu MD5 đã không còn được phát triển trong các ứng dụng chữ ký số. Trong khi đó SHA1 là dạng hàm băm được sử dụng rộng rãi nhất hiện nay, được hỗ trợ bởi hầu hết các ứng dụng: Microsoft Office 2007, Adobe Acrobat professional,...

2.3. Thuật toán RSA

Hình 2.3. Quy trình hoạt động trong RSA



Sơ lược về RSA:

- Chọn 2 số nguyên tố p và q sao cho kích thước tính theo bits gần tương đương với n = p*q (|n| là chiều dài của giá trị khóa cần tạo), p≠q.
- Tính φ(n) = (p-1)(q-1).
- Chọn e, sao cho 1 < e < φ(n), và gcd(e, φ(n))=1.
- Tính toán giá trị d thỏa mãn biểu thức đồng dư: ed ≡ 1 (mod φ(n)).[2]
- Độ bảo mật của RSA là việc tính được p,q từ n. Tuy nhiên việc phân tích số n ra p,q theo phương pháp và hệ thống máy tính hiện nay có thể mất cả hàng trăm năm với n đủ lớn. Do đó độ bảo mật của thuật toán được đảm bảo.

RSA dùng trong mã hóa dữ liệu:

- Quy tắc mã hóa: c = RsaPublic(m) = m^e mod n, 1 < m < n-1.
- Quy tắc giải mã: m = RsaPrivate(c) = c^d mod n.[2]

Trong đó c: dữ liệu đã mã hóa, m: dữ liệu ban đầu

RSA dùng trong chữ ký số:

- Chữ ký: s = RsaPrivate(m) = m^d mod n, 1 < m < n-1.
- Xác nhận: v = RsaPublic(s) = s^e mod n.[2]

Trong đó s: là chữ ký, m: văn bản cần ký

3. CẢI TIẾN RSA ĐỂ ỨNG DỤNG VỚI SỐ NGUYÊN LỚN

Điều quan trọng nhất để đảm bảo độ bảo mật chữ ký số ứng dụng trong thực tế là chuỗi ký mã (chuỗi ký số định danh người dùng) phải lớn với độ dài từ hàng ngàn bit trở lên. Tuy nhiên việc mã hóa và thao tác tính toán trên số nguyên lớn với các giải thuật thông thường thì thời gian thực thi để tạo khóa bí mật và chữ ký số là rất lớn. Dưới đây là các nguyên tắc và giải thuật quan trọng để tăng tốc độ tính toán số lớn và độ bảo mật cho chữ ký số:

Nguyên tắc chọn ngẫu nhiên 2 số nguyên tố lớn p, q:

- Tạo ngẫu nhiên p đảm bảo p là số lẻ và $|p| = \frac{b}{2}$
- Tạo ngẫu nhiên q đảm bảo q là số lẻ và $|q| = b - \frac{b}{2}$

Trong đó b là chiều dài khóa cần tạo, kiểm tra tính nguyên tố của p, q dùng giải thuật Miller-Rabin.[2]

Trong hệ mật mã RSA, độ bảo mật của thuật toán là dựa trên việc tính toán số lớn với n=p*q. Việc chọn 2 số nguyên tố p, q nếu dùng giải thuật kiểm tra số nguyên tố thông thường sẽ không đáp ứng được yêu cầu về thời gian thực thi.

Giải thuật:

Mã giả:

```
while not Prime(p) do,
 p = p+2
return p
```

Trong trường hợp:

- Nếu p = q, tăng p lên 2 và thực hiện lại đoạn mã trên.
- Nếu như p<q, hoán đổi p và q (nếu sử dụng định lý số dư trung quốc)

Giải thuật Miller-Rabin trích từ Wikipedia:

Đầu vào: n > 3, k và a xác định xác suất chính xác của thuật giải Đầu ra: composite: n không là số nguyên tố, probably prime: n là số giả nguyên tố.

Mã giả:

```
Loop: repeat k times:
   pick a random integer a in the range [2, n − 2]
   x ← a^d mod n
   if x = 1 or x = n − 1 then do next Loop
   repeat s − 1 times:
      x ← x^2 mod n
      if x = 1 then return composite
      if x = n − 1 then do next Loop
   return composite
return probably prime
```

Xác suất để n nguyên tố là: $(1 - \left(\frac{1}{4}\right)^k)$; với k là số lần thực thi, [2].

Ngoài ra còn có một số giải thuật kiểm tra số nguyên tố khác như giải thuật Fermat, giải thuật Solovay–Strassen. Tuy nhiên các giải thuật này tốc độ thực thi chậm nên không còn dùng nữa.

Nguyên tắc chọn e:

Công thức nguyên thủy chọn e trong RSA: $1 < e < \varphi(n)$, và $\gcd(e, \varphi(n)) = 1$. Tuy nhiên do giá trị của $\varphi(n)$ tương đối lớn (khoảng vài ngàn bits) nên tốc độ tính toán tương đối chậm.

Để tăng tốc độ thực thi ta chọn e theo công thức Fermat: $f(x) = 2^{2^x} + 1$, được gọi là số nguyên tố Fermat. [2]

- Do $\varphi(n) = (p-1)(q-1)$ nên thay vì kiểm tra $\gcd(e, \varphi(n)) = 1$ ta kiểm tra $\gcd(e,p-1) = 1$ và $\gcd(e,q-1) = 1$. Với kích thước (p-1) và (q-1) chỉ bằng một nửa, $\varphi(n)$ nên tốc độ sẽ giảm đi đáng kể.
- Khi e > 2 để kiểm tra $\gcd(p-1,e) = 1$, ta dễ dàng sử dụng công thức (p mod e) != 1.

Giải thuật:

Đầu vào: 2 số nguyên tố p, q. Đầu ra: e thỏa mãn $1 < e < \varphi(n)$, và $\gcd(e, \varphi(n)) = 1$.

Mã giả:

set x = 0

do then

$e = 2^{2^x} + 1$

x = x + 1

if (p mod e) != 1 and (q mod e) != 1 then break

while true

return(e).

Nguyên tắc tính d:

Để tính giá trị d trong hệ mật mã RSA ed ≡ 1 (mod φ(n)) có nhiều giải thuật thường dùng nhất là hệ thức Bezout và Euclid mở rộng. Tuy nhiên hệ thức Bezout chỉ phù hợp trong tính toán bằng tay với các giá trị nhỏ. Trong khi đó giải thuật Euclid mở rộng ứng dụng trong tính toán biểu thức ax + by = gcd(a, b) bằng cách dùng thêm các giá trị phụ, lại phù hợp với việc tính toán với số lớn.[2]

Công thức tính d trong RSA: $d = e^{-1} \bmod \varphi(n)$, tương đương $de + k\varphi(n) = 1$. Vì ed ≡ 1 (mod φ(n)) thỏa mãn với d = d + kφ(n) với k∈ Z và gcd(e, φ(n)) = 1, dưới đây là giải thuật sửa đổi từ Euclid mở rộng để đảm bảo d > 0:

Giải thuật:

Đầu vào: 2 giá trị e, φ(n).

Đầu ra: $d = e^{-1} \bmod \varphi(n)$, d > 0.

Mã giả:

set a = e, b = φ(n)

set x2 = 1, x1 = 0

while b > 0, do

q = floor(a/b), r = a - qb, x = x2 - qx1.

a = b, b = r, x2 = x1, x1 = x.

set x = x2.

if x < 0 then set x = x + b;

set d = x

and return(d)

Ứng dụng định lý số dư Trung Quốc trường hợp đặc biệt và công thức Garner:

Để tạo chữ ký, ta dùng private key là cặp (n,d) công thức:

$s = RsaPrivate(m) = m^d \bmod n$

Các giá trị khóa được tính trước:

- $dP = \frac{1}{(e)} \bmod (p-1)$
- $dQ = \frac{1}{(e)} \bmod (q-1)$
- $qInv = \frac{1}{(q)} \bmod p$

Công thức tính chữ ký s:

- $s_1 = m^{dP} \bmod p$
- $s_2 = m^{dQ} \bmod q$
- $h = qInv.(s_1 - s_2) \bmod p$
- $s = s_2 + h.q$

private key sẽ gồm 5 giá trị là *(p, q, dP, dQ, qInv)*. [2]

Khi dùng định lý số dư Trung Quốc vào trong RSA, thì các giá trị dP, dQ và qInv được tính trước nên giảm đáng kể thời gian khi tạo chữ ký. Việc thực hiện tính toán trên hàm mũ sẽ tăng theo hệ số k mũ 3 (k là số bits của giá trị mod), do đó việc thực hiện phép toán mod trên p và q sẽ hiệu quả hơn nhiều khi dùng mod n vì p và q chỉ có kích thước gần bằng một nửa n. Do đó, toàn bộ chi phí thời gian tính toán trên p và q tiết kiệm là khoảng: $k^3 \Big/ 2\left(\frac{k}{2}\right)^3$

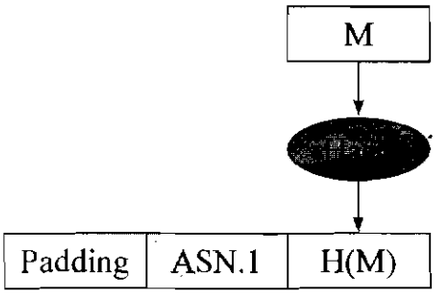
Các công thức trên được phát triển dựa vào cơ sở của 2 định lý: trường hợp đặc biệt của định lý số dư Trung Quốc, định lý Euler và công thức Garner để xây dựng công thức chia nhỏ khóa bí mật, tăng tốc độ thực thi khi tạo chữ ký số.[2]

4. SO SÁNH PKCS#1V1.5 VÀ RSA-PSS

Các chuẩn chữ số được phòng thí nghiệm RSA Data Security Inc phát triển. Nó dựa vào các cấu trúc ASN.1 và thiết kế cho phù hợp với chứng thư X.509, các tiêu chuẩn này do ANSI thiết kế, theo đó dữ liệu được chia thành từng khối nhỏ nhất là 8 bit (octet), để tăng cường độ bảo mật của dữ liệu khi mã hóa hoặc chữ ký số.

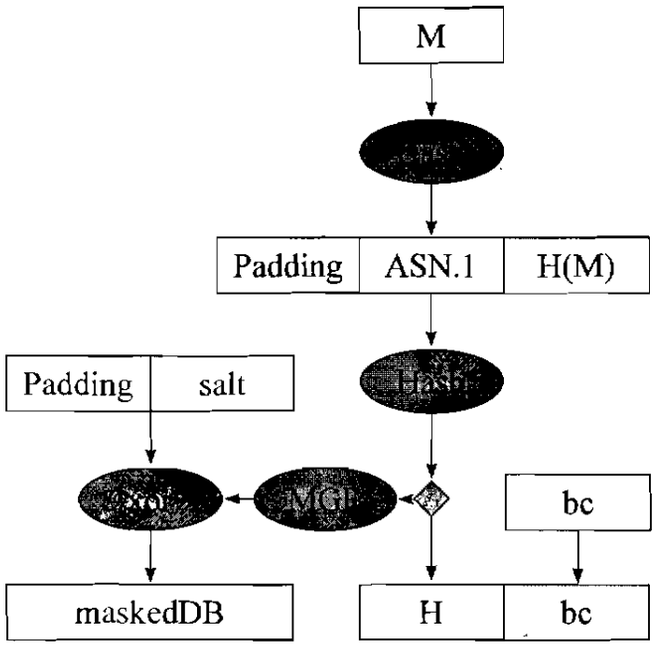
Hiện nay đa phần các hệ thống ứng dụng chữ ký số đều sử dụng chuẩn PKCS#1v1.5. Tuy nhiên chuẩn này vẫn còn tồn tại những điểm yếu và có thể bị tấn công. Trong khi đó RSA-PSS là một chuẩn chữ ký mới đã khỏa lấp các điểm yếu của PKCS#1v1.5 thông qua tính năng động khi tạo chữ ký.

Hình 4.1. Sơ đồ hoạt động chuẩn chữ ký PKCS#1v1.5



Chuẩn chữ ký PKCS#1v1.5 vẫn được xem là an toàn khi áp dụng để tạo chữ ký số. Tuy nhiên, Những điểm yếu bị tấn công thường là do trong quá trình triển khai thực thi PKCS#1v1.5 không đáp ứng được các yêu cầu về độ bảo mật, một nguyên nhân khác là đến từ mô hình thiết kế PKCS#1v1.5 không đáp ứng yêu cầu động. Hai hình thức tấn công vào PKCS#1v1.5 điển hình là **Tấn công Bleichenbacher (2006) của** nhà mật mã học Deniel Bleichenbacher và **Tấn công bằng phương pháp gây lỗi** (2006) nhà mật mã học Eric Brier cùng các cộng sự.

Hình 4.2. Sơ đồ hoạt động chuẩn chữ ký RSA-PSS



Chuẩn chữ ký RSA-PSS thay vì chỉ băm rồi tạo chữ ký thì giá trị hàm băm phải trải qua một quy trình ký mã trước khi thực sự tạo chữ ký. Chuẩn RSA-PSS được phát minh bởi hai nhà khoa học là Mihir Bellare và Phillip Rogaway. Do tính ngẫu nhiên trong RSA-PSS, những kẻ tấn công không thể biết trước dữ liệu sẽ được ký mã như thế nào. Do đó dẫn tới hoàn toàn thất bại trong tấn công bằng cách gây lỗi hệ thống, thống kê hay thử sai. [6]

5.HIỆN THỰC ỨNG DỤNG

5.1. Phương pháp tính toán số lớn

- **Biểu diễn số nguyên lớn:**

Hầu hết các hệ mật mã bất đối xứng đều yêu cầu sử dụng các số nguyên lớn có số bit từ 1024 bit trở lên để đáp ứng độ bảo mật. Do đó việc xây dựng một kiểu dữ liệu lớn, hỗ trợ các phép toán như số nguyên thông thường là cần thiết.

Trong thực tế các số nguyên được biểu diễn theo phương trình đa thức sau:

$$n = a_k b^k + a_{k-1} b^{k-1} + ... + a_1 b^1 + a_0$$

trong đó, b đại diện cho hệ số b >= 2, và giá trị đơn vị a thuộc không gian 0 <= a <= b-1. [8]

Quy trình tính toán số nguyên lớn chính là việc thao tác giữa các đa thức với nhau:

$$n = a_k b^k + a_{k-1} b^{k-1} + ... + a_1 b^1 + a_0$$
$$m = c_k b^k + c_{k-1} b^{k-1} + ... + c_1 b^1 + c_0$$
$$n + m = (a_k + c_k) b^k + (a_{k-1} + c_{k-1}) b^{k-1} + ... + (a_1 + c_1) b^1 + (a_0 + c_0)$$

Trong ngôn ngữ lập trình, một mảng các byte liên tiếp dùng để đại diện giá trị cho số nguyên lớn, một phần tử của mảng chính là giá trị đơn vị của số nguyên lớn.

Giải thuật cộng với b = 2^8 ;

Đầu vào: 2 giá trị nguyên lớn bi1, bi2

Đầu ra: result = bi1 + bi2

Mã giả:

```
set carry = 0, i = 0
while i < len, do
begin
    sum = bi1[i] + bi2 + carry
    carry = sum >> 8
    result[i] = sum
   i = i + 1;
end
return (result)
```

- **Phương pháp tính lũy thừa modulo:**

Lũy thừa modulo là một trong những phép toán quan trọng nhất trong các hệ thống mã hóa. Việc tối ưu việc tính toán phép tính này phụ thuộc vào rất nhiều yếu tố như kích thước thông số đầu vào, mức độ hoạt động của phần mềm, phần cứng và các thuật toán trong toán học.

Ba thuật giải có thể được ứng dụng để giải quyết các phép toán modulo trên số nguyên lớn là thuật giải Clasical, Barret và Montgomery. Tuy nhiên thuật giải Montgomery được xem là phương pháp tính phép nhân modulo hiệu quả nhất nhờ việc tránh các phép tính chia khi tính modulo. [9]

Ý tưởng cơ bản của thuật giải khi tính toán $z = x.y \bmod m$ là chuyển đổi x, y sang miền giá trị Montgomery. Trong miền giá trị Montgomery các phép tính nhân modulo được thực thi nhanh hơn nhiều so với vùng giá trị ban đầu. Do đó, đối với việc tính toán lũy thừa modulo gồm một tập hợp các phép nhân modulo được thực hiện, chi phí tính toán sẽ giảm hơn rất nhiều. Đặc biệt các phép nhân modulo Montgomery rất dễ tính toán trong hệ nhị phân với các phương pháp đẩy n bit qua trái hoặc qua phải.

Định lý Montgomery:

- Cho 2 giá trị x, y and $0 \le x,y \le m$
- Vùng giá trị Montgomery: $\tilde{x} = xR \bmod m$, $\tilde{y} = yR \bmod m$.
- Phép tính nhân modulo Montgomery: $\tilde{z} = \tilde{x}\tilde{y}R^{-1} \bmod m = xR(yR)R^{-1} \bmod m = xyR \bmod m$.
- Kết quả giá trị trả về trong vùng Montgomery được định nghĩa bởi hàm $MP(\tilde{x},\tilde{y})$.
- Kết quả: $z = \tilde{z}R^{-1} \bmod m = (zR)R^{-1} \bmod m$.
- Giá trị domain của x có thể được tính bằng $MP(\tilde{x}, R^2 \bmod m)$. Vì $MP(x, R^2) = (xRR)/R \pmod m = xR \pmod m$.
- Kết quả $z = x*y \bmod m$ được tính bằng $MP(\tilde{z}, 1)$. Vì $MP(\tilde{z}, 1) = MP(zR, 1) = (zR)/R \pmod m = z \pmod m$. [10]

Thuật giải tính toán hàm MP(x,y) (trích từ handbook of applied cryptography):

Giá trị a trả về của hàm MP sẽ được cộng với bội số của m, việc thực hiện phép tính diễn ra trên từng giá trị đơn vị u_i của số nguyên lớn khi biểu diễn trong hệ số b.

Đầu vào: số nguyên dương $m = (m_{n-1}\dots m_0 m_1)_b$, $x = (x_{n-1}\dots x_0 x_1)_b$, $y = (y_{n-1}\dots y_0 y_1)_b$ với $x>=0$, $m > y$, $R = b^n$ thỏa mãn $\gcd(m,b) = 1$, $m' = -m^{-1} \bmod b$. [2]

Đầu ra: $xyR^{-1} \bmod m$

Mã giả:

```
set a = 0
for i from 0 to n-1do
begin
    u_i = (a_0 + x_i y_0)m' mod b
    a  = (a + x_i y + u_i m) / b
end
if a >=m then a = a-m
return (a)
```

Thuật giải tính lũy thừa modulo: là sự kết hợp của phép nhân modulo và phép duyệt lũy thừa theo cơ số 2 (nhị phân) từ trái qua phải $e = (e_t e_{t-1} \ldots e_0 e_1)_2$. [2]

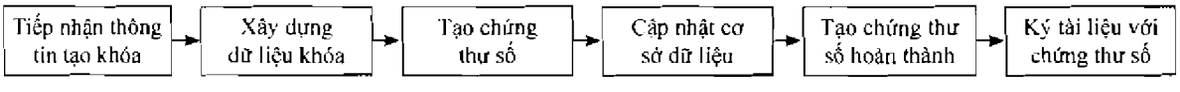
Mã giả:

```
x̃ = MP(x, R^2 mod m), ã = MP(1, R^2 mod m)
for i in t downto 0
    ã = MP(ã, ã)
    if e_i = 1 then ã = MP(ã, x̃)
a = MP(ã, 1)
return(a)
```

5.2. Mô phỏng ứng dụng cấp phát chứng thư số

Đây là ứng dụng mô phỏng quá trình cấp phát chứng thư số của các tổ chức có pháp nhân, được cấp phép của Bộ Thông tin và Truyền thông. Ứng dụng được phát triển trên platform window form được tích hợp với hệ thống các thuật toán RSA và SHA1, với các chức năng thu thập thông tin khách hàng, kích thước khóa, tạo và quản lý chứng thư số trong cơ sở dữ liệu SQL Server.

Hình 5.1. Quy trình cấp phát chứng thư số



Các chứng thư số được tạo thành file .p12 dùng được cài trực tiếp vào trong window hay lưu trữ trong smart card hoặc USB Token để tạo chữ ký số trong các ứng dụng: Microsoft office, Adobe Acrobat Professional, Open Office,...

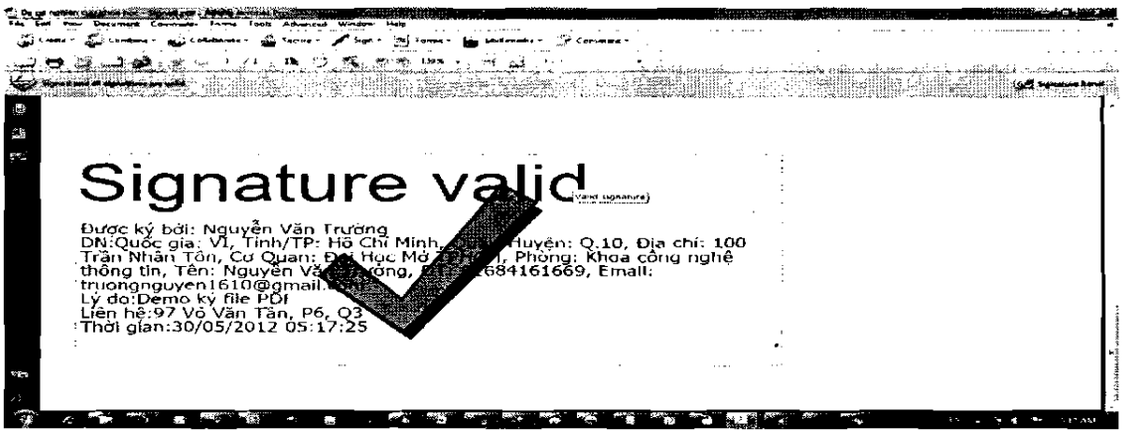
Hình 5.2. Chứng thư số được xây dựng với khóa 2048 bits trong cơ sở dữ liệu

Name	Date modified	Type	Size
10e00d000b0000a0000000900000f001.p12	4/20/2012 7:19 PM	Personal Information Exchange	4 KB
10e00d000b0000a0000000900000f002.p12	4/20/2012 7:19 PM	Personal Information Exchange	4 KB
10e00d000b0000a0000000900000f003.p12	2/11/2012 1:18 PM	Personal Information Exchange	4 KB
10e00d000b0000a0000000900000f004.p12	2/13/2012 2:40 PM	Personal Information Exchange	4 KB
10e00d000b0000a0000000900000f005.p12	2/13/2012 2:40 PM	Personal Information Exchange	4 KB
10e00d000b0000a0000000900000f006.p12	2/13/2012 5:43 PM	Personal Information Exchange	4 KB
10e00d000b0000a0000000900000f007.p12	4/20/2012 7:19 PM	Personal Information Exchange	4 KB

Hình 5.3. File chứng thư số tạo ra từ ứng dụng

	Số HĐ	ID Chứng Chỉ Số	PublicKey
1	HD01	10e00d000b0000a0000000900000f001	30818902818100C9D14E9EB7147683A4BCEFB2DF8FB8DB4...
2	HD02	10e00d000b0000a0000000900000f002	30818902818100EAB503EBA4F17DB18FF995DB035176A838...
3	HD03	10e00d000b0000a0000000900000f003	30818902818100D229E8E6099C3C444291CA9047B9F98C8D...
4	HD04	10e00d000b0000a0000000900000f004	30818902818100BE6316EA24F1383E743E1CFD0B02B7BA6A...
5	HD05	10e00d000b0000a0000000900000f005	30818902818100AE7EA0EF64C129D6736F79EFCFA6DF3D0D...
6	HD06	10e00d000b0000a0000000900000f006	30818902818100B489FB9FB3B8D6238794C70FD9164232503...
7	HD08	10e00d000b0000a0000000900000f007	30818902818100A8119CE043E5A52CB2C74E04B301FCF987B...

Hình 5.4. Chữ ký số trong Adobe Acrobat Professional

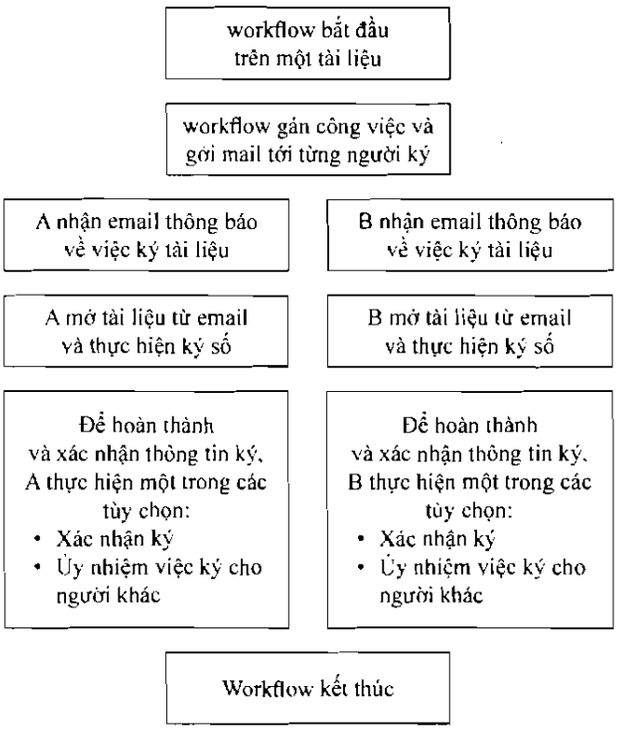


5.3. Collect Signature workflow SharePoint

Thực hiện thu thập chữ ký số tự động dùng công nghệ sharepoint: đây là một công nghệ mới của Microsoft, hỗ trợ khả năng quản lý và thu thập chữ ký tự động trên các loại tài liệu Microsoft Office như word, excel,...

Collect Signature Workflow trong Sharepoint quản lý và theo dõi tự động tất cả tiến trình của từng cá nhân như thông tin người đã ký, thời gian ký, trạng thái thu thập chữ ký. Collect Signature workflow cho phép tùy chỉnh danh sách người ký tài liệu, việc thu thập chữ ký là đồng thời hoặc theo thứ tự người ký trong danh sách. Điều này đặc biệt có ích trong trường hợp tài liệu cần chữ ký của nhiều cấp, phòng ban.

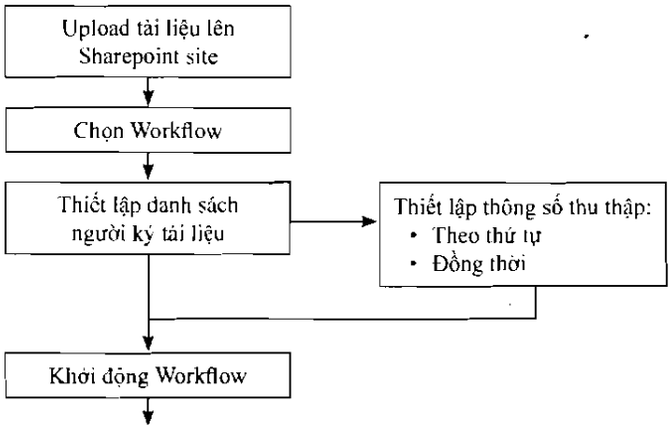
Hình 5.5. Sơ đồ hoạt động Collect Signature workflow trong Sharepoint



Tiến trình của Collect Signature workflow:

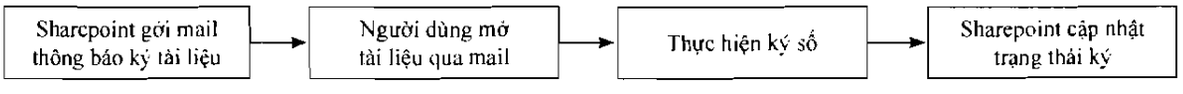
• Bắt đầu workflow: Tải tài liệu lên SharePoint Server site, cập nhật danh sách những người ký tài liệu, khởi động workflow. Workflow bắt đầu tự động thực hiện thu thập chữ ký tự động.

Hình 5.6. Sơ đồ tiến trình bắt đầu collect signature workflow



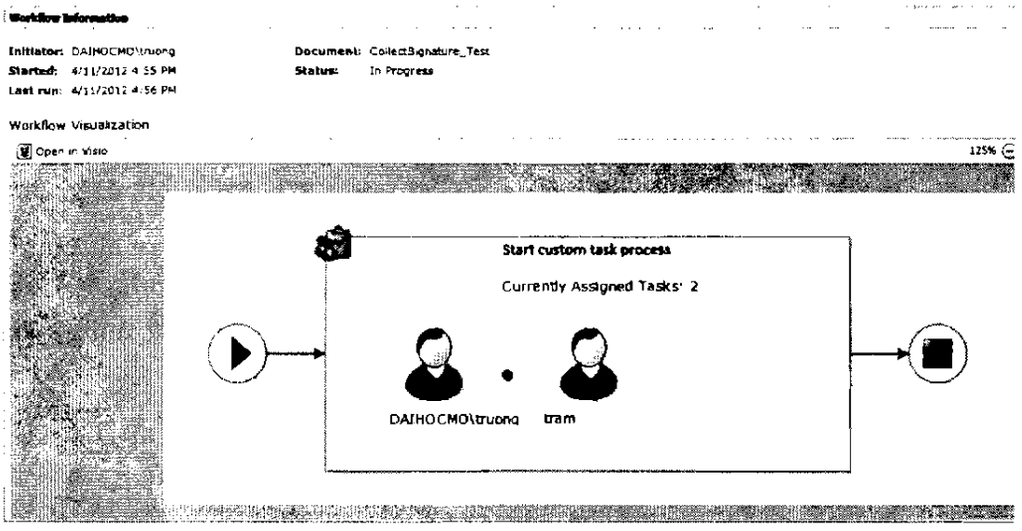
• Thực hiện ký tài liệu: Email sẽ được gởi tới từng người tham gia ký, mỗi người mở tài liệu của mình thông qua link liên kết trong Outlook, ký tài liệu.

Hình 5.7. Sơ đồ tiến trình nhận và ký tài liệu

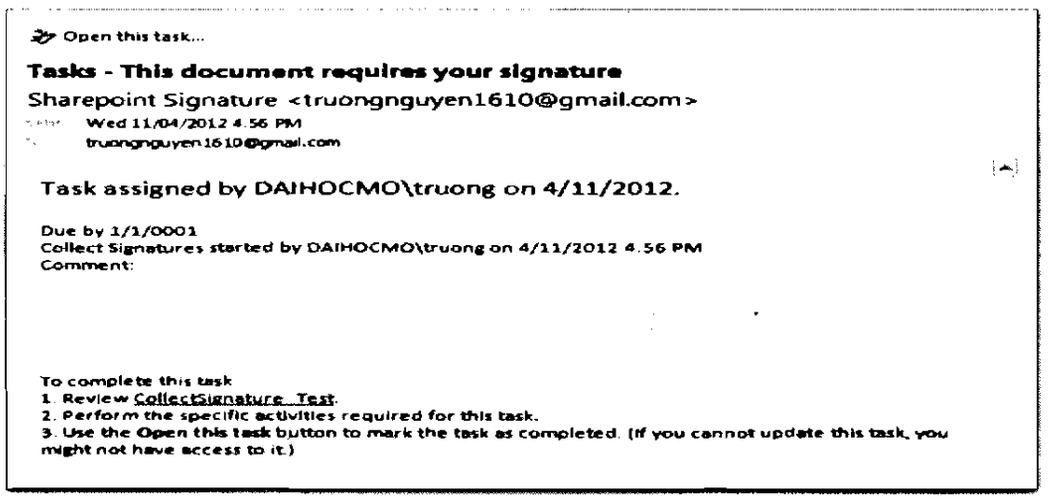


Trạng thái ký của tài liệu: khi workflow đang trong quá trình hoạt động, mỗi người có thể xem thông tin, trạng thái ký của từng người tham gia. Khi tất cả chữ ký đã được thu thập workflow kết thúc.

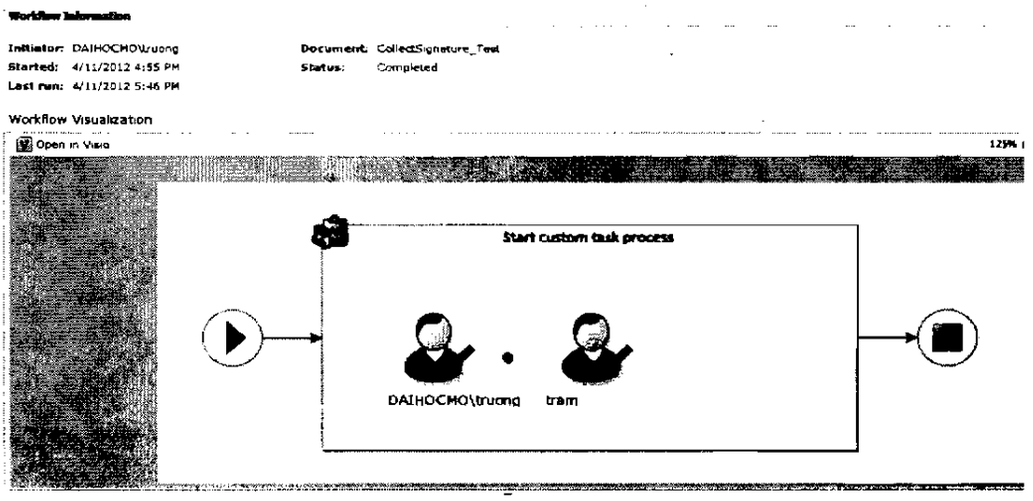
Hình 5.8. Trạng thái workflow khi bắt đầu thu thập chữ ký



Hình 5.9. Mail yêu cầu thực hiện ký số tài liệu.



Hình 5.10. Trạng thái workflow kết thúc thu thập chữ ký



6. KẾT QUẢ ĐẠT ĐƯỢC VÀ ĐÁNH GIÁ

Hệ mật mã RSA được xem là một trong những hệ DSA phổ biến ứng dụng cho chữ ký số. Trong bài báo này đã mô tả quy trình xây dựng mô hình chữ ký số RSA, với các thuật giải Miller-Rabin để kiểm tra tính nguyên tố của số nguyên tố lớn p, q; Euclid mở rộng, Fermat để tối ưu việc sinh các giá trị khóa (e, d, n); định lý số dư Trung Quốc dùng chia nhỏ dữ liệu để giúp quá trình tạo chữ ký nhanh hơn, cùng với đó là sự so sánh giữa hai chuẩn chữ ký PKCS#1v1.5 và RSA-PSS.

Bên cạnh đó, bài báo biểu diễn hình thức tồn tại của các giá trị số nguyên với một hệ số b bất kỳ, để xây dựng một dãy các byte liên tiếp nhau đại diện cho giá trị của số nguyên lớn. Việc tính toán trên số nguyên lớn, đặc biệt là phép tính lũy thừa modulo được trình bày qua thuật giải Montgomery. Các giá trị x,y,...sẽ được chuyển qua vùng Montgomery để thực hiện phép nhân modulo dựa vào dịch chuyển n bit giá trị trong hệ nhị phân.

Bằng việc ứng dụng RSA và SHA1, bài báo mô tả ứng dụng mô phỏng dùng để tạo chứng thư số, một loại file đặc biệt (.p12 hoặc .pfx) dùng để ký số trên các loại tài liệu. Đồng thời, bài báo giới thiệu và mô tả quy trình thu thập chữ ký tự động bằng workflow sharepoint ứng dụng trong cơ quan, doanh nghiệp để tăng hiệu quả làm việc.

7. KẾT LUẬN VÀ TRIỂN VỌNG

Nghiên cứu đã đưa ra những phương pháp để hiện thực thuật giải RSA trên số nguyên lớn. Qua đó ứng dụng RSA xây dựng chương trình cấp phát chứng thư số, thực hiện việc ký số trên file word, excel, pdf,...

Hiện nay, nhiều nước trên thế giới không chỉ triển khai ứng dụng chữ ký số trên mạng máy tính mà còn phát triển chữ ký số trên mạng điện thoại di động để thực hiện các giao dịch điện tử. Hướng đi này giúp đẩy nhanh giao dịch, đơn giản hóa mua sắm trực tuyến và giúp người dùng có thể truy cập mọi lúc, mọi nơi để xử lý công việc.

TÀI LIỆU THAM KHẢO

1. David Hook, Begin cryptography with Java 2005, chapter 4 – 7.
2. Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone, Handbook of applied cryptography 1996, chapter 3 - 5, 7 - 9, 11.
3. David Ireland, RSA_Theory Report, 5 May 2007.
4. RSA Laboratories. PKCS #1 v2.1: RSA Encryption Standard. June 2002.
5. Burton Kalinski. Some Examples of the PKCS Standards, RSA Laboratories, 1999.
6. Christina Lindenberg, Kai Wirt, Johannes Buchmann, Report Formal Proof for the correctness of RSA-PSS, 2006.
7. Johannes, Report RSA-PSS – Provably secure RSA Signatures, November 10, 2011.
8. A. Bosselaers, R. Govaerts, and J. Vandewalle, "Comparison of Three Modular Reduction Functions", Proc. CRYPTO'93.
9. M. Johnson, B. Phung, T. Shackelford, S. Rueangvivatanakij, " Modular Reduction of Large Integers Using Classical, Barrett, Montgomery Algorithms".
10. Allen Michalski, "Montgomery Multiplication Algorithmic and Hardware Implementations" November 6th, 2002.

11. Klaus Schmed, "Cryptography and Public key Infrastructure on the Internet" –WILAY.
12. Wenbo Mao, "Mordern Criptography-Theory and Practic", HP company 2004.

(Ngày nhận bài: 27/12/2012; Ngày chấp nhận đăng: 19/02/2013).