**Học viện công nghệ bưu chính viễn thông**

**Khoa công nghệ thông tin I**

-----🙞🙜🕮🙞🙜-----



**BÁO CÁO BÀI TẬP LỚN**

**Môn: An Toàn Bảo Mật Hệ Thống Thông Tin**

***Đề tài* : Tìm hiểu về giải thuật tạo chữ ký số sử dụng RSA**

**Giáo viên hướng dẫn: TS. Hoàng Xuân Dậu**

**Nhóm sinh viên:**

**1.Trần Văn In**

**2.Nguyễn Đăng Khiêm**

**3.Nguyễn Đức Mạnh**

**4.Phạm Văn Mạnh**

**5. Trần Việt Mạnh**

**Lớp: D11CNPM4**

Hà Nội, 23/ 09/ 2014

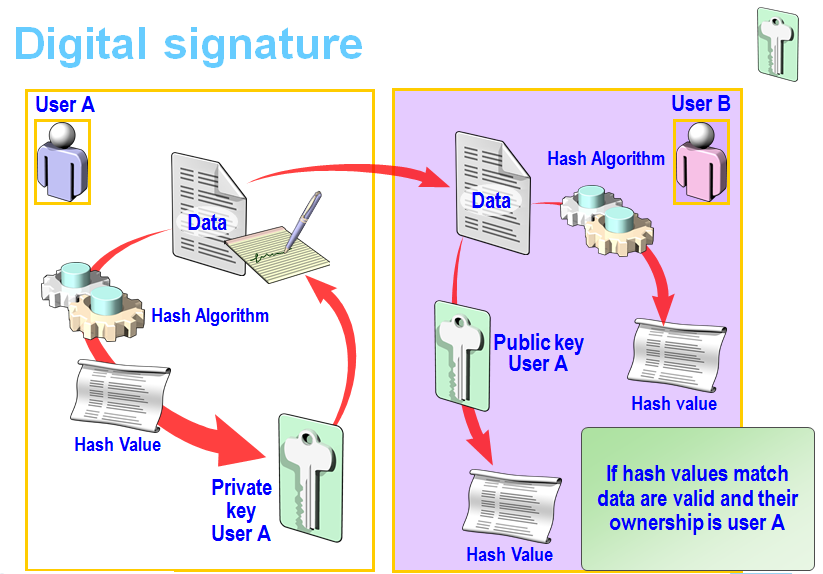
**Mục lục:**

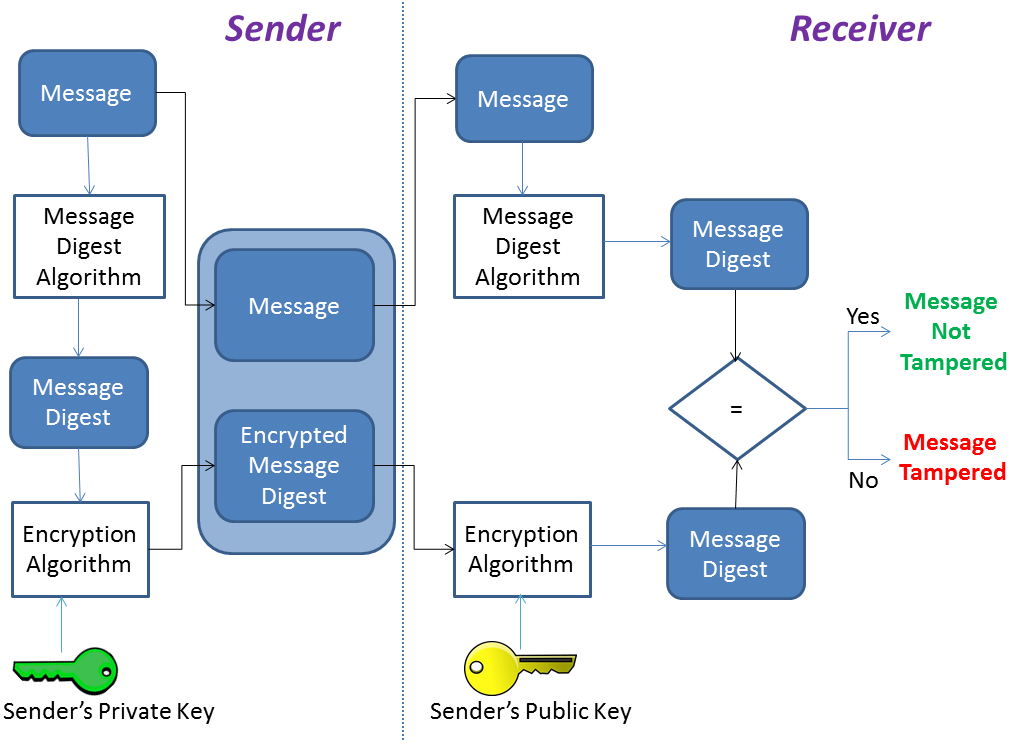
1. Giới thiệu chữ ký số
2. Nội dung
   1. Kiến trúc tổng quát chữ ký số và chữ ký số sử dụng giải thuật RSA
   2. Giải thuật và cài đặt giải thuật.
   3. Các điểm yếu chữ ký số RSA
   4. Các dạng tấn công
   5. Ứng dụng chữ ký số RSA
3. Kết luận
4. Tài liệu tham khảo
5. **Giới thiệu**

* Một số khái niệm:
  + Chữ kí số (Digital Signature ) là một chuỗi dữ liệu liên kết với một thông điệp (message) và thực thể tạo ra thông điệp.
  + Giải thuật tạo ra chữ ký số (Digital Signature generation algorithm) là một phương pháp sinh chữ ký số.
  + Giải thuật kiểm tra chữ ký số (Digital Signature verification algorithm ) là một phương pháp xác minh tính xác thực của chữ ký số, có nghĩa là nó thực sự được tạo ra bởi 1 bên chỉ định.
  + Một hệ chữ ký số (Figital Signature Scheme) bao gồm giải thuật tạo chữ số và giải thuật kiểm tra chữ kỹ số.
  + Quá trình tạo chữ ký số (Digital Signature signing process) bao gồm:
    - Giải thuật tạo chữ ký số.
    - Phương pháp chuyển dữ liệu thông điệp thành dạng có thể ký được
  + Quá trình kiểm tra chữ ký số (Digital signature verification process) bao gồm:
    - Giải thuật kiểm tra chữ ký số, và
    - Phương pháp khôi phục dữ liệu từ thông điệp
  + Hàm băm (Hash Funtion) làm hàm toán học chuyển đổi thông điệp (message) có độ dài bất kỳ (hữu hạn) thành một dãy bít có độ dài cố định (tùy thuộc vào thuật toán băm). Dãy bít này được gọi là thông điệp rút gọn.(message disgest) hay giá trị băm (hash value), đại diện cho thông điệp ban đầu.
    - Hàm băm SHA-1: Thuật toán SHA-1 nhận thông điệp ở đầu vào có chiều dài k<264 bit, thực hiện xử lý và đưa ra thông điệp thu gọn (message digest) có chiều dài cố định 160 bits. Quá trình tính toán cũng thực hiện theo từng khối 512bits, nhưng bộ đệm xử lý dùng 5 thanh ghi 32-bits. Thuật toán này chạy tốt với các bộ vi xử lý có cấu trúc 32 bits.
* Lược đồ chữ ký số RSA.
  + Trong phần này mô tả lược đồ chữ ký RSA. Độ an toàn của lược đồ chữ ký RSA dựa vào độ an toàn của hệ mã RSA. Lược đồ bao gồm cả chữ ký số kèm theo bản rõ và tự khôi phục thông điệp từ chữ ký số.
    - Thuật toán sinh khóa cho lược đồ chữ ký RSA
    - Thuật toán sinh chữ ký RSA
    - Thuật toán chứng thực chữ ký RSA

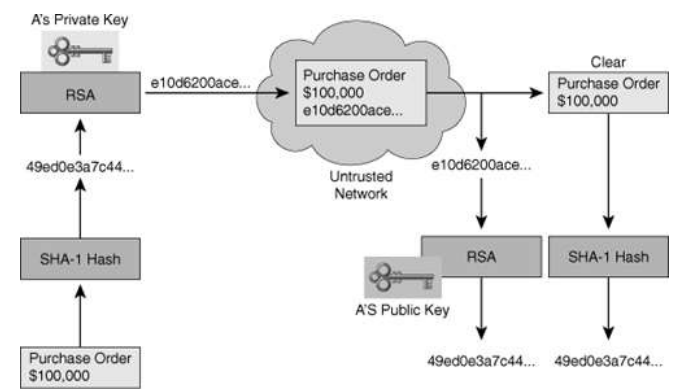
1. **Nội dung**
2. **Kiến trúc**

**1.1 Kiến trúc chữ ký số tổng quát**

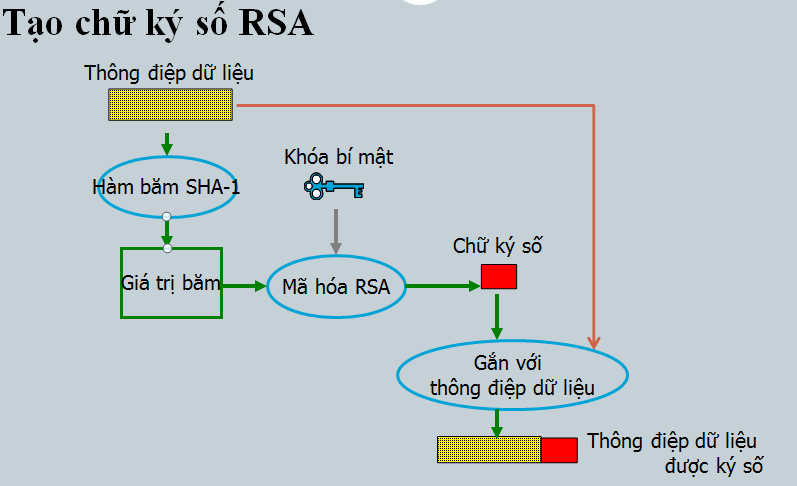


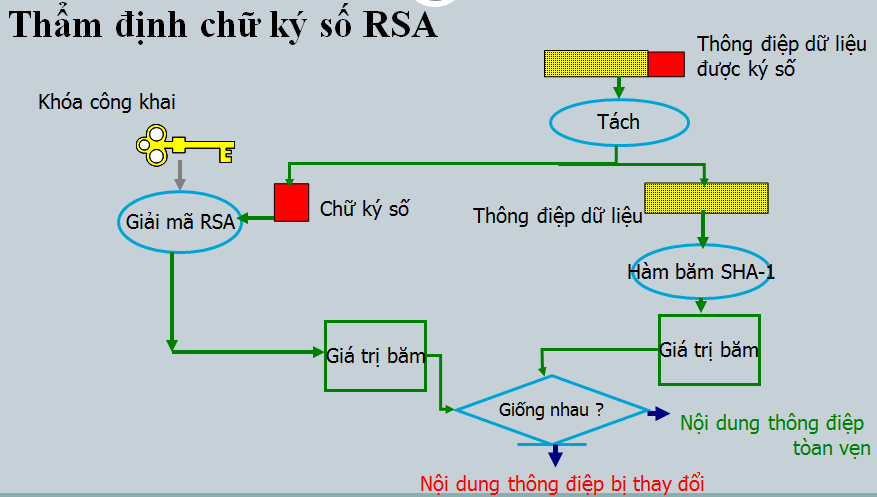


* 1. Quá trình ký (bên gửi)
* Tính toán chuỗi đại diện (message digest/ hash value) của thông điệp sử dụng một giải thuật băm (Hashing algorithm)
* Chuỗi đại diện được ký sử dụng khóa riêng (Priavte key) cảu người gửi va 1 giải thuật tạo chữ ký (Signature/ Encryption algorithm). Kết quả chữ ký số (Digital signature) của thông điệp hay còn gọi là chuỗi đại diện được mã hóa (Encryted message digest)
* Thông điệp ban đầu (message) được ghép với chữ ký số( Digital signature) tạo thành thông điệp đã được ký (Signed message)
* Thông điệp đã được ký (Signed message) được gửi cho người nhận
  1. Quá trình kiểm tra chữ ký (bên nhận)
* Tách chữ ký số và thông điệp gốc khỏi thông điệp đã ký để xử lý riêng;
* Tính toán chuỗi đại diện MD1 (message digest) của thông điệp gốc sử dụng giải thuật băm (là giải thuật sử dụng trong quá trình ký)
* Sử dụng khóa công khai (Public key) của người gửi để giải mã chữ ký số -> chuỗi đại diện thông điệp MD2
* So sánh MD1 và MD2:
  + Nếu MD1 =MD2 -> chữ ký kiểm tra thành công. Thông điệp đảm bảo tính toàn vẹn và thực sự xuất phát từ người gửi (do khóa công khai được chứng thực).
  + Nếu MD1 <>MD2 -> chữ ký không hợp lệ. Thông điệp có thể đã bị sửa đổi hoặc không thực sự xuất phát từ người gửi.
  1. **Kiến trúc chữ ký số RSA**



Cụ thể hơn:



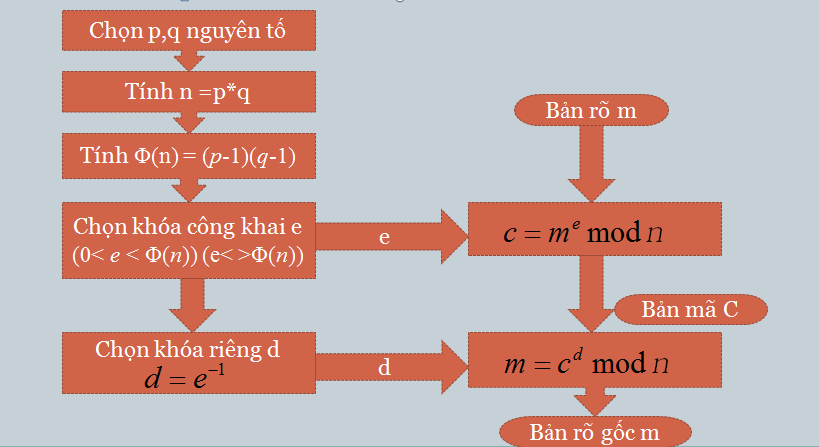


* 1. Quá trình ký (bên gửi)
* Tính toán chuỗi đại diện (message digest/ hash value) của thông điệp sử dụng một giải thuật băm (Hashing algorithm) SHA-1
* Chuỗi đại diện được ký sử dụng khóa riêng (Priavte key) của người gửi và giải thuật tạo chữ ký (Signature/ Encryption algorithm) RSA. Kết quả chữ ký số (Digital signature) của thông điệp hay còn gọi là chuỗi đại diện được mã hóa bởi giải thuật RSA (Encryted message digest)
* Thông điệp ban đầu (message) được ghép với chữ ký số( Digital signature) tạo thành thông điệp đã được ký (Signed message)
* Thông điệp đã được ký (Signed message) được gửi cho người nhận
  1. Quá trình kiểm tra chữ ký (bên nhận)
* Tách chữ ký số RSA và thông điệp gốc khỏi thông điệp đã ký để xử lý riêng;
* Tính toán chuỗi đại diện MD1 (message digest) của thông điệp gốc sử dụng giải thuật băm (là giải thuật sử dụng trong quá trình ký là SHA-1)
* Sử dụng khóa công khai (Public key) của người gửi để giải mã chữ ký số RSA-> chuỗi đại diện thông điệp MD2
* So sánh MD1 và MD2:
  + Nếu MD1 =MD2 -> chữ ký kiểm tra thành công. Thông điệp đảm bảo tính toàn vẹn và thực sự xuất phát từ người gửi (do khóa công khai được chứng thực).
  + Nếu MD1 <>MD2 -> chữ ký không hợp lệ. Thông điệp có thể đã bị sửa đổi hoặc không thực sự xuất phát từ người gửi.

1. **Giải thuật và cài đặt giải thuật**

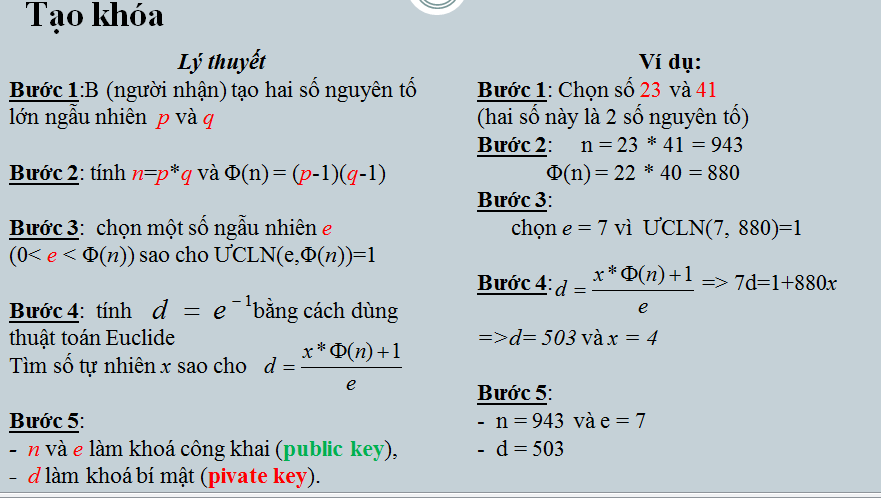
**2.1 Giải thuật RSA được dùng trong việc tạo khóa, mã hóa, giải mã.**

Sơ đồ giải thuật :

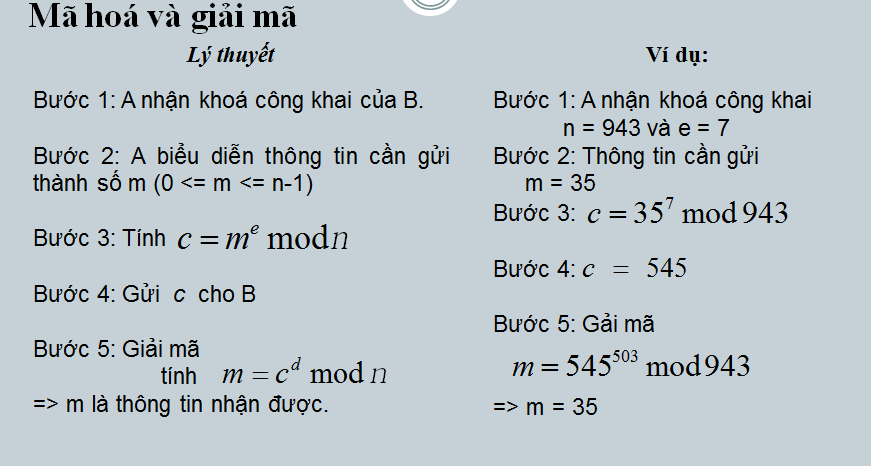


|  |  |
| --- | --- |
| Thuật toán RSA có hai Khóa:  **- *Khóa công khai*** (**Public key**): được công bố rộng rãi cho mọi người và được dùng để mã hóa  **-** ***Khóa bí mật*** (**Private key**):  Những thông tin được mã hóa bằng khóa công khai chỉ có thể  được giải mã bằng khóa bí mật tương ứng | C:\Users\Mr Khiem\Desktop\RSA\Báo cáo\TQRSA.png |

* 1. Tạo khóa



* 1. Mã hóa và giải mã

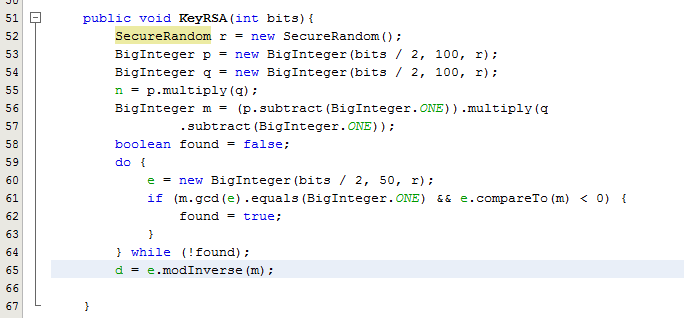


**2.2 Giải thuật SHA-1 được dùng trong việc băm chia nhỏ dữ liệu.**

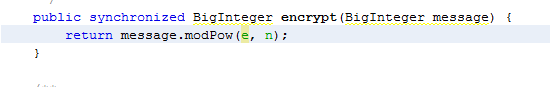
**2.3 Cài đặt giải thuật trong ngôn ngữ JAVA**

* Sử dụng BigInteger trong gói java.math.\* cung cấp hầu hết các hàm dựng và các hàm số học cho phép thao tác thuận lợi với số nguyên lớn
* Một số hàm:
  + Hàm dựng BigInteger (int bitLength, int certainty, Random rnd): sinh số nguyên tố ngẫu nhiên với số bit cho trước;
  + Hàm BigInteger add(BigInteger val): cộng 2 số nguyên lớn;
  + Hàm BigInteger subtract(BigInteger val): Trừ 2 số nguyên lớn;
  + Hàm BigInteger multiply(BigInteger val): nhân 2 số nguyên lớn;
  + Hàm gcd(BigInteger val): Tìm USCLN của 2 số lớn
  + Hàm mod(BigInteger m): Tính modulo (phần dư) của phép chia nguyên;
  + Hàm BigInteger modInverse(BigInteger m): tính modulo nghịch đảo (this-1mod m);
  + Hàm BigInteger modPow(BigInteger exponent, BigInteger m): Tính (thisexponent mod m).
* Cụ thể code RSA

Tạo khóa:

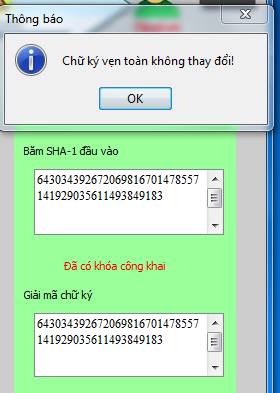


Mã hóa: (code này trong RSA , còn trong chữ ký số thì d bí mật được mã hóa)



Giải mã: (code này trong RSA , còn trong chữ ký số thì e công khai được mã hóa)

C:\Users\Mr Khiem\Desktop\RSA\Báo cáo\giai ma code.PNG

Giao diện DEMO kết quả chữ ký số sử dụng RSA: 

1. **Các điểm yếu của chữ ký số sử dụng giải thuật RSA**
   1. **Chữ ký số nói chung**

Sự xuất hiện của chữ ký số và chức năng tiền định của nó, đặc biệt là vai trò của nó như là một công cụ trong việc xác định tính nguyên gốc, xác định tác giả, bảo đảm tính toàn vẹn của tài liệu số, đã đóng một vai trò vô cùng quan trọng trong việc xác định địa vị pháp lý của tài liệu số trong giao dịch số. Việc sử dụng chữ ký số trong phần lớn trường hợp là cơ sở khẳng định giá trị pháp lý của những văn bản điện tử tương đương với tài liệu giấy. Hiện nay, chữ ký số là phương tiện duy nhất để xác nhận giá trị pháp lý của tài liệu điện tử.

Như vậy, với sự xuất hiện của chữ ký số, vấn đề giá trị pháp lý của tài liệu điện tử, có thể coi như đã được giải quyết. Việc sử dụng chữ ký số trong giao dịch cũng có những ưu điểm và bất cập nhất định. Dưới đây là những hạn chế của chữ ký số:

- Sự lệ thuộc vào máy móc và chương trình phần mềm: chữ ký số là một chương trình phần mềm máy tính. Để kiểm tra tính xác thực của chữ ký cần có hệ thống máy tính và phần mềm tương thích. Đây là hạn chế chung khi sử dụng văn bản điện tử và chữ ký số.  
- Tính bảo mật không tuyệt đối: Nếu chữ ký bằng tay được thực hiện trên giấy, được ký trực tiếp và luôn đi kèm với vật mang tin, chữ ký tay không thể chuyển giao cho người khác, thì chữ ký số không như vậy. Chữ ký số là một bộ mật mã được cấp cho người sử dụng, đây là phần mềm máy tính không phụ thuộc vào vật mang tin. Chính vì vậy, trở ngại lớn nhất khi sử dụng chữ ký số là khả năng tách biệt khỏi chủ nhân của chữ ký. Nói cách khác, chủ nhân của chữ ký số không phải là người duy nhất có được mật mã của chữ ký. Tồn tại một số nhóm đối tượng có thể có được mật mã, đó là: bộ phận cung cấp phần mềm; bộ phận cài đặt phần mềm, những người có thể sử dụng máy tính có cài đặt phần mềm. Ngoài ra, mật mã có thể bị đánh cắp. Cũng có thể, chủ nhân chữ ký số chuyển giao cho người khác mật mã của mình. Như vậy, tính bảo mật của chữ ký số không phải là tuyệt đối.  
- Vấn đề bản gốc, bản chính: Nếu đối với tài liệu giấy, chữ ký được ký một lần và chỉ có một bản duy nhất (được coi là bản gốc). Bản gốc được ký bằng chữ ký sẽ không thể cùng lúc ở hai chỗ khác nhau. Có thể tin tưởng rằng, nếu bản gốc duy nhất mất đi thì sẽ không thể có bản thứ hai giống hệt như vậy. Nhưng với văn bản điện tử đã được ký bằng chữ ký số, người ra có thể copy lại và bản copy từ bản chính và bản copy từ bản copy không có gì khác biệt so với bản chính duy nhất được ký. Đây là một thách thức đối với công tác văn bản và cả nền hành chính. Khái niệm bản gốc, bản chính trong văn bản hành chính sẽ phải xem xét lại đối với văn bản điện tử.  
- Sự có thời hạn của chữ ký điện tử. Chữ ký điện tử là chương trình phần mềm được cấp có thời hạn cho người sử dụng. Về lý thuyết, văn bản sẽ có hiệu lực pháp lý khi được ký trong thời hạn sử dụng của chữ ký. Tuy nhiên, thực tế hiệu lực pháp lý của văn bản hoàn toàn có thể bị nghi ngờ khi chữ ký số hết thời hạn sử dụng. Đây cũng là một hạn chế và thách thức rất lớn đối với việc sử dụng chữ ký số.

**3.2 Chữ ký số sử dụng RSA**

**3.2.1 Hiệu** **suất** **thực** **hiện** **của** **thuật** **toán** **RSA**

Tốc độ thực hiện của hệ RSA là một trong những điểm yếu so với các hệ mật mã khóa đối xứng.

Theo ước tính, thực hiện mã hóa và giải mã bằng hệ mật mã RSA chậm hơn 100 lần so với hệ mã khóa đối xứng DES (Khi thực hiện bằng phần mềm). Và chậm hơn 1000 lần so với DES (Khi thực hiện bằng phần cứng).

**3.2.2 Chi** **phí** **và** **tốc** **độ** **thực** **hiện** **của** **thuật** **toán** **RSA**

**1.** **Chi** **phí**

Để thực hiện thuật toán RSA phần lớn tốn chi phí thực hiện các phép tính cơ bản như : Tạo khóa, mã hóa, giải mã. Quá trình mã hóa, giải mã tương được với chi phí thực hiện các phép tính lũy thừa module n. Để đảm bảo cho khóa bí mật được an toàn thì thường chọn mũ công khai e nhỏ hơn nhiều so với số mũ bí mật d, do đó chi phí thời gian để thực hiện mã hóa dữ liệu nhỏ hơn nhiều so với thời gian giải mã.

**2.Tốc độ của hệ RSA**

Tốc độ của RSA là một trong những điểm yếu của RSA so với các hệ mã đối xứng, so với hệ mã DSA thì RSA chậm hơn từ 100 đến 1000 lần, vì vậy RSA không được dùng để mã hóa khối lượng dữ liệu lớn mà thường dùng để mã hóa dữ liệu nhỏ.

1. **Các dạng tấn công**

**4.1.** **Tấn** **công** **lặp**

Simons và Norris đã chỉ ra rằng hệ thống RSA có thể bị tấn công khi sử dụng tấn công lặp liên tiếp. Đó là khi kẻ tấn công biết khóa công khai (*e*, *n*) và bản mã C thì anh ta có thể tính chuỗi các bản mã sau:

C1 = Ce (mod *n*) C2 = C1e (mod *n*) …………………

Ci = Ci-1e (mod *n*)

Nếu có một phần tử Cj trong chuỗi C1, C2, …, C, … sao cho Cj = C thì khi đó anh ta sẽ tìm được M = Cj-1 bởi vì:

i

*e*

Cj = *Cj*-1 (mod *n*)

C = Me (mod *n*)

**4.2.** **Kiểu** **tấn** **công** **module** **n** **dùng** **chung**

Simons và Norris cũng chỉ ra rằng hệ thống RSA có thể bị tấn công khi sử dụng module n dùng chung, thực vậy nếu một thông điệp M được mã hoá bằng hai khoá công khai e1 và e từ hai thành viên trong hệ thống thì được:

2

C1 = *M* *e*1 (mod n)

C2 = *M* *e*2 (mod n)

Sau đó người tấn công dùng thuật toán Euclide mở rộng:

e­1\*a +e2\*b = 1 sao cho gcd(e1, e2 ) = 1 thì M được khôi phục lại như sau: M = *C1a* . *C2b* mod n.

**4.3.** **Tấn** **công** **khi** **khoá** **công** **khai** **e** **nhỏ**

Hastad đã đưa ra kiểu tấn công khi khoá công khai e nhỏ (e =3) của hệ mã công khai RSA như sau:

Giả sử để gửi thông điệp M đến các người dùng P1, P2 …,Pk với khoá công khai là (ei , ni). A mã hoá M bằng khoá công khai (ei , ni) và gửi các bản mã Ci đến người dùng Pi, biết M < n với i = 1, 2,…, n.

Ta có thể nghe trộm kết nối ra ngoài của A và thu thập được k bản mã Ci.

Giả sử các khoá công khai ei = 3 thì có thể khôi phục M nếu k ≥ 3.

i

Thực vậy, nếu có được C1, C2, C3 với C1= M3 mod ni; C2= M3 mod n2; C3= M3 mod n3 và gcd(ni, nj) = 1, i ≠ j. Áp dụng định lý số dư Trung Hoa với C1,C2,C3 tìm được:

C’ thuộc Zn1n2n3 thỏa C’ = M3 mod n1n2n3 -> M3 là số nguyên

Vậy M=

**5.Ứng dụng chữ ký số sử dụng giải thuật RSA**

-Sử dụng trong việc đảm bảo vẹn toàn dữ liệu: Chữ ký, công văn, file, tệp tin của người gửi qua môi trường Internet của các cá nhân , cơ quan tổ chức ….

**III. Kết luận**

Bài báo cáo đã giới thiệu được kiến trúc, cài đặt giải thuật, các điểm yếu, các dạng tấn công của chữ ký số sử dụng giải thuật RSA.

**IV.Tài liệu tham khảo**

**1.**Giáo trình An Toàn Bảo Mật Hệ Thống Thông Tin\_TS Hoàng Xuân Dậu\_HVCNBCVT.(Chương 4: Các kỹ thuật mã hóa thông tin)

**2.**http://vi.wikipedia.org/wiki/RSA\_(m%C3%A3\_h%C3%B3a)

**3.** http://en.wikipedia.org/wiki/RSA

**4.**http://vi.wikipedia.org/wiki/Ch%E1%BB%AF\_k%C3%BD\_s%E1%BB%91

**5.** http://en.wikipedia.org/wiki/Digital\_signature

**6.**http://docs.oracle.com/javase/7/docs/api/java/math/BigInteger.html

**7.** http://vi.wikipedia.org/wiki/SHA

**8.**http://www.java2s.com/Tutorial/Java/0490\_\_Security/UseSHA1.htm

**9.**Internet