ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

BÁO CÁO TIỂU LUẬN

MẬT MÃ VÀ AN TOÀN DỮ LIỆU

ĐỀ TÀI: HỆ MÃ HÓA RC5

GIẢNG VIÊN: PGS.TS TRỊNH NHẬT TIẾN

HỌC VIÊN: VŨ THỊ NHẠN - K20HTTT

LỚP: INT 6010 2

Mã số: 13025011

Hà Nội, tháng 5 năm 2014

Mục Lục

HỆ MÃ HÓA RC5

1. Giới thiệu

Thuật toán mã hóa RC5 do giáo sư Ronald Rivest của đại học MIT công bố vào tháng 12 năm 1984. Đây là thuật toán mã hóa theo khóa bí mật. Ngay từ khi được giới thiệu RC5 được quan tâm rất nhiều do tính an toàn của nó.

Ngày nay truyền dữ liệu thông qua một kênh yêu cầu bảo mật hơn. An ninh đạt được tầm quan trọng hơn chỉ đơn giản là truyền. Đảm bảo truyền yêu cầu giải thuật mã hóa. Các yêu cầu thực hiện phần cứng của các thuật toán tiêu thụ điện năng ít hơn, phân bổ nguồn lực, tái cấu hình, và kiến trúc hiệu quả và hiệu quả chi phí. Mã hóa RC5 có yêu cầu công suất thấp và độ phức tạp thấp và độ trễ thấp, độ xử lý nhanh được ứng dụng nhiều trong giao dịch mạng và thương mại điện tử

2. Thuật toán

2.1. Định nghĩa các giá trị

RC5 được xác định là RC5-w/b/r trong đó:

+w : kích thước khối cần được mã hóa (giá trị chuẩn là 32 bit, ngoài ra ta có thể chọn 16 hay 64 bit).

+r : số vòng lặp (giá trị từ 0,1,...,255)

+b : chiều dài khóa theo byte (0 đến 255)

Các giá trị thường dùng là : w = 32, r = 20, còn chiều dài khóa có thể 16, 24, hay 32 byte.

Đối với tất cả các biến, các thao tác RC5-w-r-b trên khối w-bit sử dụng các toán tử cơ bản sau:

a + b : phép cộng module 2^w

a - b : phép trừ module 2^w

a xor b : phép toán xor

a <<< b : phép toán quay trái a sang trái ít nhất $\log_2 w$ bit của b

Trong thuật toán RC5 quá trình mã hóa và giải mã đều cần qua một quá trình quan trọng là quá trình mở rộng khóa.

2.2. Mở rộng khóa

Để tăng độ an toàn cũng như việc bảo vệ khóa bí mật cho người dùng. Việc mở rộng khóa là một chiều nên không thể suy ngược lại giá trị của khóa K khi biết được các giá trị của khóa mở rộng. Đây cũng chính là một đặc điểm nổi bật của thuật toán RC5.

Thuật toán mở rộng cho khóa K của người sử dụng thành một tập gồm 2(r+1) các khóa trung gian. Các khóa trung gian này được điền vào một bảng khóa mở rộng S. Do vậy, S là một bảng của t = 2(r+1) các giá trị nhị phân ngẫu nhiên được quyết định bởi khóa K. Nó sử dụng hai hằng số lý tưởng được định nghĩa :

P_w = Odd ((e - 2)2^w)

Q_w = Odd ((Ø- 1)2^w

Trong đó :

e = 2.178281828459... (dựa trên số logarithms tự nhiên)

Ø = 1.618033988749... (tỉ lệ vàng)

Odd (x) là số nguyên lẻ gần x nhất

Một số giá trị khác :

t = 2(r + 1) : số phần tử của bảng khóa mở rộng S.

u = w/8 : u là số lượng các byte của khối w

c = b/u

Quá trình mở rộng khóa bao gồm các bước sau:

+Bước 1 :

Chép khóa bí mật K[0,...,b-1] vào mảng L[0,...,c-1].

Thao tác này sử dụng u byte liên tục nhau của khóa K để điền vào cho L theo thứ tự từ byte thấp đến byte cao. Các byte còn lại trong L được điền vào giá trị 0.

Trong trường hợp b = c = 0, chúng ta sẽ đặt c về 1 và L[0] về 0.

+ **Bước 2 :**

Khởi tạo mảng S với một mẫu bit ngẫu nhiên đặc biệt, bằng cách dùng một phép tính số học module 2^w được quyết định bởi hằng số lý tưởng P_W và Q_w.

$S[0] = P_w$

For i = 1 to t - 1 do

$S[i] = S[i-1] + Q_w$

+ **Bước 3 :**

Trộn khóa bí mật của người sử dụng vào mảng L và S.

A = B = 0

i = j = 0

v = 3 * max{c,t}

For s=1 to v do {

A = S[i] = (S[i] + A + B) <<<3

B = L[j] = (L[j] + A + B) <<< (A + B)

i = (i + 1) mod (t)

j = (j + 1) mod (c)

}

Lưu ý rằng: hàm mở rộng khóa là một chiều, do vậy không dễ dàng tìm ra khóa K từ S.

Thuật toán mở rộng :

Input : khóa b được nạp và mảng c phần tử L[0,...,c-1]

Số vòng lặp r

Output : mảng khóa S[0,...,2r + 1]

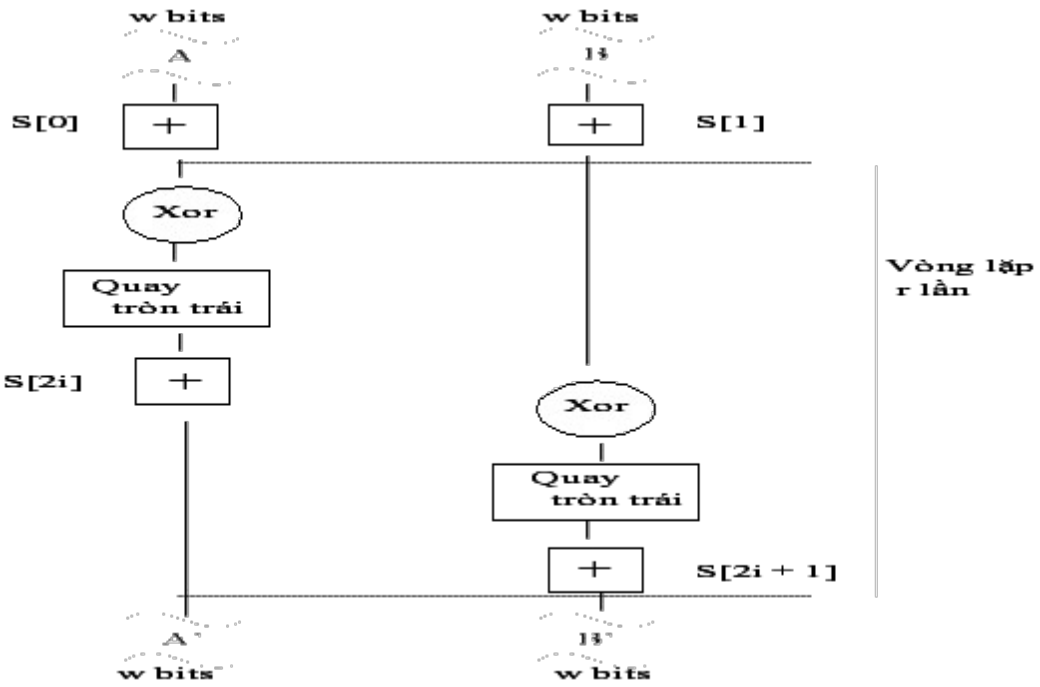
```
S[0] = Pw
For i = 1 to t - 1 do
        S[i] = S[i - 1] + Qw
A = B = 0
i = j = 0
V = 3 * max {c, t}
For s = 1 to v do {
          A = S[i] = (S[i] + A + B) <<< 3
          B = L[j] = (L[j] + A + B) <<< (A + B)
          i = (i + 1) mod (t)
          j = (j + 1) mod (c)
}
```

2.3. Quá trình mã hóa

Thuật toán sẽ mỗi lần mã hóa trên hai khối w bit, giả sử là A và B. Và sau quá trình mã hóa sẽ cho ra hai khối đã được mã hóa A' và B'.

Ban đầu A sẽ được cộng với giá trị khóa mở rộng S[0] và B sẽ được cộng với S[1]. Sau đó quá trình mã hóa sẽ thực hiện biến đổi A dựa vào giá trị của B bằng các phép toán Xor và quay tròn trái. Tiếp tục giá trị này sẽ được cộng tiếp với giá trị khóa mở rộng S[2]. Kết quả này được dùng để tiếp tục biến đổi giá trị của B giống như trên. Toàn bộ quá trình này sẽ được thực hiện r lần. Kết quả cuối cùng ở bước r sẽ là giá trị đã được mã hóa A', B'.

Quá trình mã hóa có thể được minh họa như sau :



Hình 3.6: Sơ đồ mã hóa với RC5

Thuật toán mã hóa:

Input : giá trị gốc được lưu trữ trong hai khối w-bit A, B

Số vòng lặp r

w-bit khóa vòng lặp S[0,...,2*r + 1]

Output : giá trị mã được lưu trong hai khối w-bit A', B'

```
                                    A = A + S[0]
B = B + S[1]
                                    For i = 1 to r do {
        A = ((A XOR B) <<< B) + S[2i]
        B = ((B XOR A) <<< A) + S[2i + 1]
}
A' = A
B' = B
```

Thuật toán giải mã :

Quá trình giải mã chính là quá trình đi ngược lại quá trình mã hóa để có được cái giá trị gốc.

Thuật toán giải mã như sau :

Input : giá trị mã được lưu trữ trong hai khối w-bit A', B'

Số vòng lặp r

w-bit khóa vòng lặp S[0,...,2r + 1]

Output : giá trị giải mã được lưu trong hai khối w-bit A, B

```
For i = r downto 1 do {
        B' = ((B' - S[2i + 1]) >>> A') XOR A'
        A' = ((A' - S[2i]) >>> B' XOR B'
}
B = B' - S[1]
A = A' - S[0]
```

2.4. **Đánh giá**

Thám mã RC5 :

+Theo kết quả đánh giá độ an toàn của các thuật toán thì RC5 với 12 vòng lặp và mã hóa khối 64-bit thì cung cấp độ an toàn tương đương với thuật toán DES khi thử với phương pháp giả mã, 2^{44} cho RC5 và 2^{43} DES.

Bảng mô tả số thao tác cần thực hiện để thám mã RC5 mã hóa 64 bit

Số vòng lặp	4	6	8	10	12	14	16	18
Thám mã Differential (với thông tin nguồn được chọn)	2^7	2^{16}	2^{28}	2^{36}	2^{44}	2^{52}	2^{61}	>

+ Khi số vòng lặp lên đến 18 thì việc thám mã trên lý thuyết là không thể thực hiện được (do đòi hỏi khoảng 2^{128} thao tác cho khối 64 bit). Do việc tăng thêm số vòng lặp là tăng thêm độ an toàn cho RC5. Người ta nhận xét rằng RC5 với 16 vòng lặp và mã hóa khối 64 bit có thể cung cấp độ an toàn rất tốt để chống lại các thuật toán thám mã.

Ưu điểm :

+RC5 là một thuật toán mã hóa khối với tốc độ nhanh được thiết kế cho việc sử dụng dễ dàng cho cả phần cứng lẫn phần mềm.

+ RC5 là một thuật toán được tham số hóa với : một biến mô tả kích thước khối, một biến cho số vòng quay, và một cho chiều dài khóa.

+ RC5 thì rất đơn giản : cơ chế mã hóa dựa trên ba toán tử chính : cộng, exclusive-or và quay. Vì thế, RC5 dễ cài đặt và phân tích hơn các thuật toán mã hóa khối khác.

+Một đặc điểm nỗi bật khác của RC5 là các thao tác quay sử dụng chặt chẽ các dữ liệu phụ thuộc với nhau nhằm tránh được các phép thám mã tuyến tính và vi phân.

+ Cơ chế mở rộng khóa của RC5 là một chiều. Do vậy các hacker khó có thể phục hồi lại khóa chính ngay cả khi đã xác định được bộ khóa mở rộng.

+ Mỗi quá trình mã hóa và giải mã của RC5 được thực hiện trên hai khối w bit do vậy có thể tăng tốc độ mã hóa.

Khuyết điểm :

Trên thực tế cho đến năm 1998 thì chưa có cách thám mã nào có thể giải mã được RC5. Tuy nhiên một vài nghiên cứu lý thuyết đã cung cấp một vài cách thám mã có thể thực thi. Họ dựa vào đặc điểm là số lượng vòng lặp trong RC5 thì không phụ thuộc vào tất cả các bit trong một khối. Bên cạnh đó RC5 được thiết kế rất đơn giản do cơ chế mã hóa chỉ dựa vào các phép toán cộng, exclusive-or và quay.