THUẬT TOÁN MÃ HÓA THÔNG TIN AFFINE VÀ PHẦN MỀM ỨNG DỤNG MÃ AFFINE SAU NÂNG CẤP

TS. Nguyễn Đăng Minh[1], Ths. Vũ Anh Tú[2]
[1]Khoa Công nghệ thông tin và Điện tử viễn thông, Trường Đại học Hòa Bình
[2] Trung tâm Công nghệ thông tin, Đại học Công nghiệp Hà Nội
Tác giả liên hệ: ndminh@daihochoabinh.edu.vn

Ngày nhận: 13/8/2022
Ngày nhận bản sửa: 22/9/2022
Ngày duyệt đăng: 26/9/2022

Tóm tắt

Bài viết trình bày khả năng bảo mật của một thuật toán mã hóa affine và cách đánh giá độ bảo mật - nguyên tắc Kerckhoffs. Độ bảo mật được xem là khả năng không thể giải mã văn bản đã được mã hóa bằng thuật toán đã biết và chỉ không biết chìa khóa mã.

Độ bảo mật phụ thuộc vào số chìa khóa mã có thể có và thời gian để phần mềm phá khóa có thể duyệt hết tất cả khóa mã có thể để tìm ra văn bản mã hóa từ văn bản mã.

Độ bảo mật còn phụ thuộc vào độ bất định (obscurity - trạng thái không rõ ràng, độ bất định) của bản giải mã khi duyệt tất cả khóa mã có thể cho dù có thể biết trước hướng nội dung của văn bản mã hóa.

Độ bảo mật được nâng cấp nhờ biến đổi cả hệ số a và hệ số b trong khi m tăng lên đến 256 đã làm cho số khóa mã có thể trở thành quá lớn và xác suất xuất hiện ký tự không còn giúp gì cho việc phá khóa.

Công bố một phần mềm áp dụng thuật toán có biến đổi a, m, b để chứng minh khả năng tăng độ bảo mật.

Từ khóa: *Bảo mật, Affine.*

The Affine cipher encrypting algorithm and modified Affine application software

Abstract

The article presents the confidentiality of the Affine cryptographic algorithm and the use of Kerckhoffs principles in assessing security capabilities. Security capability is defined as undecodable capacity of an encrypted text coded by a known algorithm providing that its encryption key stays unknown. Security level depends on the number of possible encryption keys and the time length for the key cracking software to looking up all possible encryption keys for the ciphertext within encrypted documents. The security also depends on the obscurity of decryption when all possible encryption keys are traversed given that the ciphertext has its content direction predictable. The enhanced security by transforming both a and b coefficients while m increases to 256 made the number of possible encryption keys excessively large and the probability of a character occurrence was no longer helpful to decryption. Introducing a software that applies an algorithm with transformed a, m, b could help demonstrate higher confidentiality.

Keywords: *Security, affine.*

Mở đầu

Thế giới bước vào kỷ nguyên cách mạng khoa học công nghệ 4.0. Hầu hết thông tin cho dù là thông tin chung hay thông tin thuộc cơ sở dữ liệu đều được lưu trữ trên máy tính và chúng có thể được truyền từ người này sang người khác, cơ quan này đến cơ quan khác dưới dạng văn bản hoặc dưới dạng nào đó khác. Nhu cầu cần giữ bí mật để người khác không thể hiểu

được nhiều văn bản có tính cá nhân hoặc tập thể ngày càng cao. Việc xuất hiện những thông tin mà lộ bí mật các thông tin đó có thể kéo theo những thiệt hại khó lường. Ngành khoa học mã hóa đã phát triển từ rất sớm và là ngành nghiên cứu các kỹ thuật toán học cho tất cả các khía cạnh của an toàn thông tin bao gồm:

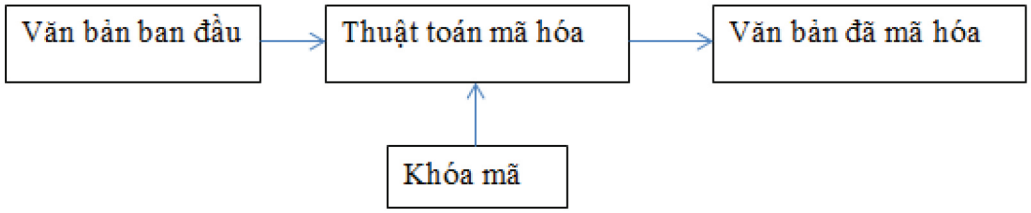
- Tính bảo mật hoặc quyền riêng tư;
- Toàn vẹn dữ liệu;
- Xác thực;
- Không thể bác bỏ.

Có nhiều cách mã hóa khác nhau. Trong [1], có đề xuất một phương pháp mã hóa XOR1 với các tính năng khác biệt. Theo một cách nhìn nhận nào đó, thuật toán XOR1 cho ta khả năng bảo mật tối đa. Khả năng giải mã văn bản đã mã hóa bằng phương pháp này gần như không thể nếu không có khóa mã [2-4].

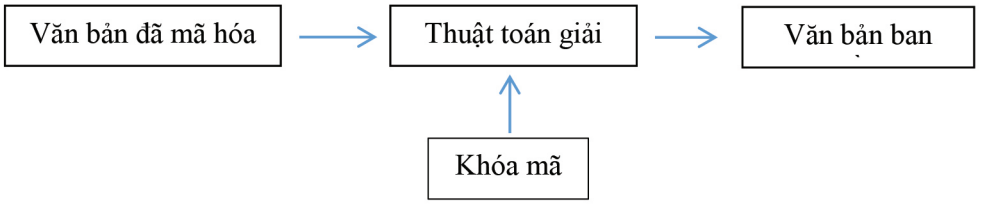
Về nguyên tắc, để bảo mật thông tin, cần giữ kín phương pháp mã hóa. Thông thường, mỗi phương pháp mã hóa sẽ bao gồm bản thân thuật toán mã hóa và khóa mã. Vì văn bản sau mã hóa phải truyền từ điểm này đến điểm khác hoặc giữ trên máy tính nên về nguyên tắc, nó không được bảo mật. Đó là lí do phải mã hóa các thông tin quan trọng và cần bảo mật. Để thấy độ mật của một phương pháp mã hóa, cần xét lại cả quá trình mã hóa, truyền tin và giải mã.

Như thế, hệ thống mã hóa biến một văn bản tường minh (văn bản nguồn) thành văn bản mà người thường không hiểu được văn bản gốc, gọi là văn bản mã hóa. Để giải mã, người giải mã cần văn bản đã được mã hóa, nguyên tắc mã hóa (thuật mã hóa, thuật toán mã hóa) và khóa mã.

Mã hóa thông tin và giải mã thông tin đã mã hóa xảy ra theo sơ đồ sau:



Hình 1. Quá trình mã hóa



Hình 2. Quá trình giải mã

Mục đích của mã hóa là giấu thông tin nhờ giữ bí mật cách (phương pháp) mã hóa và dấu khóa mã. Người ta cho rằng việc giữ bí mật phương pháp mã hóa khó thực hiện vì những lý do rất khác nhau. Có thể đối phương biết được nhờ những biện pháp phi khoa học như tình báo hay biện pháp khác. Vì vậy, để đánh giá độ bảo mật của phương pháp mã hóa, người ta cho rằng chỉ đánh giá khả năng giữ được bí mật khi công khai phương pháp mã hóa.

Thảo luận trên đây dựa trên nguyên tắc Kerckhoffs. Nguyên tắc Kerckhoffs là một quy tắc để phát triển các hệ thống mật mã, theo đó, khả năng bảo mật của một thuật toán được xác định khi chỉ khóa mã được giữ bí mật. Nghĩa là, thuật toán mã hóa phải được công khai. Nguyên tắc này được xây

dựng lần đầu tiên vào thế kỷ 19 bởi nhà mật mã người Hà Lan Auguste Kerkgoffs và được áp dụng cho đến ngày nay [6].

N**guyên tắc Kerkgoff**.

Nguyên tắc Kerkgoffs có 6 điểm cơ bản:

(1). Hệ thống mã phải là hệ thống thực sự, nếu như nó không phải là một hệ thống toán học, không thể bẻ khóa được;

(2). Không cần phải giữ bí mật hệ thống mã hóa. Khi những bí mật của hệ thống rơi vào tay đối phương thì cũng không gây hại gì cho việc bảo mật thông tin;

(3). Giữ và truyền khóa mã không nhất thiết phải viết trên giấy, những người sử dụng hệ thống có thể thay đổi khóa mã cho phù hợp với nhu cầu sử dụng;

(4). Hệ thống có thể dùng được cho thiết bị truyền telex;

(5). Hệ thống có thể dễ dàng di chuyển. Hệ thống có thể không cần phải nhiều người đồng thời tham gia;

(6). Hệ thống phải dễ sử dụng. Không cần quá nhiều công sức để hiểu và vận hành.

Mã Affine

Trong mật mã Affine, đầu tiên, bảng chữ cái của thông điệp cần mã hóa có kích thước m sẽ được chuyển thành các con số tự nhiên từ 0 đến m-1. Sau đó, dùng một hàm modul để mã hóa và chuyển thành bản mã.

Hàm mã hóa cho một ký tự theo thuật mã hóa affine thực hiện theo hàm:

y=E(x)=(ax+b) mod m(1). Trong đó, x là số cần mã hóa; a, b là các tham số mã hóa; y là số mã hóa của x; E(x) gọi là hàm mã hóa.

Ví dụ: Trong bảng mã ascii thường dùng, nếu chọn a=2, b=14 thì với ký tự "A" có mã ascii=65, áp dụng hàm trên ta được: E(65)= 2*65+14=144; 144 là ascii của ký tự "É".

Trong các ngôn ngữ lập trình khác nhau, hàm modul có thể có các ký hiệu khác nhau. Với m là kích thước của bảng chữ cái (ví dụ trong tiếng Anh có 26 chữ cái, nên m=26), a và b quyết định thứ tự của bảng chữ cái mã hóa nên nó quyết định việc chuyển đổi từ bản rõ sang bản mã hóa. Để việc mã hóa và giải mã luôn xảy ra đơn trị (mỗi ký tự ban đầu chỉ có thể đổi thành một ký tự tương ứng mã hóa nào đó), giá trị a cần được chọn sao cho a và m là các số nguyên tố cùng nhau, nghĩa là, ước số chung lớn nhất của (a,m)=1. Giá trị ước số chung lớn nhất của a, m=1. Khi đó, hàm giải mã sẽ là:

x=D(y)=(a^{-1}(y-b) mod m) (2).

Các tham số như trong (1).

với a^{-1} là nghịch đảo của a theo mô đun m. Tức là, aa^{-1}mod m=1.

Mã hóa, cho dù là bằng phương pháp nào thì thực chất cũng là sự biến đổi tương ứng từ một bảng chữ cái này sang một bảng chữ cái khác nó nhưng khác nhau về thứ tự. Sự khác nhau đó dẫn đến nguyên bản đọc và hiểu nhưng bản mã hóa đọc không hiểu.

Độ bảo mật của mã affine

Trong nhiều tài liệu, người ta cho rằng độ bảo mật của mã affine không cao vì các lý do sau đây:

- Số khóa mã quá ít.

Thực vậy, khi chọn bảng chữ cái, người ta thường chọn bảng chữ cái tiếng Anh và chỉ chọn chữ in hoa [2-4]. Kết quả có: m=26(ABCDEFGHIJKLMNOPQRSTUVWXYZ), dẫn đến chỉ có thể chọn a<m. Với điều kiện (a,m)=1, a chỉ có thể là: 3, 5, 7, 9, 11, 15, 17, 19, 21, 25, tức 10 giá trị khác nhau. Như vậy, ta có: 10*25 lựa chọn, tức là, có 250 khóa mã. Số khóa mã như vậy không lớn, dẫn đến độ mật của thuật toán mã hóa affine không cao.

- Độ bảo mật không cao còn xuất phát từ nguyên nhân tiềm ẩn ngay trong phương pháp mã hóa. Khi mỗi ký tự ở bản rõ luôn tương ứng với một ký tự trong bản mã hóa thì có nhiều cách không cần khóa mã, vẫn có thể giải mã [2] chí ít là căn cứ vào tần số xuất hiện của các ký tự để phỏng đoán dấu cách, các từ, rồi dẫn đến giải mã cả văn bản.

Nâng cao độ bảo mật của mã affine

Để nâng cao độ bảo mật của mã affine, cần tránh hiện tượng mỗi ký tự giống nhau ở bản chính sẽ chuyển hóa thành ký tự giống nhau ở bản mã hóa. Ví dụ:

BẢN RÕ: ASDFG ASDFG

Bản mã hóa: CUFHI"CUFHI

Chữ A ở bản chính sẽ chuyển thành chữ C ở bản mã hóa, chữ S ở bản chính sẽ biến thành chữ U ở bản mã hóa.

Để tránh hiện tượng này, cần bảo đảm cho các chữ ở những vị trí khác nhau có thể sẽ có các tham số hệ số a hoặc b khác nhau hoặc cả a và b khác nhau. Điều này dẫn đến cùng một ký tự như nhau ở bản chính nhưng ở vị trí khác nhau sẽ biến thành các chữ khác nhau. Độ bảo mật sẽ được tăng lên nhờ không còn nguyên nhân độ bảo mật thấp thứ 2.

Khi thay đổi a và b, ta đã tạo nên các khóa mã mới. Với m=256, ta có thể chọn b bất kỳ sao cho 256>b>0 (255 số), còn a có thể chọn bất kỳ số nào trong danh sách sau: a=3, 5, 7, 9,…, 255 (127 số) để chúng là các số nguyên tố cùng nhau với m. Với một ký tự ở một vị trí bất kỳ như thế, ta có, 255*127 = 32385. Một từ có 10 ký tự thì số khóa mã có thể là 3238519^{10}. Với số khóa mã lớn như vậy, độ bảo mật tăng lên rất đáng kể.

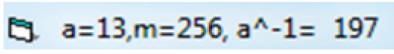
Phần mềm thực hiện thuật toán mã hóa nâng cao

Phần mềm được xây dựng dựa trên các phân tích trên đây trong môi trường Microsoft Visual Studio.

- Phần mềm tính a^{-1} modul m[1]

```
Private Sub Invert1_Click()
 Me.Caption = ""
 For i = 1 To So_m
 If (13 * i Mod 256 = 1) Then
 Me.Caption = " a=13,m=256, a^-1= "
+ Str(i)
 Invert = i
 End If
 Next
End Sub
```

Kết quả:Khi chạy Private Sub Invert1_Click(), ta được:



a^{-1} modul 256=197.

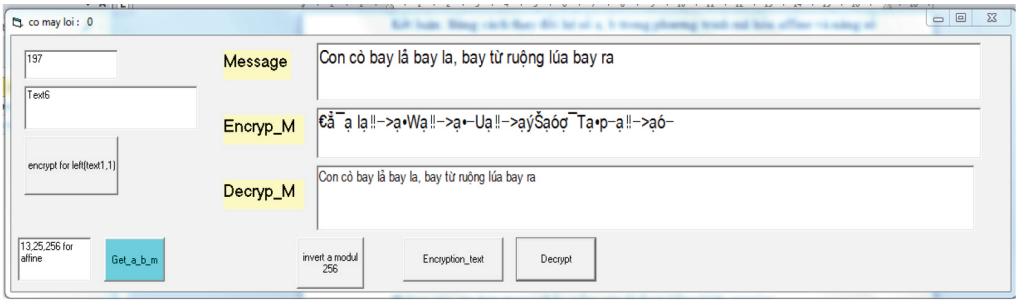
- Phần mềm affine thông thường

Để tránh nhầm lẫn giữa các biến, phần mềm này sử dụng các ký hiệu sau:

```
Private Sub Command6_Click()
' Tìm a-1
For i = 1 To So_m
If (He_so_a * i Mod So_m = 1) Then
Me.Caption = Str(i) + Me.Caption
Invert = i
Text1.Text = i
i = So_m
End If
Next
Van_ban = Text3.Text
Me.Caption = ""
Text4.Text = ""
For i = 1 To Len(Van_ban)
Ma_hoa = Chr((He_so_a *
Asc(Mid(Van_ban, i, 1)) + So_b) Mod
So_m)
Text4.Text = Text4.Text + Ma_hoa
Next
Ma_hoa = Text4.Text
```

End Sub Invert là nghịch đảo của a theo modul m, a=13.

[1] Invert là nghịch đảo của a theo modul m, a=13.



Phần mềm thay đổi b để nâng cao độ bảo mật

Giao diện của phần mềm:

Trong giao diện: Message là thông tin ban đầu, Encryp_M là thông tin được mã hóa, Decryp_M là giải mã. Khi phần mềm không lỗi, Message và Decryp_M phải như nhau.

Gõ dòng chữ "con cò bay lả bay la, bay từ ruộng lúa bay ra" có các chữ "c" giống nhau, "co" giống nhau, "bay" giống nhau, nhưng ta thấy các ký tự mã hóa cho các ký tự đó khác hẳn nhau ở mục Encryp_M. Phần mềm cho chương trình trên như sau[2]:

```
Dim x, y, m As Integer
Dim Van_ban As String
Dim Ma_hoa As String
Dim Giai_ma As String
Dim Invert As Integer
Dim giai_Ma1 As Double
Dim giai_Ma2 As Double
Dim so As Single
Dim k As Integer
Dim He_so_a, So_b, So_m As Integer

Private Sub Command1_Click()
m1 = (13 * Asc(Left(Text1.Text, 1)) + 25)
m2 = m1 Mod 256
Text6.Text = Str(Asc(Left(Text1.Text, 1))) + " : " + Str(m1) + " : " + Str(m2)
End Sub

Private Sub Form_Load()
He_so_a = 13
So_b = 25
So_m = 256
End Sub

Private Sub Get_a_b_m_Click()
Dim tg As Variant
key_word = Text2.Text
tg = Split(key_word, ",")
He_so_a = Val(tg(0))
So_b = Val(tg(1))
So_m = Val(tg(2))
Me.Caption = "he so a= " + Str(He_so_a) + " so b= " + Str(So_b) + " so m= " + Str(So_m)
End Sub

Private Sub Invert1_Click()
Me.Caption = ""
For i = 1 To So_m
If (13 * i Mod 256 = 1) Then
Me.Caption = " a=13,m=256, a^-1= " + Str(i)
Invert = i
End If
Next
End Sub

Private Sub Command6_Click()
'Dao a
For i = 1 To So_m
If (He_so_a * i Mod So_m = 1) Then
Me.Caption = Str(i) + Me.Caption
```

[2] Xem hình trên

```
Invert = i
Text1.Text = i
i = So_m
End If
Next
Van_ban = Text3.Text
Me.Caption = ""
Text4.Text = ""
For i = 1 To Len(Van_ban)
Ma_hoa = Chr((He_so_a * Asc(Mid(Van_ban, i, 1)) + So_b) Mod So_m)
Text4.Text = Text4.Text + Ma_hoa
Next
Ma_hoa = Text4.Text

End Sub

Private Sub Decrypt_Click()
Me.Caption = ""
Text5.Text = ""
k = 0
For i = 1 To So_m
If (He_so_a * i Mod So_m = 1) Then
Me.Caption = Str(i) + Me.Caption
Invert = i
Text1.Text = i
i = So_m
End If
Next
For i = 1 To Len(Ma_hoa)
tg = Mid(Ma_hoa, i, 1)
tg1 = Asc(tg)
If (tg1 - So_b < 0) Then
For n = 0 To 10
If n * So_m + tg1 - So_b > 0 Then
tg1 = n * So_m + tg1
End If
Next
End If

giai_Ma2 = (((tg1) - So_b) Mod So_m)
giai_Ma1 = Invert * giai_Ma2
giai_Ma1 = giai_Ma1 Mod So_m
Text5.Text = Text5.Text + Chr(giai_Ma1)
Next
Me.Caption = Text5.Text
Me.Caption = "co may loi : " + Str(k)
End Sub[3]
```

Kết luận

Bằng cách nâng bộ mã ký tự lên 256 và lựa chọn cách thay đổi hệ số a,b và áp dụng trong phương trình mã hóa affine các hàm chr() và asc()(trong microsoft VisualBasic), ta có thể nâng khả năng bảo mật thông tin khi sử dụng mã này. Phần mềm mã hóa có thể dùng để mã hóa các tài liệu để lưu trên máy tính cũng như truyền tin giữa các thiết bị trên mạng.

Phông của văn bản trong phần mềm này là font tiếng Việt.vnarial.

Khi dùng phương pháp này, cho dù công bố phương pháp mã hóa nhưng cũng thật khó thử hết các khóa mã có thể có để tìm văn bản thật.

Độ bảo mật tăng lên rõ rệt!

Tài liệu tham khảo

[1]. Nguyễn Đăng Minh, Một số ứng dụng của hàm logic trong mã hóa thông tin, *Tạp chí Khoa học và Công nghệ Trường Đại học Hòa Bình*, số 3/2022.

[2]. Nguyễn Bình, Ngô Đức Thiện (2013), Giáo trình *Cơ sở mật mã học*, NXB Bưu chính viễn thông, Hà Nội.

[3]. D.R. Stinson (1995), *Mật mã, lý thuyết và thực hành*, Người dịch: Nguyễn Bính.

[4]. *Баричев Сергей,* КРИПТОГРАФИЯ БЕЗ СЕКРЕТОВ;

[5]. В. В. Ященко, Введение в криптографию, Москва Издательство МЦНМО-2012.

[3] Phần mềm còn trong quá trình hoàn thiện nên có nhiều thông tin đưa lên form.caption

[6]. Принцип Керкгоффса

https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BD%D1%86%D0%B8%D0%BF_%D0%9A%D0%B5%D1%80%D0%BA%D0%B3%D0%BE%D1%84%D1%84%D1%81%D0%B0

[7]. Nguyễn Xuân Dũng (2007), *Bảo mật thông tin: Mô hình & ứng dụng*, NXB Thống kê.