DOI: 10.15625/vap.2016.00022

GIẢI PHÁP PHÁT TRIỂN THUẬT TOÁN MẬT MÃ KHÓA ĐỐI XỨNG TỪ CÁC HỆ MÃ LŨY THỪA VÀ MÃ OTP

Lưu Hồng Dũng [1], Nguyễn Vĩnh Thái[2], Tống Minh Đức[3], Bùi Thế Truyền[4]

[1] Khoa CNTT, Học viện Kỹ thuật Quân sự
[2] Viện CNTT, Viện Khoa học và Công nghệ Quân sự
[3] Khoa CNTT, Học viện Kỹ thuật Quân sự
[4] Viện CN Mô phỏng, Học viện Kỹ thuật Quân sự

luuhongdung@gmail.com, nguyenvinhthai@gmail.com, ductm08@gmail.com, buithetruyen@gmail.com

TÓM TẮT— *Bài báo đề xuất giải pháp xây dựng thuật toán mật mã khóa đối xứng từ việc phát triển hệ mã sử dụng khóa 1 lần - OTP (One - time Pad) kết hợp với các hệ mã lũy thừa. Ưu điểm của thuật toán mới đề xuất là có tính an toàn và hiệu quả thực hiện cao tương tự OTP, đồng thời với việc sử dụng khóa hoàn toàn giống như các hệ mã khối được sử dụng trong thực tế: DES, AES,...*

Từ khóa— *Mật mã khóa đối xứng, thuật toán mật mã khóa đối xứng, thuật toán mật mã sử dụng khóa một lần, mật mã OTP*.

I. ĐẶT VẤN ĐỀ

Hầu hết các hệ mã khóa đối xứng đều được thiết kế dựa trên 2 nguyên tắc cơ bản của Claude Shannon, đó là tính hỗn loạn (*confusion*) và tính khuếch tán (*diffusion*). Trong bài báo này, nhóm tác giả đề xuất giải pháp xây dựng hệ mã khóa đối xứng theo nguyên tắc mã hóa của hệ mã sử dụng khóa 1 lần (OTP) [1-5] kết hợp với hệ mã lũy thừa như: RSA [6], ElGamal [7],... nhằm giải quyết các yêu cầu sau:

- Tốc độ thực hiện cao, dễ cài đặt trên các hệ nền khác nhau, cũng như cho phép tích hợp hiệu quả trên các thiết bị có kích thước, dung lượng nhớ nhỏ và năng lực tính toán hạn chế.
- Có khả năng loại trừ các dạng tấn công đối với các hệ mã khóa đối xứng đã biết trên thực tế [8].

Bài báo cũng đề xuất 2 thuật toán xây dựng theo giải pháp mới đề xuất, cho thấy tính khả thi của giải pháp cũng như về cơ bản, các thuật toán ở đây có thể đáp ứng tốt các yêu cầu đã đặt ra.

II. PHÁT TRIỂN THUẬT TOÁN MẬT MÃ KHÓA ĐỐI XỨNG TỪ CÁC HỆ MÃ LŨY THỪA VÀ MÃ OTP

A. Các hệ mã cơ sở

1. Hệ mã sử dụng khóa 1 lần OTP

Mật mã sử dụng khóa 1 lần – OTP (One – Time Pad) được đề xuất bởi Gilbert Vernam và Joseph Mauborgne vào năm 1917. Nguyên tắc căn cản của mã OTP là sử dụng 1 khóa mật có độ dài bằng với độ dài của bản tin cần mã hóa (bản rõ), các bit của bản mã nhận được từ việc cộng loại trừ (XOR) trực tiếp các bit của bản rõ với các bit của khóa mật:

$$C = P \oplus K$$

Trong đó:

$P = (P_1 P_2 ... P_i ... P_n)$: Bản rõ n bit.

$K = (K_1 K_2 ... K_i ... K_n)$: Khóa mật n bit.

$C = (C_1 C_2 ... C_i ... C_n)$: Bản mã n bit.

Lý thuyết của Claude E. Shannon [9] đã chỉ ra OTP là một loại mã có độ mật hoàn thiện (*Perfect Secrecy*). Hiện tại, mật mã OTP vẫn được xem là loại mã an toàn tuyệt đối và chưa có kết quả nào được công bố cho thấy có thể phá được loại mã này nếu mỗi khóa chỉ dùng để mã hóa 1 bản tin duy nhất và các khóa được chọn có tính chất ngẫu nhiên.

Trong bài báo đề xuất phát triển một hệ mật mã có nguyên tắc mã hóa và giải mã tương tự OTP, nhằm giải quyết các yêu cầu cao về tính an toàn bảo mật và tốc độ cũng như hiệu quả khi thực hiện.

2. Các hệ mã lũy thừa

Mật mã OTP có độ an toàn rất cao, song độ an toàn của OTP lại phụ thuộc vào một thực tế là mỗi khóa chỉ được sử dụng cho 1 lần mã hóa. Với cơ chế mã hóa của OTP, rõ ràng nó không thể đứng vững trước tấn công với chỉ bản rõ đã biết, vì khóa mật dễ dàng tính được từ phép cộng loại trừ bản rõ và bản mã tương ứng:

$$K = P \oplus C$$

Do vậy, cần phải tạo một khóa mới và thông báo nó trên một kênh an toàn với mỗi bản tin trước khi gửi đi. Điều đó gây khó khăn cho vấn đề quản lý khóa và hạn chế khả năng sử dụng rộng rãi OTP. Để khắc phục hạn chế trên

của OTP, các thuật toán ElGamal và RSA áp dụng nguyên tắc mã hóa của các hệ mã lũy thừa nhằm cho phép sử dụng khóa mật nhiều lần tương tự các hệ mã khóa đối xứng khác.

a) Hệ ElGamal

Đây là 1 hệ mật mã khóa công khai được T. ElGamal đề xuất năm 1985, hệ mật này được xây dựng dựa trên tính khó của bài toán logarit rời rạc như sau:

- Các thành viên trong cùng hệ thống chọn chung một số nguyên tố p và phần tử sinh g của nhóm Z_p^*.
- Mỗi thành viên chọn cho mình 1 khóa bí mật *x* trong khoảng (1,p) và tính khóa công khai tương ứng: $y = g^x \bmod p$
- Giả sử A muốn gửi cho B bản tin M với: $M < p$, A chọn ngẫu nhiên 1 giá trị *k* trong khoảng (*1,p-1*). A tính: $r = g^k \bmod p$, $C = M \times (y_B)^k \bmod p$, trong đó: $y_B = g^{x_B} \bmod p$ là khóa công khai của B, rồi gửi cho B cặp: (*r,C*).
- B sử dụng khóa bí mật x_B của mình để giải mã bản tin bằng cách tính: $r^{-x_B} \bmod p$ rồi nhân với *C*.

Tính an toàn của hệ ElGamal dựa trên giả thiết không thể tính được $g^{k.x_B} \bmod p$ nếu chỉ biết $g^k \bmod p$ và $g^{x_B} \bmod p$. Trên lý thuyết, có thể có cách sử dụng tri thức về $g^k \bmod p$ và $g^{x_B} \bmod p$ để tính $g^{k.x_B} \bmod p$. Nhưng hiện tại, chưa có cách nào để tính $g^{k.x_B} \bmod p$ mà không phải giải bài toán logarit rời rạc.

b) Hệ RSA

RSA cũng là 1 hệ mã khóa công khai do R. Rivest, A. Shamir và L. Adleman phát minh năm 1977, hệ này có nguyên tắc hoạt động như sau:

- Chọn 2 số nguyên tố *p, q* lớn và mạnh. Tính: $n = p \times q$ và $\phi(n) = (p-1) \times (q-1)$
- Chọn e trong khoảng $(1, \phi(n))$ và $\gcd(e, \phi(n)) = 1$
- Tính $d = e^{-1} \bmod \phi(n)$. Công khai: (e,n), giữ bí mật: d và hủy các giá trị: *p, q,* $\phi(n)$.
- Để gửi thông điệp P (P < n) cho người có khóa công khai (e,n), người gửi tính: $C = P^e \bmod n$.
- Để giải mã, người nhận sử dụng khóa bí mật của mình tính: $P = C^d \bmod n$.
- Trong hệ mật RSA, bài toán phân tích một số nguyên lớn ra các thừa số nguyên tố được sử dụng để hình thành cặp khóa công khai/bí mật (e,d). Thực vậy, do *p, q,* $\phi(n)$ được giữ bí mật, nên chỉ có thể tìm được khóa bí mật *d* từ khóa công khai (*e,n*) nếu có thể phân tích được: $n = p \times q$. Như vậy, tính an toàn của hệ RSA được thiết lập dựa trên giả thiết về tính khó giải của bài toán này.

B. Nguyên tắc xây dựng

1. Mã hóa và giải mã theo khối với thuật toán OTP

Tuy có độ an toàn và tốc độ mã hóa cao cũng như khả năng cài đặt dễ dàng, nhưng mã OTP đòi hỏi không gian khóa và bản rõ phải bằng nhau và kích thước của khóa cũng phải bằng kích thước của bản rõ đã gây khó khăn cho việc quản lý khóa trong các ứng dụng thực tế.

Bài báo đề xuất giải pháp xây dựng một hệ mã khóa đối xứng theo nguyên tắc của mật mã OTP, nhằm giải quyết các yêu cầu cao về tính an toàn bảo mật và tốc độ cũng như hiệu quả khi thực hiện. Ở đây, bản rõ *P* được mã hóa dưới dạng *n* khối dữ liệu P_i có kích thước *m* bit:

$$P = \{P_1, P_2, ..., P_i, ..., P_n\},\ i = \overline{1,n},\ |P_i| = m\ bit$$

Do đó, bản mã *C* cũng được giải mã dưới dạng *n* khối dữ liệu C_i có kích thước *m* bit:

$$C = \{C_1, C_2, ..., C_i, ..., C_n\},\ i = \overline{1,n},\ |C_i| = m\ bit$$

Thuật toán mã hóa và giải mã cơ bản chỉ dựa trên phép XOR tương tự mật mã OTP:

$$C_i = P_i \oplus K_i, i = \overline{1,n}$$

và:

$$P_i = C_i \oplus K_i, i = \overline{1,n}$$

Trong đó K_i là khóa mã hóa/giải mã sử dụng 1 lần tương ứng với mỗi khối dữ liệu P_i và C_i. Có thể thấy rằng, về mặt hình thức việc mã hóa/giải mã ở đây được thực hiện theo khối *m* bit như các hệ mã khối thông thường (DES, AES,...) thay vì từng bit như ở mã OTP. Tuy nhiên, nếu có thể tạo ra các khóa K_i là khác nhau đối với mỗi khối dữ liệu cần mã hóa/giải mã thì việc mã hóa và giải mã ở hệ mã được đề xuất và mã OTP là hoàn toàn như nhau, và hoàn toàn khác với việc sử dụng 1 khóa để mã hóa và giải mã cho tất cả các khối dữ liệu của bản tin trong các hệ mã khối thông thường.

2. Sử dụng 2 khóa khác nhau để mã hóa/giải mã bản tin

Như đã đề cập ở mục trước, khóa K_i trong thủ tục mã hóa được sinh ra từ các khối dữ liệu đứng trước P_{i-1} bằng một hàm sinh số ngẫu nhiên F_1:

$$K_i = F_1(P_{i-1}),\ i = \overline{2,n}$$

Với phương pháp tạo khóa này, khóa K_{OT} sử dụng 1 lần cho việc mã hóa bao gồm:

$$K_{OT} = \{K_2, K_3, ..., K_i, ..., K_n\}$$

Do đó, việc mã hóa theo OTP với K_{OT} cũng chỉ thực hiện từ khối thứ 2 trở đi:

$$C_i = P_i \oplus K_i, i = \overline{2,n}$$

Như vậy sẽ đặt ra vấn đề tạo khóa và mã hóa cho khối dữ liệu đầu tiên của bản rõ. Hơn nữa, để thủ tục mã hóa và giải mã có thể thực hiện với cùng phép XOR như mã OTP thì các khóa K_i tương ứng với các khối bản mã C_i trong thủ tục giải mã cũng phải được sinh ra theo cùng 1 phương pháp với thủ tục mã hóa. Điều này có thể thực hiện được nếu khối dữ liệu đầu tiên của bản tin được mã hóa và giải mã theo một phương pháp an toàn nào đó. Giải pháp ở đây là sử dụng các hệ mã lũy thừa có các tham số được giữ bí mật hoàn toàn và chính các tham số này sẽ được sử dụng làm khóa bí mật chia sẻ K_S để mã hóa cho khối dữ liệu đầu tiên của bản rõ:

$$C_1 = F_2(P_1, K_s)$$

và cũng chính K_S sẽ được sử dụng để giải mã cho khối dữ liệu đầu tiên của bản rõ:

$$P_1 = F_2^{-1}(C_1, K_s)$$

ở đây: F_2^{-1} là hàm ngược của F_2.

Sau khi khối dữ liệu đầu tiên của bản mã được giải mã, các khóa K_i để giải mã cho các khối tiếp theo sẽ được sinh ra theo chính phương pháp đã sử dụng trong thủ tục mã hóa:

$$K_i = F_1(P_{i-1}),\ i = \overline{2,n}$$

và các khối còn lại của bản mã được giải mã theo thuật toán OTP:

$$P_i = C_i \oplus K_i, i = \overline{2,n}$$

Như vậy, ở hệ mã được đề xuất khóa bí mật K sẽ bao gồm 2 thành phần có chức năng phân biệt:

$$K = \{K_S, K_{OT}\}$$

Trong đó: K_S là khóa bí mật chia sẻ giữa các đối tượng tham gia trao đổi thông tin mật, khóa này được sử dụng để chỉ mã hóa và giải mã cho riêng khối dữ liệu đầu tiên của bản tin, khóa này được sử dụng dài hạn tương tự khóa bí mật chia sẻ của các hệ mã khối khác như DES, AES,... Trong khi đó, K_{OT} là khóa sử dụng chỉ 1 lần với 1 bản tin và khóa này được sử dụng để mã hóa và giải mã cho các khối dữ liệu từ thứ 2 trở đi của bản tin.

3. Khóa mã hóa sử dụng 1 lần là khóa tự sinh

Mục đích của việc mã hóa bản tin theo các khối bit là để tạo các khóa K_i từ các khối dữ liệu đứng trước P_{i-1} bằng một hàm sinh số ngẫu nhiên F_1:

$$K_i = F_1(P_{i-1}),\ i = \overline{2,n}$$

Hơn nữa, ở thủ tục giải mã, sau khi khối đầu tiên đã được giải mã, khóa K_i để giải mã cho các khối tiếp theo cũng được tạo ra bằng chính phương pháp này. Do đó, thủ tục mã hóa và giải mã của hệ mã đề xuất ở đây có thể được thực hiện với cùng một thuật toán tương tự các hệ mã khối điển hình như DES, AES,...

Thực tế, trong 1 bản tin cần mã hóa có thể bao gồm nhiều khối P_i có giá trị giống nhau, để K_i không bị lặp lại thì việc chỉ sử dụng hàm F_1 là không đủ, khi đó K_i cần phải được tạo ra từ P_{i-1} và 1 giá trị ngẫu nhiên V nhờ hàm F_1:

$$K_i = F_1(P_{i-1}, V),\ i = \overline{2,n}$$

C. Xây dựng thuật toán mật mã khóa đối xứng theo giải pháp đề xuất

Mục này đề xuất xây dựng 2 dạng thuật toán khác nhau. Thuật toán thứ nhất – ký hiệu: MTA 16.5 – 01, được thiết kế để làm việc ở chế độ mã dòng, thuật toán dạng 2 – ký hiệu: MTA 16.5 – 02, làm việc như các hệ mã khối thông thường nhưng hỗ trợ khả năng xác thực nguồn gốc và tính toàn vẹn của bản tin được mã hóa.

1. Thuật toán MTA 16.5 – 01

a) Dữ liệu:

III. BẢN RÕ *P* ĐƯỢC MÃ HÓA DƯỚI DẠNG CÁC KHỐI DỮ LIỆU P_I CÓ KÍCH THƯỚC 128 BIT:

$$P = \{P_1, P_2, ..., P_i, ..., P_n\},\ i = \overline{1,n},\ |P_i| = 128\ bit$$

Bản mã *C* cũng được giải mã dưới dạng các khối dữ liệu C_i 128 bit:

$$C = \{C_1, C_2, ..., C_i, ..., C_n\},\ i = \overline{1,n},\ |C_i| = 128\ bit$$

a) Khóa:

Khóa bí mật bao gồm 2 phân khóa riêng biệt:

$$K = \{K_S, K_{OT}\}$$

Trong đó:

- Khóa bí mật chia sẻ K_S được sử dụng để mã hóa/giải mã khối dữ liệu đầu tiên của bản tin, bao gồm các thành phần:

$$K_S = (p, g, x)$$

Trong đó: *p* là 1 số nguyên tố lớn có $|p| = 128$ bit, *g* là phần tử sinh của nhóm Z_P^* và *x* là một giá trị được chọn ngẫu nhiên trong khoảng (*1, p*).

- K_{OT} là khóa sử dụng 1 lần để mã hóa/giải mã cho các khối còn lại của bản tin:

$$K_{OT} = \{K_2, K_3, ..., K_i, ..., K_n\},\ i = \overline{2,n},\ |K_i| = 128\ bit$$

Trong thuật toán đề xuất ở đây, K_{OT} là khóa tự sinh được tạo ra từ chính bản tin cần mã hóa/giải mã. Trong đó, các khóa con K_i để mã hóa/giải mã cho khối dữ liệu P_i/C_i được tạo ra từ khối dữ liệu đứng trước P_{i-1} và 1 vector khởi tạo *V* nhờ hàm băm MD5 [10] như sau:

$$K_i = MD5(P_{i-1}, V),\ i = \overline{2,n}$$

Ở đây: *V* là vector khởi tạo có giá trị được chọn ngẫu nhiên cho mỗi lần mã hóa bản tin, nhằm loại bỏ các trường hợp: $P_{1i} = P_{1j}$ dẫn tới: $K_{1i} = K_{1j}$. Ở đây: *i, j* là chỉ số định danh các bản tin khác nhau được mã hóa.

b) Thuật toán mã hóa:

- Sinh khóa mã hóa sử dụng 1 lần K_{OT}:
  [1]. Chọn ngẫu nhiên một giá trị *k* trong khoảng (*1,p*)
  [2]. Tính giá trị vector khởi tạo: $V = g^k \bmod p$
  [3]. Thủ tục sinh khóa K_{OT}:

```
for i =2 to n do
   begin
      K_i = MD5(P_{i-1} || V || P_{i-1} || V)
   end
```

- Mã hóa khối đầu tiên của bản rõ:
  [1]. Tính giá trị: $C_0 = P_1 \times (V)^x \bmod p$
  [2]. Tính giá trị: $E = MD5(P_1 \| V) \bmod p$
  [3]. Tính giá trị: $S = x^{-1} \times (k + E) \bmod p$
  Khối đầu tiên của bản mã: $C_1 = (C_0, E, S)$
- Mã hóa các khối từ 2 đến n:

```
for i = 2 to n do
   begin
      C_i = P_i ⊕ K_i
   end
```

- Bản mã nhận được:

$$C = \{C_1, C_2, ..., C_i, ..., C_n\},\ i = \overline{1,n},\ |C_i| = 128\ bit$$

Chú ý:

- Toán tử "||" sử dụng ở thủ tục sinh khóa K_{OT} và bước [2] của thủ tục mã hóa khối C_0 là phép toán ghép nối 2 xâu bit.

- Điều kiện để giải mã đúng khối đầu tiên là: $P_1 \le p$. Trong thực tế, có thể xảy ra một số trường hợp mà: $P_1 > p$ và kết quả giải mã sẽ bị sai. Do đó, ở bước [1] của thủ tục mã hóa khối đầu tiên của bản rõ có thể tính: $C_0 = (p - P_1)_2 \times (V)^x \bmod p$ thay vì: $C_0 = P_1 \times (V)^x \bmod p$, trong đó: $(p - P_1)_2$ là dạng mã bù 2 của $(p - P_1)$.

c) Thuật toán giải mã:

- Giải mã khối thứ nhất cuả bản mã:

[1]. Tính giá trị: $\overline{V} = g^{x.S} \times g^{-E} \bmod p$

[2]. Tính: $\overline{P_1} = C_0 \times (\overline{V})^{-x} \bmod p$

[3]. Tính: $\overline{E} = MD5(\overline{P_1} \| \overline{V}) \bmod p$

[4]. Nếu: $\overline{E} = E$ thì: $\overline{P_1} = P_1$. Khi đó sẽ chuyển sang thực hiện thủ tục sinh khóa và giải mã các khối từ 2 đến n. Ngược lại, nếu $\overline{E} \ne E$: kết thúc việc giải mã.

- Thủ tục sinh khóa và giải mã các khối từ 2 đến n được:

for i = 2 to n do

begin

$$K_i = MD5(P_{i-1} \| \overline{V} \| P_{i-1} \| \overline{V})$$

$$P_i = C_i \oplus K_i$$

end

Chú ý:

- Giá trị $g^x \bmod p$ có thể tính 1 lần và lưu trữ như 1 thành phần của K_S: $K_S = (p, g, x, y)$, ở đây: $y = g^x \bmod p$.
- Khi đó, giá trị $\overline{V}$ ở bước [1] của thuật toán giải mã được tính theo: $\overline{V} = y^S \times g^{-E} \bmod p$.

d) Tính đúng đắn của MTA 16.5 – 01

Điều cần chứng minh ở đây là: p số nguyên tố, $MD5: \{0,1\}^* \mapsto Z_q$ với: $|p| = |q| = 128\ bit$, $1 < x, g, k < p$, $y = g^x \bmod p$, $V = g^k \bmod p$, $E = MD5(P_1 \| V) \bmod p$, $S = x^{-1} \times (k + E) \bmod p$, $C_0 = P_1 \times (V)^x \bmod p$. Nếu: $\overline{V} = g^{-E} \times y^S \bmod p$, $\overline{P_1} = C_0 \times (\overline{V})^{-x} \bmod p$, $\overline{E} = MD5(\overline{P_1} \| \overline{V}) \bmod p$ thì: $\overline{P_1} = P_1$ và $\overline{E} = E$.

Chứng minh:

Ta có:

$$\overline{V} = g^{-E} \times y^S \bmod p = g^{-E} \times g^{x.(x)^{-1}.(k+E)} \bmod p$$
$$= g^{-E+k+E} \bmod p = g^k \bmod p = V$$

Nên:

$$\overline{P_1} = C_0 \times (\overline{V})^{-x} \bmod p = P_1 \times (V)^x \times (V)^{-x} \bmod p = P_1$$

Và:

$$\overline{E} = MD5(\overline{P_1} \| I\overline{V}) \bmod p = MD5(P_1 \| IV) \bmod p = E$$

Đây là điều cần chứng minh.

2. Thuật toán MTA 16.5 – 02

a) Dữ liệu và khóa:

IV.BẢN RÕ CẦN MÃ HÓA P BAO GỒM N KHỐI DỮ LIỆU CÓ ĐỘ DÀI 128 BIT:

$P = \{P_1, P_2, ..., P_i, ..., P_n\}$, $i = \overline{1,n}$, $|P_i| = 128\ bit$

- Bản rõ được mã hóa P_m là bản rõ P được bổ sung khối P_0:

$$P_m = \{P_0, P\} = \{P_0, P_1, P_2, ..., P_i, ..., P_n\}$$, ở đây: $P_0 = MD5(P)$

- Khóa bí mật chia sẻ K_S bao gồm 2 thành phần:

$$K_S = (p, x)$$

Trong đó: *p* là 1 số nguyên tố lớn có $|p| = 128$ bit, *x* là một giá trị được chọn ngẫu nhiên trong khoảng (*1,p*) và thỏa mãn: $\gcd(x, p-1) = 1$.

a) Thuật toán mã hóa:

- Thủ tục sinh khóa K_{OT}:

```
for i =1 to n do
    begin
        K_i = MD5(P_{i-1} || P_0 || P_{i-1} || P_0)
    end
```

- Mã hóa khối đầu tiên của bản rõ P_m:

$$C_0 = (P_0)^x \bmod p$$

- Mã hóa các khối còn lại:

```
for i =1 to n do
    begin
        C_i = P_i ⊕ K_i
    end
```

- Bản mã nhận được:

$$C = \{C_0, C_1, C_2, ..., C_i, ..., C_n\},\ i = \overline{0,n},\ |C_i| = 128\ bit$$

<u>*Chú ý*</u>:

- Đối với các trường hợp mà: $MD5(P) > p$, dẫn đến: $P_0 > p$ và việc giải mã sẽ bị sai. Vì thế, ở thủ tục mã hóa khối đầu tiên của P_m cần tính: $C_0 = (P_0)^x \bmod p$ với: $P_0 = (p - MD5(P))_2$ thay vì: $P_0 = MD5(P)$, ở đây: $(p - MD5(P))_2$ là dạng mã bù 2 của $(p - MD5(P))$.

b) Thuật toán giải mã:

- Giải mã khối C_0 của bản mã nhận được:

$$\overline{P}_0 = (C_0)^{x^{-1}} \bmod p$$

- Thủ tục sinh khóa và giải mã các khối từ 1 đến n:

```
for i = 1 to n do
  begin
      K̄_i = MD5(P̄_{i-1} || P̄_0 || P̄_{i-1} || P̄_0)
      P̄_i = C_i ⊕ K̄_i
  end
```

- Bản rõ nhận được:

$$\overline{P} = \{\overline{P}_1, \overline{P}_2, ..., \overline{P}_i, ..., \overline{P}_n\},\ i = \overline{1,n}$$

- Thủ tục xác thực bản tin nhận được:

[1]. Tính: $H = MD5(\overline{P})$

[2]. Nếu: $H = \overline{P}_0$ thì: $\overline{P} = P$. Khi đó bản tin được xác thực về nguồn gốc và tính toàn vẹn.

<u>*Chú ý*</u>:

- Việc tính giá trị $x^{-1} \bmod(p-1)$ trong thủ tục giải mã khối C_0 có thể thực hiện 1 lần và lưu trữ như 1 thành phần của K_S: $K_S = \{p, x, y\}$, ở đây: $y = x^{-1} \bmod(p-1)$.
- Khi đó, giá trị $\overline{P}_0$ được tính theo: $\overline{P}_0 = (C_0)^y \bmod p$.

c) Tính đúng đắn của MTA 16.5 – 02

Điều cần chứng minh ở đây là: p số nguyên tố, $MD5: \{0,1\}^* \mapsto Z_q$ với: $|p| = |q| = 128\ bit$, $1 < x < p$, $y = x^{-1} \bmod(p-1)$, $P = \{P_1, P_2, ..., P_i, ..., P_n\}$, $P_m = \{P_0, P\} = \{P_0, P_1, P_2, ..., P_i, ..., P_n\}$ với: $|P_i| = 128\ bit$ và: $P_0 = MD5(P)$, $K_i = MD5(P_{i-1} \| P_0 \| P_{i-1} \| P_0)$ với: $i = \overline{1,n}$, $C_0 = (P_0)^x \bmod p$, $C_i = P_i \oplus K_i$ với: $i = \overline{1,n}$. Nếu: $\overline{P}_0 = (C_0)^y \bmod p$, $\overline{K}_i = MD5(\overline{P}_{i-1} \| \overline{P}_0 \| \overline{P}_{i-1} \| \overline{P}_0)$, $\overline{P}_i = C_i \oplus \overline{K}_i$ với: $i = \overline{1,n}$, $H = MD5(\overline{P})$ thì: $H = \overline{P}_0$ và $\overline{P} = P$ với: $\overline{P} = \{\overline{P}_1, \overline{P}_2, ..., \overline{P}_i, ..., \overline{P}_n\}$.

Chứng minh:

Thật vậy, ta có:

$$\overline{P}_0 = (C_0)^y \bmod p = \left((P_0)^x \bmod p\right)^{x^{-1}} \bmod p = (P_0)^{x.x^{-1}} \bmod p = P_0$$

Nên:

$$\overline{K}_1 = MD5(\overline{P}_0 \| \overline{P}_0 \| \overline{P}_0 \| \overline{P}_0) = MD5(P_0 \| P_0 \| P_0 \| P_0) = K_1$$

Do đó:

$$\overline{P}_1 = C_1 \oplus \overline{K}_1 = C_1 \oplus K_1 = P_1$$

Tiếp theo:

$$\overline{K}_2 = MD5(\overline{P}_1 \| \overline{P}_0 \| \overline{P}_1 \| \overline{P}_0) = MD5(P_1 \| P_0 \| P_1 \| P_0) = K_2$$

Và:

$$\overline{P}_2 = C_2 \oplus \overline{K}_2 = C_2 \oplus K_2 = P_2$$

Tương tự:

$$\overline{K}_3 = K_3, ...,\ \overline{K}_n = K_n$$

Và:

$$\overline{P}_3 = P_3, ...,\ \overline{P}_n = P_n$$

Suy ra:

$$\overline{P} = P$$

Và:

$$H = MD5(\overline{P}) = MD5(P) = P_0 = \overline{P}_0$$

Đây là điều cần chứng minh.

1. Một số đánh giá về độ an toàn và hiệu quả thực hiện của các thuật toán mới đề xuất

a) Mức độ an toàn và hiệu quả thực hiện của MTA 16.5 – 01

Mức độ an toàn: Việc sử dụng 2 khóa phân biệt để mã hóa/giải mã bản tin, trong đó khóa K_{OT} được sử dụng tương tự như hệ mã OTP cho phép loại trừ hầu hết các dạng tấn công đã được biết đến trong thực tế: thám mã vi sai, thám mã tuyến tính, tấn công bản mã có lựa chọn, tấn công bản rõ đã biết,… Các phương pháp tấn công này hoàn toàn không có tác dụng với thuật toán mới đề xuất do K_{OT} chỉ sử dụng 1 lần cùng với bản tin được mã hóa, hơn nữa với kích thước 128 bit thì phương pháp vét cạn là không khả thi để tấn công các khóa con K_i. Mặt khác, khóa bí mật chia sẻ K_S trong thuật toán này chỉ sử dụng để mã hóa và giải mã cho khối dữ liệu đầu tiên của bản tin và các thuật toán mã hóa/giải mã ở đây được thực hiện theo phương pháp của các hệ mã lũy thừa (RSA, ElGamal,...) nên khóa bí mật chia sẻ có thể sử dụng nhiều lần hoàn toàn như các hệ mã khối thông thường khác: DES, AES,… Ngoài ra, thuật toán mã hóa/giải mã khối đầu tiên của bản tin với khóa K_S còn có tác dụng cho phép xác thực nguồn gốc của bản tin nhận được.

Hiệu quả thực hiện: Ngoại trừ khối đầu tiên được mã hóa và giải mã theo phương pháp của các hệ mã lũy thừa như: RSA, ElGamal,... cho hiệu quả thực hiện không cao, các khối còn lại của bản tin được mã hóa/giải mã hoàn toàn theo nguyên tắc của hệ mã OTP. Vì vậy, về căn bản hiệu quả thực hiện của thuật toán mới đề xuất là tương đương với hệ mã OTP.

b) Mức độ an toàn và hiệu quả thực hiện của MTA 16.5 – 02

Mức độ an toàn và hiệu quả thực hiện của MTA 16.5 – 02 về cơ bản có thể đánh giá tương tự thuật toán MTA 16.5 – 01, ngoại trừ 2 điểm khác biệt chủ yếu:

- Có khả năng xác thực nguồn gốc và tính toàn vẹn của bản tin nhận được. Vì thế ngoài khả năng chống được các dạng tấn công đối với các hệ mã khối thông thường khác, thuật toán còn có thể chống lại một số dạng tấn công giả mạo trong thực tế.

- Chỉ thực hiện với các loại bản tin có kích thước xác định, nói cách khác thuật toán này không làm việc được với các dòng dữ liệu mà kích thước chưa được xác định tại thời điểm tiến hành mã hóa như MTA 16.5 – 01.

V. KẾT LUẬN

Bài báo đề xuất giải pháp xây dựng một hệ mã khóa đối xứng hiệu năng cao từ việc phát triển hệ mã sử dụng khóa 1 lần OTP kết hợp với các hệ mã lũy thừa khác nhằm đáp ứng các yêu cầu về độ an toàn và hiệu quả thực hiện. Với giải pháp thiết kế khóa mật từ 2 phân khóa tách biệt, các thuật toán được xây dựng theo giải pháp được đề xuất ở đây có khả năng loại trừ hầu hết các dạng tấn công đối với các hệ mã khóa đối xứng, đây là một ưu điểm rất quan trọng được kế thừa từ hệ mã OTP. Ngoài ra, do có cơ chế xác thực nguồn gốc và tính toàn vẹn của bản tin được mã hóa, các thuật toán này còn có khả năng chống các dạng tấn công giả mạo đã biết trên thực tế. Những ưu điểm khác của các thuật toán này là có tốc độ và hiệu quả thực hiện có thể so sánh với hệ mã OTP, song khóa mật chia sẻ có thể dụng nhiều lần như các hệ mã khóa đối xứng khác. Đây là những đặc tính rất quan trọng để ứng dụng các thuật toán mới trong việc thiết kế - chế tạo các thiết bị bảo mật thông tin trong thực tế.

TÀI LIỆU THAM KHẢO

[1] SharadPatil , Ajay Kumar, "*Effective Secure Encryption Scheme(One Time Pad) using Complement Approach",* International Journal of Computer Science & Communication, Vol.1,No.1,January-June 2010,pp.229-233.

[2] Raman Kumar, Roma Jindal, Abhinav Gupta, SagarBhalla, HarshitArora, "*A Secure Authentication System-Using Enhanced One Time Pad Technique*", IJCSNS International Journal of Computer Science and Network Security, VOL.11 No.2, February 2011.

[3] SharadPatil, ManojDevare, Ajay Kumar, "*Modified One Time Pad Data Security Scheme: Random Key Generation Approach",* International Journal of Computer Science and Security (IJCSS) ,Volume (3): Issue(2).

[4] N.J.Croft and M.S.Olivier, "*Using an approximated One-Time Pad to Secure ShortMessaging service (SMS)",* SATNAC 2005 Proceedings.

[5] Jeff Connelly, "*A Practical Implementation of a One-time Pad Cryptosystem",* CPE 456, June 11, 2008.

[6] R. L. Rivest, A. Shamir, and L. M. Adleman, "*A Method for Obtaining Digital Signatures and Public Key Cryptosystems*", Communications of the ACM, Vol. 21, No. 2, 1978, pp. 120-126.

[7] T. ElGamal, "*A public key cryptosystem and a signature scheme based on discrete logarithms*", IEEE Transactions on Information Theory. 1985, Vol. IT-31, No. 4. pp.469–472.

[8] Mark Stamp, Richard M. Low, "*Applicd cryptanalysis: Breaking Ciphers in the Real World*", John Wiley & Sons, Inc., ISBN 978-0-470-1.

[9] Shannon C.E., "*Communication Theory of Secrecy Systems*", Bell System Technical Journal, Vol.28-4, pp 656-715, 1949.

[10] Menezes A., Van Oorschot P. and Vanstone S., "*Handbook of Applied Cryptography*", Boca Raton, Florida: CRC Press. 1996.

A SOLUTION FOR DEVELOPING SYMMETRIC - KEY CRYPTOGRAPHIC ALGORITHMS BASED ON THE OTP AND EXPONENTIAL CIPHERS

Luu Hong Dung, Nguyen Vinh Thai, Tong Minh Duc, Bui The Truyen

ABSTRACT*— This paper proposes a solution for developing Symmetric-key cryptographic algorithms based on the OTP cipher combined with the exponential ciphers. Advantages of the new algorithm have high safety and efficient implementation as OTP cipher, but the use of secret keys are exactly the same as DES/AES algorithms.*

Keywords *— Symmetric-Key Cryptography, Symmetric-Key Cryptographic Algorithm, One - Time Pad Algorithm, OTP Cipher.*