PHÁT TRIỂN THUẬT TOÁN CỦA POLLARD TÍNH CẤP CỦA PHẦN TỬ TRONG $\mathbb{Z}_n^*$

Lê Văn Tuấn[1*], Lều Đức Tân[2]

Tóm tắt: *Bài viết này giới thiệu thuật toán của Pollard để giải bài toán phân tích số và bài toán logarit rời rạc. Dựa trên thuật toán của Pollard, chúng tôi xây dựng thuật toán để tính cấp của một phần tử trong $\mathbb{Z}_n^*$. Kết quả nghiên cứu được ứng dụng vào phân tích độ an toàn của các lược đồ chữ ký số có độ an toàn dựa trên tính khó giải của bài toán logarit rời rạc trong vành Z_n.*

Từ khóa: Bài toán phân tích số; Bài toán logarit rời rạc; Cấp của phần tử.

1. MỞ ĐẦU

Hiện nay, hệ mật khoá công khai cùng các biến thể của nó, có độ an toàn dựa trên tính khó giải của bài toán logarit rời rạc trên vành Z_n, đang được các nhà khoa học trong nước và trên thế giới nghiên cứu và phát triển [1] [2] [4]. Đồng thời, việc nghiên cứu và phát triển các thuật toán giải bài toán logarit rời rạc trên vành Z_n cũng được quan tâm bởi các nhà khoa học. Bởi vì những thuật toán giải bài toán logarit rời rạc trên vành Z_n là cơ sở để xây dựng tham số an toàn cho hệ mật cùng các biến thể có độ an toàn dựa trên tính khó giải của bài toán này. Việc nghiên cứu ra các thuật toán để giải bài toán logarit rời rạc trên trường GF(p) đã được nhiều nhà khoa học nghiên cứu, phát triển và đang được ứng dụng trên thực tế, tiêu biểu là tác giả Pollard với thuật toán mang tên Rho Pollard [6]. Tuy nhiên, cho đến nay vẫn chưa có thuật toán nào trên thế giới và trong nước được công bố để giải bài toán logarit rời rạc trên vành Z_n. Một điều hiển nhiên rằng, nếu tính được cấp của các phần tử trên $\mathbb{Z}_n^*$ thì hầu hết các thuật toán giải bài toán logarit rời rạc trên trường GF(p) có thể áp dụng để giải bài toán logarit rời rạc trên vành Z_n. Vậy, để giải bài toán logarit rời rạc trên vành Z_n, thì việc tính cấp của của phần tử thuộc nhóm $\mathbb{Z}_n^*$ có vai trò quyết định.

Trong lý thuyết số thì hai bài toán "phân tích hợp số n ra thừa số nguyên tố" và "xác định cấp của phần tử trong $\mathbb{Z}_n^*$" đã được chứng minh là có độ phức tạp tính toán tương đương. Nếu như bài toán phân tích số có được một số thuật toán tìm các ước nguyên tố p với độ phức tạp tính toán $O(\sqrt{p})$ chẳng hạn như thuật toán Rho-Pollard [5] thì với bài toán còn lại chỉ với thuật toán duyệt dần (vét cạn) có thể tìm được cấp của các phần tử có cấp m trong thời gian O(m). Trong bài viết này, chúng tôi phỏng theo phương pháp Rho-Pollard tính logarit rời rạc trên trường GF(p) [3] [5] để xây dựng thuật toán tính cấp m của phần tử trong $\mathbb{Z}_n^*$ với độ phức tạp tính toán $O(\sqrt{m})$ và độ phức tạp không gian là $O(\sqrt{m})$ (bit).

2. CƠ SỞ TOÁN HỌC

2.1. Kết quả về xác định cấp của phần tử

Định nghĩa 2.1. Cho G là một nhóm nhân với phần tử đơn vị là 1 và $g \in G$. Nếu tồn tại số tự nhiên d sao cho:

$$g^d = 1 \tag{1}$$

thì g được gọi là "có cấp hữu hạn" và giá trị d nhỏ nhất thỏa mãn đẳng thức (1) được gọi là "cấp của g trong G" và ký hiệu là ord(g) mod G hay $\mathrm{ord}_G(g)$.■

Lưu ý. Nếu G là nhóm hữu hạn thì mọi phần tử trong G đều có cấp hữu hạn và $\mathrm{ord}_G(g)$ luôn là ước của #G, ngoài ra, nếu số tự nhiên d thỏa mãn (1) thì $\mathrm{ord}_G(g)$ cũng là ước của d.

Thuật toán 2.1: Thuật toán tính cấp của phần tử.

Input: Cho G là một nhóm nhân và g ∈ G. Nếu $g^M = 1$ và M = $\prod_{i=1}^{k} r_i^{a_i}$ với r_i là các số nguyên tố khác nhau.

Output: d = $ord_G(g)$.

```
1. d ← M;
2. for i from 1 to k do
        begin
        j =1;
        ok=1;
        while ((j ≤ a_i) and ok) do
        begin
        n = d/r_i;
        ok = (g^n == 1);
        if (ok)
        {d = n;
        j←j+1;
        }
        end while;
        end for;
3 return d.■
```

Định lý 2.1. Thuật toán 2.1 là đúng đắn.■

Từ định lý trên, ta dễ dàng thu được hệ quả sau.

Hệ quả 2.2. Cho G là một nhóm nhân và g ∈ G. Nếu $g^M = 1$ và M = $R\prod_{i=1}^{k} r_i^{a_i}$ với r_i là các số nguyên tố khác nhau và R|$ord_G(g)$. Khi đó, áp dụng thuật toán 2.1 với các ước nguyên tố r_i trong phần phân tích được của M ta cũng tìm được $ord_G(g)$.■

2.2. Thuật toán phân tích số và thuật toán tính logarit rời rạc của Pollard

Thuật toán 2.2: Thuật toán phân tích số [5].

Trong thuật toán này, Pollard đã sử dụng hàm giả ngẫu nhiên F(x) = ($x^2 + a$) mod n. Thuật toán phân tích số trình bày như sau:

Input: Hợp số n.

Output: p là ước không tầm thường của n.

```
1. [Choose seeds]
        Choose random a∈ [1,n-1];
        Choose random s ∈ [1,n-1];
        U = V = s;
2. [Factor search]
        U = F(U);
        V = F(V);
        V = F(V);
        p = gcd(U – V,n);
        if (p == 1) goto 2 [Factor search];
3. [Bad seed]
       if (p == n) goto 1 [Choose seeds];// U và V gặp nhau
4. [Success]
return p;■
```

Định lý 2.3. Thuật toán 2.2 tìm được thừa số nguyên tố p của n trong O($\sqrt{p}$) bước.■

2.3. Một số kết quả bổ trợ

Để phục vụ cho thuật toán được trình bày trong mục 3, chúng tôi đưa ra một kết quả sau:

Định lý 2.4. Cho s số tự nhiên ngẫu nhiên t bít $M_1, M_2, \dots, M_s$ ($M_i < 2^t$, i = 1, 2, ..., s), ký hiệu d = gcd($M_1, M_2, \dots, M_s$). Khi này, ta có xác suất để trong d có ước nguyên tố r lớn hơn k bít ($r \geq 2^k$), ký hiệu là ρ, được đánh giá theo bất đẳng thức sau

$$\rho \leq \left(1 + \frac{1}{2^{(s-1)k}}\right)^u - 1 \qquad (2)$$

với $u = \left\lfloor \frac{t}{k} \right\rfloor$.

Chứng minh:

Nhớ lại rằng với r là một số nguyên tố bất kỳ thì xác suất để r là ước của một số tự nhiên N đúng bằng $\frac{1}{r}$. Như vậy, với mỗi ước nguyên tố r của M_1 cũng là ước của $s-1$ các $M_2, \dots, M_r$ (do đó, cũng là ước của gcd($M_1, M_2, \dots, M_s$)) xảy ra với xác suất là $\frac{1}{r^{(s-1)}}$.

Từ M_1 là số t bít nên nó chỉ chứa cùng lắm là $u = \left\lfloor \frac{t}{k} \right\rfloor$ thừa số nguyên tố trên k bít, vậy, ta có ước lượng sau:

$$\rho = C_u^1 \frac{1}{r^{(s-1)}} + C_u^2 \frac{1}{r^{2(s-1)}} + \dots + C_u^u \frac{1}{r^{u(s-1)}}$$

Do $r \geq 2^k$, ta có ước lượng sau:

$$\rho \leq C_u^1 \frac{1}{2^{(s-1)k}} + C_u^2 \frac{1}{2^{2(s-1)k}} + \dots + C_u^u \frac{1}{2^{u(s-1)k}}$$
$$= \left(1 + \frac{1}{2^{(s-1)k}}\right)^u - 1.$$

Định lý đã được chứng minh.■

3. THUẬT TOÁN TÍNH CẤP CỦA PHẦN TỬ TRONG $\mathbb{Z}_n^*$ THEO PHƯƠNG PHÁP RHO-POLLARD

3.1. Thuật toán

Lựa chọn hàm F: $\mathbb{Z}_n^* \times \mathbb{Z}_n^* \to \mathbb{Z}_n^*$

Bản chất của phương pháp Rho-Pollard là chọn trước một hàm giả ngẫu nhiên theo biến thứ 2 F: $\mathbb{Z}_n^* \times \mathbb{Z}_n^* \to \mathbb{Z}_n^*$ trên cơ sở một phân hoạch tập $\mathbb{Z}_n^*$ thành k tập con có lực lượng đồng đều. Để thêm phong phú trong thuật toán được trình bày dưới đây, chúng tôi chọn phân hoạch theo các số đồng dư theo mô đun k=3, nói một cách khác hai hàm F(x,y) và E(a,b,y) được xác định theo công thức sau:

$$F(x,y) = \begin{cases} (x.y) \bmod n & \text{if} \quad y \bmod 3 = 1 \\ y^2 \bmod n & \text{if} \quad y \bmod 3 = 0 \\ (g.y) \bmod n & \text{if} \quad y \bmod 3 = 2 \end{cases} \qquad (3)$$

Và do cấp của g chưa xác định nên ta không thể lập hàm tính số mũ tương ứng giống như công thức (3) mà hàm E: $\mathbb{N}\times\mathbb{N}\times\mathbb{Z}_n \to \mathbb{N}\times\mathbb{N}$ phải được xác định như sau:

$$E(a,b,y) = \begin{cases} (a+1, b) & \text{if} \quad y \bmod 3 = 1 \\ (2a, 2b) & \text{if} \quad y \bmod 3 = 0 \\ (a, b+1) & \text{if} \quad y \bmod 3 = 2 \end{cases} \qquad (4)$$

Thuật toán 3.1

Input: n, g ∈ $\mathbb{Z}_n^*$.

Output: d = ord(g) (mod n).

1. d = 0;
2. Chọn mầm s//[Choose seeds]
   Chọn ngẫu nhiên s, s ∈ [1,n-1];
   x = g^s mod n; //
   y = z = 1;
   a = b = u = v = 0;
3. Tìm mối liên hệ giữa y và z.//[Colection search]
   (a,b) = E(a,b,y); y = F(x,y);
   (u,v) = E(u,v,z); z = F(x,z);
   (u,v) = E(u,v,z); z = F(x,z);
   if (y ≠ z) goto 3//[Colection search];
4. M = $v - b + s(u - a)$;
5. [Bad seed]
   if (M == 0) goto 2//[Choose seeds];
6. d = gcd(M,d);
7. Phân tích d theo mẫu:
   d= R. $\prod_{i-1}^{k} r_i^{a_i}$ với gcd(R, $\prod_{i-1}^{k} r_i^{a_i}$) = 1 //theo thuật toán 2.2; //
8. Kiểm tra kết quả phân tích//[Bad factoring]
   Nếu R là hợp số thì goto 2// [Choose seeds];
9. Tính toán bậc của g là d, d = ord(g) (mod n) bởi thuật toán 2.1//
10 return d.■

3.2. Chứng minh tính đúng đắn của thuật toán 3.1

Bổ đề 3.1 Với mọi giá trị y và z tính được tại bước 3 đều có dạng:

$$y = g^{s.a+b} \bmod n \text{ và } z = g^{s.u+v} \bmod n. \quad (5)$$

Chứng minh:

Do việc tính y và (a,b) giống hệt như việc tính z và (u,v) nên ở đây ta chỉ cần chứng minh đẳng thức y = $g^{s.a+b}$ mod n là đủ. Quả vậy, từ bước 2 ta có x = g^s mod p, y= 1, a= b= 0. Nên theo hai công thức (3) và (4) các giá trị y và (a,b) tính được tại bước lặp đầu tiên của bước 3 sẽ là:

y= (x.y) mod n= x= g^s mod n, a= a+1= 1 và b= b= 0. Suy ra y = $g^{s.a+b}$ mod n hay (3-3) đã đúng tại bước lặp đầu tiên của bước 3.

Giả thiết quy nạp là (3-3) đã đúng đến bước lặp thứ i của bước 3. Ký hiệu các giá trị y, (a,b) tính được tại vòng thứ i+1 là y*, (a*,b*). Lại theo hai công thức (3) và (4):

Trường hợp y mod 3 = 0, kết quả tính y* như sau:

y* = (x.y) mod n = $(g^s.y)$ mod n, a* = a+1 và b* = b. Theo đẳng thức (5) ta có

$(g^s.y) = (g^s.g^{s.a+b}) = g^{s.(a+1)+b} = g^{s.a^*+b^*}$ (mod n)

Trường hợp y mod 3 = 1, kết quả tính y* như sau:

y* = (y^2) mod n, a* = 2a và b* = 2b. Theo đẳng thức (5) ta có

$y^2 = g^{2(s.a+b)} = g^{s.(2a)+2b} = g^{s.a^*+b^*}$ (mod n)

Trường hợp y mod 3 = 2, kết quả tính y* như sau:

y* = (g.y) mod n , a* = a và b* = b+1. Theo đẳng thức (5) ta có

$(g.y) = (g.g^{s.a+b}) = g^{s.a+(b+1)} = g^{s.a^*+b^*}$ (mod n)

Như vậy (5) đã đúng tại vòng thứ i+1 nên theo nguyên lý quy nạp ta có (3-3) đúng tại mọi bước lặp của bước 3 và bổ đề đã được chứng minh.■

Định lý 3.2. Thuật toán 3.1 là đúng đắn.

Chứng minh

Trước hết, ta chứng tỏ rằng các giá trị M tìm được tại bước 4 là bội của ord(g).

Quả vậy, do bước 4 được thực hiện chỉ khi y= z nên theo bổ đề 3.1 ta có đồng dư thức sau:

$g^{s.a+b} = g^{s.u+v}$ (mod n) hay $g^{s.(u-a)+(v-b)} = g^{M} = 1$ (mod n). Theo tính chất của cấp của g là số nguyên dương nhỏ nhất thỏa $g^{ord(g)}=1$ mod n, từ đẳng thức $g^{M} = 1$ (mod n) dẫn đến M là bội của ord(g) ■.

Từ điều chứng minh được trên, ta có, d xác định tại bước 6 cũng là bội của ord(g) nên theo định lý 2.1 ta có giá trị d tìm được tại bước 9 của thuật toán (và cũng là đầu ra của thuật toán 3.1) đúng bằng ord(g) vậy định lý đã được chứng minh.■

3.3. Độ phức tạp thời gian và không gian của thuật toán 3.1

Bổ đề 3.3. Nếu hàm E cho theo công thức (4) là giả ngẫu nhiên theo (a,b,y) thì bước 3 của thuật toán 3.1 thực hiện trong khoảng $O(\sqrt{ord(g)})$ bước.

Chứng minh:

Trong mỗi bước trong vòng 3 của thuật toán kiểm tra điều kiện y = z, điều kiện trên theo bổ đề 3.1 là tương đương với điều kiện:

$$s.a+b = s.u+v \pmod{ord(g)}. \quad (6)$$

Theo giả thiết của bổ đề, hai vế của (6) được coi là các số ngẫu nhiên nên theo định lý ngày sinh thì trong khoảng $\sqrt{ord(g)}$ cặp (s.a+b, s.u+v) sẽ tồn tại ít nhất một cặp thỏa mãn (6) với xác suất $1-\frac{1}{e}$ (e là số tự nhiên ≈ 2...). Điều trên có nghĩa trong khoảng $O(\sqrt{ord(g)})$ bước ta sẽ tìm được 1 cặp thỏa mãn (6) và điều này đã chứng minh xong bổ đề.■

Hệ quả 3.4. Kích thước của các biến a, b, u, v và do đó, các biến d, M trong thuật toán 3.1 cho theo công thức:

$$O(\sqrt{ord(g)}) \text{ (bits)} \quad (7)$$

Quả vậy, theo công thức (4) thì sau mỗi vòng trong bước 3 các giá trị a, b được tăng lên tối đa là 1 bít còn hai gia trị u, v được tăng lên tối đa 2 bít (ứng với y và z đều chia hết cho 3)). Số vòng của bước này, theo bổ đề 3.3 là $O(\sqrt{ord(g)})$ với a, b, u, v xuất phát đều bằng 0 cho nên kích thước của 4 biến nêu trên được cho bởi (7) và hệ quả đã được chứng minh.■

Bổ đề 3.5. Sau s giá trị M tìm được thì giá trị d tính trong bước 6 sẽ qua được bước 8 trong thuật toán 3.1 với xác suất bằng $1-\frac{1}{k.2^{(s-1.5)k}}$. (8)

Chứng minh:

Trong giải tích cổ điển ta có $\lim_{x\to 0}\frac{(1+x)^u-1}{x} = u$, điều này có nghĩa khi x đủ nhỏ thì $(1+x)^u - 1 \approx u.x$.

Theo hệ quả 3.4, giá trị M là số cỡ $O(\sqrt{ord(g)})$ bít nên trong phạm vi ord(g) đủ nhỏ để thuật toán 3.1 có thể tìm được các giá trị M, chẳng hạn, cỡ k bít thì cỡ của biến M sẽ là cỡ $O(2^{k/2})$ bits. Lại từ định lý 2.3 cho thấy với khả năng độ phức tạp tính toán là $O(2^{k/2})$ thì có thể tìm được các ước không tầm thường cỡ $O(2^{k})$ (tức là các ước cỡ k bít), vì vậy, có thể coi giá trị t = $2^{k/2}$ trong định lý 2.4 và theo định lý này ta có xác suất để giá trị d tìm được trong bước 6 của thuật toán 3.1 sau s lần tính M có chứa ước nguyên tố lớn hơn 2^k

(ngưỡng phân tích được của thuật toán 2.2) nhưng không là ước của org(g), cũng chính là xác suất d có thể không qua được bước 8 là:

$$\rho = \left(1 + \frac{1}{2^{(s-1)k}}\right)^u - 1 \approx \frac{1}{2^{(s-1)k}}\left\lfloor\frac{2^{k/2}}{k}\right\rfloor \le \frac{1}{k.2^{(s-1.s)k}}. \quad (9)$$

Đến đây bổ đề đã được chứng minh.■

Với nhận định 2.4 về khả năng thực hiện được thuật toán 2.2 (giá trị k lớn hơn 40) nên với s = 2 thì biểu thức (8) sẽ cho giá trị lớn hơn 0.999999, với s= 3 sẽ là lớn hơn 0.999999999999999999 ... nên ta có thể kết luận như sau:

Định lý 3.6. Thuật toán 3.1 có độ phức tạp thời gian là $\sqrt{\mathrm{ord}(g)}$ và độ phức tạp không gian là $\sqrt{\mathrm{ord}(g)}$ bít để tìm được ord(g).■

4. VÍ DỤ

Ứng dụng thuật toán 3.1 để tính bậc của g với g = 2 trên vành Z_n, với n = 4292870399 = 65519 * 65521. Tại bước 2 của thuật toán, ta chọn mầm s = 2. Sau 12058 vòng lặp bước 3, các giá trị y và z được tìm thấy như sau:

y = 2739898200
z = 2739898200

Với s = 2, tại bước 4 của thuật toán đã tìm ra M với len(M)=502*16=8032 bit ứng với số 2414 chữ số thập phân:

2440362158605110709450446867850191936970442659756942891320869465033050332835011756603533747150762031373495855735839791869089565772846233197223578906307367047288054855828091183078204451904328068270100859524152419185291622375253488633739449718796941773218217316285182111649431255613765850431931178141679201880339451926985597371971869304564180385101066181647982979769199693035572500706487026812730485897223870139982817655675887365354998015350160329498507894533643241603375521954494661079280922232601422106120044401500995820834732524437189139267953372862897155200082408112885789761383595314143318122662397080896223398021567631763246654426348382351402497209682863511075446626941558366047241194265960898746919615321501517924423005553445998972364023822171309339078929171581689909308710782725463578684165376349427918128814301606550654805288415836107760272756874108309061484890714075024550162109497080801110335497774475681863823483556580616288140356905765290023246160989531195986411850987278795752119960745468166848171019590731823224993513078858327581435279423955975632939598368017606982123226557138481927589283973133498830837178264226741509156297980910124063092548086903678689588967342690814311662916398142773956816147424966666994201841866637623153576913869880592809077167835322878750664729806285759973488780566853467944009319895883261016019511499302034671971405775066451264482712221117975582082644240196392263956189408521418455979635246149918255648721778321811457606218861275215709502579429984342698309713502305782314257225517761385766744300219520397784527792338411840341494326998947485625528403096184130103703114159262090357829112828230002878976542458713246841967432348787706206676069547364380640907648221413469529693349226142161822944307633977588403741993998867078167669673669724849576089433577547617209172628083149245948693169579929992130464106479552537318133479418647982246448915566863571129561446578166739844915501586002334115567084724946827629680935908840961253442772181059541967432689320115864102936105205499371837125777140543498884689899348539228939838999597680600343301171648544320356912405940869510687103773874378448215758315551361982598857798317229442434171073059716328809499782562762040054817531565673746793012124863236434813546409930698589204059627100515149160577711335798773336894294421625847586601669152900

670270927414107491836475655606940256381491866296956152166755301220452320125685194052334261714812928 0.

Bước 5 của thuật toán 3.1 tìm ra M≠0 nên thuật toán chuyển sang bước 6. Vì d được khởi tạo giá trị 0 nên kết quả bước 6 tìm ra ngay giá trị d= gcd(M,d) chính là M. Bước 7, thuật toán phân tích số d về dạng chuẩn như sau: d = R. $\prod_{i=1}^{k} r_i^{a_i}$, với gcd(R, $\prod_{i=1}^{k} r_i^{a_i}$) = 1. Kết quả phân tích là:

d=2^3086*3^2*5*7*13*17*41*47*293*1627*4009109770683278830350296430707012825368753073686519561729330751274879323401255449199583735087499291906205014049457178251201276552215162950892103229357395733958886410178822302327209501042649263170057777372585192140447798061135678800498090280781673920508366883370457917154406261567532143132762173990890413067599956328435031083638321060745874332694069974328519563150951983424245829028188561063106144456186526992242855530370378178723726280692353402693593109518696007248799459159898982465870364318128032334721622685139606661011590839956056391397064507643563658198032387301615300830998242856434265087265866826611076841525117816865830793443897583627949873295505449621662909366943334375900384084003729344337817495405501800740592422875703026476204218214667222796272654542114703156751810903211854060850342754663342981470874011268751776680562554023134257749840894997891316253036829607241856784124489105615421141901705693447166321065072841007635451093124696202511568200807663932273935821083851611950711746359637321341228392456121591891739858223279631128224971209348047765988035449060530494086878338557218896967442734026420518712857693967141423480177679128507099429723075221746522885241419381292519694696396199914474494339916521196974839381899689352990668047332379395266394261311860358511877872983295162909881015166971506707094354813436830818677981734789371254887866724753218250606749568271725406023610036582323344309356322082624288087319236115801324472872926410996189857741982967460 59.

Đồng thời tại bước 7 kiểm tra điều kiện gcd(R, $\prod_{i=1}^{k} r_i^{a_i}$) = 1. Tại bước 8 của thuật toán 3.1, ứng với s = 2, thực hiện kiểm tra giá trị R là hợp số hay số nguyên tố. Với giá trị R dưới đây:

4009109770683278830350296430707012825368753073686519561729330751274879323401255449199583735087499291906205014049457178251201276552215162950892103229357395733958886410178822302327209501042649263170057777372585192140447798061135678800498090280781673920508366883370457917154406261567532143132762173990890413067599956328435031083638321060745874332694069974328519563150951983424245829028188561063106144456186526992242855530370378178723726280692353402693593109518696007248799459159898982465870364318128032334721622685139606661011590839956056391397064507643563658198032387301615300830998242856434250872658668266110768415251178168658307934438975836279498732955054496216629093669433437590038408400372934433781749540550180074059242287570302647620421821466722279627265454211470315675181090321185406085034275466334298147087401126875177668056255402313425774984089499789131625303682960724185678412448910561542114190170569344716632106507284100763545109312469620251156820080766393227393582108385161195071174635963732134122839245612159189173985822327963112822497120934804776598803544906053049408687833855721889696744273402642051871285769396714142348017767912850709942972307522174652288524141938129251969469639619991447449433991652119697483938189968935299066804733237939526639426131186035851187787298329516290988101516697150670709435481343683081867798173478937125488786672475321825060674956827172540602361003658232334430935632208262428808731923611580132447287292641099618985774198296746059.

Đồng thời tại bước 7 kiểm tra điều kiện gcd(R, $\prod_{i=1}^{k} r_i^{a_i}$) = 1. Tại bước 8 của thuật toán 3.1, ứng với s = 2, thực hiện kiểm tra giá trị R là hợp số hay số nguyên tố. Với giá trị R dưới đây:

4009109770683278830350296430707012825368753073686519561729330751274879323401255449199583735087499291906205014049457178251201276552215162950892103229357395733958886410178822302327209501042649263170057777372585192140447798061135678800498090280781673920508366883370457917154406261567532143132762173990890413067599956328435031083638321060745874332694069974328519563150951983424245829028188561063106144456186526992242855530370378178723726280692353402693593109518696007248799459159898982465870364318128032334721622685139606661011590839956056391397064507643563658198032387301615300830998242856434250872658668266110768415251178168658307934438975836279498732955054496216629093669433437590038408400372934433781749540550180074059242287570302647620421821466722279627265454211470315675181090321185406085034275466334298147087401126875177668056255402313425774984089499789131625303682960724185678412448910561542114190170569344716632106507284100763545109312469620251156820080766393227393582108385161195071174635963732134122839245612159189173985822327963112822497120934804776598803544906053049408687833855721889696744273402642051871285769396714142348017767912850709942972307522174652288524141938129251969469639619991447449433991652119697483938189968935299066804733237939526639426131186035851187787298329516290988101516697150670709435481343683081867798173478937125488786672475321825060674956827172540602361003658232334430935632208262428808731923611580132447287292641099618985774198296746059,

Kết quả kiểm tra cho thấy giá trị R là hợp số nên việc chọn giá trị mầm s phải được thực hiện lại. Trong lần lặp tiếp theo của thuật toán 3.1, giá trị mầm s được chọn là 3. Sau 16654 vòng lặp trong bước 3 của thuật toán, đã tìm ra giá trị y và z như sau:

y = 4218529016

z = 4218529016

Tại bước 4 của thuật toán ứng với s=3, đã tìm ra giá trị M (len(M)=11600 bit), giá trị đó là:

M=116758132158988918812362428121635337510782643229146480272228568937807149717323168728647379420068680783263155002276506333834559626979597446368803361936290170504932285914321200154804184326265761671355280027423904292581797985718676501588109036873932272235673742484874380006251574618202312711093867872572759507617429924808401008729373430447777358250194545512722732477700194950684061617558733275093973917102738195885068737000831904043107639351414484230961895010996135445448156632551914149453340518140112634633198496439938474614564847187030014768675557364315512363581773475633813048210828356854176924642257366054180230153017211424575893388133666945851259496413393553894644437994096708817162863201500496561605554952616840628549413565359344558041401842670926959746946372045825855922988702854709992528707647376514137685547672075864522005855497973903327018160895589323871113656196641696548939429327614923650257193912873719411130813229715864185293386965637451589341227236914790330704352693273697583126858648762905516893051043973901351633570160552496619194142259611407874938684386967453480293128245206936608179517626496481598892427893469113438422088295186975676744078177317370162895279282909337433055286040216085938943092239778546086350870523015151880406880059316750001391104131890784001073080025638201329660376687491122944347623989972790846741625180267261480869341052167224495234348227012014180153678019709306967222777172662235258942491250055828936787545126908657869578661070817716564307434766012810480780558504131604589254432153218725115619190516696484625743719805426366925401268409992262014968005541197837658811225339274668805831055695770405747985103698384394521285028813938575228158292770081445304438439016527233441616523115950027312389148996023058953274608077403818298761315905423954457995719918944414565224629734823111298574308524772592254280828582130305049088182725367247570453342412223736977686437814845905877459786097776852170326425263137035180292771964115766033978597877581062254911342518212450525792806300710824022417982926034643944956990333185260225906684545723069226231101751262537326147600668221821679579613240778116923110417640199848296091103946059711825066245752438556379529353441996360482810535857245819954433874597336212663691910420719418504201377236264168083697686838943005559363316327810220671230311328347695751547813301837584153921832882777813854491701557864135919428472274278889874231599261769180195987524626199344008856324415385232422058357274613423086621157369669076581463054429930769111444765959077186896538249107340327122496791931773311875445415767717727413845096967234079779110758994938470652919022063243016702616185239872659666916919127560230086452739130654749176652620811854686100911434555334568154883335832394644359567894815676928438425878986612667179008256721903247901642778057119981582223526529821121310380374797411578523092000184677142919066676145427650806295801653744444516968630853694947066367707974988302608204054874138576276259500547367530978584478769931850377039587322937303969270117405662108080712105717198954998609428720021263119006237520742048892068410727257754247165063163442803612319704788876203003368999759080371934068392014717582779608032009717670025577636893868601703732351738661171131824937107453900990847026665529767839586455950158835793764631131744925448836470843276214709875567238972163755207510586557901233836779892043021029146411740448467544236893006096118824904501835885848290367583851270697117013634484536579359400921028409759667695686999919820 80.

Khi đó, kết quả tính d tại bước 6, d= gcd(M,d) như sau:

326111932265583202387923998594874386451204319942075093757961899537925218567043245084868368309912547331124732869062628637293352347270746488649114474826732039913200276956473180242714067035557336816246031700399110187690721406807880737693476419600884019097235861814452948010938204318896408917330293309448737678552402038252397785725709384641989194959539711566618988189054965540736645649370210105097400120284150733158940103263150482213778156660424713685859087858914491843097476421027344040688744208550230969880273015293674831376903064072817671789348069894005 08

374088988894357334657249811798474250360835270970188385844175605796529297476795488994139742617970697451492214200577890108406901107589120

Bước 7, kết quả phân tích d về dạng chuẩn là: d= 2^2263*3^2*5*13*17*41*47. Kết quả tính bậc của g theo thuật toán 3.1 là: Ord(g)=38328030

Trong ví dụ trên, giá trị M được tính trong bước 4 của thuật toán 3.1 rất lớn, lý do vì thuật toán đã sử dụng công thức (4), được phỏng theo công thức thuật toán của Pollard, tức là các toạ độ (a,b) không được rút gọn theo cấp của g, đây là sự trả giá về không gian lưu trữ mà thuật toán 3.1 yêu cầu.

5. KẾT LUẬN

Bài báo đã giới thiệu thuật toán "phân tích số" và thuật toán "tìm logarit rời rạc" của Rho Pollard. Phỏng theo phương pháp của Rho Pollard, chúng tôi đã xây dựng được thuật toán tính bậc của phần tử bất kỳ trong $\mathbb{Z}_n^*$ và cài đặt thành công thuật toán trên ngôn ngữ lập trình C++. Kết quả thử nghiệm cho thấy chương trình chạy cho kết quả chính xác, điều đó chứng tỏ thuật toán đảm bảo tính đúng đắn, tính dừng. Kết quả nghiên cứu khẳng định rằng với khả năng tính toán của máy tính hiện nay, việc tìm được toàn bộ các ước nguyên tố không quá 2^{50} của một hợp số cỡ vài ngàn chữ số thập phân là khả thi. Kết quả nghiên cứu làm cơ sở để xây dựng hệ tiêu chuẩn về tham số cho các hệ chữ ký số và biến thể trên vành Z_n.

TÀI LIỆU THAM KHẢO

[1]. Nguyễn Xuân Quỳnh, Nguyễn Hồng Quang, "*Báo cáo khoa học về độ mật chống lại đối phương tích cực của giao thức thiết lập khóa dựa theo ID",* Hội nghị toàn quốc lần thứ V về Tự động hóa 24-26/10/2002, Hà Nội.

[2]. C Meshram. "*Discrete Logarithm and Integer Factorization using IBE*". ISSN: 2089-3191, Bulletin of Electrical Engineering and Informatics Vol. 4, No. 2, June 2015, 160-168.

[3]. Douglas R. Stinson, "*Cyptography theory and practice*", 2003, 191-193.

[4]. OKAMOTO E, (1987), "*Key distribution systems based on identification information*", Proc. Of Crypto.

[5]. Richard Crandall, Carl Pomerance. "*Prime Numbers, A Computational Perspective"*, Second Edition, Springer Science + Business Media, Inc, 2005, 229-231.

ABSTRACT

DEVELOPING POLLARD ALGORITHM
TO COMPUTE ORDER OF ELEMENTS IN $\mathbb{Z}_n^*$

Abstract: *In this submission, Pollard's the large integer factoring algorithm and the algorithm for solving discrete logarithm problem have been introduced. Based on these logarithms, an algorithm to compute the order of elevents in $\mathbb{Z}_n^*$ is developed. The results are able to apply on analysing security of digital signature schemes, which are on basis of discrete logarithm problem in ring Z_n.*

Keyword: Large integer factoring Problem; Discrete Logarithm Problem; Order of elevents.

Nhận bài ngày 19 tháng 4 năm 2017
Hoàn thiện ngày 29 tháng 5 năm 2017
Chấp nhận đăng ngày 20 tháng 6 năm 2017

Địa chỉ: [1] Học viện Khoa học quân sự;
[2] Học viện Kỹ thuật Mật mã.
*Email: levantuan71@yahoo.com