TÌM LOGARIT THEO PHƯƠNG PHÁP BABY-STEP GIANT-STEP

Nguyễn Thanh Sơn*

Tóm tắt: *Bài báo này trình bày về phương pháp baby-step giant-step và phương pháp baby-step giant-step tìm logarit rời rạc trong miền cho trước (thuật toán cải biên). Và sau đó đưa ra đánh giá về khả năng thành công của thuật toán cải biên dựa vào việc định lượng xác suất thành công của thuật toán. Xác suất thành công của thuật toán được xác định phụ thuộc vào kích thước của số nguyên tố p được sử dụng trong bài toán logarit rời rạc.*

Từ khóa: Thuật toán tính logarit rời rạc, Baby-step, Giant-step, Xác suất thành công, Kích thước p.

1. MỞ ĐẦU

Việc ứng dụng bài toán logarit rời rạc (DLP) trong mật mã đã được sử dụng rộng rãi nhằm bảo mật thông tin. Việc xây dựng các thuật toán mật mã ứng dụng DLP trong bảo mật thông tin đã được triển khai phổ biến trên thế giới. Các hệ mật tiêu biểu có thể kể đến như giao thức trao đổi khóa Difie-Hellman, ElGamal,… Trong các hệ mật đó, các tham số mật mã được sử dụng trong hệ mật đóng vai trò cực kỳ quan trọng, quyết định tính an toàn của hệ mật. Các tổ chức tiêu chuẩn quốc tế đã công bố các tiêu chuẩn cho tham số cho bài toán DLP, tuy vậy, trong các tiêu chuẩn này không đưa ra cơ sở lý thuyết để lựa chọn các tham số như vậy. Nhằm đánh giá một cách định lượng về độ an toàn của tham số *p*, bài báo sẽ trình bày việc đánh giá xác suất thành công khi giải bài toán logarit rời rạc trên trường hữu hạn bằng thuật toán baby-step giant-step và thuật toán baby-step giant-step cải biên.

Đầu tiên, chúng tôi nhắc lại định nghĩa bài toán logarit rời rạc.

Định nghĩa: (Bài toán Logarit rời rạc - DLP) Cho số nguyên tố p, một phần tử sinh α của nhóm nhân Z_p^* và một phần tử $\beta \in Z_p^*$. Hãy tìm số nguyên x, $2 \le x \le p-2$, sao cho $\alpha^x \equiv \beta \pmod p$. Ta có thể tìm x bằng phép tính $x = \log_\alpha \beta$.

Độ khó giải của bài toán DLP là độc lập với việc chọn phần tử sinh. Thật vậy, giả sử α và γ là hai phần tử sinh của nhóm nhân G cấp n, và $\beta \in G$. Giả sử $x = \log_\alpha \beta, y = \log_\gamma \beta, z = \log_\alpha \gamma$. Khi đó, $\alpha^x = \beta = \gamma^y = (\alpha^z)^y$. Do đó, $x = zy \pmod n$, ta có: $\log_\gamma \beta = (\log_\alpha \beta)(\log_\alpha \gamma)^{-1} \bmod n$.

Điều này có nghĩa rằng thuật toán bất kỳ tính logarit theo cơ số α cũng có thể sử dụng để tính lôgarit theo cơ sở γ của nhóm nhân G. Đây là bài toán logarit rời rạc truyền thống và cho đến nay chưa có một thuật toán nào giải được bài toán này trong thời gian đa thức.

Phương pháp baby-step giant-step do Daniel Shanks tìm ra [5] dùng để phân tích số nguyên và sau đó dùng để tìm logarit trên nhóm cyclic hữu hạn. Ý tưởng chính của phương pháp khi tính giá trị $\alpha = \log_g a$ với $a \in \langle g \rangle$ có $\#\langle g \rangle = M$ là tìm α dưới dạng $\alpha = u + vm$ với $m = \lceil \sqrt{M} \rceil$ theo theo thuật toán sau:

Thuật toán BSGS.

Input: $a \in \langle g \rangle$, $M = \#\langle g \rangle$.

Output: $\alpha = \log_g a$.

1. $m = \lceil \sqrt{M} \rceil$;

2. $S \leftarrow \{\}$; $u \leftarrow 0$; $b \leftarrow 1$;

3. while (u<m) {$S \leftarrow S \cup \{(b,u)\}$; $b \leftarrow b*g$; $u \leftarrow u+1$;}

4. $v \leftarrow 0$; $c \leftarrow a$; $d \leftarrow g^{-m}$;

5. while ((v<m)&&(c not in head_S)) {$c \leftarrow c*d$; $v \leftarrow v+1$;}

6. assume $c = b$ with $(b,u) \in S$, $\alpha = (u+v*m) \bmod M$;

7. return α;.□

Với S là tập các cặp (b,u) và "head_S" là tập các phần tử đầu (b) của các cặp trong S.

Phương pháp Baby-step giant-step là một phương pháp tất định có chi phí tính toán là $O(\sqrt{M})$ phép toán nhóm và cần đến không gian lưu trữ là $O(\sqrt{M})$ phần tử nhóm. Chính xác hơn, chi phí cho bước 3 và chi phí tối đa của bước 5 của thuật toán là xấp xỉ bằng: $m = \sqrt{M}$ (phép toán nhóm) (1.1)

Để chống lại tấn công theo phương pháp trên, trong quá trình thiết kế các hệ mật có độ an toàn dựa vào bài toán logarit trên nhóm $\langle g \rangle$, theo [2] độ phức tạp của thuật toán baby-step giant step là $\sqrt{\#\langle g \rangle}$, với $2^{s(y)}$ là số phép toán cơ bản mà người tấn công không thể thực hiện được cho đến năm ***y*** và được gọi là "*ngưỡng an toàn*" cho đến năm ***y***, ta suy ra:

$$\sqrt{\#\langle g \rangle} > 2^{s(y)} \rightarrow \#\langle g \rangle > 2^{2s(y)} \quad (1.2)$$

Ký hiệu số phép toán nhóm có thể thực hiện được cho đến năm ***y*** của kẻ tấn công là ***AttackPower***, đại lượng ***AttackPower*** còn được gọi là "*sức mạnh của người tấn công*", thì với điều kiện (1.2) về kích thước nhóm cho nên nếu tuân theo phương pháp Baby-step giant-step truyền thống thì thậm chí ***AttackPower*** = $\#\langle g \rangle/2$, kẻ tấn công cùng lắm là hoàn thành được bước 3 và do đó, không thể tính được giá trị $\log_g a$ nào.

2. DÙNG PHƯƠNG PHÁP BABY-STEPS, GIANT-STEPS TÌM LOGARIT TRONG MIỀN [A, A+m^2)

2.1. Hàm *Logarit*(a, g, M, m, A)

Hàm ***Logarit***(a, g, M, m, A) trình bày trong phần này gồm 5 tham số đầu vào đó là: a, g là hai phần tử của nhóm G nào đó với $a \in \langle g \rangle \subset G$; M, m và A là các số nguyên với $M = \#\langle g \rangle$, $A \geq 0$, $m > 0$.

Giá trị hàm là $\log_g a$ khi $\log_g a \in [A, A+m^2)$ và là "*Failure*" trong trường hợp ngược lại. Ở đây, cần lưu ý một điểm đó là nếu $A+m^2 \leq M$ thì $[A, A+m^2)$ được hiểu như nghĩa thông thường, ngược lại nó là $[A, M) \cup [0, A+m^2 \bmod M)$.

Việc tính ***Logarit***(a, g, M, m, A) được thực hiện theo thuật toán sau:

Thuật toán 1. (tính ***Logarit***(a, g, M, m, A))

Input: a, g, M, m, A.

Output: $\alpha = \log_g a$ if $(\log_g a \in [A, A+m^2])$ else α = "*Failure*".

1. $S \leftarrow \{\}$; $u \leftarrow 0$; $b \leftarrow e$; //ở đây e là phần tử trung hòa của nhóm

2. while (u<m) $\{S \leftarrow S \cup \{(b,u)\}$; $b \leftarrow b*g$; $u \leftarrow u+1;\}$

//ở trên "*" là ký hiệu phép toán nhóm

3. $v \leftarrow 0$; $c \leftarrow a*g^{-A}$; $d \leftarrow b^{-1}$; //đến đây $b=g^m$.

4. while ((v<m)&&(c not in head_S)) $\{c \leftarrow c*d$; $v \leftarrow v+1;\}$

5. if (v==m) α = "*Failure*"

else {assume c = b with $(b,u) \in S$, $\alpha = (A+u+v*m) \bmod M;\}$

6. return α;.□

2.2. Phân tích thuật toán 1

Tính đúng đắn và chi phí tính toán của thuật toán 1 được cho trong kết quả dưới đây.

Kết quả 1. *Thuật toán 1 có chi phí tối đa, ký hiệu là C_{max} được đánh giá như sau:*

$C_{max} \leq 2m + 4\log_2 M$ *(phép toán nhóm)* (2.1)

và sẽ tìm được $\log_g a$ khi và chỉ khi $\log_g a \in [A, A+m^2)$.

Chứng minh:

Rõ ràng trong bước 2 cần thực hiện đúng m phép toán nhóm. Việc tính c ở bước 3 cần 1 phép toán nhóm, một phép lũy thừa và một phép tính phần tử ngược trong nhóm. Bước 4 thực hiện tối đa là m vòng lặp, trong mỗi vòng lặp cần đúng 1 phép toán nhóm. Biết rằng mỗi phép lũy thừa nhóm hoặc tính phần tử ngược (theo công thức $x^{-u} = x^{M-u}$) trong nhóm cần không quá $2\log_2 M$ phép toán nhóm. Tóm lại, chi phí tối đa của thuật toán là $2m + 4\log_2 M$ phép toán nhóm.

Nếu $\alpha = \log_g a \in [A, A+m^2)$ điều này tương đương với sự tồn tại $0 \leq u, v < m$ sao cho: $\alpha = A+u+vm$ (2.2)

Đẳng thức trên tương đương với: $g^{\alpha} g^{-A} g^{-vm} = g^u$

hay $a g^{-A} (g^{-m})^v = g^u$ (2.3)

Các giá trị c tính trong bước 3 và 4 của thuật toán chính là giá trị trong vế phải của (2.3) với các v tương ứng còn tập S xác định theo bước 1 và 2 chứa toàn bộ các giá trị vế phải của (2.3) cho nên bước 5 của thuật toán luôn được thực kiện với điều

kiện v<m và đầu ra của thuật toán được xác định theo (2.2). Nói một cách khác thuật toán 1 là đúng đắn.□

3. TẤN CÔNG CÁC HỆ MẬT CÓ ĐỘ AN TOÀN DỰA VÀO TÍNH KHÓ GIẢI CỦA BÀI TOÁN LOGARIT TRÊN NHÓM $\langle g \rangle$

Trong phần này, chúng tôi đưa ra chiến thuật tấn công của người có sức mạnh là ***AttackPower*** phép toán nhóm nhằm giải bài toán logarit trên nhóm $\langle g \rangle$ với $\#\langle g \rangle$=M. Công cụ sử dụng của người tấn công là thuật toán 1 và để đơn giản trong trình bày, ở đây, ta luôn giả thiết rằng để thực hiện được thuật toán 1 thì người sử dụng chỉ cần chi phí tối đa là 2m phép toán nhóm.

3.1. Thuật toán tấn công

Thuật toán 2.(giải bài toán logarit)

Input: a, g where $a \in \langle g \rangle$, $M = \#\langle g \rangle$.

Output: $\alpha = \log_g a$ if find out. Else α = "*Failure*".

1. m ← ***AttackPower***/2;

2. A ← random[0, M);

3. return ***Logarit***(a, g, M, m, A);.□

3.2. Phân tích thuật toán 2

Với giả thiết đưa ra trên, việc chọn tham số m như trong bước 1 của thuật toán thì người tấn công luôn tính được ***Logarit***(a, g, M, m, A) với kết quả là $\log_g^a$ hoặc "Failure", như vậy, phân tích của chúng ta ở đây chỉ đánh giá khả năng tìm được $\log_g a$ của thuật toán tức là giá trị α ở đầu ra khác "*Failure*". Theo như đã được phân tích về hàm ***Logarit***(a, g, M, m, A) thì điều trên xảy ra khi và chỉ khi $A \le \log_g a < A+m^2$, điều này có nghĩa có đúng m^2 trong tổng số M giá trị A thỏa mãn điều kiện trên và theo công thức xác suất cổ điển ta có xác suất thành công của thuật toán, ký hiệu là $Prob_{succ}$, cho bởi đẳng thức sau: $Prob_{succ} = \frac{m^2}{M}$ (3.1)

3.3. Sự an toàn của các hệ mật trước tấn công theo phương pháp cải tiến

Theo định nghĩa về ngưỡng an toàn ta có quan hệ giữa s và ***AttackPower*** theo bất đẳng thức sau $\textbf{\textit{AttackPower}} < \frac{2^s}{t_G}$ (3.2)

Ở trên, t_G là số các phép toán cơ bản để thực hiện một phép toán trên nhóm G.

Từ việc xác định m = ***AttackPower***/2 trong bước 1 của thuật toán 2 và từ bất đẳng thức (1.2) về việc chọn $M = \#G \ge 2^{2s}$ ta được: $m < \frac{\sqrt{M}}{t_G}$ (3.3)

Thay (3.3) vào (3.1) ta có bất đẳng thức sau: $Prob_{succ} < t_G^{-2}$ (3.4)

Như đã biết, các nhóm hữu hạn mà bài toán logarit trên đó được làm cơ sở an toàn cho các hệ mật bao gồm: Nhóm con cấp q nguyên tố của trường F_p hoặc F_{2^k} [4], nhóm con cấp nguyên tố M các điểm của đường cong elliptic trên trường F_p hoặc F_{2^k} [3]. Để có những đánh giá về tính an toàn của các hệ mật nêu trên, chúng ta cần đưa vào các thông tin về tham số t_G tương ứng.

3.4. Đánh giá độ an toàn của các tham số dùng cho hệ mật Diffie-Hellman đối với các tấn công dựa trên thuật toán GSBS và GSBS cải tiến

Ký hiệu t_G là số phép toán cơ bản để thực hiện một phép toán nhóm trên GF(p). Phép toán nhóm trên GF(p) chính là phép $b \cdot g (\mathrm{mod}\, p)$. Chính vì vậy, ta sẽ ước lượng số lượng phép toán cơ bản để thực hiện phép toán $b \cdot g (\mathrm{mod}\, p)$.

Theo lý thuyết về độ phức tạp tính toán khi thực hiện phép nhân $b \cdot g$ có độ phức tạp $O(n^2)$, với n là độ dài chữ số của thừa số [6],[7]. Khi có kết quả của phép nhân $b \cdot g$, phép chia lấy số dư theo modulo p cũng có độ phức tạp tính toán $O(n^2)$,[6],[7].

Như vậy, phép toán $b \cdot g (\mathrm{mod}\, p)$ cần khoảng $2n^2$ phép toán cơ bản, với n là số chữ số của các toán hạng.

Vì b và g đều là phần tử của GF(p) nên độ dài theo bit của b,g phải nhỏ hơn hoặc bằng độ dài theo bit của p. Ta gọi độ dài theo bit của p là l_p.

Với kiến trúc máy tính hiện nay, khi thực hiện tính toán với số lớn, các số được lưu theo cơ số 2^{32}, nghĩa là một chữ số (một word) có kích thước là 32 bit, suy ra $n = \frac{l_p}{32}$. Số lượng phép toán cơ bản để thực hiện một phép toán nhóm là:

$$t_G = 2n^2 = 2\left(\frac{l_p}{32}\right)^2 = \frac{{l_p}^2}{512} \tag{3.5}$$

Vậy xác suất thành công để thực hiện thuật toán cải tiến là:

$$Prob_{succ} < \frac{1}{4} {t_G}^{-2} = \frac{1}{4}\left(\frac{l_p^2}{512}\right)^{-2} = \frac{512^2}{4l_p^4} = \frac{2^{16}}{l_p^4} \tag{3.6}$$

(với l_p là độ dài số nguyên tố p theo bit)

Nếu $l_p \geq 1024$ thì $Prob_{succ} < \frac{2^{16}}{2^{40}} = \frac{1}{2^{24}} < 10^{-6}$. Như vậy, với kích thước tối thiểu của p là 1024 bit, thì xác suất thành công của thuật toán cải tiến đã rất nhỏ, nhỏ hơn một phần triệu.

4. KẾT LUẬN

Bài báo đã trình bày về thuật toán baby-step giant-step và thuật toán cải tiến của nó. Sau đó, chúng tôi đưa ra đánh giá về xác suất thành công của thuật toán trong

việc tấn công lên các hệ mật có độ an toàn dựa vào tính khó giải của bài toán logarit trên nhóm $\langle g \rangle$. Từ đó, hoàn toàn có thể định lượng được xác suất thành công của thuật toán trên với việc xác định kích thước số nguyên tố p (được khuyến nghị trong các bộ tiêu chuẩn [8]) để sử dụng trong các hệ mật để đảm bảo an toàn. Trong các nghiên cứu tiếp theo, chúng tôi sẽ công bố một số kết quả liên quan đến phương pháp tính logarit rời rạc dựa trên các số mũ có trọng số thấp.

TÀI LIỆU THAM KHẢO

[1]. Darrel Hankerson, Alfred Menezes, Scott Vanstone, "*Guide to Elliptic Curve Crytography"*, Springer-Verlag New York, Inc. 2004.

[2]. Arjen K. Lenstra, Key Lengths, "*Contribution to The Handbooks of Information Security"*, Lucent Technologies and technische Universiteit Eindhoven, June 30, 2004.

[3]. V. Miller. "*Use of elliptic curves in cryptography"*. In H. Williams, editor, Advances in Cryptology, Proc. Eurocrypt '85, volume 218 of Lecture Notes in Computer Csience, pages 417-426, Springer-Verlag,1985.

[4]. A. Odlyzko. "*Discrete logarithms in finite fields and their cryptographyc significance"*. In Advances in Cryptology, Proc. Eurocrypt '84, volume 209 of Lecture Notes in Computer Csience, pages 224-313, Springer-Verlag,1985.

[5]. D. Shanks, "*Class number, a theory of factorization, and genera"*. In 1969 Number Theory Institue, Stony Brook, N. Y., volume 20 of Proc. Sympos. Pure Math., pages 415-440. Amer. Math. Soc., 1971.

[6]. E. B. Makhovenko, "*Lý thuyết số trong mật mã",* Moscow, 2006, chương *4.*

[7]. Victor Shoup, "*A Computational Introduction to number theory and Algebra, (version 2.2)",* mục 3.3

[8]. *"NIST SP800-57 Part 1 Revision 4: Recommendation for Key Management",* Part 1: General, 1/2016.

ABSTRACT

SOLVE DISCRETE LOGARITHM PROBLEM BY BABY-STEPS, GIANT-STEPS METHOD

In this paper, the algorithm baby-step giant-step and modified algorithm is described. Then the evaluation of success of modified algorithm is given. This evaluation is based on success probability of algorithm. Success probability depends on size of prime number p, used in discrete logarithm problem.

Keywords: Discrete logarithm, Baby-steps, giant-steps, Success probability, Size of p.

Nhận bài ngày 15 tháng 9 năm 2016
Hoàn thiện ngày 01 tháng 11 năm 2016
Chấp nhận đăng ngày 14 tháng 12 năm 2016

Địa chỉ: Học viện Kỹ thuật Mật mã, Ban Cơ yếu chính phủ;
* Email: sonngt2002@yahoo.com