Mathematics and Computer Science
2023; 8(2): 57-61
http://www.sciencepublishinggroup.com/j/mcs
doi: 10.11648/j.mcs.20230802.14
ISSN: 2575-6036 (Print); ISSN: 2575-6028 (Online)

Analysis and Optimization of Improved Index Calculus Algorithm

Hu Jianjun

School of Digital Media, Lanzhou University of Arts and Science, Lanzhou, China

Email address:
Hujj518@126.com

To cite this article:
Hu Jianjun. Analysis and Optimization of Improved Index Calculus Algorithm. *Mathematics and Computer Science*.
Vol. 8, No. 2, 2023, pp. 57-61. doi: 10.11648/j.mcs.20230802.14

Received: February 22, 2023; **Accepted**: March 30, 2023; **Published**: April 13, 2023

Abstract: IC (Index Calculus) algorithm is the most effective probability algorithm for solving discrete logarithm of finite prime fields, and IICA (improved Index Calculus algorithm) is an improved algorithm based on IC in the third stage. The essence of IICA is to convert the number required to solve the discrete logarithm into the product of the power of prime factors, and then multiply every prime factor larger than the smooth bound by a smooth number approximating a large prime p from the right end, that is, to perform congruence transformation for every prime factor larger than the smooth bound. If all prime factors larger than the smooth bound fall within the smooth bound, the number required to solve the discrete logarithm is successfully solved. Unfortunately, given a large prime number p, some prime factors do not have congruent transformations of smooth numbers. For this, this paper analyzes the features of the IICA algorithm, based on the characteristics of IICA algorithm, is given when the IICA algorithm cannot undertake congruence transformation termination conditions, namely when $\lceil p/p_i \rceil$ not smooth algorithm is terminated, where p_i is greater than a smooth boundary element factor. According to the $\lceil p/p_i \rceil$ not smooth algorithm was terminated when judging conditions, optimized the IICA algorithm, and the correctness of the optimization algorithm is verified by an example.

Keywords: Discrete Logarithm, Decomposition Base, Smooth Boundary, Factorization, Prime Finite Fields

1. Introduction

IC algorithm is the most effective method to solve the discretized logarithm problem in a finite field of prime numbers [1-9]. IC algorithm also plays an important role in solving discrete logarithms of elliptic curve cryptosystems [15-17]. IC algorithm is divided into three stages to solve the discrete logarithm. The first stage is to randomly select $t+d$ smooth numbers, where t is the number of prime numbers in the smooth bound. The second stage is to solve the discrete logarithm of all prime factors in the smooth boundary, and the third stage is to solve the discrete logarithm of the given number.

IICA uses the approximation method to optimize the random trial of IC algorithm in stage 3 and achieves good results [10]. However, what is the end condition of IICA, what is the probability of success, and what is the size of the smooth boundary need to be further discussed. In fact, the key to the success of the IICA algorithm is that the maximum prime factor after each congruence transformation must be decreasing, and the congruence transformation assumes that the quotient of p divided by the prime number q is smooth. Therefore, it is of great value to analyze the characteristics of IICA and optimize IICA algorithm for improving the application of IICA.

2. Preliminaries

2.1. Prime Finite Fields

Let p is a large prime number, $N = p-1$ is the order of the cyclic group about Z_p^*, and $g \in Z_p^*$ is the generator of the cyclic group, that is $g^N \bmod p = 1$. For $\forall y \in Z_P^*$, find $x = \log_g y$, this is the discrete logarithm solution problem over prime finite fields F_p.

In addition, according to literature [10], Equation (1) is established.

$$\begin{cases} g^{p-1} \equiv 1(\bmod p) \\ g^{(p-1)/2} \equiv -1(\bmod p) \end{cases}, \quad (0 < g < p) \tag{1}$$

2.2. IC Algorithm Overview

The basic idea of IC algorithm is to construct a set of small prime numbers called factor basis on the cyclic group Z_P^*, expressed as $M = \{p_1, p_2, \cdots, p_t\}$, the set M is the set of initial prime numbers. The IC algorithm is divided into two stages (Literature [12] is divided into three stages). The first stage constructs and solves $t+d$ (d is an appropriate constant) linear equations to find the discrete logarithm of each element in the factor base; In the second stage, a number *m* is randomly selected. If all the prime factors of yg^m after decomposition fall into the decomposition basis, then the discrete logarithm of y is calculated.

Definition 1: A smooth integer is an integer that has only small prime factors $p_1, p_2, \cdots, p_t$ that satisfies relation $p_1 < p_2 < \cdots < p_t$. An integer is $T-smooth$ if all of its prime factor are less then or equal to T . T is called a smooth bound, $p_t \le T$.

Each equation of a system of linear equations is congruence to a modular P. In the first stage, the process of constructing the system of linear equations is as follows: $\forall z \in Z_P^*$ is selected randomly. If g^z is smooth, then g^z is decomposed and the next equation is found until $t+d$ linear equations are found. Solve the system of linear equations and find the discrete logarithm of all the elements in the decomposition basis.

In the second stage, $\forall m \in Z_P^*$ is randomly selected and yg^m calculated, and yg^m is tried to be written as the product of elements in set M , i.e., equation (2).

$$yg^m = \prod_{i=1}^{t} p_i^{d_i}, d_i \ge 0 \tag{2}$$

If successful, equation (3) is the discrete logarithm.

$$\log_g y = (\sum_{i=1}^{t} d_i \log_g p_i - m) \bmod (p-1) \tag{3}$$

Otherwise, the second stage is repeated until yg^m is written as the product of the elements in the set M .

2.3. Overview of IICA [10]

Let $y = \prod_{j=1}^{k} q_j^{w_j}$, for $\forall q_j \in Z_P^*(j = 1, 2, \cdots, k)$ be a prime and $q_1 < q_2 < \cdots < q_k$. $r = \left\lVert \frac{p}{q_j} \right\rVert$ represents the product of the prime powers in set M , r is a smooth number.

Definition 2: Suppose $q_j > p_t$ (p_t is the largest element in the set M), the symbol " $\Rightarrow$ " represents a congruence transformation, then the operation $y \bmod P \Rightarrow \frac{(r * y) \bmod P}{r} \bmod P$ is said to be a congruence transformation of $y \bmod P$.

The basic idea of IICA is: the first stage is the same as IC, and the solution process of the second stage is as follows Algorithm 1:

Algorithm 1: IICA

1) If y mod $p = -1$, then output $\log_g y = \frac{N}{2}$, go to 7).
2) If y mod $p = 1$, then output $\log_g y = N$, go to 7).
3) Decompose $y = \prod_{j=1}^{k} q_j^{w_j}$ and save $q_1 < q_2 < \cdots < q_k$.
4) If y is smooth, output $\log_g y = (w_1 \log_g q_1 + w_2 \log_g q_2 + \cdots + w_k \log_g q_k) \bmod N$, go to 7).
5) The congruence transformation is applied to the primes in y greater than p_t until all the primes fall into the set M .
6) Compute the discrete logarithm of y and output $\log_g y = (\log_g (U \bmod P) - \log_g V) \bmod N$, where U (the numerator in the congruence transformation) is the product of the prime powers of the elements in the set M transforming y, and V (the denominator in the congruence transformation) is the product of the prime powers of the elements in the set M with the transformation y.
7) The algorithm is complete.

3. IICA Analysis and Optimization

Definition 3: Let $L = \left\lceil \frac{p}{p_i} \right\rceil$, then L is the strict right approximation of p_i with respect to p.

IICA adopts $\left\lceil \frac{p}{p_i} \right\rceil p_i > p$ method, so IICA is also called strict right approximation method.

Theorem 1: The number of successful or unsuccessful round of congruence transformation by IICA is at most $\log_2 p$.

Proof: Since 2 is the smallest prime in the smooth bound T , and all primes of the cyclic group Z_p^* are less than P, so any prime p_i less than P, if $\left\lceil \frac{p}{p_i} \right\rceil$ is smooth, the prime of p_i after a congruence transformation reduces at least one prime below $p_i / 2$, and thus the number of successful or failed congruence transformations of a round of IICA is at most $\log_2 p$.

Lemma 1 [11]*:* Suppose $\Psi(T,p)$ represents the number of smooth integers from T to P, let $\mu=\log_T p$, when $p\to\infty$, then $\Psi(T,p)\ge p\cdot\exp\left[\left(-1+o(1)\mu\ln\ln p\right)\right]$.

Theorem 2: The probability of IICA smooth number is at least $\exp\left[\left(-1+o(1)\mu\ln\ln p\right)\right]$.

Proof: Since the smooth number in IC must be the smooth number in IICA, it can be seen from lemma 1 that the probability of IC algorithm being the smooth number is $\Psi(T,p)/p$, namely $\exp\left[\left(-1+o(1)\mu\ln\ln p\right)\right]$. In addition, in the case of successful IICA congruence transformation, there are non-smooth numbers that can all fall into T for $\log_2 p$ transformations at most, so the probability of IICA smooth number is at least $\exp\left[\left(-1+o(1)\mu\ln\ln p\right)\right]$.

Theorem 3: The premise that IICA can congruence transformation every time is that $\left\lceil \frac{p}{p_i} \right\rceil$ is smooth, where $i>t$.

Proof: Suppose $\left\lceil \frac{p}{p_i} \right\rceil$ is non-smooth, where $i>t$, then there are discrete logarithms of unknown prime factors in the denominator of congruence transformation. These unknown prime factors cannot be eliminated at the end of congruence transformation, and IICA fails. Therefore, the premise that IICA can congruence transformation every time is that $\left\lceil \frac{p}{p_i} \right\rceil$ is smooth.

Theorem 4: If $\frac{p}{T}<x<p$ is a prime number, the prime factor of x must decrease after a congruence transformation.

Proof: Because $\frac{p}{T}<x<p$, so $1<\frac{p}{x}<T$, since T is a smooth bound, according to the rules of congruence transformation, there is $p<Tx<pT$, so $0<Tx-p<pT$, assuming x after a congruence transformation of the prime factor does not decrease, means $Tx-p>x$, that is $x>\frac{p}{T-1}$. $x>\frac{p}{T-1}$ is in contradiction with the assumption $\frac{p}{T}<x<p$, so the conclusion is valid.

According to Theorem 4, there are $\pi(p)-\pi(p/T)$ prime numbers that must decline after the first congruence transformation, where $\pi(x)$ represents the number of primes that do not exceed the real number *x*. Prime numbers from T to $\frac{p}{T}$, in the implementation of IICA's strict right approximation, because $\left\lceil \frac{p}{p_i} \right\rceil$ is not necessarily smooth, so the congruence transformation of IICA may not exist. If the congruence transformation does not exist, it needs to perform the second and third stage operations similar to IC, which is the reason why IICA algorithm is still a probabilistic algorithm.

IICA does not give a method for dealing with prime factors when congruence transformations do not exist. In order to ensure that IICA effectively ends or stops invalid congruences transformation, the following algorithm 2 of optimization is carried out on IICA according to theorems 3 and 4 as well as the solution methods of the second and third stages of IC:

Algorithm 2: Optimization of IICA

1) Let y_0=y, and m=0.
2) Let $y=y_0 \mathrm{g}^m \bmod \mathrm{p}$, if $y \bmod P=-1$, then output $\log_g y=\frac{N}{2}-m$, go to 8).
3) If $y \bmod P=1$, then output $\log_g y=N-m$, go to 8).
4) Decompose $y=\prod_{j=1}^{k} q_j^{w_j}$ and save $q_1<q_2<\cdots<q_k$.
5) If y is smooth, output $\log_g y=(w_1\log_g q_1+w_2\log_g q_2+\cdots+w_k\log_g q_k-m) \bmod N$, go to 8).
6) Carry out congruence transformation for all prime factors in y greater than p_t until all prime numbers fall into set M. If no congruence transformation exists for prime factors in the congruence transformation process, then select an integer of $0<m<N$ at random and go to 2), repeat 2)-6).
7) Compute the discrete logarithm of y and output $\log_g y=(\log_g(U \bmod P)-\log_g V) \bmod N$, where U (the numerator in the congruence transformation) is the product of the prime powers of the elements in the set M transforming y, and V (the denominator in the congruence transformation) is the product of the prime powers of the elements in the set M with the transformation y.
8) The algorithm is complete.

4. Comparison of IICA and IC

Lemma 2 [11]*:* Assuming a smooth bound $T=\exp\left[\left(\sqrt{2}/2\right)\left(\sqrt{\ln p\ln\ln p}\right)\right]$, then the probability of IC algorithm failure is at most 1/2, and the expected running time is $\exp\left[\left(2\sqrt{2}+o(1)\right)\sqrt{\ln p\ln\ln p}\right]$.

IICA is an improvement of IC algorithm, in essence, it is an improvement of a non-smooth prime factorization. IC algorithm directly uses subexponential time probability algorithm to solve the non-smooth integer, while IICA needs to use subexponential time probability algorithm to solve the non-smooth number that cannot be transformed to a smooth bound. Therefore, IICA still uses the smooth bound assumed by Lemma 2.

IC and IICA have the same operation process in the first stage, and IICA is only the optimization of the second and third stages of IC, so IC and IICA have the same time complexity. The time complexity of the first stage is $O\left(\left(\sqrt{\ln p \ln\ln p}\right)^{2+o(1)}\right)$, and the time complexity of the second and third stages is $O\left(\left(\sqrt{\ln p \ln\ln p}\right)^{3/2+o(1)}\right)$ [12-14].

According to theorem 2, the probability of success of IICA is greater than that of IC algorithm.

According to Theorem 4, prime numbers from p_t to $\frac{p}{T}$ and numbers containing prime factors from p_t to $\frac{p}{T}$, their discrete logarithms are solved by IICA algorithm, with a very low probability of success.

5. Algorithm Verification

Let $p = 14087$ and $g = 5$ be a generator of Z_P^* order $N = 14086$, $M = \{2,3,5,7,11,13\}$, calculate the discrete logarithm of $y = 4909$.

Assume that the discrete logarithms of prime numbers 2, 3, 7, 11 and 13 in smooth bound M have been obtained by IC algorithm, that is $\log_g 2 = 3028$, $\log_g 3 = 5018$, $\log_g 7 = 8542$, $\log_g 11 = 5446$, $\log_g 13 = 4729$. The calculation process of $\log_g y$ is as follows:

Since 4909 is prime, and $\left[\left\lceil \frac{14087}{4909} \right\rceil\right] = 3$ is a smooth number, a congruence transformation of y is $\Rightarrow \frac{4909 \cdot 3}{3} \bmod p = \frac{640}{3} \bmod p = \frac{2^7 \cdot 5}{3} \bmod p$.

Now all prime numbers fall into the smooth bound, so $\log_g 4909 = 7\log_g 2 + \log_g 5 - \log_g 3 = 2093$.

Verify $g^{2093} \bmod p = 4909$, which is consistent with the algorithm analysis. The congruence transformation only goes through one congruence transformation.

Using the same parameters as above, the discrete logarithm of $y = 5872$ is calculated. The calculation process is as follows:

Decompose $y = 5872 = 2^4 \cdot 367$. Since 367 is prime and cannot be decomposed again, 5872 is a non-smooth integer. Since $r = \left\lceil \frac{14087}{367} \right\rceil = 31$ is a non-smooth integer, an integer m greater than 0 needs to be randomly selected. If $m = 3$ is randomly selected, then $y = 5872 \cdot 5^9 \bmod 14087 = 1476 = 2^2 \cdot 269$.

Since $r = \left\lceil \frac{14087}{269} \right\rceil = 53$ is a non-smooth integer, the solution fails and m needs to be selected again. Suppose that $m = 9$ is selected by luck, then $y = 5872 \cdot 5^9 \bmod 14087 = 2081$.

Since 2081 is prime and $r = \left[\left\lceil \frac{14087}{2081} \right\rceil\right] = 7$ is smooth, then the first congruence transformation of y is $\Rightarrow \frac{2081 \cdot 7}{7} \bmod 14087 \Rightarrow \frac{480}{7} \bmod 14087 = \frac{2^5 \cdot 3 \cdot 5}{7} \bmod 14087$. Now all the prime numbers fall into the smooth bound, so $\log_g 5872 = 5\log_g 2 + \log_g 3 + \log_g 5 - \log_g 7 - 9 = 11608$.

Verify $g^{11608} \bmod p = 5872$, the discrete logarithm of 5872 is successfully solved through a congruence transformation under the circumstance that $m = 9$ is randomly selected, which is consistent with the analysis of the algorithm.

The number of congruence transformation for solving the two discrete logarithms given by this example does not exceed $\log_2 14087$. The first example does not choose m and is successfully solved by one congruence transformation; the second example chooses m twice. After selecting 9, it is successfully solved by one congruence transformation, which verifies the correctness of theorem 1. In this example, the smooth bound of $M = \{2,3,5,7,11,13\}$ is 16, according to the smooth bound $T = \exp\left[\left(\sqrt{2}/2\right)\left(\sqrt{\ln p \ln\ln p}\right)\right] = 26.65$ assumed by Lemma 2, and then 23 should also be included in the smooth bound. In fact, the larger the smooth bound, the more smooth number, and the greater the probability of success for IICA and IC.

It is found that for any prime number from 528 to 14086, the maximum prime factor decreases after the first congruence transformation, while for any prime number from 26 to 528, some have no smooth numbers to multiply, such as 367, which verifies the correctness of theorem 4. That is to say, given this example, the discrete logarithm of the primes from 17 to 828 are solved using IICA as well as IC. The solution of discrete logarithm of 367 confirms this.

The above two examples were used in IICA to solve the discrete logarithm of the non-smooth number in polynomial time. However, IC can only be solved using probabilistic algorithm, which verifies that the probability of success of IICA algorithm is greater than that of IC.

6. Conclusion

The solution of discrete logarithm in prime finite field has an important impact on the security of cryptosystem based on prime finite field. Therefore, the solution of discrete logarithm in prime finite field is a hot topic in computer and mathematics circles. IC algorithm has few restrictions on group structure, and has sub-exponential running time, so it is favored by people. However, IC algorithm is a probabilistic algorithm, and there are a lot of trial problems. IICA is an improved algorithm of IC. IICA uses approximation method instead of trial method to achieve better results. The essence of IICA algorithm is analyzed, the correlation analysis model is established, and the correctness of the model is verified by an example.

In the future, the safety impact of IICA on elliptic curves will be further studied.

Acknowledgements

This paper is one of the phased achievements of the project "Research on Key Techniques of Elliptic Curve Cryptography" (2021FWDF15) funded by Lanzhou University of Arts and Sciences, which serves the local Economic and Social Development Plan Fund.

References

[1] Adleman L M. A subexponential algorithm for discrete logarithms with applications to cryptography [C]. Proc. 20th IEEE Found. Comp. Sei. Symp., IEEE Computer Society, Long Beach, CA, 1979, 55-60.

[2] Adleman L M, Demarrais J. A subexponential algorithm for discrete logarithms over all finite fields [J]. Mathematics of computation, 1993, 61 (203): 1-15.

[3] Andreas E, Pierrick G. A general framework for subexponential discrete logarithm algorithms [J]. Acta Arithmetica, 2002, 102: 83-103.

[4] Enge A, Gaudry P, Thomé E. An L (1/3) discrete logarithm algorithm for low degree curves [J]. Journal of Cryptology, 2011, 24 (1): 24-41.

[5] Padmavathy R, Bhagvati C. Index calculus method based on smooth numbers of ± 1 over Z_p^* [J]. International Journal of Network Security, 2013, 15 (1): 210-218.

[6] Gretel S S. A brief survey of the discrete logarithm problem [D]. University of Hawaii at Manoa, 2011.

[7] Hu Jianjun, Wang Wei, Li Hengjie. An improved Index Calculus algorithm [J]. Journal of Nanchang University (Engineering & Technology), 2016, 38 (3): 286-289.

[8] Mukhopadhyay M. Aspects of Index Calculus Algorithms for Discrete Logarithm and Class Group Computations [D]. Indian Statistical Institute, Kolkata, 2021.

[9] Zhang Mingyao, Zhang Fan. Concrete Mathematics: Fundamentals of Computer Science (Second Edition) [M]. Beijing: Posts and Telecommunications Press, 2013.

[10] Hu Jianjun. A discrete logarithm solution method based on ICA [J]. Engineering Journal of Wuhan University, 2021, 54 (9): 874-878.

[11] Victor Shoup. A Computational Introduction to Number Theory and Algebra (BETA version 3) [M]. London: Cambridge University Press, 2004.

[12] Howell J S. The Index Calculus Algorithm for Discrete Logarithms [D]. Clemson University, 1998.

[13] Silverman J H, Suzuki J. Elliptic curve discrete logarithms and the index calculus [C]. International Conference on the Theory and Application of Cryptology and Information Security. Springer, Berlin, Heidelberg, 1998: 110-125.

[14] Padmavathy R, Bhagvati C. Performance analysis of index calculus method [J]. Journal of Discrete Mathematical Sciences and Cryptography, 2009, 12 (3): 353-371.

[15] Galbraith S D, Zobernig L. Obfuscated fuzzy hamming distance and conjunctions from subset product problems [C] // Theory of Cryptography: 17th International Conference, TCC 2019, Nuremberg, Germany, December 1–5, 2019, Proceedings, Part I. Cham: Springer International Publishing, 2019: 81-110.

[16] De Micheli G, Gaudry P, Pierrot C. Asymptotic complexities of discrete logarithm algorithms in pairing-relevant finite fields [C] // Advances in Cryptology–CRYPTO 2020: 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17–21, 2020, Proceedings, Part II 40. Springer International Publishing, 2020: 32-61.

[17] Mukhopadhyay M, Sarkar P. Pseudo-Random Walk on Ideals: Practical Speed-Up in Relation Collection for Class Group Computation [J]. Cryptology ePrint Archive, 2021.