A New Method for Solving Discrete Logarithm Based on Index Calculus

Jianjun Hu

School of Digital Media, Lanzhou University of Arts and Science,Lanzhou 730010, China

Abstract: Index Calculus (IC) algorithm is the most effective probabilistic algorithm for solving discrete logarithms over finite fields of prime numbers, and it has been widely applied to cryptosystems based on elliptic curves. Since the IC algorithm was proposed in 1920, the research on it has never stopped, especially discretization of prime numbers on the finite fields, both the algorithm itself and its application have been greatly developed. Of course, there has been some research on elliptic curves, but with little success. For the IC algorithm, scholars pay more attention to how to improve the probability of solving and reduce the time complexity of calculation. It is the first time for the IICA to study the optimization problem of the IC by using the method of integer. However, the IICA only studies the case of integer up, and fails to consider the case of integer down. It is found that the integer direction of the IICA can be integer up or integer down, but the concept of modular multiplication needs to be used when integer down. After optimizing the IICA, the probability of successful solution of discrete logarithm is increased by nearly 2 times, and the number of transformations is also reduced to a certain extent, thus reducing the time complexity of solution. The re-optimized the IC algorithm greatly improves the probability of successful the IC solution. This research result poses a serious challenge to cryptosystems based on finite fields of prime numbers.

Key words: Discrete logarithm; decomposition base; smooth boundary; factorization; prime finite fields

1 Introduction

Index Calculus algorithm is the most effective probabilistic algorithm for solving discrete logarithms over finite fields of prime numbers. Since the IC algorithm was proposed in 1920, the research on it has never stopped, and both the algorithm itself and its application have been greatly developed. Initially,the IC was mainly used in the finite fields of prime numbers [1–12], after 1985, with the application of elliptic curve cryptosystem, the IC calculation has been extended to the solution of elliptic curve cryptosystem discrete logarithm [13–17]. As far as the IC is concerned, scholars pay more attention to how to improve the probability of solving linear equations and discrete logarithm and reduce the time complexity of calculation. The IICA of [10] improves the probability of solving discrete logarithm, but the IICA only studies the case of integer up, and fails to consider the case of integer down [10, 11]. It is found that the integer direction of the IICA can be integer up or integer

Received date: 2025-01-04 Final date:

down, but the concept of modular multiplication needs to be used when integer down. Therefore, the IICA is optimized again. After optimization, the probability of successful solution of discrete logarithm is increased by nearly 2 times, and the number of transformations is reduced to a certain extent, thus reducing the time complexity of solution.

2 Preparatory knowledge

2.1 Prime finite field

Set p is a large prime Numbers, N=$p-1$ is a cyclic group of order, $g \in Z_p^*$ is a generator about cyclic group Z_p^*, namely $g^N mod \quad P$=1,for $\forall y \in Z_p^*$, and x=$\log_g y$, this is the solution problem of discrete logarithm in prime finite field Z_p^*.

In addition, from the reference [9] we can see that formula (1) is true.

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

The following are some of the concepts and properties used later in the article.

Definition 1 If all prime factor products of positive integer m do not exceed positive integer B, then m is said to be B smooth, that is,m is a smooth number and B is a smooth bound.

Modular operations have the following properties [20].

Proposition 1 If $k \neq 0$, and $a \equiv b (mod \quad p)$, then $ka \equiv kb (mod \quad |k|p)$.

Proposition 2 $ca \equiv cb (mod \quad p) \Longleftrightarrow a \equiv b (mod \quad p/(c,p))$.

Proposition 3 If $a \equiv b (mod \quad p)$,$c \equiv d (mod \quad p)$, then $ac \equiv bd (mod \quad p)$,$a+c \equiv b+d (mod \quad p)$.

The bottom and top functions have the following characteristics [18] :

$$\lfloor x \rfloor = n \Leftrightarrow n \leq x < n+1 \tag{2}$$

$$\lfloor x \rfloor = n \Leftrightarrow x - 1 < n \leq x \tag{3}$$

$$\lceil x \rceil = n \Leftrightarrow n - 1 < x \leq n \tag{4}$$

$$\lceil x \rceil = n \Leftrightarrow x \leq n < x + 1 \tag{5}$$

For any real number, there is formula (6),

$$x = (mod \quad p) = x - y \lfloor \frac{x}{y} \rfloor, y \neq 0 \tag{6}$$

Lemma 1 [11] If p、r are primes, and $p > r$, then $\exists s \in [\lceil \frac{p}{r} \rceil, \lfloor \frac{2p}{r} \rfloor]$,make $p < sr < 2p$ true.

Lemma 2 [11] If p、r are primes, and $p > r$,$\exists s \in [\lceil \frac{p}{r} \rceil, \lfloor \frac{2p}{r} \rfloor]$,$t$=$sr (mod \quad p)$,then $t < r$.

2.2 IC algorithm description

The basic idea of the IC algorithm is: if the desires of positive integer y is a smooth number, direct decomposition y, can calculate y discrete logarithm, otherwise in cyclic group Z_p^* construct a called factor based on a collection of small primes, expressed as B =$\{p_1, p_2, \cdots, p_t\}$,set B is the set of initial t prime numbers. IC algorithm is divided into two stages, the first stage constructs and solves $t + d$ (d is an appropriate constant) linear equations, each equation of the linear equations is the congruent of a module p, and solves the discrete logarithm of each prime number in the smooth bound B; In the second stage, a number g^m is randomly selected and multiplied by y. If all the decomposed prime

factors fall in the decomposition basis, the discrete logarithm of y is calculated. Otherwise, a number is re-selected and so on is repeated until the discrete logarithm of y is calculated.

The specific process of the two stages is as follows: in the first stage, randomly select $\forall z \in Z_p^*$, calculate and decompose $yg^z(mod \quad p)$, if all the prime factors of the decomposition are in$B$, find the next equation until $t+d$linear equation is found. Solve a system of linear equations and find the discrete logarithm of all elements in the resolution basis.

in the second stage, randomly select $\forall k \in Z_p^*$, and calculate yg^k, trying to write yg^k as the product of elements in the set B, that is, formula (7).

$$yg^k = \prod_{i=1}^{t} p_i^{d_i}, d_i \geq 0 \tag{7}$$

If successful, equation (8) is the desired discrete logarithm

$$\log_g y = ((\sum_{i=1}^{t} \log_g p_i) - k)(mod \quad N) \tag{8}$$

Otherwise, repeat the second stage until yg^k is written as the product of the elements in the set B.

2.3 Overview of IICA

Set $y = \prod_{j=1}^{k} q_j^{w_j}$, for $\forall q_j \in Z_p^* (j = 1, 2, \cdots, k)$ are prime Numbers, and $q_1 < q_2 < \cdots < q_k$.$r = [\frac{p}{q_j}]$ said Integer, the direction of r integer is based on the principle of whether it can be decomposed into the product of prime powers.

Definition 2 If $q_j > p_t$(p_t is the largest element set B), the symbol "$\Longrightarrow$" said a congruence transformation, called operation $y(mod \quad p) \Longrightarrow \frac{(r*y)(mod \quad p)}{r}(mod \quad p)$ is a congruence transformation of y.

According to definition 2, if r is a smooth number, the transformation can be repeated until all non-smooth factors are transformed into the smooth bound B, ending the transformation. The idea of the IICA is: the first step is the same as the ICA, that is, to find the discrete logarithm of all prime factors in the smooth bound B; the second step is to carry out a finite transformation according to definition 2. If all prime factors fall into the smooth bound B after a finite transformation, the discrete logarithm of y is found. For the specific algorithm of the above process, see algorithm 1.

algorithm 1 IICA algorithm

Input: g, p,B, y

Output: $\log_g y$

1 N=p-1,$N_s = 0$,$M_r = 0$.
2 If $y(mod \quad p) = -1$ then $\log_g y = N/2$,goto 8
3 If $y(mod \quad p) = 1$ then $\log_g y = N$,goto 8
4 Decompose $y = \prod_{j=1}^{k} q_j^{w_j}$,and save $q_1 < q_2 < \cdots < q_t$
5 If $q_k < p_t$ then $\log_g y = (w_1 \log_g q_1 + w_2 \log_g q_2 + \cdots + w_k \log_g q_k)(mod \quad N)$,goto 8
6 If after a congruence transformation of q_k, the largest prime factor is greater than q_k,indicating that the congruence transformation does not exist at this time, go to 9.Otherwise the congruence transformation is performed on the primes in p_k greater than p_t ,until all the primes fall into the set B.
7 compute $log_g y = (\log_g M_r - \log_g N_s)(mod \quad N)$,where M_r(the numerator in the congruence transformation) is the product of the prime powers of the elements in the set B, and N_s(the denominator in the congruence transformation) is the product of the prime powers of the elements in the set B with the transformation of M_r.
8 Output $\log_g y$, the algorithm ends.
9 The algorithm ends.

3 Algorithm optimization

The references [10] and [11] use the method of rounding up, that is, the result of each transformation is a positive integer. This method limits the solution of IC discrete logarithm. In addition, according to definition 2, if r is not a smooth number, the transformation terminates immediately. However, it is found that for some transformations, even if r is not a smooth number, it is still possible to solve, such as r with the same maximum prime factor as y. Therefore, the algorithm is optimized by taking advantage of the invariance of the multiplication of two modules in Proposition 3.

make $M_r = \prod_{i=1}^{t} p_i^{d_i} \geq 1$, $d_i \geq 0$, $N_s = \prod_{j=1}^{t} p_i^{c_j} \geq 1$, $c_j \geq 0$.

Theorem 1 If $N_s = \lceil \frac{p}{y} \rceil$ and $M_r = yN_s(mod \quad p)$ are smooth, then $\log_g y = (\log_g M_r - \log_g N_s)(mod \quad N)$.

Proof: Because $y = \frac{y\lceil p/y \rceil}{\lceil p/y \rceil}$,$yN_s - p > 0$,so $M_r > 0$,$N_s > 0$.Again $N_s = \lceil \frac{p}{y} \rceil$ and $M_r = yN_s(mod \quad p)$ are smooth.Thus $yN_s \equiv M_r(mod \quad p)$.For For identity $yN_s \equiv M_r(mod \quad p)$, take the log of both sides modulo N,acquirability $\log_g y = (\log_g M_r - \log_g N_s)(mod \quad N)$. □

Theorem 2 Supposing $M_r > 0$ is a smooth number.If $y \equiv \frac{M_r}{y}(mod \quad p)$, then $\log_g y = 2^{-1}(\log_g M_r - \log_g N_s)(mod \quad N/2)$.

Proof: Because $M_r > 0$ is a smooth number,and $y \equiv \frac{M_r}{y}(mod \quad p)$.According to property 1,$y^2 \equiv M_r(mod \quad yp)$.Again taking the logarithm of the mod N of both sides of the congruence $y^2 \equiv M_r(mod \quad yp)$,get $\log_g y = 2^{-1}(\log_g M_r - \log_g N_s)(mod \quad N/2)$. □

Theorem 3 Supposing $N_s = \lfloor p/y \rfloor$ and $M_r = |yN_s - p|$ are smooth.If $y \equiv \frac{M_r}{N_s}(mod \quad p)$, then $\log_g y = (\log_g M_r - \log_g N_s)(mod \quad N/2)$.

Proof: Because $N_s = \lfloor p/y \rfloor$ and $M_r = |yN_s - p|$ are smooth.$yN_s - p < 0$,so $y \equiv \frac{yN_s - P}{N_s}(mod \quad p) < 0$,According to proposition 1,$N_s y \equiv (yN_s - p)(mod \quad N - sp) < 0$.Again using proposition 3,taking the logarithm of the mod N of both sides of the congruence $N_s y \equiv (yN_s - p)(mod \quad N - sp) < 0$,get $2\log_g y = 2(\log_g M_r - \log_g N_s)(mod \quad N)$.Using proposition 2,get $\log_g y = (\log_g M_r - \log_g N_s)(mod \quad N/2)$. □

Theorem 4 Supposing $N_s = \lfloor p/y \rfloor$ and $M_r = |yN_s - p|$ are smooth.If $y \equiv \frac{yN_s - p}{yN_s} (mod \quad p)$, then $\log_g y = 2^{-1}(\log_g M_r - \log_g N_s)(mod \quad N/4)$.

Proof: Because $N_s = \lfloor p/y \rfloor$ and $M_r = |yN_s - p|$ are smooth.$yN_s - p < 0$,so $y \equiv \frac{yN_s - P}{N_s}(mod \quad p) < 0$,According to proposition 1,$N_s y^2 \equiv (yN_s - p)(mod \quad N - syp) < 0$.Again using proposition 3,taking the logarithm of the mod N of both sides of the congruence $N_s y^2 \equiv (yN_s - p)(mod \quad N - syp) < 0$,get $4\log_g y = 2(\log_g M_r - \log_g N_s)(mod \quad N)$.Using proposition 2,get $\log_g y = 2^{-1}(\log_g M_r - \log_g N_s)(mod \quad N/4)$. □

Lemma 3 [11] If $p/t < x < p$ is a prime number, then the prime factor of x must decrease after a congruence transformation.

According to Lemma 2, such a transformation exists, but according to Lemma 3, every time the multiplier is a smooth transformation and must exist, and theorems 3 and 4 show that this phenomenon occurs. When y goes through a congruence transformation, the prime values greater than p_t in its decomposition prime factors will decrease, and after finite transformations, all the prime values greater than p_t in the decomposition of y's prime factors will fall in the set B, then the decomposition is successful. When y is in the process of congruence transformation, the prime factor in its resolution basis does not decrease, then the transformation fails.

In fact, it can be solved in the first stage of ICA using the method introduced in Theorems 1-4. The specific method is described in Algorithm 2.

algorithm 2 IC optimization algorithm

Input: g, p,B, y
Output: $\log_g y$
1 N=p-1.
2 Choose a integer $k_1 \in Z_p$.
3 Decompose $y_1 = g^{k_1}(mod \quad p)$ using theorems 1-4.If y_1 is smooth,then save $k_1 \equiv \log_g y_1(mod \quad N)$.
4 Repeat steps 2 and 3 until you have t expressions as shown in step 3.
5 Solve t linear equations to obtain the discrete logarithm of all elements in the resolution basis.
6 The algorithm ends.

Combining theorems 1-4, algorithm 1 is optimized to algorithm 3.

algorithm 3 IC optimization algorithm

Input: g, p,B, y
Output: $\log_g y$
1 N=p-1,$N_s = 0, M_r = 0$.
2 If $y(mod \quad p) = -1$ then $\log_g y = N/2$,goto 8
3 If $y(mod \quad p) = 1$ then $\log_g y = N$,goto 8
4 Decompose $y = \prod_{j=1}^{k} q_j^{w_j}$,and save $q_1 < q_2 < \cdots < q_t$
5 If $q_k < p_t$ then $\log_g y = (w_1 \log_g q_1 + w_2 \log_g q_2 + \cdots + w_k \log_g q_k)(mod \quad N)$,goto 8
6 If $q_j > p_t (j = 1, 2, ..., k)$,then transform all q_j using Theorem 1-4.If the prime factor after one congruence transformation is still greater than p_j, indicating that the congruence transformation does not exist,the conversion fails,goto 9.Otherwise the congruence transformation is performed on the primes in p_j greater than p_t ,until all the primes fall into the set B.
7 compute $log_g y = (\log_g M_r - \log_g N_s)(mod \quad N)$,where M_r(the numerator in the congruence transformation) is the product of the prime powers of the elements in the set B, and N_s(the denominator in the congruence transformation) is the product of the prime powers of the elements in the set B with the transformation of M_r.
8 Output $\log_g y$, the algorithm ends.
9 The algorithm ends.

4 Algorithm analysis

It is well known that for a prime number greater than the smooth bound B, it must be a non-smooth number. The number of non-smooth numbers can be roughly estimated according to Chebyshev's Theorem, see Lemma 4.

Lemma 4 (Chebyshev's Theorem) $\pi(p) = \Theta(x/\ln x)$.

Assuming that $\Psi(x,p)$ represents the number of smooth numbers over the finite field F_p, the number of smooth numbers can be found by Lemma 5.

Lemma 5 [19] Supposing that R is a function,for some $\epsilon > 0$,satisfy $R = \Omega((\ln p)^{1+\epsilon})$, and when $p \to \infty, \mu = \ln p/\ln B \to \infty$, $\pi(R,p) = p \cdot \exp\{[-1+o(1)]\mu \ln \mu\}$.

The reference [13] indicates $\pi(y,p) \sim \frac{1}{\sqrt{2\pi t}}[\frac{e\ln p}{t\ln t}]^t$,Where t is the rank of the largest prime in R, that is, the tth prime.If $t \approx \exp(\sqrt{\ln p})$, then $\pi(y,p) > p \cdot \exp(-\frac{1}{2}\sqrt{\ln p})$. From this point, if $t \approx \exp(\sqrt{\ln p})$, then is the number of smooth probability greater than $\frac{1}{\exp(\frac{1}{2}\sqrt{\ln p})}$, this is the IC algorithm is used to decompose the probability of success.By reference [21], The relation between smooth bound B and p is $B = \exp\{(1/2+o(1))\sqrt{\ln p \ln\ln p}\}$.

From lemma 5 we know that the probability of ICA success is $\pi(R,p)/p$. While IICA increases the number of successfully solved non-smooth numbers obtained by rounding up, lemma 3 shows that such a transformation exists, so the probability of success of IICA is greater than $\pi(R,p)/p$. Algorithm 3 complements the case where the discrete logarithm is successfully solved by rounding down to obtain a non-smooth number, and according to theorems 2-4, such a transformation also exists. Therefore, the probability of successfully solving the discrete logarithm of the optimized IICA is nearly 2 times higher than that of IICA.

5 Case verification

In order to investigate the implementation process of different sizes of finite fields, two examples are given for verification. Although the size of the two finite fields given below is far from the actual size, or is too small, it is sufficient to verify the correctness and effectiveness of the algorithm.

5.1 Validation of small finite fields

Let $p = 229$,$g = 6$ is a genertor on order 228 in Z_p^*,$B = \{2,3,5\}$,computing discrete logarithm $y = 227$.

In the first stage, the discrete logarithm of the elements in basis B is calculated by algorithm 1. Select an integer 10 at random and compute $g^{10}(mod \quad p) = 100 = 2^2 * 5^2$. Nextly,select another integer 42 at random and compute $g^{42}(mod \quad p) = 2^2$, so we get $\log_g 2 = 21$, $\log_g 5 = 98$. In turn,randomly pick another integer 29, calculate $g^{29}(mod \quad p) = 31 = \frac{19}{2^3} = -\frac{1}{2^5 * 3}$.We can see from theorem 3 that $g^{2*29}(mod \quad p) \equiv \frac{1}{(2^5*3)^2}(mod \quad p)$. , and then $g^{2*29}(mod \quad p) \equiv \frac{1}{(2^5*3)^2}(mod \quad p)$ take the pairs on both sides and combine modules N, can be known $29 \equiv -(5\log_g 2 + \log_g 3)(mod \quad p)$,thus $\log_g 3 \equiv 94(mod \quad 114)$.So $\log_g 3 = 208$.The dispersion pairs of all the elements of this solution are obtained, i.e. $\log_g 2 = 21$,$\log_g 3 = 208$,$\log_g 5 = 98$.

In the second stage, logg y is calculated as follows:

The first step is to decompose $y = 227$, which can no longer be decomposed since 227 is prime.

The second step, as a result of $r = [\frac{229}{227}] = 2$, so a transformation of y is $\frac{ry(mod \quad p)}{r}(mod \quad p) = 225 = 3^2 \cdot 5^2$. Since the transformed prime factorization already falls into the set $\{2,3,5\}$.

The third step, calculate $\log_g 227 = (2\log_g 3 + 2\log_g 5 - \log g 2)(mod \quad p) = 135$.

Since the equation $6^{135}(mod \quad p) = 227$ holds,the algorithm is correct. 227 is not smooth in IC, but it is smooth in IICA from the example calculation, which verifies the correctness of the algorithm

analysis in Section 4.

The congruence transformation of $y = 227$ is solved correctly once.

Assuming $v = 7$,the IICA algorithm is used to solve the problem.Because $7(mod\ \ p) = \frac{7*3*11}{3*11}(mod\ \ p) = \frac{2}{3*11}(mod\ \ p)$, number 11 of the denominator is not smooth, the algorithm fails. But if use IC algorithm optimization, there are $7(mod\ \ p) = \frac{7*32}{2^5}(mod\ \ p) = \frac{-5}{2^5}(mod\ \ p)$,at this point mode by itself, according to the Theorem 3 can get $7^2 \equiv (\frac{-5}{2^5} mod\ \ p)$, which has a $\log_g 7 = (\log_g 5 - 5\log_g 2)(mod\ \ 114)$,即$\log_g 7 = 107$. Since the equation $6^{107}(mod\ \ p) = 7$ holds,the algorithm is correct, which also verifies the correctness of the algorithm analysis in Section 4. This shows that the optimized improved algorithm has a higher success rate than the original improved algorithm.

It is proved that in this example, the discrete logarithm of all numbers can be solved by transformation, and the probability of success is 100%.

5.2 Validation of larger finite fields

Choose$p = 14087$,$g = 5$ is a generator on order 14086 in Z_p^*,$B = \{2, 3, 5, 7, 11, 13\}$,computing discrete logarithm $y = 4999$.

In the first stage, the discrete logarithm of the elements in basis B is calculated by algorithm 1. Select an integer 19 at random and compute $g^{19}(mod\ \ p) = 3000 = 2^3 * 3 * 5^3$. Nextly,select another integer 32 at random and compute $g^{32}(mod\ \ p) = 2^6 * 3^2$, so we get $\log_g 2 = 3028$, $\log_g 3 = 5018$. In turn,randomly pick another integer 10117, calculate $g^{10117}(mod\ \ p) = @^2 * 7 * 41 = \frac{2^5*3}{7^2}$.We can see that $\log_g 7 = 8542$.Similarly, by randomly selecting 11530 and 11793 and combining the known discrete logarithms of 2, 3, 5, and 7, we get $\log_g 11 = 5446$,$\log_g 13 = 4279$.The dispersion pairs of all the elements of this solution are obtained, i.e. $\log_g 2 = 2028$,$\log_g 3 = 5018$,$\log_g 5 = 1$,$\log_g 7 = 8542$,$\log_g 11 = 5446$,$\log_g 13 = 4729$.

In the second stage, logg y is calculated as follows:

Assuming discrete logarithm of elements in B is known by ICA,namely $\log_g 2 = 3028$,$\log_g 3 = 5018$, $\log_g 5 = 1$,$\log_g 7 = 8542$, $\log_g 11 = 5446$,$\log_g 13 = 4729$.$\log_g y$ is calculated as follows:

For $4999(mod\ \ p) \Rightarrow \frac{3*4999(mod\ \ p)}{3}(mod\ \ p) = \frac{910(mod\ \ p)}{3}(mod\ \ p) = \frac{2*5*7*13(mod\ \ p)}{3}(mod\ \ p)$,so $\log_g 4999 = (\log_g 2 + \log_g 5 + \log_g 7 + \log_g 13 - \log_g 3)(mod p) = 11282$.Since the equation $5^{11282}(mod p) = 4999$ holds,the algorithm is correct. 4999 is a non-smooth number in ICA, but from the example calculation, 4999 is smooth in IICA, which verifies the correctness of the algorithm analysis in Section 4.

Let's look at the solution for the discrete logarithm of y=14077.

$14077(mod\ \ p) = \frac{2*14077(mod\ \ p)}{2}(mod\ \ p) = \frac{14067(mod\ \ p)}{2}(mod\ \ p) \Rightarrow \cdots \Rightarrow \frac{3847}{2^{10}}(mod\ \ p) \Rightarrow \frac{11*1301}{11*2^{12}}(mod\ \ p) = \frac{7*2^5}{11*2^{12}}(mod\ \ p) = \frac{7}{11*2^7}(mod\ \ p)$.

Sequentially $\log_g 14077 = (\log_g 7 - 7\log_g 2 - \log_g 11)(mod\ \ 14086) = 10074$. Since the equation $5^{10074}(mod\ \ p) = 14077$ holds,the algorithm is correct, which verifies the correctness of the algorithm analysis in Section 4.

Another example of Theorem 2 is given. Suppose we compute the discrete logarithm of 19. The specific process is as follows:

$19(mod\ \ p) = \frac{-8}{3*13*19}(mod\ \ p) \Rightarrow 19^2(mod\ \ p) = \frac{-8)}{3*13}(mod\ \ p) \Rightarrow 19^4(mod\ \ p) = \frac{2^6}{3^2*13^2}(mod\ \ p)$.

Sequentially $4\log_g 19 = (6\log_g 2 - 2\log_g 3 - 2\log_g 13)(mod\ \ 14086) = 12760$. Since the equation $5^{3190}(mod\ \ p) = 19$ holds,the algorithm is correct, which verifies the correctness of Theorem 2.

Similarly, 53 and 19 have similar characteristics, and the specific process is as follows:

$53(mod\ \ p) = \frac{53*265}{265}(mod\ \ p) \Rightarrow \frac{-42}{5*53}(mod\ \ p) \Rightarrow 53^4(mod\ \ p) = \frac{42^2(mod\ \ p)}{5^2}(mod\ \ p)$.

Sequentially $4\log_g 53 = (2\log_g 2 + 2\log_g 3 + 2\log_g 7 - 2\log_g 5)(mod\ \ 14086) = 5002$,namely

$2\log_g 53 = 2501$, thus $\log_g 53 = \frac{2501}{2}(modp) = \frac{2501}{2}(mod \quad 7043) = 2501 * 3522(mod \quad p) = 4722$.

Since the equation $5^{4772}(mod \quad p) = 53$ holds, the algorithm is correct, which further verifies the correctness of Theorem 2.

Let's look at an example of using IICA and optimizing IICA after a round of transformation failure. Suppose we compute the discrete logarithm of 17.

First look at IICA's solution process, because $17(mod \quad p) = \frac{17*829}{829}(mod \quad p)$, 829 is a prime number greater than 17, is a non-smooth number, so IICA's solution for this round fails.

Next look at the optimization IC algorithm solution process, because $17(mod \quad p) = \frac{17*828}{828}(mod \quad p) \Rightarrow \frac{-11}{2^2*3^2*23}(mod \quad p)$, 23 is a prime number greater than 17 and is a non-smooth number, the optimized IC algorithm fails this round.

This shows that the probability of solving the discrete logarithm of the optimized IC algorithm does not reach. In other words, the optimized IC algorithm is still a probabilistic algorithm, but it only improves the probability of success.

6 Closing remarks

Solving discrete logarithms over finite fields of prime numbers plays an important role in ensuring the security of cryptosystems based on finite fields of prime numbers, so solving discrete logarithms over finite fields of prime numbers is a hot topic in computer and mathematics circles. ICA is favored by many people because of its limited group structure and sub-exponential running time. However,ICA is a probabilistic algorithm with a lot of trial problems. IICA is an improved algorithm of ICA, which improves the probability of solving discrete logarithms. The optimized IICA algorithm extends the solution range of IICA, and the probability of solving discrete logarithms is higher than that of IICA. However, whether the IICA algorithm or the optimized IICA algorithm fails to establish the mathematical model of success probability analysis, whether the improved algorithm and the improved algorithm of optimization can be used in the attack of elliptic curve cryptosystem still needs further research.

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] ANDREAS E,PIERRICK G, EMMANUEL T.An (1/3) discrete logarithm algorithm for low degree curves[J]. J. Cryptology,2011,24(1):24-41.

[5] PADMAVATHY R,BHAGVATI C.Index calculus method based on smooth numbers of ±1 over [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] HOWELL J S. The Index Calculus Algorithm for Discrete Logarithms[D].Clemson University, 1998.

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

[9] HU J J, WANG W,LI H J. An improving algorithm of index calculus[J].Journal of Nanchang University(Engineering and Technology Edition), 2016, 38(3):286-289.

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

[11] Hu J J.Analysis and Optimization of Improved Index Calculus Algorithm. Mathematics and Computer Science, 2023,8(2):57-61.
[12] DONALD E K.The Art of Computer Programming(Vol.3)[M].New Jersey:Addison-Wesley,1973.
[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] THERIAULT N. Index calculus attack for hyperelliptic curves of small genus[C].International Conference on the Theory and Application of Cryptology and Information Security. Berlin, Heidelberg: Springer Berlin Heidelberg,2003,75-92.
[15] FAUGERE J C, PERRET L, PETIT C, et al. Improving the complexity of index calculus algorithms in elliptic curves over binary fields[C].Annual international conference on the theory and applications of cryptographic techniques. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012,27-44.
[16] JOUX A, VITSE V. Cover and Decomposition Index Calculus on Elliptic Curves Made Practical: Application to a Previously Unreachable Curve over[C].Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin, Heidelberg: Springer Berlin Heidelberg,2012,9-26.
[17] MUKHOPADHYAY M. Aspects of Index Calculus Algorithms for Discrete Logarithm and Class Group Computations[D]. Indian Statistical Institute, Kolkata, 2021.
[18] ZHANG M R,ZHANG F. Mathematics:AFoundation for Computer Science[M]. Second Edition.Beijing:Post and Telecom Press,2013.
[19] VICTOR S. A Computational Introduction to Number Theory and Algebra(BETA version 3)[M]. London:Cambridge University Press,2004.
[20] PAN C D, PAN C B. Elementary Number Theory (Third edition)[M]. Beijing:Peking University Press,2013.
[21] JOE B, PETER S.Algorithmic Number Theory: Lattices, Number Fields, Curves and Cryptography[M]. Cambridge University Press,2005.