Index Calculus Method Based on Smooth Numbers of ±1 over Z_p^*

R. Padmavathy[1] and Chakravarthy Bhagvati[2]

(Corresponding author: R. Padmavathy)

Department of Computer Science and Engineering, National Institute of Technology Warangal,India[1]

Department of Computer and Information Sciences University of Hyderabad, Hyderabad, Andhra Pradesh, India[2]

(Email: r_padma3@rediffmail.com, chakcs@uohyd.ernet.in)

(Received Aug. 24, 2010; revised and accepted Nov. 12, 2010)

Abstract

The paper presents a variant of ICM on integer field when the factors of the group are known and small. This is achieved through the properties of Smooth numbers of ±1 over Z_p^*. The ICM has two steps, such as a pre-computation and an individual logarithm computation. The pre-computation step is to compute the logarithms of a subset of a group and the individual logarithm step is to find the DLP using the pre-computed logarithms. The algorithm presented in the paper for ICM is a combination of Pohlig-Hellman, which is the popular attack on the groups of order with all small factors and the traditional ICM. In the present study we show the substantial performance improvement of ICM for the problems of size upto ≈ 150 bits on Pentium 4 machine. The analysis presented in the paper is considered as useful to recover ephemeral keys used in the cryptosystems like text book ElGamal and Chang and Chang three party password key exchange protocol to name a few. One way of recovering the ephemeral key is to solve the DLP. Since the ephemeral keys are dynamic and change for every session, once the discrete logarithms of a subset of a group is known, the DLP for the ephemeral key can be obtained by using the individual logarithm step. Therefore, the ephemeral keys are recovered by using the individual logarithm step proposed in the present study.

Keywords: Index calculus method, Pohlig-Hellman method, smooth numbers of ±1 *over* Z_p^*

1 Introduction

The Index Calculus Method (ICM) is the most effective method to solve the Discrete Logarithm Problem (DLP). Many public key cryptosystems are based on the intractability of DLP. The DLP defined over a prime field $\mathbb{Z}_p^*$ of random prime (p) is considered in the present study.

For a given prime p, a generator $g \in Z_p^*$ and an element $y \in Z_p^*$, the problem of finding x, in the range of $0 \leq x \leq p-2$, such that $g^x = y \ (mod\ p)$, is known as the DLP. The security assumption of cryptosystem based on DLP is, one way function : an attacker cannot recover x from g and g^x. Many cryptographic schemes rely on the assumption that the DLP is hard, to name a few, Diffie-Hellman key exchange [7], ElGamal public key cryptosystems [8] and the digital signature algorithm. The comparison between signatures are narrated in [28]. Some of the attacks on the DLP are discussed below. Apart from the exponential time algorithms the Pohlig-Hellman method is a popular attack [23], which reduces the DLP in a field to small subgroups and combines the results using Chinese Remainder Theorem. The DLP can be computed in the sub exponential time using the ICM. The ICM uses a fixed small set called the factor base B and tries to write the elements as a product of members of the factor base B [15]. The base consists of objects which are small and irreducible. In a prime field F_p, where we identify the field elements with integers in $0, 1, \cdots, p-1$, a factor base consists of all prime numbers less than some prescribed bound. In a field of characteristic 2, F_{2^n}, where we write field elements as polynomials of degree $< n$, a factor base consists of all irreducible polynomials of degree less than some prescribed bound. The algorithm has two steps:

- A pre-computation step, where the logarithms of log_g^b of all members of the factor base is obtained.
- A computation step, which tries enough $g^a y$ until the result factors over the factor base, thus providing the requested logarithm $log_g y$ [24, 25].

The pre-computation step itself has two phases:

1) First phase is to find linear relations relating the logarithms of the primes in the factor base.

2) Second phase is to solve these logarithms using techniques from linear algebra.

Coppersmith and Odlyzko [5] presented three versions of index calculus method. Later LaMacchia and Odlyzko [13] reported the implementation of two of these three versions namely, linear sieve and Gaussian integer methods. An implementation of cubic sieve method is reported by Abhijit Das and Veni Madhavan [6]. Even for the primitive approach one can obtain running time bound of the form $\exp((c+o(1))(\log p)^{\frac{1}{2}}(\log\log p)^{\frac{1}{2}})$ for some constant c [14]. Research is in progress in obtaining better values for c. For fields $GF(q)$ with $q = p^n$ for small p, Coppersmith's algorithm offered running time $\exp((C-o(1)))(\log q)^{\frac{1}{3}}(loglogq)^{\frac{2}{3}})$ for a positive constant C [18]. For some fields $GF(q)$ with $q = p^n$ in which both p and n grew even bounds of the first form were not available [17, 19]. Another variant of index calculus method is number sieve field and it has heuristic running time of the form $\exp((c+o(1))(\log p)^{\frac{1}{3}}(\log\log p)^{\frac{2}{3}})$ [9, 24, 26, 27].

The main concept associated with index calculus method is smoothness property of integers. The distribution of smooth integers are studied extensively [10, 11, 12]. Berstein presented a tight bounds on the distribution of smooth integers [2], a linear time algorithm to list y-smooth integer up to x and smooth part of integers [1] and several algorithms for number of integers free of large prime factors.

The present study extends the Smooth number definition on integers and presents a detailed study on the distribution of Smooth numbers of ±1 over Z_p^* on different types of primes, which are classified based on the order of the group. These characteristics lead to develop a new method for solving the DLP with the combination of Pohlig-Hellman and Index Calculus Method. It is known that Pohlig-Hellman can be applicable when factors of $p-1$ are small, in the present work the adaptability of ICM on the above group is studied. The main concept of index calculus method is, with known logarithms of a subset of a group, logarithm of any element can be computed. In the present work the pre-computation phase is designed and analyzed in such a way to built an efficient individual logarithm phase with the help of the properties of Smooth numbers of ±1 over Z_p^*. This leads to recover the ephemeral keys used Chang and Chang three party password key exchange protocol.

The rest of the paper is organized as follows, the following section reviews the Pohlig-Hellman and Index Calculus Methods. Section 3 describes the concept of Smooth integers of ±1 over Z_p^* and the analysis on the smoothness property of different types of primes. Section 4 presents the algorithms for index calculus method to solve the DLP. Section 5 discusses the results and the concluding remarks are made in Section 6.

2 General Algorithm for Pohlig Hellman and Index Calculus Method

2.1 Pohlig-Hellman Algorithm

This is an algorithm introduced by Pohlig-Hellman. If the order of the group is known along with the complete factorization and the factors are relatively small then this attack is possible.
Let $p-1 = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$ and g be the generator of order $p-1$. Then

$$g^x = y \bmod p.$$

The above equation can be reduced into

$$\alpha^x = \beta \bmod p,$$

where α is $g^{(\frac{p-1}{p_1^{e1}})}, g^{(\frac{p-1}{p_2^{e2}})}, \cdots, g^{(\frac{p-1}{p_k^{e_k}})}$ and β is $y^{(\frac{p-1}{p_1^{e1}})}, y^{(\frac{p-1}{p_2^{e2}})}, \cdots, y^{(\frac{p-1}{p_k^{e_k}})}$.

The logarithm in the small subgroups are solved by using one of the popular square root algorithms. Later the Chinese Remainder Method is used to combine the results $x_i \bmod p_i^{e_i}$ to retrieve $x \bmod p$.

2.2 Index Calculus Method

The Index Calculus Methods are the most prominent collection of algorithms that have successfully used additional knowledge of the underlying groups to provide sub exponential algorithms. The basic idea, which goes back to Kraitchik [15] is that if

$$\prod_{i-1}^{m} x_i = \prod_{j-1}^{n} y_i$$

for some elements of $GF(q)^*$, then

$$\sum_{i-1}^{m} \log_g x_i \equiv \sum_{j-1}^{n} \log_g y_j \bmod (q-1).$$

If we obtain many equations of the above form, and they do not involve too many x_i and y_i, then the system can be solved.

The algorithm has two steps:

- A pre-computation step, where the logarithms of $\log_g b$ of all members of the factor base is obtained, where g is the generator and b is the element in the factor base.
- A computation step, which tries enough $g^a y$ until the result factors over the factor base, thus providing the requested logarithm $\log_g y$, where y is the element for which the logarithm to be computed [25].

The pre-computation step itself has two phases:

- First phase is to find the linear relations relating the logarithms of the primes in the factor base.
- Second phase is to solve this linear system using techniques from linear algebra.

The general algorithm for traditional ICM is described below [3].

2.2.1 General Algorithm for ICM

INPUT a generator g of a cyclic group G of order n i.e., $p-1$ and an element y.
OUTPUT $\log_g^y$.

- pre-computation step:
  - Select a factor base S={ $p_1, p_2, ...p_t$ }, which belongs to G such that a significant portion of elements of G can be efficiently expressed as a products of elements from S.
  - Find a linear system using the procedure as given below:
    1) Select a random integer k, such that $0 \leq k \leq n-1$ and compute g^k.
    2) Try to write g^k as a product of elements in S as

       $$g^k = \prod_{i=1}^{t} p_i^{c_i}, c_i > 0, \quad \text{for any } k.$$

       Then,

       $$k \equiv \sum_{i=1}^{t} c_i \log_g p_i.$$

    3) Repeat the above steps to get the value of $t+c$ equations.
  - The linear system is reduced into smaller size using structured Gaussian method. This step is optional one and used when a large system is generated in the previous step.
  - Solve this linear system to obtain $\log_g p_i$.
- Computation step:
  - Compute $\log_g y$:
    1) Select a random integer, k, $(0 \leq k \leq n-1)$ and compute yg^k.
    2) Try to write yg^k as a product of elements in S:

       $$yg^k = \prod_{i=1}^{t} p_i^{d_i}, \quad \text{for any } k.$$

       Then,

       $$\log_g y = (\sum_{i=1}^{t} d_i \log_g p_i - k) \bmod n.$$

3 Properties of Smooth Numbers of ±1 over Z_p^*

Definition 1. *Smooth number.*
An integer is called as Y smooth, if it has no prime divisors larger than some bound Y. For example 2,3,4,6,8,9 are 3 smooth numbers.

Definition 2. *Let R_1 is the relation defined on field elements of Z_p^* such that for any $a \in Z_p^*$, $b \in Z_p^*$ $ab \equiv 1$ mod p. Let R_2 be the relation defined on Z_p^* element such that for any $a \in Z_p^*$, $b \in Z_p^*$ $ab \equiv -1$ mod p.*

Definition 3. *A quadruple is defined on Z_p^* field elements based on the relations R_1 and R_2.*

Example 1. *Let s be the set of quadruple $\{a_1, a_2, a_3, a_4\}$ such that $a_1R_1a_2$ $a_3R_1a_4a_1R_2a_3a_2R_1a_4$.*

Definition 4. *Smooth number of ±1 over Z_p^*:*
Let Z_p^* is a multiplication group of prime field and B be a factor base $\in Z_p^*$. Then a quadruple s is B smooth on ±1 over Z_p^*, if it satisfies the following relations. Let $a \in B$ and $b, e, f \in Z_p^*$ then $ab \equiv 1$, $ef \equiv 1$, $ae \equiv -1$, $bf \equiv -1$. S =$\{s|s$ is B smooth on ±1 over Z_p^* $\}$ is denoted as a Smooth number set on ±1 over Z_p^*.

Theorem 1. *The field elements of Z_p^* forms a set of quadruples of disjoint sets.*

Proof. Let $x, y, z \in Z_p^*$ and $x \neq \pm 1$ and $xy \equiv 1; y \equiv x^{'}$; $xz \equiv -1; z \equiv -y; y.z \equiv -1; z \equiv -x; -x.z \equiv 1; z \equiv -y$. If $x = \pm 1; 1.1 \equiv 1, -1. -1 \equiv 1, 1. -1 \equiv -1$. Since $x^{'} \neq -x^{'}, x \neq -x, y^{'} \neq -y^{'}, y \neq -y$ and $x \neq y$ except for ±1 over Z_p^*. Hence $xR_1y; xR_2-y$, $yR_1x; yR_2-x$ and $-xR_1-y$. □

Proposition 1. *Let $x \in Zp*$ and $y \in Zp*$. If $x \neq y$ and xR_1y then $xR_2-y, yR_2-x, -xR_1-y$.*

Proof. Follows the above proof. □

Proposition 2. *Let $x \in Z_p^*$ and $y \in Z_p^*$. If x=y and $xy \equiv 1 modp$, then xR_1xxR_2-x.*

Proof. Follows the above proof. □

Next we require a function from the quadruple set in $Z_p^*/\sim$ to some set of representatives R. Since the present study is on the prime field, a suitable function for our purpose is the elements generated from the generator g, each one represents one quadruple and a map, say ϕ on $Z_p^*/\sim$. Let us denote $[g]$ as a quadruple of the generator and the experiments show that the map

$$\phi : [g] \rightarrow Z_p^*/\sim$$

can be defined by

$$g^i \rightarrow R,$$

where R is a representative of a quadruple and i is in the range $1 \leq i \leq \lceil (p-1-2)/4 \rceil$.
The following section describes the characteristics of smooth numbers ±1 over Z_p^*.

3.1 Distribution of Smooth Numbers ±1 over Z_p^* on Different Types of Primes

The distribution of smooth numbers ±1 over Z_p^* is studied on different types of primes. The primes are classified based on the factors of $p-1$ as described below.

Type 1: The prime p with $p-1=2q$,where q is a prime.

Type 2: The prime p with $p-1=2q_1q_2$,where q_1 and q_2 are prime.

Type 3: The prime p with $p-1=2q^n$, where q is a prime.

Type 4: The prime p with $p-1=2^nq$, where q is a prime.

Type 5: The prime p with $p-1=2^mq^n$,where q is a prime.

Theorem 2. *Let p be a prime of the form $2q+1$. Let S be the smooth number set over Z_p^*. Let Q be a quadruple $\in S$. If one pair of elements in Q consists of generators then the other pair contains non generators of order q. If one pair is 1 the other pair is -1.*

Proof. Let Q be the quadruple $((x,y)(-x,-y))$ with relation R_1 R_2 such that $xR_1y, -xR_1-y, xR_2-y, yR_2-x$.

1) Let x be a non generator:

$$\begin{aligned} x^q &\equiv 1 \mod p; \\ -x^q &\equiv -1. \end{aligned}$$

This shows $-x$ is generator. Let y be a generator.

$$\begin{aligned} y^q &\equiv 1 \mod p; \\ -y^q &\equiv -1. \end{aligned}$$

where, $-y$ is a non generator.

2) Let x be an identity element, the quadruple is $((1,1)\ (-1,-1))$.

□

Corollary 1. *Let p be a prime of the form $2q_1q_2+1$ and S be the smooth number set over Z_p^*. Let Q be a quadruple $\in S$. Then the pairs in Q have the following characteristics:*

1) If one pair of elements in Q consists of generators then the other pair contains non generators of order q_1q_2.

2) If one pair of elements in Q is of order q_1 or q_2 then the other pair contains non generators of order $2q$ or $2q_2$.

3) If one pair is 1 another pair is -1.

This can be easily proven from 1 and 2 of Theorem 2.

Corollary 2. *Let p be a prime of the form $2q^n+1$. Let S be the smooth number set over Z_p^*. Let Q be a quadruple $\in S$.*

1) If one pair of elements in Q is of order q^i, where i varies from 1 to $n-1$ then the other pair is of order $2q^i$.

2) If one pair of elements in Q is of generators then the other pair is of order q^n.

3) If one pair is 1, another pair is -1.

This also can be proven from 1 and 2 of Theorem 2.

Theorem 3. *The prime p of the form 2^nq+1. Let S be the smooth number set over Z_p^*. Let Q be a quadruple $\in S$.*

1) If one pair of elements in Q is of order q then the other pair is non generators of order $2q$.

2) If one pair of elements in Q is of order 2^iq where i varies from 2 to n then the other pair is also of order 2^iq.

3) If one pair is of order 2^i, where i varies from 1 to n then another pair is also of order 2^i.

4) If one pair is 1, another pair is -1.

Proof.

1) This follows the Proof 1 of Theorem 2.

2) Let x be the element of order 2^iq where i varies from 2 to n.

$$\begin{aligned} x^{\frac{2^iq}{2}} &\equiv x^{2^{i-1}q} \equiv -1 \mod p; \\ -x^{\frac{2^iq}{2}} &\equiv -1. \end{aligned}$$

This shows $-x$ is an element of order 2^iq. Let y be the element of order 2^iq where i varies from 2 to n:

$$\begin{aligned} ry^{\frac{2^iq}{2}} &\equiv x^{2^{i-1}q} \equiv -1 \mod p; \\ -y^{\frac{2^iq}{2}} &\equiv -1. \end{aligned}$$

This shows $-y$ is an element of order 2^iq.

3) Let x be the element of order 2^i where i varies from 1 to n:

$$\begin{aligned} x^{\frac{2^i}{2}} &\equiv x^{2^{i-1}} \equiv -1 \mod p; \\ -x^{\frac{2^i}{2}} &\equiv -1. \end{aligned}$$

This shows $-x$ is an element of order 2^i. Let y be the element of order 2^i where i varies from 1 to n:

$$\begin{aligned} y^{\frac{2^i}{2}} &\equiv x^{2^{i-1}} \equiv -1 \mod p; \\ -y^{\frac{2^i}{2}} &\equiv -1. \end{aligned}$$

This shows $-y$ is an element of order 2^i.

4) If one pair is 1, another pair is -1.

□

Corollary 3. *Let p be a prime of the form $2^n q_1^m + 1$; S be the smooth number set over Z_p^*; Q be a quadruple $\in S$:*

1) *If one pair of elements in Q is of order 2^i, where i varies from 1 to n then the other pair is also of order 2^i.*

2) *If one pair of elements in Q is of order q^i, where i varies from 1 to m then the other pair is the order $2q^i$.*

3) *If one pair of elements in Q is of order $2^i q^k$, where i varies from 2 to n and k varies from 1 to m. then the other pair is the order $2^i q^k$.*

4) *If one pair is 1, another pair is -1.*

Proof.

1) This is easily proven from 3 of Theorem 3.

2) This follows from 1 of Theorem 2.

3) This follows from 2 of Theorem 3.

□

3.2 Results and Discussion

From the above theorems, it is observed that, in the first type of primes, the set of quadruples are formed by the pair of generators and non generators. In the second types of primes, when one pair of the quadruple is of order q_1 or q_2 then the another pair is of order $2q_1$ or $2q_2$. Similarly, when one pair of the quadruple is a generator, then the another pair is non generator of order $2q_1q_2$. In the third type of problems the quadruples are formed by the pairs of generator and the elements of order q^n or elements of order $2q^i$ and the elements of order q^i, where i varies from 1 to $n-1$. The fourth kind of primes exhibit different characteristics, such as the classes are pairs of either generators or non generators. The non generators of order $2^i q$ forms quadruple with non generators of order $2^i q$, where i varies from 2 to $n-1$. The other choices are the combination of elements of order q with $2q$ and the pairs of elements of order 2^i, where i varies from 2 to n. The final type problems are combination of all the above. If one pair of elements is (2^i),where i varies from 1 to n then the another pair is of order 2^i. If one pair of elements is (q^i),where i varies from 1 to m then the another pair is of order $2q^i$. If one pair of elements is $(2^i q^k)$, where i varies from 2 to n and k varies from 1 to m then the another pair is of order $2^i q^k$.

The main conclusion is the quadruples exhibit different characteristics on different types of primes. Through our experimental results we found that the quadruples can be mapped using ϕ and R from the generators of the group as well as the generators of subgroups. A class $((a,b)(e,f))$ forms the relations as $\log a + \log b \equiv 0$ or $\log e + \log f \equiv 0$ and $\log a + \log e \equiv \log -1$ or $\log b + \log f \equiv \log -1$. Once the logarithm of any one of the above elements is known, the logarithms of other elements can be solved easily, since the logarithm of ±1 is known. At the same time to solve a set of, say m, quadruples with $4m$ elements, logarithm of m unknowns to be solved. This principle is used in the following methods to solve the DLP.

In the present study two ways of performing precomputation phase and three ways of computing individual logarithm phase is studied. In the finite field Z_p^* the pre-computation is to compute the logarithms of first-t primes, which are treated as m unknowns as discussed above. In the first type of pre-computation phase, the logarithms of first-t primes are computed using Pohlig-Hellman method and the individual logarithm phase is performed either using Pohlig-Hellman or the general third phase of index calculus method. In the second type, the logarithms of elements of each subgroups are stored in a list using the distribution of smooth numbers of ±1 over Z_p^* and the individual logarithm is only the Chinese Remainder Theorem phase. The following Table 1 presents the details of methods studied in the present work.

The above analysis is useful in obtaining the value of ephemeral key k used in the cryptosystem like text book El Gamal for every communication between Bob and Alice. Once the pre-computation phase is completed. The value of k can be retrieved easily using above mentioned methods. Since g and y are public, c_1 and c_2 are known and the value of k is computed using the individual logarithm phase, the message can be obtained easily.

The following algorithms presents the two types of precomputation phase. The second method is designed with the help of distribution of smooth number of ±1 over Z_p^*. Since the quadruples are disjoint sets, the logarithms of elements of subgroups are computed by forming the quadruples of subgroup elements. These logarithms are stored in a list. The number of group operation needed to compute the quadruples for each subgroup is relatively less, since from the properties of smooth numbers of ±1 over Z_p^*, it is observed that the quadruples can be formed by the generator.

4 Algorithms for ICM

In this section the above discussed methods for ICM are addressed. The Algorithm 1 presents the Method-1 and Algorithm 2 addresses the Method-2. Algorithm 1 is a naive approach to find the logarithms of first-t primes using Pohlig-Hellman method. The Steps 1 to 3 form the quadruples from the primes in the factor base. Steps 4 to 6 form the relations from the quadruples. Steps 7 to 14 are for finding the logarithms of primes in the factor base using Pohlig-Hellman method. Finally step 15 is to solve the logarithms of elements in the relations by using the logarithms of primes in the factor base. The Algorithm 2 takes the advantage of the distribution of smooth num-

Table 1: Methods for index calculus method

Methods	Pre-computation phase	Individual logarithm phase		
		General third phase	Pohlig-Hellman method	CRT
Method-1	Pohlig-Hellman	✓	✓	×
Method-2	Using smooth number ±1 over Z_p^*	×	×	✓

bers over Z_p^*. The logarithms of elements of subgroups are computed by using the classes formed from the generator of the subgroups. This is achieved through the mapping function ϕ and the representative R. Steps 1 to 7 are to find the number of iteration needed to form the quadruples for each subgroups. This is based on the order of the subgroups. The number of iterations, say A, is $\frac{p_i}{4}$, when the subgroup is of order 2^n and $\frac{p_i}{2}$ for other cases, where p_i is the order of the subgroup. Step 9 is to generate the subgroup element from the generator of subgroup. Steps 10 to 13 are to form the quadruples and to find the logarithms of elements in the quadruples. Final Step 13 is to store them in a list. In the first method, the individual logarithm is computed by either the general individual logarithm step of ICM, since the logarithms of first-t primes along with the classes are known or by using a simple Pohlig-Hellman method. The individual logarithm step of second method is a simple Chinese Remainder method (CRT).

Algorithm 1 To find the logarithm of first t primes when factors of $p-1$ are small

INPUT: Problem of size p. FB factor base consist of first t primes. Factors of $p-1$.
OUTPUT: Logarithm of first t primes and quadruples.

```
 1: for every $e$ of $FB$ do
 2:    Find the quadruple $((e,b)(c,d))$ where $e \times b \equiv 1 \ mod\ p$; $c \times d \equiv 1 \ mod\ p$; $e \times c \equiv -1 \ mod\ p$; $b \times d \equiv -1 \ mod\ p$
 3: end for
 4: for every pair $(a,b)$ in the quadruple do
 5:    relation is formed as $\log a + \log b \equiv 0$ or $\log a + \log b \equiv \log -1$
 6: end for
 7: for each element in $FB$ do
 8:    for  each subgroup $p_i$ do
 9:       Find the generator $g_i$ of order $p_i$ as $g^{\frac{p-1}{p_i}}$
10:       Assign $e_i$ as $e^{\frac{p-1}{p_i}}$
11:       Find the logarithm of $e_i$ mod $p_i$ using Pollard-Rho or Shanks method
12:    end for
13:    Find logarithm of $e$ mod $p$ using Chinese Remainder Theorem
14: end for
15: Solve the relations of quadruples using the logarithm of unknowns of $FB$
```

The individual logarithm step of ICM in Method-1 de-

Algorithm 2 To find logarithm of all small subgroup elements in the order of subgroup using the distinct set of quadruples formed with relation ±1 over Z_p^*

INPUT: Problem of size p and factors of $p-1$.
OUTPUT: Logarithm of all small subgroup elements in the order of subgroup.

```
 1: for every subgroup of order $p_i$ do
 2:    Assign $G = g^{p-1/p_i}$
 3:    if order of the subgroup is $2^n$ then
 4:       Assign $A = p_i/4$
 5:    else
 6:       Assign $A = p_i/2$
 7:    end if
 8:    for $i$ in $1..A$ do
 9:       Assign $H = G^i$
10:       Assign $\log H = i * \frac{p-1}{p_i}$
11:       Find the quadruple $((H,b)(c,d))$ where $H \times b \equiv 1 \ mod\ p$; $c \times d \equiv 1 \ mod\ p$; $H \times c \equiv -1 \ mod\ p$; $b \times d \equiv -1 \ mod\ p$.
12:       $\log b = -\log H; \log c = \log -1 - \log H; \log d = -\log c; \log d = \log -1 - \log b$
13:       Store them in the list
14:    end for
15: end for
```

pends on the number of elements in the factor base FB. To achieve maximum performance in the individual logarithm step a larger factor base is to be chosen. Since the size of the factor base is larger, the Method-1 needs substantially more time than Method-2. Method-1 performance is based on the size of the subgroups as well as the size of the factor base. On the other hand Method-2 depends only on the size of the subgroups. This leads to achieve a considerable performance increase in Method-2.

The usage of Pohlig-Hellman in the individual logarithm step of Method-1 is, by considering the y is $\notin FB$ and the factors of $p-1$ are small. Similarly, the general individual logarithm step of ICM is used, by considering y is $\notin FB$ and factors of $p-1$ are relatively large. Since the factors are large, the Pohlig-Hellman needs more time to solve the problem. Irrespective of the methods used in the individual logarithm step of Method-1, the CRT of Method-2 is more advantageous for a class of problems where the factors of $p-1$ are small.

4.1 Experimental Results

This section presents the results and analysis for the new method discussed above. The present problem is described as follows: First a data file is produced, which contains a list of tuples. A tuple is of the form (m, p, q) with the following properties:- m lies between 13 and 50 digits, p is a prime, q is the list of factors of $p-1$. Based on these properties the tuples are computed as follows:

- Choose k as 100.
- m is selected between 13 and 50 digits.
- A prime number is selected of size m and checked for the factors of $p-1$.
- The above step is repeated till a prime of required form is obtained.
- The factors are stored in the list q.
- Store the tuple in the data file.
- Repeat the above steps for k number of times.

Having built up the above file, the following algorithm is implemented:

- Read a tuple (m, p, q)
- Execute the Method-1 for the above tuple.
- Execute the Method-2 for the same tuple.
- Keep track of the computed run time.
- Repeat the above steps until all the tuples are calculated.

The Table 2 shows the difference in running time between Method-1 and Method-2.

4.2 Ephemeral Key Recovery

The ephemeral keys may be unique for each session or they may be reused for different sessions of a same party. For example, the ANSI X9.42 standard, which specifies several Diffie-Hellman protocols states that an ephemeral key is a "private or public key that is unique for each execution of a cryptographic schemes". Other protocols do not place any restrictions on the reuse of ephemeral keys [16]. The ephemeral keys, which are unique for each session is considered in the present study.

The DLP of ephemeral key can be solved efficiently, once the logarithms of a subset of group is known. The ephemeral keys are solved by using Pohlig-Hellman method in [21] and other efficient methods to recover ephemeral keys are discussed in [20, 22]. Assume the logarithms of a subset of a group i.e., the logarithms of subgroup elements, are computed by using *Algorithm-1* or *Algorithm-2*. Since the logarithms of subgroup elements are known, the DLP for the ephemeral key can be obtained by using any one of the methods in individual logarithm phase of ICM as mentioned in Table 1. This is possible due to the fact that the prime field and the generator are shared between the communicators before starting the sessions. The assumption is that the logarithms of subgroup elements are computed before starting the sessions. The DLP for ephemeral key is to be computed once the session get started. The Table 3 presents the running time of three methods as mentioned in the Table 1, such as general individual logarithm phase, Pohlig-Hellman and CRT.

The traditional individual logarithm step needs substantially more time due to the fact that the DLP is solved without considering the additional information such as the factors of $p-1$. The Pohlig-Hellman and the CRT work with the additional information regarding the factors of $p-1$ and the running time depend on the size of the factors of $p-1$. Similarly, the DLP for the ephemeral keys N_A and N_B used in Chang and Chang key exchange protocol is solved [4]. The ephemeral keys are solved in fraction of seconds.

5 Conclusion

In the present work the smoothness concept over integer is extended and smooth numbers of ±1 over Z_p^* is defined. The properties of smooth number of ±1 over Z_p^* are analyzed on different types of primes, which are classified based on the order of the group. They exhibit different characteristics that correspond to the elements in the factor base. These characteristics lead to develop a new method for pre-computation phase of index calculus method. The index calculus method is studied, when factors of p-1 are known and small. The analysis leads to perform the individual logarithm phase efficiently, which in turn helps in obtaining the ephemeral key, N_A and N_B used in the cryptosystem called as Chang and Chang password key exchange protocol.

Table 2: Running time of Method-1 and Method-2

Problem	Method-1 running time in sec	Method-2 running time in sec
10000000000000000087	151	81
10000000000000000763	35	9
10000000000000001347	155	220
10000000000000000000 0000008211	296	13
10000000000000000000 000000000037303	778	95
43241221044344476653 62908248086967822904858	1597	55
40500691568928903388 50331494359177651620 3	1393	88
10548813247704246266 317485054480132114947	808	31
22750475822981512251 147389834477659827887	861	29
31023376122247516706 110077047014990674390	987	26
16483888118310633603 1117884004666831843	315	6
29565696133579269116 146450939411987039	306	3
17566082229403199021 601313164759909980664 11	2516	202
11604511787937964282 8484543996278641093	1715	181
96834800300461810039 999000830418556963	2327	651
40687382369048479475 232114904989637283	1891	633
24774781676535030702 2689158418187059	1194	171
38174514779333277109 099448511590627	1141	160

Table 3: Running time to solve the DLP for ephemeral keys

Problem size in digits	CRT	General Individual logarithm step	Pohlig-Hellman method
21	75ms	451ms	90ms
22	8ms	605ms	65ms
23	56ms	5s	57ms
24	41ms	7s	19ms
25	.3ms	6s	13ms
26	22ms	41s	66ms
27	4ms	62s	25ms
28	.5ms	216s	16ms
29	11ms	810s	78ms

References

[1] D. J. Berstein. *Enumerating and smooth integer.* PhD thesis, 1995.

[2] D. J. Berstein, "Arbitrarily tight bounds on the distribution of smooth integers," in *Number theory for the millinium I*, pp. 49–66, 2002.

[3] J. Buchmann and D. Weber. "Discrete logarithms recent progress,". Tech. Rep. T1-12/98, 1998.

[4] C. C. Chang and Y. F. Chang, "A novel three party encrypted key exchange protocol," *Computer Standards and Interfaces*, vol. 26, no. 5, pp. 471–476, 2004.

[5] D. Coppersmith, A. M. Odlyzko, and R. Schroeppel, "Discrete logarithms in gf(p)," *Algorithmica*, 1986.

[6] A. Das and V. Madhavan, "On the cubic sieve method for computing discrete logarithms over prime fields," *International Journal of Computer Mathematics*, vol. 82, no. 12, pp. 148–495, 2005.

[7] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Information Security*, vol. 22, no. 6, pp. 644–654, 1976.

[8] T. ElGamal, "A public key cryptosystem and signature based on discrete logarithms," *IEEE information Theory*, pp. 469–472, 1985.

[9] D. M. Gordon, "Discrete logarithms in gf(p) using the number field sieve," *SIAM Journal of Discrete Mathematics*, no. 6, pp. 124–138.

[10] A. Hildebrand, "On the number of positive integers $\leq$ x and free of prime factors>y," *Journal of Number Theory*, vol. 22, pp. 289–307, 1986.

[11] A. Hildebrand and G. Tenenbaum, "On the integer free of large prime factors," *Transactions on AMS*, pp. 65–90, 1986.

[12] S. Hunter and J. Sorrenson, "Approximating the number of integers free of large prime factors," *Mathematics of Computation*, vol. 66, no. 220, pp. 1729–1741, 1997.

[13] B. A. LaMacchia and A. M. Odlyzko, "Computation of discrete logarithms in prime fields," *Design codes and Cryptography*, vol. 1, pp. 46–62, 1991.

[14] B. A. LaMacchia and A. M. Odlyzko, “Computation of discrete logarithms in prime fields,” *LNCS*, vol. 537, 1991.

[15] K. S. McCurely, “The discrete logarithm problem,” *Cryptology and Computational Number Theory*, vol. 42, pp. 49–74.

[16] A. Menezes and U. Berkant, “On reusing ephemeral keys in diffie-hellman key agreement protocols,” *International Journal of Applied Cryptography*, vol. 2, no. 2, pp. 154–158, 2010.

[17] A. M. Odlyzko, “Discrete logarithms in finite fields and their cryptographic significance,” in *Eurocrypt' 84*, vol. LNCS 209, pp. 224–317. Spriger-Verlag, 1984.

[18] A. M. Odlyzko, “Discrete logarithms:the past and the future,” *Designs Codes and Cryptography*, pp. 129–145, 2000.

[19] A. M. Odlyzko, “On the complexity of computing discrete logarithms and factoring integers,” *Open problems in communication and computation*, pp. 113–116, 2000.

[20] R. Padmavathy and C. Bhagvati, “Ephemeral key recovery using index calculus method,” *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 13, no. 1, pp. 29–43, 2009.

[21] R. Padmavathy and C. Bhagvati, “A key recovery attack on chang and chang password key exchange protocol,” in *International Conference on Computer and Network Technology*, pp. 176–181, 2009.

[22] R. Padmavathy and C. Bhagvati, “Methods to solve discrete logarithm problem for ephemeral keys,” in *International Conference on ARTCOM*, pp. 704–708, 2009.

[23] S. Pohlig and M. Hellman, “An improved algorithm fro computing logarithms over gf(p) and its cryptographic significance,” *IEEE Transactions on Information Theory*, vol. 24, pp. 106–110, 1978.

[24] O. Schirokauer, D. Weber, and T. Denny, “Discrete logarithms,the effectiveness of the index calculus method,” vol. 1122, pp. 337–361.

[25] C. Studholme, “Discrete logarithm problem,” *Research paper requirement (milestone) of the PhD program at the University of Toronto*, 2002.

[26] D. Weber, “Computing discrete logarithms with the general number field sieve,” *LNCS*, vol. 1122, pp. 99–114, 1996.

[27] D. Weber and T.Denny, “The solution of mccurleys discrete log challenge,” *LNCS*, vol. 1462, pp. 458–471, 1996.

[28] M. Zhang, B. Yang, Y. Zhong, P. Li, and T. Takagi, “Cryptanalysis and fixed of short signature scheme without random oracle from bilinear parings,” *International Journal of Network Security*, vol. 12, no. 3, pp. 130–136, 2011.

R. Padmavathy received an M.tech degree from Andhra University and a Ph.D from the University of Hyderabad, India. At present she is working as a faculty member at the National Institute of Technology, Warangal. Her research interests include Information security, Cryptology and Network Security.

Chakravarthy Bhagvati received his Ph.D deree from RPI Newyork, USA. At present he is a professor at University of Hyderabad, Hyderabad, India. He published a number of papers in International Journals. His research interests include Image Processing, Computer Vision, Pattern Recognition, OCR for telugu and Cryptography.